Xiaodong Lin
Ali Ghorbani
Kui Ren
Sencun Zhu
Aiqing Zhang (Eds.)

LNICST

239

LNICST

Security and Privacy in Communication Networks

SecureComm 2017 International Workshops, ATCS and SePrIoT
Niagara Falls, ON, Canada, October 22–25, 2017
Proceedings

EAI

Springer

Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 239

More information about this series at http://www.springer.com/series/8197

Xiaodong Lin · Ali Ghorbani
Kui Ren · Sencun Zhu
Aiqing Zhang (Eds.)

Security and Privacy in Communication Networks

SecureComm 2017 International Workshops, ATCS and SePrIoT
Niagara Falls, ON, Canada, October 22–25, 2017
Proceedings

Springer

Editors

Xiaodong Lin
Wilfrid Laurier University
Waterloo, ON
Canada

Ali Ghorbani
University of New Brunswick
Fredericton, NB
Canada

Kui Ren
University at Buffalo
Buffalo, NY
USA

Sencun Zhu
Pennsylvania State University
Philadelphia, PA
USA

Aiqing Zhang
Anhui Normal University
Wuhu
China

ISSN 1867-8211 ISSN 1867-822X (electronic)
Lecture Notes of the Institute for Computer Sciences, Social Informatics
and Telecommunications Engineering
ISBN 978-3-319-78815-9 ISBN 978-3-319-78816-6 (eBook)
https://doi.org/10.1007/978-3-319-78816-6

Library of Congress Control Number: 2018940136

Printed on acid-free paper

This Springer imprint is published by the registered company Springer International Publishing AG
part of Springer Nature
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

The 13th EAI International Conference on Security and Privacy in Communication Networks (SecureComm) was held during October 22–25, 2017, in the beautiful Niagara Falls, Canada. SecureComm is one of the premier conferences in cyber security, which provides an opportunity for researchers, technologists, and industry specialists in cyber security to meet and exchange ideas and information.

We were honored to have hosted keynote speeches by two world renowned cyber security researchers, Dr. Patrick McDaniel, Pennsylvania State University, and Dr. Ninghui Li, Purdue University. Their topics include security and privacy in machine learning, and differential privacy, which are currently hot research topics in cyber security research.

The conference program included technical papers selected through peer reviews by the Program Committee members, invited talks, and special sessions. Out of a total number of 105 submissions, 31 were selected as full papers and 15 as short papers. Besides the main conference, there were two workshops on emerging research topics in the field of security and privacy. This volume comprises the proceedings of these two workshops. The first one, the International Workshop on Applications and Techniques in Cyber Security (ATCS), has been affiliated with SecureComm for many years, focusing on all aspects on techniques and applications in cyber security research. In all, 17 papers were accepted and are included in the proceedings. The second one was the First Workshop on Security and Privacy in the Internet of Things (SePrIoT). SePrIoT is intended to reflect the importance of addressing security and privacy in the Internet of Things (IoT). Five papers were accepted and included in the proceedings. The technical program thus comprised a total of 68 papers.

We would like to thank all of our authors as well as our dedicated organizing team and volunteers for their hard work. SecureComm would not be successful without the dedication and passion of its contributors. Many people worked hard to make SecureComm 2017 a success. We would like to express our gratitude to them. It is also impossible to list here all those individuals whom we are grateful to. But we would like to thank particularly the EAI, especially Prof. Imrich Chlamtac of EAI, for their strong support of this conference. Also, we thank the members of the conference committees and the reviewers for their dedicated and passionate work. In particular, we thank the Program Committee co-chairs, Dr. Kui Ren and Dr. Sencun Zhu, for their great leadership in creating such a wonderful program. We also thank Ms. Dominika Belisová of EAI for her hard work and dedication in taking great care of the conference organization. Without the extremely generous support of EAI, this conference could not have happened. Last but not least, we thank the Steering Committee of SecureComm for having invited us to serve as the general chairs of SecureComm 2017.

We hope you enjoy the proceedings of SECURECOMM 2017 as much as we enjoyed the conference.

March 2018

Xiaodong Lin
Ali Ghorbani

Organization

5th International Workshop on Applications and Techniques in Cyber Security, ATCS 2017

General Chairs

Jemal H. Abawajy	Deakin University, Australia
Rafiqul Islam	Charles Sturt University, Australia
Kim-Kwang Raymond Choo	University of Texas at San Antonio, USA

Program Committee

Junbin Gao	The University of Sydney, Australia
M. Alamgir Hossain	Northumbria University, UK
Md Enamul Karim	Louisiana Tech University, USA
V. Vijayakumar	VIT University, Chennai, India
Faisal Hasan	University of Illinois, USA
Tanveer Zia	Charles Sturt University, Australia
Maumita Bhattacharya	Charles Sturt University, Australia
Mizanur Rahman	King Saud University, KSA
Morshed Chowdhury	Deakin University, Australia
Mohd Farhan Fudzee	Universiti Tun Hussein Onn Malaysia, Malaysia
Zahid Islam	Charles Sturt University, Australia
Syed S. Islam	University of Western Australia, Australia
Andrei Kelarev	Federation University, Australia
Mohammad Kausar	Ittehad University, KSA
Hasannuzzaman	University of Dhaka, Bangladesh
Quazi Mamun	Charles Sturt University, Australia
Ben Martini	University of South Australia, Australia
Sumon Sahriar	CSIRO, Australia
Steve Versteeg	CA Labs, Australia
Matthew Warren	Deakin University, Australia
John Yearwood	Federation University, Australia
Samsul Huda	Deakin University, Australia
Xun Yi	RMIT University, Australia
Arif Khan	Charles Sturt University, Australia
Nasimul Noman	University of Newcastle, Australia
Chowdhury Farhan Ahmed	University of Strasbourg, France

Organization

First Workshop on Security and Privacy in the Internet of Things, SePrIoT 2017

General Chair

José María de Fuentes	Computer Security Lab (COSEC), Carlos III University of Madrid, Spain

General Co-chairs

Lorena González-Manzano	Computer Security Lab (COSEC), Carlos III University of Madrid, Spain
Pedro Peris-López	Computer Security Lab (COSEC), Carlos III University of Madrid, Spain

Technical Program Committee

Jemal Abawajy	Deakin University, Australia
Mu'awya Al Dalaien	HCT-Khalifa City Women's College, United Arab Emirates
Nasour Bagheri	Shahid Rajaee Teachers Training University, Iran
Rahmat Budiarto	Surya University, Indonesia
Carmen Cámara	Universidad Carlos III de Madrid, Spain
Kim-Kwang Raymond Choo	University of Texas at San Antonio, USA
Juan Estévez-Tapiador	Universidad Carlos III de Madrid, Spain
Flavio García	University of Birmingham, UK
Ana Isabel González-Tablas	Universidad Carlos III de Madrid, Spain
Flavio Lombardi	Università Roma Tre, Italy
Javier López	University of Malaga, Spain
Samuel Marchal	Aalto University, Finland
Katerina Mitrokotsa	Chalmers University of Technology, Sweden
Ana Nieto	Universidad de Málaga, Spain
Panos Papadimitratos	KTH, Sweden
Julian Schütte	Fraunhofer AISEC, Germany
Jetzabel Serna-Olvera	Goethe Universität, Frankfurt, Germany
Agusti Solanas	Rovira I Virgili University, Spain

Contents

International Workshop on Applications and Techniques in Cyber Security (ATCS)

1st Workshop on Security and Privacy in the Internet of Things (SePrIoT)

International Workshop on Applications and Techniques in Cyber Security (ATCS)

Privacy in Social Media

Erdal Ozkaya and Rafiqul Islam

School of Computing and Mathematics, Charles Sturt University, Sydney, Australia
erozkaya@microsoft.com, mislam@csu.edu.au

Abstract. This paper researched two issues; targeted advertising and politics in social media. In its findings, it was apparent that many social media platforms actively collected user data and used big data to store and analyze it so as to profile the users. From these profiles, targeted adverts were served to the users on their social media feeds. The data collection process was however found to be unethical and full of privacy intrusions. Concerning politics, it was found out that social media had been used as a political tool. There were findings that most users were annoyed with political posts being shown on their feeds. The culprits were found to be the algorithms behind social media feeds and suggestions and the users themselves. The paper gave recommendations said to be temporary solutions for the issues surrounding target advertising and politics in social media.

Keywords: Social media issues · Platform · Target advertising · Big data
Social media · Social media platform · Social media issue · Targeted advertising
Social medium user · Collected user data · Political view

1 Introduction

Eyebrows have been raised concerning the power that different social media platforms hold. The currently used social media platforms have been built using sophisticated systems and are powered by big data and complex algorithms. These platforms know almost everything about each of their users, from their actual locations to their likes and shopping habits (Shankar 2011). Mainly, this knowledge has been applied in marketing whereby they have set up advertising platforms for sellers. Using their rich knowledge of users from big data, these platforms are able to analyze each user's likes and interests (Shankar 2011). From this analysis, they then advertise to the user's products that they would be likely to buy (Shankar 2011). Social media marketing has easily crept up the list of the most commonly used digital marketing techniques and today it battles for the first place with Google advertising.

Apart from advertising, there are fears that social media is getting tangled in politics. There are fears that they are also being used to slowly brainwash users by politicians. Social media platforms are becoming politicized by users and the algorithms. Users currently have to tolerate politically oriented posts flooded all over their newsfeed either shared by, commented on or posted by their friends. The sophisticated social media algorithms have also been set in such a way that they will continually fill on a user's timeline political posts based on what they read, like or comment on. However, the

X. Lin et al. (Eds.): SecureComm 2017 Workshops, LNICST 239, pp. 3–10, 2018.
https://doi.org/10.1007/978-3-319-78816-6_1

algorithms are not perfect and thus have been a cause of frustration for many users. There are also questions as to whether social media have been used to stumble governments. After all, there have been some uprisings witnessed that were fueled by social media. It has also been observed that some governments have had to result to ban these platforms in tumultuous times.

1.1 Research Questions

(a) How are social media platforms doing target advertising with collected user data?
(b) Is social media being used for political reasons?

2 Literature Review

There have been a few previous works done on the same topic. Wolfsfed, Segev and Sheafar have done a paper on the role that social media played in some events of the Arab Spring (Wolfsfeld et al. 2013). The three discuss social media as a low-cost tool that is both powerful and speedy and could be used to recruit, raise funds and distribute inciting information to the masses. The three say that these are the right ingredients to start a protest in restrictive governments such as the non-democracies of Arab nations (Wolfsfeld et al. 2013). They say that in 2009, Twitter was used by protestors in Iran to fuel a revolution. Protestors are said to have ganged up on Twitter to organize coordinated protests (Wolfsfeld et al. 2013). The researchers, however, try to distance the social media platforms from the protests and say that there were several other factors at play. They bring to light the political environment at the time of the said social media powered revolutions. They point out that the revolutions in Iran took place with only 8,000 of 70 million citizens registered on Twitter (Wolfsfeld et al. 2013). At the same time, in other Arab countries where more people were on social media, no protests happened (Wolfsfeld et al. 2013). Therefore, in as much as social media was to be blamed, it was rather a tool used, just the same way the protestors would have gone to the streets and picketed. They seem to convince readers to move away from assuming that social media has anything to do with collective actions done by citizens with political, social or economic consequences. Another relevant piece of work is a paper by Broyles and Slater about social media advertising (Broyles and Slater 2014). In the paper, it is found out that big data has been heavily employed in advertising. It is what social media platforms use to power their advertising tools. They explain that with big data, the platforms are able to collect user data, consumption patterns and purchase behaviors (Broyles and Slater 2014). This information is derived from what users post, comment, like or even privately chat about with others. The two give examples of how big data is being used today in monitoring what people say about different brands on Twitter (Broyles and Slater 2014). They also explain that it is used to keep details of how the said consumers interact with various devices. They try to bring out a picture that a lot is done in the background, with the power of big data, to ensure that the right advert gets to the right consumer (Broyles and Slater 2014). The two finish off by explaining some ethical issues surrounding the whole targeted advertising ambitions of social media platforms. They give an example of a user that posts about a medical condition being targeted with different pharmaceutical adverts (Broyles and Slater 2014). They also highlight the question of the validity of big data. User tastes, likes and preferences change often and big data is not able to keep up. In the end, it ends up generating the wrong adverts to users whose likes have changed (Broyles and Slater 2014). They also slightly bring to light the privacy issues being brought up concerning the issue of intrusion into users' private lives in order to target them with adverts (Broyles and Slater 2014).

3 Research Method

The chosen research method was a qualitative analysis of data. The data that used was composed of opinions from different respondents. Therefore, the best type of analysis for the data collected would be qualitative. The research was totally depended on secondary sources. It used, analyzed and discussed the findings obtained by previous researches. Two secondary data sources were used. They were from the highest ranked research institute in the world. The first research carried out in 2016 had 4579 respondents (Duggan and Smith 2017). The respondents were asked to fill a web survey or alternatively to download a questionnaire, fill and send back via mail. From the second research carried out in 2012, 1729 respondents were asked questions concerning target advertising. The entire research was conducted through online surveys and phone interviews. The respondents were both Americans and Hispanics and therefore the questions were asked in either English or Spanish. The data obtained from the two secondary sources was analyzed using Microsoft Excel to give graphical representations of the data obtained wherever possible.

4 Results

The first secondary data source considered was from Pew Research institute and it concerned the issues of social media involvement in politics. In the research, the participants had been asked to give their feedback concerning the political feeds they got from their social media accounts. 37% said that they felt worn out, 22% were enjoying and 41% said that they neither felt angered nor joyed by the political feeds (Duggan and Smith 2017). When asked about how they felt when they came across opposing political views, 59% of the respondents said that they felt annoyed while 35% said that they just found them informative. The said that that they were neither infuriated nor interested by such discussions.

As for the impacts of the confrontations between people of opposing political views, 64% said that they felt a larger rift between themselves and the opponents (Duggan and Smith 2017). 36% felt that the encounters left them on more common grounds with the supporters of the opposing political views (Duggan and Smith 2017). When asked about their reactions to political posts of opposing views as theirs, 83% of the participants said that they ignore them while the rest said that they leave a comment (Duggan and Smith 2017). 31% of the respondents said that they changed their settings in order to see fewer political posts from volatile political enthusiasts while 27% unfriended such people instantly (Duggan and Smith 2017). When asked about the most used platforms for pushing political content to users, the respondents said that Facebook and Twitter shared the top position.

The second data source was another research, still by Pew Research institute, concerning the issue of targeted advertising. When asked whether they approved or disapproved of being subjected to targeted advertising, 68% of the respondents disapproved (Purcell et al. 2017) (Fig. 1).

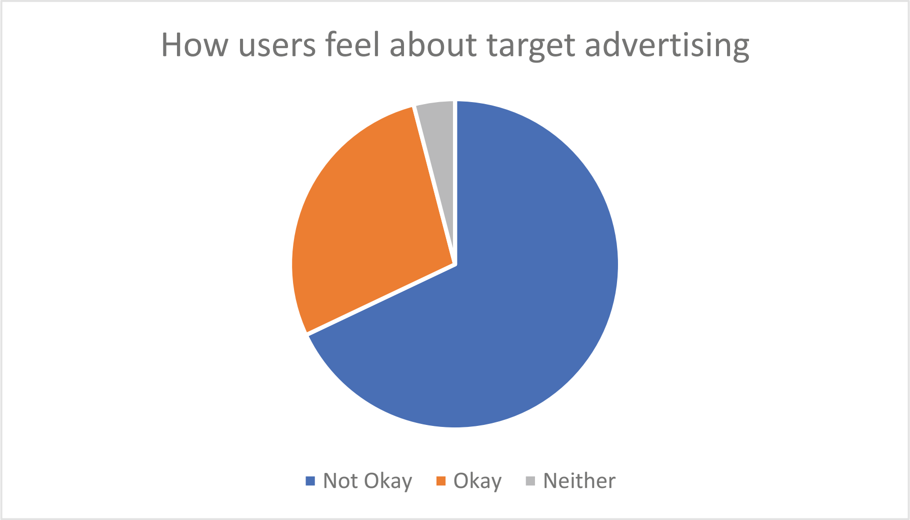


Fig. 1. User's reaction to target advertising (Purcell et al. 2017)

How do you feel about target advertising	
Not okay	1175
Okay	484
Neither	70

28% of the respondents said that they were okay with targeted advertising while 4% neither had a positive or negative reaction to target advertising (Purcell et al. 2017). From the respondents that were okay, it was observed that most of these were young or came from households with low income (Purcell et al. 2017). Most of the respondents that disapproved target advertising were male, under the age of 65, with college degrees and came from high-income households (Purcell et al. 2017) (Fig. 2).

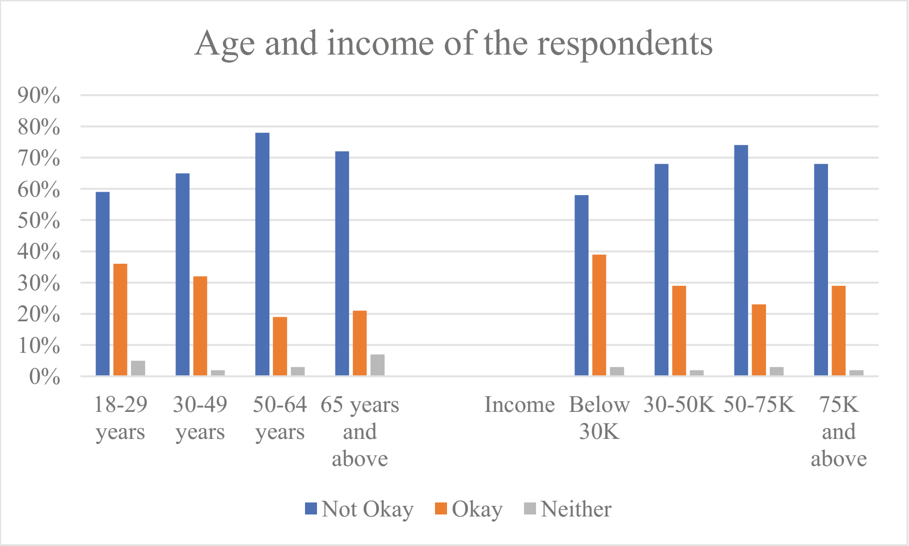


Fig. 2. The age and income brackets of the respondents (Purcell et al. 2017)

Age and income of respondents			
Age	Not okay	Okay	Neither
18–29 years	59%	36%	5%
30–49 years	65%	32%	2%
50–64 years	78%	19%	3%
65 years and above	72%	21%	7%
Income	Not okay	Okay	Neither
Below 30K	58%	39%	3%
30–50K	68%	29%	2%
50–75K	74%	23%	3%
75K and above	68%	29%	2%

5 Discussion

From the first research, it was observed that social media users were growing concerned with the number of politically-oriented posts on their timelines. Whilst some enjoyed using social media for political debates and engaging in political discussions, there was quite a number that was frustrated with such content. As was found out, 37% of the respondents said that they were growing tired of having to come across these posts. They claimed that social media was being used for polarization and to promote animosity between rival political groups. They, therefore, opted to steer clear of such posts and discussions but were growing tired of doing so. They found it exhausting to try and avoid all the political posts that were being flooded on their timelines. Unfortunately, these posts were being fed to their timelines even when they made efforts to try and avoid them. On the other hand, 22% said that they enjoyed participating in heated political discussions on social media. They said that they found the informative and interesting. 41% said that they neither felt angered nor pleased with the political posts and discussions on social media.

When respondents were asked about the impacts of confrontations with people of different political opinions on social media, most respondents said that they were both frustrating and stressful. Mostly, they claimed that instead of there being meaningful discussions, there was an exchange of insults between the conflicting sides. It can already be seen that a majority of social media users are against political posts and discussions on social media. Most users also claim that these are non-helpful discussions on social media and they leave more damage than they do good. The next question that respondents were asked amplifies this. When asked about the end result of engaging in discussions with people of opposing political views, 64% said that they felt a larger rift than before.

It is evident that most political arguments on social media turn out to be heated. Such kind of discussions end up giving opposing sides more reasons to hate each other. Most users also said that they try as much as possible to ignore political posts while only a few go ahead to read them and leave comments. This is because the political posts on social media are targeted at swaying the thoughts of anyone that reads them. It is something that social media users try to avoid. 37% of the respondents said that they try to change their social media settings in order to avoid seeing political posts. Social media platforms have provided different ways to do so. One can hide such posts or block people who post or share such articles. There is also an option of reporting a post so that the social media companies can take further actions on a particular user. Some respondents said that they will instantly block or unfollow anyone who posts or share aggressive political posts.

Throughout the whole research, it can be seen that a majority of social media users are actually against political content on social media. It is so unfortunate that social media platforms, on the other hand, are trying to push towards politics. If one watches just one video clip concerning politics, the social media platforms have complex algorithms to keep feeding such a person with more political feeds. It is desperate for users that do not want to continue getting such type of content because it annoys them. Even on YouTube, a click on the wrong political video clip triggers off a stream of political video suggestions. Users have already said that they are exploring ways to avoid interacting with such feed on their timelines.

From the second research, three-quarters of the respondents said that they were not okay with having their information collected for target advertising. In target advertising, social media platforms continue to monitor all the activities of a user on and outside social media. This data is collected and stored in large repositories supported by big data (Holtzhausen 2016). These repositories contain data about what each social media user posts, likes, comments on, his or her physical location and even what is in the private chat (Holtzhausen 2016). Some social media platforms such as Facebook go ahead and collect device information and read all the contacts that a user has on his phone.

With this information, these platforms employ complex algorithms to profile each user into a certain category to be sent adverts most suitable for him or her (Holtzhausen 2016). This is what users are trying to avoid, the loss of their privacy with a selfish aim by social media platforms of being targeted with adverts. There is a huge outcry concerning the ways social media platforms have intruded the personal lives of people in order to get more information about them for the purposes of marketing. In this research, most of the respondents that said they did not like being spied on for targeted adverts were below 65 years of age and earned high incomes. Some of the young and low-income earners said that they were okay with their data being collected for target advertising. It can be seen that, still in all the different age and income brackets, most of the respondents were against the collection of their data.

6 Conclusion

The findings and discussions from the two researches have succinctly answered the research questions. As concerns the way targeted adverts are made out of the collected user data, the issue of big data and complex algorithms has been brought to light. It has been found out that however much that targeted advertising is benefiting the social media companies, users are crying. So as to amass maximum revenues from social media advertising, many infringements to the privacy of users have been done. Users are feeling betrayed by these social media platforms. The few that are oblivious to the extent of the privacy issues said that they were okay with targeted advertising. Unknown to them is that data unrelated to social media is being siphoned off their devices.

The second question that was researched on it the issue of entanglement of social media platforms with politics. It has been proven that social media has been made a political tool. There are many culprits to blame. One of them is the newsfeed algorithm for each social media platform. They have increasingly been suggesting more politically-oriented feeds to users. Watching one of these suggestions has led to an aggressive flooding of one's feed with more political posts. The other culprits of political entanglement in social media are the users themselves. There are users that take social media as the field to wage battles against their political opponents. There are others who actively create infuriating posts designed to anger their political rivals. There are yet others that have used social media to fuel protests against governments. That is why governments such as in Turkey have sought to block off such media during politically heated moments such as the failed coup.

7 Recommendations

There are several recommendations that this paper sees as temporary solutions to the two issues that were researched on. Concerning target advertising, governments are encouraged to come up with legislations barring the intrusion of the privacy of their citizens. They should restrict the amount of data that social media platforms can collect from users. Preferably, these platforms should only collect data from posts, likes and comments that users make. Only governments can resolve this since the social media platforms have shown reluctance to address the cries of their users. As concerns the issue of political entanglement in social media, it is upon the platforms to devise new ways for users to turn off any politically related posts. Some platforms have come up with ways to block content that is considered not safe for work. The same way that they have implemented this, they should be able to come up with a way to block all political posts when a user wants to. This will help users that do not want to come across such feeds on their timelines.

References

Broyles, S., Slater, J.: Big thinking about teaching advertising. J. Advert. Educ. **18**(2), 46–50 (2014). https://search.proquest.com/docview/1645350994

Duggan, M., Smith, A.: The political environment on social, media. Pew Research Center: Internet, Science & Tech. (2017). http://www.pewinternet.org/2016/10/25/the-political-environment-on-social-media/. Accessed 14 Mar 2017

Holtzhausen, D.: Datafication: threat or opportunity for communication in the public sphere? J. Commun. Manag. **20**(1), 21–36 (2016). https://search.proquest.com/docview/1761612578

Purcell, K., Brenner, J., Rainie, L.: Main findings. Pew Research Center: Internet, Science & Tech. (2017). http://www.pewinternet.org/2012/03/09/main-findings-11/. Accessed 14 Mar 2017

Shankar, R.: Harnessing the power of social media data with master data management. Bus. Intell. J. **16**, 27–33 (2011). https://search.proquest.com/docview/912872475

Wolfsfeld, G., Segev, E., Sheafer, T.: Social media and the Arab spring: politics comes first. Int. J. Press/Polit. **18**(2), 115–137 (2013)

SecControl: Bridging the Gap Between Security Tools and SDN Controllers

Li Wang and Dinghao Wu(✉)

College of Information Sciences and Technology,
The Pennsylvania State University, University Park, PA 16802, USA
{lzw158,dwu}@ist.psu.edu

Abstract. Software-defined networking (SDN) is a promising paradigm to improve network security protections. A lot of security enhancements through SDN have been proposed. However, current SDN-based security solutions can hardly provide sufficient protections in a real SDN network, due to several reasons: (1) they are implemented at either the centralized SDN controllers or the decentralized network devices, which are subject to a performance limitation; (2) their designs are confined by SDN network characteristics and can only provide limited security functions; (3) many solutions have deployment challenges and compatibility issues. In this paper, we propose SecControl, a practical network protection framework combining the existing security tools and SDN technologies, to produce a comprehensive network security solution in an SDN environment. By employing the capabilities of existing security tools, SecControl is able to perceive the real-time security events dynamically and adjust the protected network environment correspondingly. It can be easily extended with various methods for different security threats. With SecControl, we construct a traditional-security-tool-friendly network security solution for software-defined networks. We implement a SecControl prototype with OpenFlow and evaluate its effectiveness and performance. Our experiment shows that SecControl can cooperate with many mainstream security tools and provide effective defense responses over SDN-supported networks.

Keywords: Software-defined networking (SDN)
Network Function Virtualization (NFV) · OpenFlow
SDN security application · SDN controller

1 Introduction

Software-defined networking (SDN) has gained much attention in both academia and industry [22]. By decoupling the control logic from the closed and predesigned network devices, SDN enables the reprogramming capability of network devices. Previously, traditional network devices can only work as they are manufactured, and all their traffic control and forwarding functions are not changeable once produced. With SDN, the traffic control functions and traffic forwarding

X. Lin et al. (Eds.): SecureComm 2017 Workshops, LNICST 239, pp. 11–31, 2018.
https://doi.org/10.1007/978-3-319-78816-6_2

functions are divided as *control plane* and *data plane*. The separation of *control plane* and *data plane* provides a powerful and flexible network structure for various network applications.

A lot of network-related research has been conducted with SDN, such as network management [8,9,20], network QoS [14], network load balancing [17,35], and content delivery system [36]. Similarly, researchers tried to take advantage of SDN technologies to devise new network security solutions as well. Many innovations [31–34] tried to provide better security services over software-defined networks, and they are provided either at the centralized controllers or the distributed inline network devices.

However, the existing SDN-based security solutions can hardly compete with traditional security solutions due to various reasons. First, they are designed with limitations inherently. When security functions are implemented at centralized controllers [32], the processing capabilities of controllers will become a potential bottleneck; when security functions are deployed at network devices [34], it can hardly provide a comprehensive protection over the network. Second, most of them are focusing on maximizing the control flexibility of SDN. Maximizing network control flexibility does not necessarily lead to strengthened network protection ability. Third, the existing SDN-based security solutions are mainly on a certain aspect of network protection [19], which can hardly satisfy the general network protection requirements. Last, many of them have deployment challenges and compatibility issues.

As a result, the current SDN-based security solutions cannot provide the same protection capabilities as traditional security tools can provide over SDN networks. Actually, the key innovations brought by SDN are over network control instead of security processing capability. Network protection demands more powerful security processing capabilities, such as packet payload inspection, traffic pattern analysis, and so on. Therefore, we need a practical network security solution which can provide competitive security protection and, at the same time, can take advantage of the flexible control over SDN networks.

Traditional security tools, like firewalls and intrusion detection systems, have strong security processing capabilities in protecting traditional network infrastructures, and each type of security tool is specialized to deal with a certain type of security threat. They are composed together to form a comprehensive network security solution. However, traditional security tools can hardly be used directly in software-defined networks because of the following reasons: (1) existing security tools are designed under the traditional network infrastructure, which does not fit into SDN network structure; (2) most security tools are devised to deal with a certain type of security threat. Their exclusive designs decide they can only be used individually and cannot cooperate with each other; and (3) there is no interface on existing security tools to let them take advantage of SDN benefits.

In this paper, we propose SecControl, a new network protection framework bridging the gap between security tools and SDN technologies, to provide sufficient protection capabilities in an SDN environment. Our goal is to design a practical and comprehensive network security solution over SDN networks by leveraging existing security tools and SDN control flexibility. Unlike existing SDN-based security solutions, SecControl is designed on a new security control layer above SDN controllers, which releases SDN controllers from security processing pressure. SecControl is able to perceive the real-time security threats, generate real-time defense reactions, and adjust corresponding network behaviors dynamically. With SecControl, security engineers can easily add different security tools into the protection boundary and make use of their detection abilities to serve the entire network. Our method can be applied on mainstream SDN platforms without difficulty.

In summary, the main contributions of SecControl are as follows:

- We propose a novel network protection framework for software-defined networks, which combines the existing security tools and SDN technologies. Our framework retrofits and reuses the existing security tools in the SDN context, which avoids re-development of many security defense functionalities.
- Our method equips an SDN network with strong security processing capabilities in an economic way. Existing security tools can be used to protect SDN networks without difficulty.
- SecControl layer provides an additional layer above SDN controllers, which release controllers from security processing pressures. SecControl has a full security view of the protected network domain, which enables SecControl to offer a unified protection.
- We design a practical method to dynamically translate defense responses into SDN rules to adjust network behaviors. We provide a set of SDN primitives, namely *drop*, *forward*, *reflect*, *isolate*, and *copy*, and these primitives can be translated to OpenFlow flow rules automatically.
- SecControl separates the security processing logic from the security enforcement components. With our method, a SecControl domain can receive remote protection instructions from other SecControl domains, which enables a unified SecControl protection over different SDN networks.

The remainder of the paper is organized as follows. We introduce the challenges of SecControl in Sect. 2. In Sect. 3, we discuss SecControl's architecture and how it is designed. Section 4 describes a SecControl prototype implementing with OpenFlow. The evaluation is presented in Sect. 5. In Sect. 6, we talk about a few insights obtained from this work. We briefly summarize the related work in Sect. 7. Finally, a conclusion is given in Sect. 8.

2 Challenges

Our goal is to design a practical network security solution in SDN networks by employing the security processing capabilities of traditional security tools and SDN technologies. To achieve it, we need to answer several research questions.

RQ1. How Does SDN Improve Network Security Protection?

Network security was once regarded as a subset of network management problem [9]. The key innovation of SDN is separating *control plane* and *data plane* which maximizes the network control flexibility. However, Maximizing network control flexibility does not necessarily lead to the strengthened network protection ability. We may need to think how can we use SDN to improve security. For example, how to assign security responsibilities to *control plane* and *data plane*? How to dynamically adjust network behaviors against security threats?

RQ2. How to Fit Traditional Security Tools into SDN Networks?

Although traditional security tools have powerful security processing capabilities, they cannot be used in an SDN environment directly. The reasons are summarized as follows: (1) traditional security tools are invented for traditional network infrastructure, which can hardly fit into the SDN structure; (2) these tools do not have interaction interfaces for using SDN features to improve security; and (3) seldom can existing security tools share threat information with each other since they are designed individually, and that is a weak point in defending SDN networks. Based on the above reasons, we need to answer how to fit existing security tools into SDN networks? For example, How do we place security tools in an SDN network? How can we collect threat information from traditional security tools?

RQ3. How Can We Combine Them Together?

To make use of the protection capabilities of traditional security tools and maximize the SDN benefits in securing networks, we need to make them work together. Consider most security tools are designed for traditional networks instead of SDN networks, there are several practical issues when combining them together. The first issue is the current security tools are heterogeneous, and their detection results are not compatible. For instance, a host-based intrusion detection system will be mainly monitoring system behaviors, while a firewall will be interested in suspicious network activities. The log generated by the two tools can hardly join together for further security analysis. The second issue is we lack an interaction mechanism for security tools to communicate with SDN networks. We want to employ real-time threats information to adjust network behaviors dynamically. The last issue is we need a unified method to translate the semantics of threat information into SDN rules. For example, how do we extract effective threats information from heterogeneous security event information? Given a certain security threat, how do we adjust network behaviors for an effective defense? How do we distribute defense decisions in an SDN network?

3 Architecture and Design

In this section, we introduce the SecControl architecture and design. We first give an overview of the SecControl architecture. Then, we explain how SecControl works. Last, we explain how the SecControl components are designed and how these components cooperate with each other.

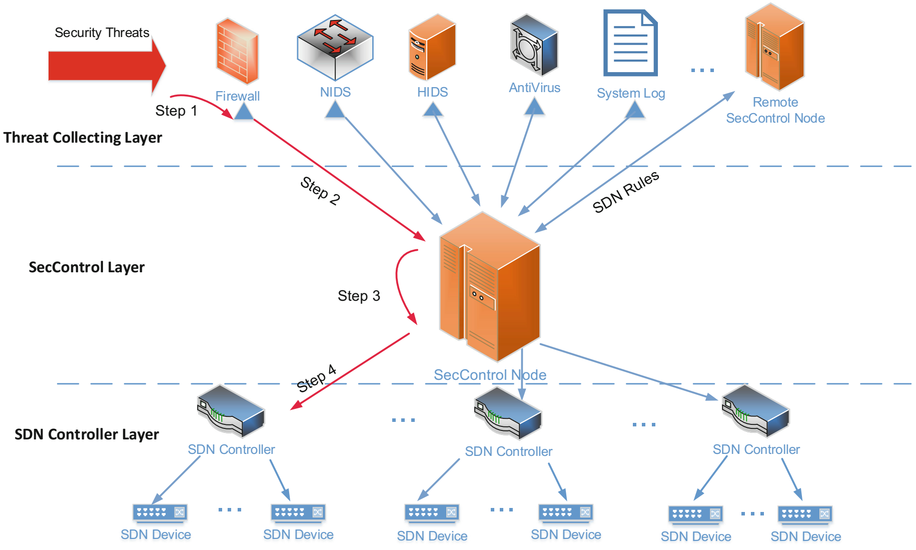


Fig. 1. The SecControl architecture

3.1 Overall Architecture

SecControl seeks to build up a practical and comprehensive protection framework for SDN networks by combining existing security tools and SDN technologies. Through collecting various threats information from various security tools, SecControl converges heterogeneous security alerts at one point. SecControl identifies attack evidence, accesses an overall security situation, and generate corresponding defense responses.

Figure 1 shows the SecControl architecture. The SecControl architecture has three layers, Threat Collecting Layer, SecControl Layer, and SDN Controller Layer. Each layer plays a different role in the SecControl protection framework. The Threat Collecting Layer is composed of various security tools and Threat Collecting Agents. Each security tool will be attached a customized Threat Collecting Agent, which is represented by a small triangle. The Threat Collecting Agent is responsible for collecting and sending threat information to the SecControl Layer. After receiving threat information, the SecControl Layer will run a series of standard steps, which includes converging all the collected security events, correlating related alerts, analyzing alert information, and abstracting attacking evidence. Then, the SecControl Layer will decide a defense response against the detected security attack, and the defense response will be translated into SDN rules and distributed to the SDN Controller Layer. The SDN Controller Layer will enforce the SDN rules and adjust network behaviors.

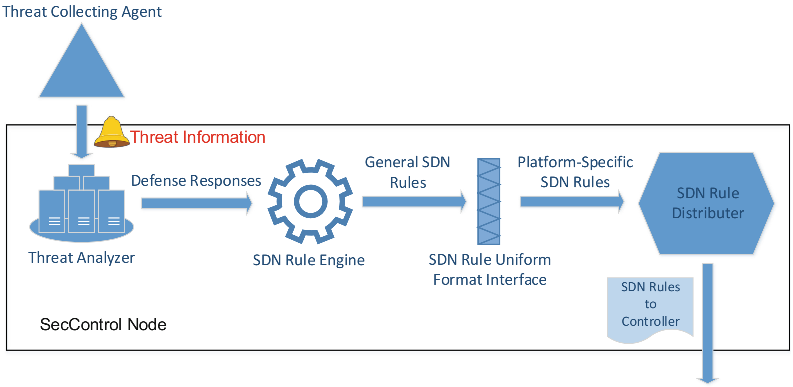


Fig. 2. The SecControl components

3.2 How SecControl Works

As Fig. 1 shows, the four working steps of SecControl are: (1) The security threats are detected by security tools and the detection results are recorded and preprocessed by Threat Collecting Agents (ThreatCA). (2) The preprocessed threat information is sent from ThreatCA to the SecControl Node. (3) The SecControl Node converges and analyzes the threat information to decide how to make a defense response over SDN networks. And, the defense responses will be translated to SDN rules. (4) The SecControl Node distributes the generated SDN rules to the corresponding SDN controllers for enforcement.

In step one, security threat information is generated by various security tools, and ThreatCA preprocesses the recorded security threat information and transforms it into a uniform format. The collected threat information will be sent to SecControl Layer in step two. ThreatCA should be able to extract effective threat information based on the main functions of security tools. For example, a processed firewall alert could be (*firewall, network position, alert level, threat source, detection time, ...*). Actually, the preprocessing of threat information can be quite complicated. More details will be given in the design section.

Step three happens inside of the SecControl Node. The SecControl Node analyzes the threat information, decides defense responses and generates corresponding SDN rules to adjust the network behaviors. In step four, the generated SDN rules will be distributed to corresponding SDN controllers. SecControl Layer is maintaining a list which records the location information of all the controllers in the protected SDN networks. The SDN rules can be sent to the related controllers based on the list. When the SDN rules are transmitted, the transmission process will be protected and secured. There will be a secure protocol between the SecControl Node and controllers to protect their communications.

3.3 SecControl Components

The SecControl framework is composed of four components, as shown in Fig. 2. The first component is Threat Collecting Agent, which is running outside of the SecControl Node and responsible for collecting various security threat information from security tools. The second one is Threat Analyzer, which is in charge of converging and analyzing the collected threat information and decides corresponding defense responses. The third component is SDN Rule Engine, whose responsibility is transforming the generated defense responses to specific SDN rules. The last component, SDN Rule Distributer, is designed for distributing the platform-specific SDN rules to SDN controllers.

Threat Collecting Agent. The inputs of the ThreatCA are various detection results of security tools, while the outputs of the ThreatCA are uniform and well-structured threat information, which can be directly used by Threat Analyzer. Consider existing security tools are separately targeting different threats, their detection results could be quite different. To handle different detection results, we need to provide each type of security tool at least one specialized ThreatCA.

The purpose of ThreatCA is to provide effective threats information to Threat Analyzer. We design a preprocess function on ThreatCA. The preprocess function is responsible for transforming the raw detection results to a unified format which can be used by Threat Analyzer for further analysis. For each ThreatCA, it is designed specially to understand the raw detection results of the security tool it is attached. To release the Threat Analyzer from tedious format details, we present the detection results in a unified format (the format is IDEMF [2]) so that Threat Analyzer can use a uniform interface to deal with all detection results.

Threat Analyzer. The preprocessed threat information will be sent to the Threat Analyzer. The Threat Analyzer will be analyzing threat information, assessing security situations, and deciding defense responses. It is designed as a configurable, adaptable, and extendable module for different protection purposes. Security engineers are able to adjust defense strategies in Threat Analyzer to practice different security analysis and detection algorithms. Analyzing threat information in a large number of detection records is quite complicated, and a lot of algorithms have been proposed [10,12,27].

With our design, security engineers can easily customize these algorithms and deploy them in SecControl Node. Once a security threat is identified, the Threat Analyzer will choose a predefined defense response as a reaction to the security threat. In different protection scenarios, defense responses may refer different reactions. For example, on a firewall, a defense response could be blocking the threaten traffic; while on a host system, a defense response could be isolating a suspicious executable file. In SecControl, we focus on network level responses, which means we adjust network behaviors through SDN technologies as defense responses.

SDN Rule Engine. SDN Rule Engine, as the name suggests, generates the corresponding SDN rules based on the received defense responses. The generated SDN rules instruct how to adjust the network behaviors at SDN network devices. We design a systematic method to achieve the generation process through using SDN primitives, which stands for the basic network operations when dealing with security threats. We define five SDN primitives based on the network flow features. They are *Drop*, *Forward*, *Reflect*, *Isolate*, and *Copy*. The five SDN primitives can be used individually or in combination against security threats.

The five SDN primitives are as follows:

1. **Drop**, which means discarding the identified network traffic. This primitive is usually used to block unwanted network traffic.
2. **Forward**, which just tells the network devices to pass the identified traffic to its destination based on the existing SDN rules. When we do not want to do any operation on the identified network traffic for passing certain network device, we use forward.
3. **Reflect**, changes the destination of the identified network traffic both for inbound and outbound directions. For example, A wants to build up a connection with B. When A's connection traffic is reflected to C, A will be connected with C instead of B. After this, C will use B's network address and communicate with A, and A knows nothing about this. Reflect primitive can be used in deploying a shadow server or a honeypot.
4. **Isolate**, limits the identified traffic to a certain host or network area. When a node (or a node group) is identified as a source of an attack, we use this primitive to confine its network activities.
5. **Copy**, duplicates the identified packets, which is usually used for monitoring or logging use. Most current network devices have been equipped with this primitive. It could be used for real-time traffic analysis and other purposes.

The five SDN primitives can be used in combination, repeatedly, and in any sequence to form a wanted defense response. Each defense response will be translated into one or several SDN primitives. For example, a defense response may require directing the suspicious source to a honeynet, where the suspicious traffic will be recorded and analyzed. In this situation, the defense response will be translated into two SDN primitives, *reflect* and *isolate*. The suspicious traffic will be first reflected to a honeynet and then isolated in the honeynet area.

Usually, each SDN rule contains one SDN primitive, which represents the specific action of this rule. Some SDN primitives, like *drop* and *forward*, have been supported on most SDN platforms. For those SDN primitives that cannot be well supported, we may need additional translation processes to turn these primitives into corresponding SDN rules. Besides, we design an SDN rule uniform format interface to transform general SDN rules to platform-specific SDN rules. The platform-specific SDN rules can be distributed to SDN controllers by SDN Rule Distributer for execution.

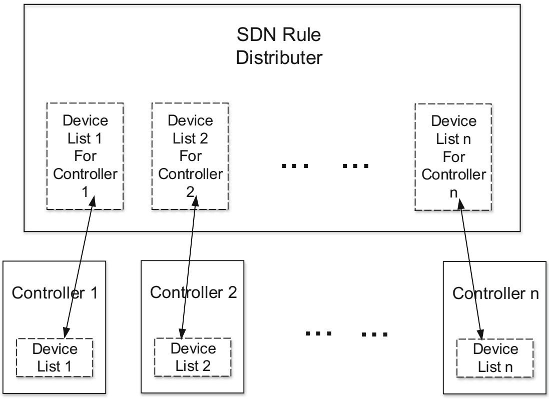


Fig. 3. The SDN rule distributer

SDN Rule Distributer. The generated SDN rules will be sent to SDN controllers through the SDN Rule Distributer. In an SDN network, network devices are divided into groups and each group will be connected and managed by a controller. Only the controller can send SDN rules to its connected SDN devices. The SDN Rule Distributer needs to distribute the SDN rules to controllers first, then have controllers send SDN rules to corresponding SDN devices.

To ensure the SDN rules can be delivered to the right controller, the SDN Rule Distributer should have a full map of the SDN networks. As can be seen in Fig. 3, the SDN Rule Distributer stores a local copy of network device lists for all the SDN controllers. In this paper, we use OpenFlow to build up SDN networks. When we dynamically update OpenFlow rules, it may cause inconsistencies [19,23] at OpenFlow devices. A lot of research has been done on verifying the consistency of OpenFlow rules. Consider SDN rules consistency is not our research focus, we assume this problem is well solved in our design.

3.4 Components Communication

Based on the workflow of SecControl, we need two communication mechanisms which reside in step two and step four separately. In step two, the Threat Collecting Agents need to communicate with the SecControl Node to send collected security threat information, and that communication can be happening all the time. The other communication happens between the SecControl Node and SDN controllers, which serves to distribute SDN rules and maintain network devices information. Besides, we also need another communication mechanism among the SecControl Nodes, which enables the exchange of SDN rules between different SecControl Nodes.

We can achieve the step two communication like any typical network application by using TCP/IP protocols. The Threat Collecting Agents can send security threat information over TCP or UDP protocol, which can both be used for typical network communication. The communication between the SecControl Node

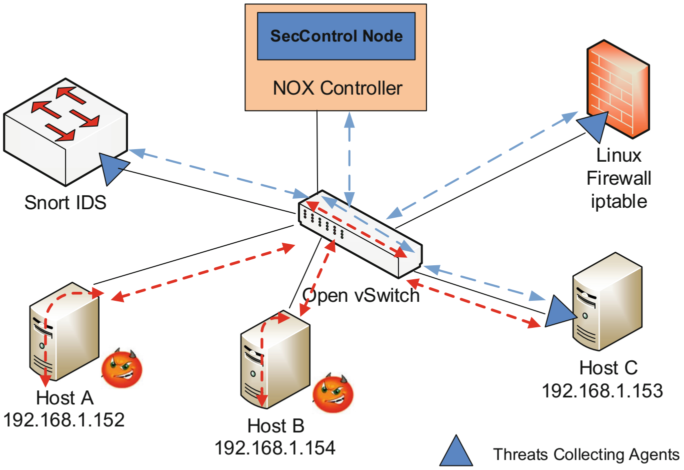


Fig. 4. A SecControl prototype (Color figure online)

and SDN controllers is a little bit different. Except for distributing SDN rules, it is also used to synchronize network device information. Because it is related to device information update on SDN controller, it should be extended with existing SDN protocols. Similarly, the communication among SecControl Nodes can be implemented like any typical network application over TCP/IP.

4 A SecControl Prototype

We develop a prototype of the SecControl framework. For a proof of concept purpose, we implement both SecControl Node and SDN controller together. We chose to modify and extend an open source SDN controller, NOX [16], to finish all the related functions. Our implementation includes all the necessary functions for the SecControl components and is able to show the effectiveness of SecControl protections.

The SecControl Node is implemented on NOX version 0.9.0 with OpenFlow v1.0. NOX is an open source OpenFlow controller in C++/Python, which can be used to manage OpenFlow switches. We implemented the Threats Analyzer in Python and SDN Rule Engine in C++. The Threat Analyzer module is running as an OpenFlow application on NOX, while the SDN Rule Engine is inserted as an extension of NOX. We modified the built-in functions, `send_openflow_command` and `install_datapath_flow`, of NOX to implement the SDN Rule Distributer.

We pick three most used security tools for a demonstration purpose. They are Snort IDS, Linux iptables, and Linux system logs. Snort IDS is a popular open source IDS; Linux iptables is a kernel-supported firewall tool on Linux system; Linux system logs are native log system of Linux system which is often used for audit purposes. Each tool is attached a customized ThreatCA. Because the three tools use different alert formats, we implement three different ThreatCAs to collect security threat information. Besides, to simplify the protection, we categorize

```
<IDMEF-Message version="1.0">
<Alert ident="abc123456789">
  <Analyzer analyzerid="analyzer1">
    <Node category="dns">
      <location>HTTP Server</location>
      <name>host.domain.org</name>
    </Node>
  </Analyzer>

  <CreateTime ntpstamp="0xbc72b2b4.0x00000000">
    2020-05-19T15:31:00-08:00
  </CreateTime>

  <Source ident="abc01">
    <Node ident="abc01-01">
      <Address ident="abc01-02" category="ipv4-addr">
        <address>192.168.1.100</address>
      </Address>
    </Node>
  </Source>

  <Target ident="vic01">
    <Node ident="vic01-01" category="dns">
    <name>www.example.com</name>
      <Address ident="vic01-02" category="ipv4-addr">
        <address>192.168.1.50</address>
      </Address>
    </Node>
    <Service ident="vic01-03">
      <portlist>1-1024</portlist>
    </Service>
  </Target>

  <Classification origin="vendor-specific">
    <name>portscan</name>
    <url>http://www.vendor.com/portscan</url>
  </Classification>
</Alert>
</IDMEF-Message>
```

Fig. 5. A scan detection in IDEMF.

security events into attack events and suspicious events. The attack events should be reacted with a defense response instantly, while suspicious events need further analysis before deciding a defense response. When a ThreatCA meets an attack event, it just tags the event and sent it to Threat Analyzer to get an instant defense response. For the suspicious events, the ThreatCA extracts the critical information of the events and put them in a unified format, Intrusion Detection Exchange Message Format (IDEMF) [2]. IDEMF provides a unified format and structure that allows the security detection results can be transferred among different parties. A scan detection involving three nodes can be demonstrated in IDEMF as shown in Fig. 5.

The collected IDEMF messages are stored in a local DB for further analysis. If a defense response is determined, it will be translated into OpenFlow flow rules. In OpenFlow, each flow rule will have a set of attributes, such as *match field*, *counter*, *timeout*, *actions*, and so on, to match network flows. The *actions* field contains an action set, which indicates the operations to be executed for the matched network traffic. To enforce the SDN primitives at the OpenFlow switches, we translate the five SDN primitives into compatible OpenFlow actions. Figure 6 shows `generateOFactions()` function translating five SDN primitives to the OpenFlow flow rule actions. Finally, the new flow rules are sent to switch through function `install_datapath_flow (self, dp_id, attrs, idle_timeout, hard_timeout, actions, buffer_id, priority, inport, packet)`.

```
FlowAction generateOFActions(defenseResponse){
    FlowAction flowaction;
    switch (defenseResponse) {
        case drop:
            addAction(flowaction,drop);
        case forward:
            addAction(flowaction,forward);
        case reflect:
            addAction(flowaction,reflect);
        case isolate:
            addAction(flowaction,isolate);
        case copy:
            addAction(flowaction,copy);
    }
    return flowaction;
}
```

Fig. 6. Translate five SDN primitives into OpenFlow flow actions

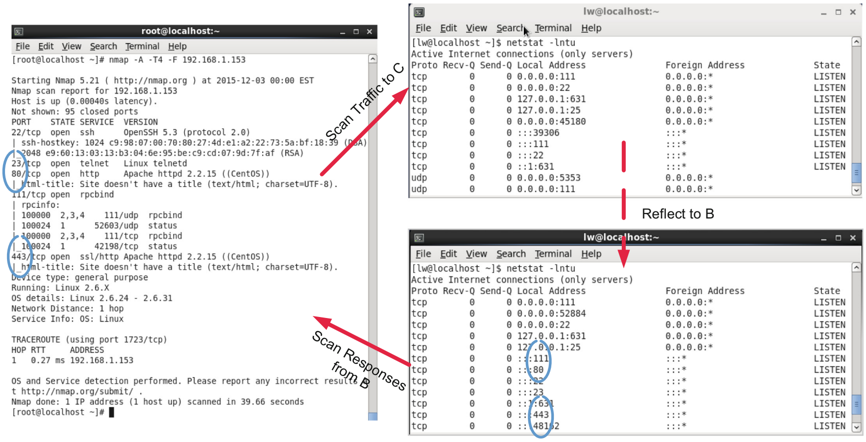
5 Prototype Evaluation

In this section, we evaluate the SecControl prototype with respect to effectiveness and extendibility. The evaluation testbed is deployed as shown in Fig. 4. It is running on a desktop with an Intel Core i7-3370 3.4 GHz processor and 16 GB RAM. We use KVM, Open vSwitch [1,29], NOX [16], Linux firewall iptables, Snort IDS, and Linux built-in log system to construct a SecControl protected virtual network. The evaluation environment is built on a virtual network 192.168.1.0/24. The physical machine is running CentOS 6.0 with kernel 2.6.32 and qemu-kvm-0.15.1 for virtualization. The three hosts are running as guest OSes with CentOS 6.0 as well. As can be seen in Fig. 4, all the nodes are in a virtual network and connected by an Open vSwitch. We have security tools, Snort 2.9.7.5, iptables 1.4.7, and Linux Syslog systems, running at host machine. Each security tool is attached with a Threat Collecting Agent (each blue triangle in Fig. 4 stands for a ThreatCA), and the ThreatCAs are communicating with the SecControl Node through the virtual network.

5.1 Effectiveness

We demonstrate the effectiveness of the SecControl framework with several security threats, regular scan threat, and payloads specific attacks. As Fig. 4 shows, host A, and host B are attacking machines, and host C is the victim machine (for some attacking scenarios, we may deploy more attacker nodes). We use attacking machines to send out attack traffic to the victim machine.

Regular Scan Threat. Regular network scan is typically conducted by a single attacker to locate easy targets in an open network environment, like a public network. In our network environment, we assume an attacker owns host A 192.168.1.152, and he wants to sniff the network status of host C 192.168.1.153. We configure Snort with a scan detection rule: `alert tcp any any -> $HOME_NET any (msg: "TCP SYN"; flow: stateless; flags:S;`


root@localhost:~
File Edit View Search Terminal Help
lw@localhost:~
File Edit View Search Terminal Help
lw@localhost:~
File Edit View Search Terminal Help
Scan Traffic to C
Reflect to B
Scan Responses from B

```
[root@localhost ~]# nmap -A -T4 -F 192.168.1.153

Starting Nmap 5.21 ( http://nmap.org ) at 2015-12-03 00:00 EST
Nmap scan report for 192.168.1.153
Host is up (0.00040s latency).
Not shown: 95 closed ports
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 5.3 (protocol 2.0)
| ssh-hostkey: 1024 c9:98:07:00:70:80:27:4d:e1:a2:22:73:5a:bf:18:39 (DSA)
|_2048 e9:60:13:03:13:b3:04:6e:95:be:c9:cd:07:9d:7f:af (RSA)
23/tcp  open  telnet   Linux telnetd
80/tcp  open  http     Apache httpd 2.2.15 ((CentOS))
|_html-title: Site doesn't have a title (text/html; charset=UTF-8).
111/tcp open  rpcbind
| rpcinfo:
| 100000  2,3,4    111/udp  rpcbind
| 100024  1      52603/udp  status
| 100000  2,3,4    111/tcp  rpcbind
|_100024  1      42198/tcp  status
443/tcp open  ssl/http Apache httpd 2.2.15 ((CentOS))
|_html-title: Site doesn't have a title (text/html; charset=UTF-8).
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.24 - 2.6.31
Network Distance: 1 hop
Service Info: OS: Linux

TRACEROUTE (using port 1723/tcp)
HOP RTT     ADDRESS
1   0.27 ms 192.168.1.153

OS and Service detection performed. Please report any incorrect results a
t http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 39.66 seconds
[root@localhost ~]# 
```

```
[lw@localhost ~]$ netstat -lntu
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address               Foreign Address             State
tcp        0      0 0.0.0.0:111                 0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:22                  0.0.0.0:*                   LISTEN
tcp        0      0 127.0.0.1:631               0.0.0.0:*                   LISTEN
tcp        0      0 127.0.0.1:25                0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:45180               0.0.0.0:*                   LISTEN
tcp        0      0 :::39306                    :::*                        LISTEN
tcp        0      0 :::111                      :::*                        LISTEN
tcp        0      0 :::22                       :::*                        LISTEN
tcp        0      0 ::1:631                     :::*                        LISTEN
udp        0      0 0.0.0.0:5353                0.0.0.0:*
udp        0      0 0.0.0.0:111                 0.0.0.0:*
```

```
[lw@localhost ~]$ netstat -lntu
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address               Foreign Address             State
tcp        0      0 0.0.0.0:111                 0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:52884               0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:22                  0.0.0.0:*                   LISTEN
tcp        0      0 127.0.0.1:631               0.0.0.0:*                   LISTEN
tcp        0      0 127.0.0.1:25                0.0.0.0:*                   LISTEN
tcp        0      0 :::111                      :::*                        LISTEN
tcp        0      0 :::80                       :::*                        LISTEN
tcp        0      0 :::22                       :::*                        LISTEN
tcp        0      0 :::23                       :::*                        LISTEN
tcp        0      0 ::1:631                     :::*                        LISTEN
tcp        0      0 :::443                      :::*                        LISTEN
tcp        0      0 :::48162                    :::*                        LISTEN
```

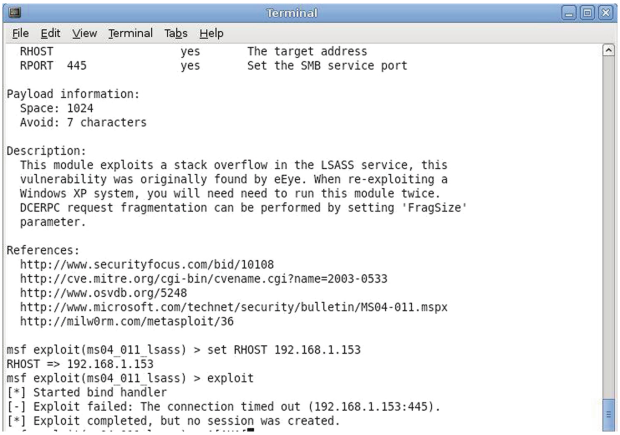
Fig. 7. A simple network scan

`detection_filter:track by_dst, count 100, seconds 5; sid:1000001; rev:1)`. We tag the detected scan threats as attack events and configure primitive *reflect* as default defense response to a scan threat. All the scan traffic for host C will be reflected to host B 192.168.1.154. We open port 22, 23, 25, 80, 111, and 443 on B, and 22, 25, and 111 on C. Figure 7 shows the reflecting process, from which we can see the scan results are from host B instead of host C. That is, the scan traffic is successfully reflected to B.

An Attack with Specific Payloads. When an attacker knows a specific vulnerability of a target machine, he can attack the target machine by sending a well-designed exploit. The attacking exploit sent through network packets is called malicious payloads. Malicious payloads can help the attacker take over the victim machine and gain an absolute control over it. We install an old Windows 2000 OS on host C 192.168.1.153 and open the vulnerable service SMB on port 445, which holds a dangerous vulnerability through which an attacker can easily obtain a remote shell with admin privileges. We configure the Snort to match the signature of the attacking payload `windows/vncinject/bind_tcp`. We choose *block* as the default defense response if any malicious payload is matched. Correspondingly, the *block* defense response is translated to primitive *drop* on the controller. We use host A 192.168.1.152 as the attacking machine. The attacking payload is sent with `metasploit`, a penetrating test tool. Figure 8 shows the `metasploit` console window. The result shows the exploit fails due to a connection timeout, which proves we successfully block the attacking traffic to host C.

5.2 Extendibility

We demonstrate the extendibility of the SecControl framework by using different security analysis principles. We use time-based threat correlation and target-



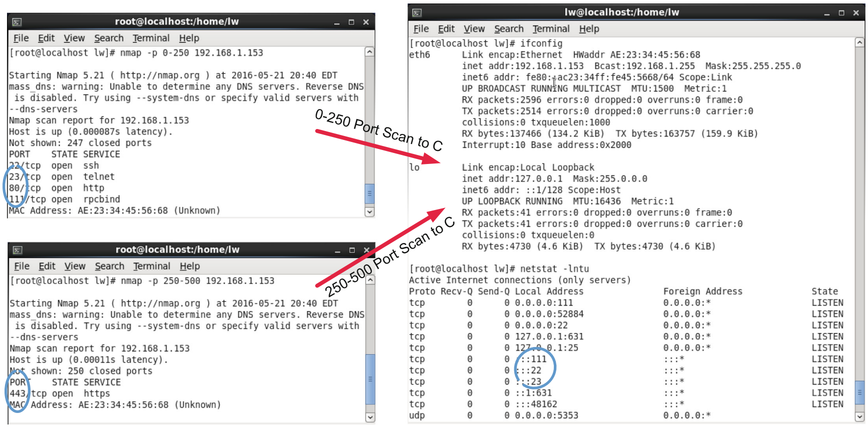
```
  RHOST                     yes       The target address
  RPORT  445                yes       Set the SMB service port

Payload information:
  Space: 1024
  Avoid: 7 characters

Description:
  This module exploits a stack overflow in the LSASS service, this
  vulnerability was originally found by eEye. When re-exploiting a
  Windows XP system, you will need need to run this module twice.
  DCERPC request fragmentation can be performed by setting 'FragSize'
  parameter.

References:
  http://www.securityfocus.com/bid/10108
  http://cve.mitre.org/cgi-bin/cvename.cgi?name=2003-0533
  http://www.osvdb.org/5248
  http://www.microsoft.com/technet/security/bulletin/MS04-011.mspx
  http://milw0rm.com/metasploit/36

msf exploit(ms04_011_lsass) > set RHOST 192.168.1.153
RHOST => 192.168.1.153
msf exploit(ms04_011_lsass) > exploit
[*] Started bind handler
[-] Exploit failed: The connection timed out (192.168.1.153:445).
[*] Exploit completed, but no session was created.
```

Fig. 8. An attack with specific payloads

based threat correlation to identify several advanced attacks, which usually may not be easily detected by existing security tools. And, we show the scalability of the SecControl framework by deploying multiple SecControl instances over different SDN networks, and our results show different SecControl instances can cooperate to offer protections across SDN networks.

Distributed Scan Threat. Distributed Scan is an advanced and hidden network scan, which is achieved by multiple scanning sources. Smart attackers can take multiple attacking sources to start a distributed scan, in order to bypass existing security tools. In this attacking scenario, we use host A and host B to start a distributed port scan on host C. Our target port range is 0–500. Host C is opening port 22, 25, and 111, while host D has port 22, 25, 80, 111, and 443 open (we add one more host D as a honeypot to communicated with the reflected scan traffic. Host D share the same configuration with host B). We choose *redirect* defense response to deal with the distributed scan, and it is translated to *reflect* primitive. To detect the distributed scan threat, we extend the security analysis process of Threat Analyzer by following the target-based threat correlation principle. We configure Snort to record all the traffic. Figure 9 shows the results of distributed scan. From the scan result of A and B, we can see the port 80 and 443 is open, which shows D is the real scanned node and the distributed scan traffic is successfully reflected to D.

Step-Stone Attack. Step-stone attack is another advanced attack [6]. To reduce the risks of being detected, attackers choose to start an attack on step-stone nodes instead of his own machine. Step-stone nodes are immediate nodes taken by attackers. Through step-stone nodes, an attacker can get more accesses or conveniences in taking over the target node. Following the time-based threat correlation principle, we design a two step-stones attack detection algorithm.



```
[root@localhost lw]# nmap -p 0-250 192.168.1.153

Starting Nmap 5.21 ( http://nmap.org ) at 2016-05-21 20:40 EDT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS
 is disabled. Try using --system-dns or specify valid servers with
--dns-servers
Nmap scan report for 192.168.1.153
Host is up (0.000087s latency).
Not shown: 247 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
23/tcp  open  telnet
80/tcp  open  http
111/tcp open  rpcbind
MAC Address: AE:23:34:45:56:68 (Unknown)
```

```
[root@localhost lw]# nmap -p 250-500 192.168.1.153

Starting Nmap 5.21 ( http://nmap.org ) at 2016-05-21 20:40 EDT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS
 is disabled. Try using --system-dns or specify valid servers with
--dns-servers
Nmap scan report for 192.168.1.153
Host is up (0.00011s latency).
Not shown: 250 closed ports
PORT    STATE SERVICE
443/tcp open  https
MAC Address: AE:23:34:45:56:68 (Unknown)
```

```
[root@localhost lw]# ifconfig
eth6      Link encap:Ethernet  HWaddr AE:23:34:45:56:68
          inet addr:192.168.1.153  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::ac23:34ff:fe45:5668/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2596 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2514 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:137466 (134.2 KiB)  TX bytes:163757 (159.9 KiB)
          Interrupt:10 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:41 errors:0 dropped:0 overruns:0 frame:0
          TX packets:41 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:4730 (4.6 KiB)  TX bytes:4730 (4.6 KiB)

[root@localhost lw]# netstat -lntu
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address               Foreign Address             State
tcp        0      0 0.0.0.0:111                 0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:52884               0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:22                  0.0.0.0:*                   LISTEN
tcp        0      0 127.0.0.1:631               0.0.0.0:*                   LISTEN
tcp        0      0 127.0.0.1:25                0.0.0.0:*                   LISTEN
tcp        0      0 :::111                      :::*                        LISTEN
tcp        0      0 :::22                       :::*                        LISTEN
tcp        0      0 :::23                       :::*                        LISTEN
tcp        0      0 ::1:631                     :::*                        LISTEN
tcp        0      0 :::48162                    :::*                        LISTEN
udp        0      0 0.0.0.0:5353                0.0.0.0:*
```

Fig. 9. A distributed network scan

We use *redirect* and *block* as the defense response for the step-stone attack. In our defense, the attacker node will be blocked, and the step-stone node will be redirected to a honeypot. We use host A as the attacker's machine and host B as the step-stone to attack host C. As the attacking side on host A, we first open and login a shell remotely on host B, then we use B as a step-stone to send malicious payloads to host C. We record all the outside connections of host B, including the connection between A and B. We configure Snort to record the SSH connections between A and B. The remote login attempt is recorded in the system log of host B. Targeted by the detection algorithm, the SSH traffic is tagged as attack traffic. The results show SecControl detected the step-stone attack and the host B's traffic is successfully reflected to the honeypot node.

Cooperations Among SecControl Nodes. We show the scalability of the SecControl framework with multiple SecControl deployments. We use two physical machines, and each physical machine is deployed with one SecControl instance. Two SecControl frameworks are running in two different virtual networks. We configure routing information of two virtual networks so that they can communicate with each other. In our evaluation, we manually send a set of OpenFlow rules from one SecControl Node to the other, and the result shows the other SecControl Node can successfully receive and enforce the OpenFlow rules. However, there could be an information inconsistency problem when we have more different SecControl Nodes. In order to send SDN rules to the proper SecControl Node, every SecControl Node should have a full picture of all other SecControl Nodes' network positions and their network device lists. A lot of algorithms studied in distributed computing can be borrowed and used in this scenario. Consider this is not the focus of this paper, we will not elaborate further on this.

5.3 Overhead

SecControl is a practical network security solution aiming to provide a comprehensive protection for SDN networks. Since SecControl uses different strategies and algorithms to deal with different security threats, we can hardly find a unified method to evaluate its overall performance. We evaluate the time interval between a SecControl flow rule leaves NOX and the flow takes effect in the network. For the *forward* primitive, the time interval is 7.542 ms; for the *drop* primitive, the time interval is 13.152 ms; for the *reflect* primitive, the time interval is 17.684 ms. Besides, consider our evaluation testbed is deployed on one physical machine and all the involved nodes share the same set of physical resources, we should be able to shorten the time interval value if it is conducted on a more powerful machine.

6 Discussion

We discuss some limitations of the SecControl framework in this section. First, SecControl may have a delay reaction issue when providing defense responses. This is a common issue for many monitor-based security tools for there is always a delay between threat detection and defense reaction. Also, the network efficiency may affect the protection effect of SecControl. Consider security events are transmitted over network between the Threat Collecting Agents and the SecControl Node, the network transmission efficiency can affect SecControl's protection effect. In some protection scenarios, security engineers may require an instant response on a detected threat. A possible way to alleviate this issue is to build an exclusive network channel between the Threat Collecting Agents and SecControl Node. Further, to improve the performance of security event collecting, we may design built-in threat collecting interfaces on security tools.

Second, SecControl relies on existing security tools to gather security events and generate defense responses. We may face an accuracy issue because the accuracy of the security threat information is not exactly guaranteed. Almost all mainstream security solutions follow a detection-based protection policy, and the protection is affected by detection accuracy. Consider the current detection algorithms are not perfect, the detection results may suffer false positive and false negative issues. Therefore, SecControl may produce inaccurate defense responses. One possible solution is to manually record the real attacks and pick up corresponding defense responses. We believe a lot of further research can be done on this issue.

Third, consider the SecControl framework relies on a distributed architecture, it may suffer all possible issues that can happen in a distributed network environment. For example, a potential issue is the single failure problem. If the SecControl Node is down, our protection will be discontinued. In fact, single failure and all other related issues have been well researched in the distributed system field. We can just take whatever comes to our protection scenarios and adopt these solutions.

7 Related Work

Security Incident and Event Management (SIEM) [26,28] is a set of technologies which are used to gather, analyze and present information from network and security devices. SIEM is designed to collect security-related information from all kinds of devices and applications such as firewalls, IDS, antivirus, and so on. When an attack happens, security engineers will turn to SIEM for a complete record of that attack for security investigations and audits. SIEM mainly focuses on monitoring and tracing purposes. Compared with SecControl, although SIEM is capable of collecting and analyzing security threats, it does not provide interaction interfaces for the latest SDN networks.

SecControl combines traditional security tools and SDN technologies to provide a practical network security solution. For one hand, SecControl makes use of security processing abilities of existing tools; for the other hand, SecControl maximizes the security benefits of taking SDN technologies. Shin et al. propose FRESCO [32], a modular security application development framework for OpenFlow networks. FRESCO provides a fine-grained framework to implement security functions as OpenFlow applications. However, it requires security engineers to reimplement all security functions to fit FRESCO design, which brings a lot of engineering work. Besides, consider FRESCO is implemented at controller side, it is greatly confined by the processing capabilities of the controller. As a result, the security functions requiring complicated computation and analysis can hardly be deployed with FRESCO. AVANT-GUARD [33] aims to improve the data plane performance in order to provide SDN security applications a more scalable and responsive OpenFlow infrastructure. It designs a *connection migrations* mechanism to improve OpenFlow's weak points and protect OpenFlow devices from saturation attacks. However, AVANT-GUARD does not change the fact that the SDN controller could be a potential bottleneck in security applications. Different from FRESCO and AVANT-GUARD, OpenFlow Extension Framework (OFX) [34] modifies the software system of network hardware devices to allow SDN applications dynamically load software modules. OFX achieves a good performance because it is running on switch hardware directly. However, not all security services can provide effective protections on a switch hardware. Compared with existing SDN security innovations, SecControl neither introduces heavy workload to SDN controller nor brings negative effects to existing security tools.

Except for the SDN security application frameworks, researchers also extended the individual security tools in SDN environments. FlowGuard [19] is designed to achieve a firewall running over SDN networks. FlowGuard is capable of checking suspicious network flows and verifying network-wide firewall policies. However, it just provides basic firewall functions and cannot be extended with other security functions. Similarly, some research modifies traditional intrusion detection systems to fit SDN environments. Mehdi et al. [25] suggest using SDN to solve home network security problems. They provide four prominent traffic anomaly detection algorithms to detect security threats on SDN controllers. This innovation provides an example of applying SDN technologies in home network security solution.

Some researchers also try to innovate security functions with Network Function Virtualization (NFV) [4]. Aaron et al. design OpenNF [15], a control plane architecture to enable the reallocation of flows within NF instances. Through OpenNF, network operators are able to create rich control applications, including firewall, NAT, traffic loadbalancer, and so on. OpenBox [7] is designed to decouple the control plane of middleboxes from their data planes and unify the data plane through service instances. It provides a set of interfaces and protocols to communicate with SDN controllers and middleboxes. OpenBox introduces a uniform platform for network admins to design network applications cross SDN network devices and middleboxes. Similarly, these NFV innovations focus on a universal network architecture for general network applications instead of security applications. SecControl can be regarded as a "controller" of the SDN controllers. It releases the security related computation logic from typical SDN controllers that should focus on managing low-level network devices. NOX [16] and POX [24] are two twin open source OpenFlow controllers implemented in C++ and Python respectively. They provide a set of APIs for upper-level network applications to dynamically change the flow tables of OpenFlow switches. However, the current OpenFlow structure is problematic and may meet some issues when deploying in a large scale network. Researchers propose different SDN controller solutions to fit existing controllers into large scale deployments, like HyperFlow [3], Pratyaastha [21], DISCO [30], ElastiCon [13], and ONOS [5]. These methods enhance the existing controllers by adding more supports on scalability, device state synchronization, controller cooperation, fault tolerance, and other functions. Relying on SDN controllers, many network relevant applications have been innovated. Heller et al. [18] propose to reduce the energy consumptions by improving network infrastructures of data centers through centralized SDN controllers. Curtis et al. [11] suggest using SDN controllers to optimize flow management to further achieve a better overall network performance.

8 Conclusion

In this paper, we propose a new network protection framework bridging the gap between existing security tools and SDN technologies, to produce a practical and comprehensive network security solution for SDN environments. SecControl integrates the capabilities of existing security tools and combines SDN controls to obtain an optimized SDN network security solution. We demonstrate the capability of SecControl by implementing a prototype with the OpenFlow protocol and evaluate its effectiveness and performance impacts with common security threats. Our experiments show that SecControl can cooperate with many mainstream security tools and provide effective defense responses over SDN-supported networks.

Acknowledgments. This work was supported in part by The Penn State Fund for Innovation.

References

1. Open vSwitch. http://openvswitch.org/
2. RFC4765. The Intrusion Detection Exchange Message Format (IDEMF). https://www.ietf.org/rfc/rfc4765.txt
3. Balis, B.: HyperFlow: a model of computation, programming approach and enactment engine for complex distributed workflows. Future Comput. Syst. **55**, 147–162 (2016)
4. Batalle, J., Riera, J.F., Escalona, E., Garcia-Espin, J.A.: On the implementation of NFV over an openflow infrastructure: routing function virtualization. In: Future Networks and Services (SDN4FNS), pp. 1–6. IEEE (2013)
5. Berde, P., Gerola, M., Hart, J., Higuchi, Y., Kobayashi, M., Koide, T., Lantz, B., O'Connor, B., Radoslavov, P., Snow, W., Parulkar, G.M.: ONOS: towards an open, distributed SDN OS. In: Proceedings of the Third Workshop on Hot Topics in Software Defined Networking, HotSDN (2014)
6. Blum, A., Song, D., Venkataraman, S.: Detection of interactive stepping stones: algorithms and confidence bounds. In: Jonsson, E., Valdes, A., Almgren, M. (eds.) RAID 2004. LNCS, vol. 3224, pp. 258–277. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30143-1_14
7. Bremler-Barr, A., Harchol, Y., Hay, D.: OpenBox: a software-defined framework for developing, deploying, and managing network functions. In: Proceedings of the 2016 Conference on ACM SIGCOMM. ACM (2016)
8. Casado, M., Freedman, M.J., Pettit, J., Luo, J., McKeown, N., Shenker, S.: Ethane: taking control of the enterprise. In: SIGCOMM Review (2007)
9. Casado, M., Garfinkel, T., Akella, A., Freedman, M.J., Boneh, D., McKeown, N., Shenker, S.: SANE: a protection architecture for enterprise networks. In: Proceedings of the 15th Conference on USENIX Security Symposium, vol. 15 (2006)
10. Cuppens, F., Miège, A.: Alert correlation in a cooperative intrusion detection framework. In: IEEE Symposium on Security and Privacy (2002)
11. Curtis, A.R., Mogul, J.C., Tourrilhes, J., Yalagandula, P., Sharma, P., Banerjee, S.: DevoFlow: scaling flow management for high-performance networks. In: Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (2011)
12. Debar, H., Wespi, A.: Aggregation and correlation of intrusion-detection alerts. In: Lee, W., Mé, L., Wespi, A. (eds.) RAID 2001. LNCS, vol. 2212, pp. 85–103. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45474-8_6
13. Dixit, A.A., Hao, F., Mukherjee, S., Lakshman, T.V., Kompella, R.R.: ElastiCon: an elastic distributed SDN controller. In: Proceedings of the 10th ACM/IEEE Symposium on Architectures for Networking and Communications Systems (2014)
14. Egilmez, H.E., Dane, S.T., Bagci, K.T., Tekalp, A.M.: OpenQoS: an OpenFlow controller design for multimedia delivery with end-to-end quality of service over software-defined networks. In: Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA (2012)
15. Gember-Jacobson, A. Viswanathan, R., Prakash, C., Grandl, R., Khalid, J., Das, S., Akella, A.: OpenNF: enabling innovation in network function control. In: ACM SIGCOMM Computer Communication Review (2015)
16. Gude, N., Koponen, T., Pettit, J., Pfaff, B., Casado, M., McKeown, N., Shenker, S.: NOX: towards an operating system for networks. Comput. Commun. Rev. **38**(3), 105–110 (2008)

17. Handigol, N., Seetharaman, S., Flajslik, M., McKeown, N., Johari, R.: Plug-n-Serve: load-balancing web traffic using OpenFlow. In: ACM SIGCOMM Demo (2009)
18. Heller, B., Seetharaman, S., Mahadevan, P., Yiakoumis, Y., Sharma, P., Banerjee, S., McKeown, N.: ElasticTree: saving energy in data center networks. In: Proceedings of the 7th USENIX Symposium, NSDI (2010)
19. Hu, H., Han, W., Ahn, G., Zhao, Z.: FlowGuard: building robust firewalls for software-defined networks. In: Proceedings of the Third Workshop on Hot Topics in Software Defined Networking, HotSDN (2014)
20. Kim, H., Feamster, N.: Improving network management with software defined networking. IEEE Commun. Mag. **51**(2), 114–119 (2013)
21. Krishnamurthy, A., Chandrabose, S.P., Gember-Jacobson, A.: Pratyaastha: an efficient elastic distributed SDN control plane. In: Proceedings of the Third Workshop on Hot Topics in Software Defined Networking, HotSDN (2014)
22. Lantz, B., Heller, B., McKeown, N.: A network in a laptop: rapid prototyping for software-defined networks. In: Proceedings of the 9th ACM Workshop on Hot Topics in Networks, HotNets (2010)
23. Mahajan, R., Wattenhofer, R.: On consistent updates in software defined networks. In: Twelfth ACM Workshop on Hot Topics in Networks, HotNets-XII (2013)
24. Mccauley, J.: POX: a Python-based OpenFlow controller (2014). http://www.noxrepo.org/pox/about-pox/
25. Mehdi, S.A., Khalid, J., Khayam, S.A.: Revisiting traffic anomaly detection using software defined networking. In: Sommer, R., Balzarotti, D., Maier, G. (eds.) RAID 2011. LNCS, vol. 6961, pp. 161–180. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23644-0_9
26. Miller, D., Harris, S., Harper, A., VanDyke, S., Blask, C.: Security Information and Event Management (SIEM) Implementation. McGraw Hill Professional, New York (2010)
27. Mukherjee, B., Heberlein, L.T., Levitt, K.N.: Network intrusion detection. IEEE Netw. **8**, 26–41 (1994)
28. Nicolett, M., Kavanagh, K.M.: Magic quadrant for security information and event management. Gartner RAS Core Reasearch Note, May 2009 (2011)
29. Pfaff, B., Pettit, J., Amidon, K., Casado, M., Koponen, T., Shenker, S.: Extending networking into the virtualization layer. In: Eight ACM Workshop on Hot Topics in Networks (HotNets-VIII) (2009)
30. Phemius, K., Bouet, M., Leguay, J.: DISCO: distributed multi-domain SDN controllers. In: IEEE Network Operations and Management Symposium (2014)
31. Porras, P., Shin, S., Yegneswaran, V., Fong, M., Tyson, M., Gu, G.: A security enforcement kernel for OpenFlow networks. In: Proceedings of the First Workshop on Hot Topics in Software Defined Networks. ACM (2012)
32. Shin, S., Porras, P.A., Yegneswaran, V., Fong, M.W., Gu, G., Tyson, M.: FRESCO: modular composable security services for software-defined networks. In: 20th Annual Network and Distributed System Security Symposium, NDSS (2013)
33. Shin, S., Yegneswaran, V., Porras, P.A., Gu, G.: AVANT-GUARD: scalable and vigilant switch flow management in software-defined networks. In: ACM SIGSAC Conference on Computer and Communications Security, CCS (2013)
34. Sonchack, J., Aviv, A.J., Keller, E., Smith, J.M.: Enabling practical software-defined networking security applications with OFX. In: 23th Annual Network and Distributed System Security Symposium, NDSS (2013)

35. Wang, R., Butnariu, D., Rexford, J.: OpenFlow-based server load balancing gone wild. In: USENIX Workshop on Hot Topics in Management of Internet Cloud, and Enterprise Networks and Services, Hot-ICE (2011)
36. Yin, H., Liu, X., Min, G., Lin, C.: Content delivery networks: a bridge between emerging applications and future IP networks. IEEE Netw. **24**(4), 52–56 (2010)

An Effective Approach for Dealing with the Pressure to Compromise Security During Systems Development

Yeslam Al-Saggaf(✉)

School of Computing and Mathematics, Charles Sturt University,
Boorooma Street, Wagga Wagga, NSW 2678, Australia
yalsaggaf@csu.edu.au.com

Abstract. This study looks into (1) the frequency with which Australian IT professionals compromise security to meet deadlines; (2) the causes of unprofessional behavior in the IT work place; (3) the best approach for tackling unprofessional behavior; and the effectiveness of this approach. These issues were addressed using a mixed research methodology that involved three data collection stages with the input of each stage being the output of the earlier stage. In the first stage, we conducted a survey of 2,315 Australian IT professionals which the Australian Computer Society helped promote. In the second stage, we interviewed 43 Australian IT professionals from six different Australian state capitals to understand the causes of unprofessional behavior in the IT work place and the best approach for tackling unprofessional behavior. Following the research participants' suggestions, I implemented the approach suggested by the majority of participants. I then shared the links of the approach I implemented with the Australian IT professionals via the Australian Computer Society. In the final stage, I interviewed 28 IT professionals to receive their feedback with regards to the effectiveness of this approach in enhancing young IT professionals' abilities to recognize unprofessional behavior. This paper presents the results from the three stages of this study.

Keywords: Compromising security · Australian organizations
Systems development · IT professionals · Professional ethics

1 Introduction

The aim of this paper is to report findings relating to the commonness of compromising security from the viewpoint of Australian IT professionals from a study that investigated unprofessional behavior in the IT work place in Australia more generally. Another aim of this paper is to investigate the causes of unprofessional behavior more generally. Third, to discover the best approach for tackling unprofessional behavior. Fourth to evaluate the effectiveness of this approach.

The study, which is part of a larger project on professionalism in the IT work place, involved collecting data during three phases. In phase one, we conducted an online survey of 2,313 members of the Australian Computer Society (ACS), which we administered using SurveyMonkey. The data from this phase indicated that compromising security is

X. Lin et al. (Eds.): SecureComm 2017 Workshops, LNICST 239, pp. 32–46, 2018.
https://doi.org/10.1007/978-3-319-78816-6_3

one of the top ten unprofessional behaviors that IT professionals commit in the IT work place (Table 2 lists the top ten unprofessional behaviors). In the next step of data analysis, we looked at the characteristics of the survey participants who selected compromising security as one of the unprofessional behaviors. This information is important for understanding the profile of the IT professionals who identified compromising security during systems development as a problem. In phase two we conducted qualitative interviews with 43 IT professionals. We selected the IT professionals we interviewed from the survey participants who indicated their willingness to participate in this phase when completing the survey. The face-to-face conversations with the IT professionals offered valuable clues into the causes of the unprofessional behavior in the IT work place in general and the best approach for tackling unprofessional behavior. In accordance with the interviewees' suggestions, I implemented the approach suggested by the greatest number of participants. I implemented the approach suggested by the majority of participants. I then shared the links of the approach I implemented with the Australian IT professionals via the Australian Computer Society. In the final stage, I interviewed 28 IT professionals to receive their feedback with regards to the effectiveness of this approach in enhancing young IT professionals' abilities to recognize unprofessional behavior.

The paper begins by introducing the research questions. Next, the research methodology and the findings from the quantitative survey are presented. The process of collecting data using semi-structured interviews is discussed next, followed by a discussion of the findings from this qualitative component of the research relating to the causes of unprofessional behavior in general and the most effective approach for tackling unprofessional behavior in the IT work place. This is followed by a discussion of the feedback about the implemented approach. The paper ends with a comparison of the findings from both approaches.

2 Background

Cybercrime costs the Australian economy more than AU$4.5 billion annually [1]. The Australian Crime Commission laments the loss of this money that they say could otherwise be used to fund services, roads, hospitals and schools in Australia [2]. The 2016 Australian Cyber Security Centre Survey found that 90% of the Australian organizations surveyed experienced a cyber security breach or threat during the 2015-16 financial year that compromised the confidentiality, integrity or availability of network data or systems [3]. The above statistics for Australia are consistent with international trends. Juniper Research predicts the cost of security breaches to reach $2.1 trillion globally by 2019 [4]. At the individuals level, more than 46,957, cybercrime incidents have been reported to the Australian Cybercrime Online Reporting Network (ACORN) in 2016 up from 39,491 in 2015 [5]. Between 1 January 2017 and 31 March 2017, 11,775 incidents were reported to ACORN up from 9,679 during the same period in 2015 [5] suggesting that cybercrime incidents are on the rise.

As cyber-criminals and hackers continue to discover and exploit vulnerabilities in information systems, the need for securing information systems has never been greater. The above statistics and Lucas and Weckert [6] study findings that suggested that compromising security to meet deadlines or make things work is a problem facing IT professionals, raise the following question:

RQ1: How often is security compromised to meet deadlines?

There are several reasons behind unprofessional behavior in the IT work place more generally and compromising security during the development of information systems more specifically. One reason could be IT professionals' lack of awareness when it comes to recognising ethical problems in the work place or providing solutions to them especially when two or more priorities are at tension with each other (i.e. the interest of the employer versus the interest of the client etc.) [7, 8]. A study by Lucas and Weckert [6], for example, found that "ethical awareness in the IT profession requires some strengthening" (p. 42) and that "IT professionals do not have a conscious awareness of the ethical notions that are most important in their work" (p. 47). There is evidence that suggests that ethics awareness has led to a higher level of professionalism and ethical behavior among IT employees (see Al-Saggaf and Burmeister [9], Cappel and Windsor [10], and Van den Bergh and Deschoolmeester [11]). Higher levels of professionalism resulted in improvement in the performance of the IT industry and the quality and delivery of IT products and services [12]. This has led scholars to argue for the need to raise awareness of ethical issues among IT professionals [12]. Another reason is IT professionals' selfishness. Cappel and Windsor [10] argue that IT professionals may be tempted to view ethical issues from an egocentric point of view, thereby either oversimplify situations or fail to consider alternatives, stakeholders, consequences, or one's duties. A third reason is pressure. While the IT industry tries to address its shortcomings through the use of rigorous software and application development methodologies, quality assurance initiatives, risk management approaches, and external assessment processes, these largely tend to get ignored when management and personnel are under pressure to perform (see [6, 13, 14]). Lucas and Bowern [15] note that pressure to complete projects on time can make IT professionals compromise ethical standards and policies or even break the law. The second research question is therefore:

RQ2: What are the reasons behind unprofessional behavior in the IT work place?

While many studies focused on ethical decision-making (see, for example, Anderson et al. [16], Al-Saggaf and Burmeister [9], and Fleischmann [17]), few studies focused on how IT professionals actually recognize and solve ethical problems in their workplaces (see Lucas and Bowern [15], and Khanifar et al. [18]). Lucas and Mason [13] conducted a study to determine the ethical attitudes of Australian IT professionals, however, they did not focus on how these professionals identify problems or employ strategies to resolve the ethical dilemmas they face in their day to day work. That said, with the exception of Nielsen [19] and Jamil and Susanto [20], who proposed changing organisational culture as a approach to avoid unprofessional behavior, most of the studies that offered recommendations relating to how to identify and resolve ethical problems were for use by students in the classroom. These include using codes of ethics (see Anderson [16], Burmeister and Weckert [21], and Gotterbarn, [22]); case studies and scenarios (see Ferguson et al. [23] and Maslin et al. [24]); the use of role play (see Johnson [25] and Fleischmann, [17]); the doing ethics technique (see Seach et al. [26]); and critical thinking and argument mapping using Rationale (see Al-Saggaf and Burmeister [9]). The third and fourth research questions are therefore:

RQ3: What is the best approach for tackling unprofessional behavior in the IT work place?

RQ4: How effective is this approach?

3 Methodology and Results

3.1 Stage 1: Survey

Survey Procedure. The first stage of the data collection comprised administering a survey via SurveyMonkey so respondents can complete it electronically. The survey design was informed by the design of Lucas and Weckert's [6] survey study which they conducted in 2006 survey. We invited all recipients of the ACS Information Age to complete the questionnaire by a direct email sent to them by the ACS in 2013. The survey was closed within less than two months when the response rate reached 12.4%. We prefaced the online survey by an ethics information statement which included a description of the study. Questions were both closed and open-ended. This paper reports on only the closed questions.

Sample. 2,315 respondents filled the questionnaire. The average number of years of work experience for the participants was 19 years; however, the average number of years of work experience for the respondents who selected compromising security as one of the common unprofessional behaviors was 20.3 years. Table 1 shows a summary of the demographic information for the respondents overall as well as for those who identified compromising security as one of the common unprofessional behaviors. As can be seen from Table 1, the overall profile of the survey participants is similar to the profile of those who selected compromising security as one of the one of the common unprofessional behaviors.

The Commonness of Compromising Security in the Australian IT Work Place. We asked respondents to the survey to select from among different unprofessional behaviors that they witnessed in their work places. Given we allowed respondents to select more than one answer, we judged multiple response frequency analysis to be suitable for analyzing this question. We also performed cross tabulations to find out if there were variations in answers based on the characteristics of participants. The results from the multiple response frequency analysis and the cross tabulations are shown below.

The multiple response frequency analysis shown that compromising security was ranked tenth in a list of the most common unprofessional behaviors witnessed by IT professionals (n = 611, 26.4%). Table 2 shows the top 10 unprofessional behaviors along with the number of responses and their proportions. This paper focusses on compromising security; thus discussions of other unprofessional behaviors listed in Table 2 are outside the scope of this paper.

Table 1. Demographic characteristics of the survey participants and those who selected compromising security as an ethical issue in the survey.

Demographic information		Survey participants		Participants who selected compromising security	
		N	%	N	%
Gender	Female	356	15.5	84	13.7
	Male	1,940	83.9	524	85.8
	N/A	17	0.7	3	0.5
Age	<35	692	30	166	27.1
	36–45	516	22.3	161	26.4
	46–55	576	25	157	25.7
	>56	524	22.7	126	20.6
	N/A	5	0.2	1	0.2
State	ACT	247	10	72	11.8
	NSW	696	30.4	170	27.8
	NT	27	1.2	6	1
	QLD	279	12.2	71	11.6
	SA	120	5.5	39	6.4
	TAS	42	1.8	16	2.6
	VIC	581	25.4	161	26.4
	WA	218	9.5	60	9.8
	Overseas	80	3.5	13	2.1
	N/A	23	1.0	3	0.5
Occupation	Administrator	134	6.5	49	8
	Consultant	502	24.3	153	25
	Developer	307	14.8	83	13.6
	Education	150	7.3	27	4.4
	Manager	698	33.8	182	29.8
	Technical Support	277	13.3	64	10.5
	Other	215	10.39	51	8.3
	N/A	247	11.9	2	0.3
Geographical location	Capital city	2,069	89.5	550	90.0
	Regional area	215	9.43	57	9.3
	N/A	29	1.3	4	0.7
Job classification	Business owner with employed staff	57	2.7	13	2.1
	Fixed term contractors	251	11.8	76	12.4
	Indefinite contractors	34	1.6	13	2.1
	Permanent full-time	1,388	65.4	406	66.4
	Permanent part-time	90	4.2	18	2.9
	Self-employed	112	5.3	31	5.1
	Temporary full-time	61	2.9	15	2.5
	Temporary part-time	63	3.0	8	1.3
	Volunteer	67	3.2	10	1.6
	Other	121	5.69	19	3.1
	N/A			2	0.3

Table 2. The top 10 unprofessional behaviors witnessed by Australian IT professionals

Ethical problems	Number of survey respondents	
	N	(%)
Compromising quality	1104	47.7
Blaming others for own mistakes	957	41.4
Compromising functionality	846	36.6
Overworking staff	762	32.9
Incompetence	750	32.4
Conflict of interest	682	29.5
Unprofessional behavior	633	27.4
Compromising user requirements	632	27.3
Bullying	630	27.2
Compromising security	611	26.4

We wanted to find out which participants characteristics predict participants' choice of compromising security in the survey. To answer this question, we used generalized linear models (GLMs). The responses to the compromising security question are dichotomous (recorded as a Yes/No), whereas all the demographic variables are categorical. To investigate the relationships between the predictor variables and the dichotomous response variable, we fitted GLMs. We carried out the GLMs using *R* (version 3.0.2). We verified all requirements of this analysis were.

The analysis of deviance shown a significant relationship between participants' selection of compromising security as an unprofessional behavior in the survey and occupation and job classification (see Table 3). No other demographic variables showed evidence of a relationship.

Table 3. The analysis of deviance

Demographic variable	Deviance	Degrees of freedom (DF)	P
Occupation	18.38	6	0.0053
Job classification	32.32	11	0.0007

We also used GLMs to investigate if there is a relationship between the prevalence of unethical conduct and the participants' choice of compromising security. We fitted this technique to examine this relationship since the predictor is also a categorical variable. The analysis of deviance revealed a significant relationship between the prevalence of unethical conduct and participants' choice of compromising security (*Deviance* = 91.23, *df* = 4, *p* = 0.00) suggesting this variable is likewise a predictor for participants' selection of compromising security.

The analysis of deviance also revealed that occupation predicted the choice of compromising security. Twenty-five percent of the participants who selected compromising security as a frequent unprofessional behavior were consultants and 29.8% were managers. In contrast, only 13.6% of the respondents who selected compromising security as an unprofessional conduct were developers. This shows that participants in senior positions are more worried about compromising security than participants in non-senior positions Likewise, we also found job classification to be a predictor of the choice of compromising security. A greater percentage of permanent full time employees (66.4%) and fixed term contractors (12.4%) selected this issue in the questionnaire. The full time permanent employees group is not surprising, but the fact that a large percentage of fixed term contractors selected this problem indicates that fixed term contractors are particularly worried about this issue. A future research study could provide insights with regards to the reasons for this surprising finding.

3.2 Stage 2 and 3: Qualitative Interviews

Conducting the Interviews and Analysing the Data. The survey stage was followed by two stages of semi-structured interviews (43 participants in the second stage and 28 participants in the third stage). Participants were all selected from the respondents of survey who indicated willingness to participate in the future stages of the project. We conducted the second stage interviews in 2014 and the third stage interviews in 2017. We conducted sixty-six interviews (43 interviews in the second stage and 23 interviews in the third stage) were face-to-face and took place in the six Australian state capitals. We conducted the remaining five interviews (third stage) via Skype. We audio recorded and transcribed verbatim all interviews.

We sent the invitation for participation to all survey respondents who indicated that they were willing to take part in interviews during the project's stage one. We selected the interviewees based on their characteristics and ensure that a diverse range of backgrounds were represented. The final list of participants included IT professionals from a diverse range of organizations, such as large and small and who come from all Australian state capitals and represent different ages, genders, kinds of jobs, and work experiences. Table 4 lists the characteristics of these individuals.

We analyzed the transcribed interviews using qualitative thematic analysis with the help of NVivo. We used each transcribed interview document as the unit of analysis. We performed data analysis as follows. (1) We read the interview documents several times. (2) We created nodes based on keywords and dominant phrases in the transcribed documents. (3) We located text within the interview documents with the same nodes and assigned it to these nodes. This way each node acted as a "bucket" in the sense that it held all the data related to a specific node. These nodes were then further divided into specific sub-nodes. This was done to create a hierarchy so it is easier to interpret the findings.

Table 4. Characteristics of the interviewees

Interviewees' characteristics		N
Gender	Female	12
	Male	59
Age	<35	5
	36–45	10
	46–55	26
	>56	26
Occupation	Accreditor	1
	Business analyst	5
	Consultant	12
	Database developer/coordinator	2
	Manager	19
	IT educator	4
	Retired	6
	Other	22
City	Adelaide	9
	Brisbane	8
	Canberra	8
	Melbourne	12
	Sydney	17
	Perth	12
	Skype	5
Job classification	Fixed term contractors	19
	Permanent full-time	52
IT work experience (years)	10–19	9
	20–29	15
	30–39	24
	>40	23

The Causes of Unprofessional Behavior. While the findings from the 43 interviews revealed 25 reasons behind unprofessional behavior in the IT work place only the reasons brought up during interviews by the highest number of interviewees, in this case 18 out of the 43, will be discussed below. Two reasons met this condition: bad management and pressure. Other reasons include greed, lack of respect for IT people, poor communication skills, self-interest, IT project's complexity, fear of losing job and lack of awareness, to name a few. These findings are consistent with the literature pertaining to the main reasons behind unprofessional behavior in the IT work place (see Background section above) specifically with regards to lack of awareness, self-interest and pressure.

Eighteen out of 43 participants identified bad management as one of the causes of unprofessional behavior in the IT work place. The following quote typifies their views:

> What leads to unprofessional behavior therefore is probably a poor management structure and a set of values that aren't clearly defined or at least not communicated yeah.

The 18 interviewees who raised bad management during interviews expressed a range of views about this issue including the view that when management engages in unprofessional behavior, unprofessional behavior trickles down to staff:

> I have seen where a team leader or a manager perhaps behaves in a certain way, you'll see that behavior reflected through his organization and that's not necessarily helpful.

One interviewee agreed maintaining that ethical behavior has *"got to come from management down"*, a view which a third interviewee also shared:

> it gets driven from the top. So if you've got a leader who, so a Chief Executive who acts like that, who then, the directors follow that because he's acting in that way, they all follow in that same way. Then you have the managers who also act because the CEO's demonstrates in a behavior, the directors are, the managers are, and then because the manager is the staff also believe that that's the way.

Both of the above quotes emphasise the importance of *"leading by example"* in reducing unprofessional behavior in the IT work place.

Similarly, 18 out of 43 participants identified pressure as one of the causes of unprofessional behavior in the IT work place. The 18 interviewees who raised this issue during interviews reported several examples of pressure facing IT professionals. There is pressure on project managers to provide inaccurate estimations of costs of projects:

> And there's always a pressure on there, well if I, if I made an estimate based on what I actually think it's going to take, how much … it going to take me cost, I probably wouldn't succeed. So there's, some pressure either to make it seem smaller than it really is in order to get any money at all or the other one which says I applied some weird factor.

There is pressure on salesmen to sell unwanted products and services because they are paid leveraged salaries:

> when you're sitting on a leveraged salary that's 5545 somewhere along the line something's got to give and if you're not having a good quarter, the following quarter you've got to have a good quarter otherwise you won't keep the kids. So it's a dilemma and more and more organizations are heading towards leverage state and that drives the sales behavior.

There is pressure on program managers to cut corners to secure the next contract:

> Early delivery thing where you're getting pressure from the people above to do something and the thing wasn't the pressure to do something quicker because we're doing that all the time. As a program manager you've got all these things to sort of hit but when it's the reason for doing it is because then they might get another contract, to me that was the ethical question.

It clear from the above examples that financial gain underpins all these pressures.

The Best Approach for Tackling Unprofessional Behavior. While the findings from the 43 interviews revealed 21 approaches for tackling unprofessional behavior in the IT work place only the approaches brought up during interviews by the highest number of interviewees, in this case 26 out of the 43, will be highlighted below. One approach met this condition: case studies. 26 out of 43 participants suggested the use of case studies as an effective approach for tackling unprofessional behavior in the IT work place. The literature has also identified case studies as an effective approach for tackling unprofessional behavior in the IT work place (see Background section above). Other approaches suggested include a mentoring program for young IT professionals and a 'helpline' through which young IT professionals can receive counselling.

One reason the highest number of interviewees suggested the use of case studies is because case studies can enable IT professionals to *"be in someone else's shoes"* as one interviewee argued *"Putting yourself in the shoes of one of those other stakeholders is a key."* Case studies can enable IT professionals to ask themselves: what would I do in such a situation?:

> Every now and again there's an article in the Information Age which has case studies, I enjoy case studies…. I always read them and think "Oh yeah, what would I do?"

Case studies can enable IT professionals to learn from other people's mistakes: *"In IT one can learn very well by example"* because, according to this interviewee *"we study what's come before."* But one interviewee warned:

> Scenarios shouldn't be black and white… You should know when you're in the dark grey area cause that's the problem area, that's the stuff that we need to fix.

Black and white scenarios are straightforward and thus they are not helpful. They need to be grey so individuals can relate them to their circumstances: *"I'm likely to go in on the test cases looking for the closest of what's happening to me."*

Case studies can enable IT professionals to consider a situation from multiple perspectives:

> If you can provide somebody in a situation where they're trying to make a decision, both perspectives, you'll do, like that would be really valuable.

Case studies can also enable IT professionals to consider the risks*: "we're trying to give them an understanding of some of the risks that are out there. Some real examples."* Other interviewees also shared this interviewee's suggestion regarding the use of real examples because, as another interviewee explained, *"People relate to real scenarios."*

The Effectiveness of the Implemented Approach: Interactive YouTube Videos. In response to the interviewees' recommendation regarding the use of case studies as an effective approach for tackling unprofessional behavior in the IT work place, we developed four interactive YouTube videos highlighting cases of unprofessional practice. One of the videos specifically addresses the conduct of compromising security due to pressure from above. The video, which is titled "Early Launch", shows a situation in which a project manager is put under pressure to compromise the security of a system along with three short action videos that highlight the options to tackle the behavior and the potential outcome of each option. These videos enable IT professionals to choose options and then see the outcome of their selections. The objective of the three action videos is to enable the IT professionals watching the video to question themselves what they would do in such a scenario by selecting an action for tackling the behavior and then see the outcome of that action.

Following the development of the videos, we uploaded these videos on the ACS YouTube Channel. Next, we interviewed 28 IT professionals to provide feedback on the effectiveness of these videos in improving IT professionals' ability to recognize unethical conduct at work. The majority of interviewees agreed that these interactive

YouTube videos and their outcome videos are valuable to have. The following comments summarize their views:

- *I like the approach of the scenarios and that sort of thing*
- *Well I think sometimes role play is not bad which is what this video does*
- *I liked the examples that you had in your video*
- *Well I think sometimes role play is not bad which is what this video does*
- *Some aspects of the video were good*
- *At the higher level I think where the ethics really, within organizations, where the ethics really starts to be an issue and that's why I like in that last, the video on early, the early [launch]*
- *I think it's fantastic. I think it's a really good way to create an interactive resource, especially for a DE student. Because there's a lot of resources for face to face, there's classes, you can ask questions. But with these videos, anytime, anywhere, you can constantly review it*
- *I like the approach of the scenarios and that sort of thing*
- *Yeah, it's good. I think it's a good idea. People can – you know, that can trigger people's memories, you know, and they say, oh yeah, I'll come across with you. This is something that happened here, happened there, and it's good*
- *Yes. I thought the encryption one was good. You made a clear point. It probably hit a nerve because there's been so much problems recently with the ABS and those sort of things*
- *I thought the people in the videos were very believable*
- *But I think more often than not the rules aren't there so therefore they'll do like you said in the video, they'll respond at the time to what happens*
- *But what I thought was I thought the production qualities were excellent. I thought the acting was excellent*

Similarly, the following comments that these interviewees made about the approach of selecting possible actions then see the outcome of their selections typify their views:

- *I think that that's, that's what's attractive about it*
- *I liked it a lot, the, as I say, the only negative I have was the extreme anger in the two options-*
- *I think the training videos themselves, that ability to choose an action based on a set of circumstances I think is very important because that's what makes people pay attention to what's going on. If you just go and stick a video in front of them they're not going to take it is as much as something that's interactive;*
- *Yeah that was okay, what would you do in effect, or what – I think you were saying if you were in that position what would you do?*
- *I think they're – I mean I think they're good options*
- *I think it was good that they were able to see the outcomes*
- *Yeah I think that's excellent, that's what I think, but people – especially young people because they can't necessarily imagine that"*
- *It's good. Yeah, it's a good idea; Oh I actually – the way they played out was absolutely real world*

- *I think it's useful to get, to think through that there are consequences to decisions. There are consequences to people's lives, careers, products, projects, and just cutting code. There is much more to life and in IT and health IT than cutting code*

The above quotes shows that the interviewees thought the approach of selecting possible actions then see the outcome of their selections is effective. Further analysis will be conducted on the data from this final stage to find out how the videos can assist IT professionals with recognizing unethical conduct in the IT work place.

4 Discussion

The first research question addressed the prevalence of compromising security during system development within Australian organizations. The survey we conducted with a large sample of IT professionals showed that compromising security is one of the top ten unprofessional behaviors witnessed by Australian IT professionals in their work places. The analysis revealed a significant relationship between participants' choice of compromising security and occupation and job classification. Twenty-five percent of the respondents who identified compromising security as a frequent unprofessional behavior were consultants and 29.8% were managers. In contrast, only 13.6% of the respondents who selected compromising security as an unprofessional behavior were developers. This shows that respondents in senior positions are more worried about compromising security than respondents in non-senior positions. Interestingly 13.2% of the survey respondents who chose compromising security and 13.9% of the interviewees who brought up this issue in interviews classified their jobs as fixed term contractors. That fixed term contractors are more worried about compromising security than the permanent professionals is worthy of further investigation. Why these IT professionals ordered compromising security in the survey as a frequent unethical conduct and brought up this concern during interviews. Could it be because those external contractors are more worried about compromising security than the internal staff? Additional research is needed to shed light on this issue. It is hoped, this paper will inspire undertaking such an inquiry.

Qualitative interviews were conducted over two periods of time to address the second, third and fourth research questions. The second research question was concerned with the causes of unprofessional behavior in the IT work place. The qualitative analysis from the first round of qualitative interviews identified bad management and pressure as the top two in the list of the causes of unprofessional behavior in the IT work place. In terms of bad management, the qualitative analysis revealed that ethical behavior *"gets driven from the top"* and unprofessional behavior trickles down to staff. With regards to pressure, which the literature has also identified as a main reason for unprofessional behavior, the qualitative analysis revealed that financial gain underpinned all the kinds of pressures reported by the interviewees in this study. The third research question was concerned with the best approach for tackling unprofessional behavior in the IT work place. Twenty six out of 43 participants suggested the use of case studies as an effective approach for tackling unprofessional behavior in the IT work place. The literature has also identified case studies as an effective approach. Several reasons were highlighted for why the highest number of interviewees suggested the use of case studies. The main reason however was because case studies can enable IT professionals to 'be in someone else's shoes.' Having implemented the suggested approach, the fourth research question was concerned with the effectiveness of the implemented approach (the interactive YouTube videos). The findings from the second round of qualitative interviews revealed that the majority of interviewees agreed that these interactive YouTube videos and their outcome videos are valuable to have and that enabling the viewers to make choices and then see how these choices play out is a good idea.

5 Conclusion

The aim of this study was to address the following research questions: how often is security compromised to meet deadlines? What are the causes of unprofessional behavior in the IT work place? What is an effective approach for tackling unprofessional conduct? To what extent this approach is effective? To fulfil the aims of the project we employed a mixed methodology comprising three stages of data gathering the input of each stage being the output of the earlier stage. The data collection proceeded as follows. In the first stage, we conducted a survey of 2,315 Australian IT professionals which the Australian Computer Society helped promote. The survey revealed that compromising security is one of the top ten most frequently witnessed unprofessional conducts. Based on the findings from this stage we rewrote the content of the follow-up interviews. In the second stage, we interviewed 43 Australian IT professionals from six different Australian state capitals to learn from them, the causes of unprofessional behavior in the IT work place and the best approach for tackling unprofessional behavior. The first round of qualitative interviews identified bad management and pressure as the top two in the list of the causes of unprofessional behavior in the IT work place. According to these interviews also, the highest number of interviewees suggested the use of case studies as an effective approach for tackling unprofessional behavior. In accordance with the interviewees' recommendations, I implemented the approach suggested by the majority of participants. I then shared the links of the approach I implemented with the Australian IT professionals via the Australian Computer Society. In the final stage, I interviewed 28 IT professionals to hear their comment with regards to the effectiveness of this approach in enhancing young IT professionals' abilities to recognize unprofessional behavior. The first impressions from the second round of qualitative interviews with regards to the effectiveness of this approach were all positive.

Acknowledgments. This project was supported by an Australian Research Council Linkage grant (LP130100808). The industry partner in this project was the Australian Computer Society. Professor John Weckert, A/Prof Oliver Burmeister and Mr John Ridge were the other team investigators. The author wishes to thank A/Prof Oliver Burmeister for his significant contribution to the project in general and specifically for helping with the distribution of the survey, conducting some of the qualitative interviews and his suggestions with regards to the analysis of the qualitative data. The author also wishes to thank Rachel MacCulloch (Charles Sturt University) for her suggestions.

References

1. Acumen Insurance Brokers. http://acumeninsurance.com.au/2017/03/14/cybercrime-costs-the-australian-economy-over-4-5-billion-annually-and-is-now-in-the-top-5-risks-faced-by-businesses/
2. Australian Crime Commission. https://www.acic.gov.au/sites/g/files/net1491/f/2016/06/oca2015.pdf
3. Australian Cyber Security Centre. https://www.acsc.gov.au/publications/ACSC_Cyber_Security_Survey_2016.pdf
4. Juniper Research. https://www.juniperresearch.com/press/press-releases/cybercrime-cost-businesses-over-2trillion
5. Australian Cybercrime Online Reporting Network. https://www.acorn.gov.au/resources
6. Lucas, R., Weckert, J.: Regulation in the IT industry. Centre for Applied Philosophy and Public Ethics, Canberra (2008)
7. Sherratt, D., Rogerson, S., Fairweather, B.: The challenge of raising ethical awareness: a case-based aiding system for use by computing and IT students. Sci. Eng. Ethics **11**(2), 299–315 (2005)
8. Jung, I.: Ethical judgments and behaviors: applying a multidimensional ethics scale to measuring IT ethics of college students. Comput. Educ. **53**(3), 940–949 (2009)
9. Al-Saggaf, Y., Burmeister, O.K.: Improving skill development: an exploratory study comparing a philosophical and an applied ethical analysis technique. J. Comput. Sci. Educ. **22**(3), 1–19 (2012)
10. Cappel, J.J., Windsor, J.C.: A comparative investigation of ethical decision making: Information systems professionals versus students. Database Adv. Inf. Syst. **29**(2), 20–34 (1998)
11. Van den Bergh, J., Deschoolmeester, D.: Ethical decision making in IT: discussing the impact of an ethical code of conduct. Commun. IBIMA, 1–10 (2010)
12. McLaughlin, S., Sherry, M., Carcary, M., O'Brien, C.: e-Skills and IT Professionalism: Fostering the IT Profession in Europe. Final report. Maynooth, Innovation Value Institute, National University of Ireland (2012)
13. Lucas, R., Mason, N.: A survey of ethics and regulation within the IT industry in Australia: ethics education. J. Inf. Commun. Ethics Soc. **6**(4), 349–363 (2008)
14. Ethics Resource Center. http://www.ethics.org/ecihome/research/nbes/nbes-reports/nbes-2013
15. Lucas, R., Bowern, M.: Ethics survey: haste sours quality in IT. Information Age, June/July 2007, pp. 28–30 (2007)
16. Anderson, R.E., Johnson, D.G., Gotterbarn, D., Perrolle, J.: Using the new ACM code of ethics in decision making. Commun. ACM **36**(1), 98–107 (1993)
17. Fleischmann, K.R.: Preaching what we practice: teaching ethical decision-making to computer security professionals. In: Sion, R., Curtmola, R., Dietrich, S., Kiayias, A., Miret, J.M., Sako, K., Sebé, F. (eds.) FC 2010. LNCS, vol. 6054, pp. 197–202. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14992-4_18
18. Khanifar, H., Jandaghi, G., Bordbar, H.: The professional and applied ethics constituents of IT specialist and users. Eur. J. Soc. Sci. **27**(2–4), 546–552 (2012)
19. Nielsen, R.P.: Changing unprofessional organizational behavior. Acad. Manag. Exec. **3**(2), 123–130 (1989)
20. Jamil, N., Susanto, E.: Preventing unprofessional behavior of firms' managers through shame as a corporate culture. J. US-China Public Adm. **6**(7), 57–64 (2009)

21. Burmeister, O.K., Weckert, J.: Applying the new software engineering code of ethics to usability engineering: a study of 4 cases. J. Inf. Commun. Ethics Soc. **3**(3), 119–132 (2003)
22. Gotterbarn, D., Miller, K.: The public is the priority: making decisions using the software engineering code of ethics. IEEE Comput. **42**(6), 66–73 (2009)
23. Ferguson, S., Salmond, R., Al-Saggaf, Y., Bowern, M., Weckert, J.: The use of case studies in professional codes of ethics: the relevance of the ACS experience to ALIA's code of ethics. Aust. Libr. J. **54**(3), 299–308 (2005)
24. Maslin, M., Zuraini, I., Ramlah, H., Norshidah, M.: An ethical assessment of computer ethics using scenario approach. Int. J. Electron. Commer. Stud. **1**(1), 25–36 (2010)
25. Johnson, J.: Teaching ethics to science students: challenges and a strategy. In: Rappert, B. (ed.) Education and Ethics in the Life Sciences: Strengthening the Prohibition of Biological Weapons, pp. 197–213. ANU E Press, Canberra (2010)
26. Seach, G.R., Cattaneo, M., Burmeister, O.K.: Teaching ethics to IT practitioners. In: 6th International Conference of the Australian Institute of Computer Ethics, Burwood, Victoria (2012)

Securing Websites Against Homograph Attacks

Jemal Abawajy[1](✉), A. Richard[2], and Zaher Al Aghbari[2]

[1] Faculty of Science and Technology, School of Information Technology,
Deakin University, Melbourne, Australia
jemal@deakin.edu.au
[2] Department of Computer Science, College of Sciences,
University of Sharjah, Sharjah, UAE
zaher@sharjah.ac.ae

Abstract. With the globalisation of the Internet, standard frameworks such as the Internationalized Domain Name (IDN) that enable everyone to code a domain name in their native language or script has emerged. While IDN enabled coding the domain names in different languages, it has also put users of web browsers that support IDNs at risk of homograph attacks. As IDN-based homograph attacks have recently become a significant threat in content-based attacks such as phishing and other fraudulent attacks against Internet users, an approach that could automatically thwart such attacks against web browsers is important to the Internet users. To this end, we propose a new approach to mitigate the Internationalised Domain Name homograph attacks in this paper. The proposed approach is very easy to deploy in the existing browsers and requires no change in the way the end-user interact with the web-browsers. We implemented the proposed approach as an add-on to a popular web-browser and demonstrate its effectiveness against the homograph attack. Our assessment of the proposed implementation shows that the proposed solution to the IDN-based homograph attack protects web browsers with no noticeable overhead.

Keywords: Internationalized Domain Name · Homograph attacks
Phishing attacks · Unicode attack · Homograph obfuscation
Web browsers security

1 Introduction

The development of IDN marks the departure from the English-centric web service to a web service globalization to meet the needs of the potential users' worldwide. An IDN is an Internet domain name that allows Internet users to create and use websites in many different languages. By globalizing offering of their services, companies such as PayPal Holdings that operate businesses worldwide stand to benefit from internationalized usability of the Web through the potential increase of customer base. The use of IDN is a trend that will only increase in both public and private organizations worldwide. However, with the introduction of IDNs came a slew of new security concerns, chief among them being the homograph attack [10]. Although the homograph attack is not new, it has recently resurfaced with the introduction of IDN [1] and

X. Lin et al. (Eds.): SecureComm 2017 Workshops, LNICST 239, pp. 47–59, 2018.
https://doi.org/10.1007/978-3-319-78816-6_4

has increasingly become one of the serious Web security problems [2]. With the availability and increased usage of Unicode-based web documents and non-ASCII codes in the domain name, Unicode-based homograph attacks are expected to be a severe web security problem [1, 5]. In particular, the classic scams involving identify theft, fraud, and corruption are anticipated to increase both in number and complexity.

With the increasing internationalization of the Internet, it is more important than ever to provide automated protection against IDN-based homograph attacks for the stability of the Internet. Generally, existing solutions place the burden on the end-users by requiring them to be vigilant about the attack. For example, the Unicode Consortium has been active at raising awareness of the Unicode-based homograph attacks and in providing recommended solutions [13]. While it is true that the users should be aware about the threat of homograph attack, unfortunately we cannot expect them to be vigilant all the time whether Uniform Resource Locator (URL) is legitimate or spoofed via a homograph attack. Existing techniques do not actively attempt to determine whether or not the IDN is a homograph attack. This is left to the end user to figure out which one is genuine and which one is fake. In this paper, we propose a new IDN-based homograph attack mitigation technique. The proposed technique takes the address and determines whether or not it is a spoof of another site. Because it is able to distinguish between legitimate and spoofing sites it allows the user to visit all IDNs safely without being restricted. In addition to this, the strategy proposed here alerts the user with a warning message when they visit a homograph site, which is something the currently implemented mitigation techniques do not do. In summary, this paper makes the following contributions:

- Evaluation of the existing web browsers defense mechanisms against IDN-based homograph attacks;
- A new effective mitigation technique that detects IDN-based homograph attacks and properly notify the users; and
- Implementation and evaluation of the proposed IDN-based homograph attack mitigation technique.

The reset of the paper is organized as follows. In Sect. 2, a brief background on IDN and Unicode as well as how phishers exploit them via homograph attacks to gain sensitive information from unwary users is described. In Sect. 3, the current defenses in place to stop these Unicode-based homograph attacks are discussed. In Sect. 4, a new Unicode-based attack mitigation technique is described. The implementation of the proposed Unicode-based attack mitigation technique into a Google Chrome and performance analysis are described in Sect. 4. The conclusion is given in Sect. 5.

2 Background

In this section, a brief background on IDN and Unicode as well as how phishers exploit them via homograph attacks to gain sensitive information from unwary users is described.

2.1 Internationalized Domain Name

IDN system uses Unicode as opposed to standard English ASCII characters. Unicode is a computer industry standard code that assigns unique numbers to languages in active use today. This allows using different scripts and languages in software, predominantly for applications in web links, web pages, and emails [4]. Unicode was created to replace the American Standard Code for Information Interchange (ASCII). Due to the globalization of Internet and wide penetration of information technology worldwide, ASCII become no longer sufficient. Unicode changed that by including all the characters from every writing system in the world, both current and ancient, as well as symbols and punctuation. Currently, Unicode has over 100,000 characters. Each character in Unicode has a unique number, regardless of the platform or languages, making the different languages and scripts compatible for information interchange. Because IDN is implemented on the application level, no changes are needed to the Domain Name System (DNS) protocol. All of the work is done at the application level by the browser. However, Unicode character set contain visually and semantically confusable characters, which can lead to a myriad of security risks. Specifically, it can aid in creating a bogus domain name chief of which is IDN-based homograph attack.

2.2 IDN-Based Homograph Attack

Homograph attack [9], also called Unicode attacks, visual spoofing, and homograph obfuscation [2, 4], is the type of spoofing attack where a fake website domain name that deceptively looks like a genuine one is created by substituting one or more similar but different characters in the legitimate website domain names. An example is the 'microsoft.com' and 'microsoft.com'. In the second domain name, the second 'o' character is replaced by the Greek omicron 'o' character. The two URLs look identical. While it is impossible for a user to tell them apart visually, the computer will treat them as two totally different characters. This makes homograph popular targets for malicious users to use for phishing attacks and risky for regular users.

IDN-based homograph attack is an example of web browser security risks originating from the exploitation of the Unicode character sets to create a fake IDN. Unicode-based homograph attack is usually made possible by substituting one or more characters of the legitimate website address with their equivalent homographs in the Unicode character set. Many of these characters look similar if not identical which can lead to new and hard to detect phishing attacks. These attacks come in a variety of forms, such as mixed and whole script attacks, as well as character and word level similarity attacks.

Given 100,000 characters (many of which visually indistinguishable) contained in the Unicode, the potential for creating phony URLs that can fool even the most security savvy users is great. This attack vector can enable crafting of counterfeit websites commonly used by phishers to maliciously target unwary users in an attempt to deceive them into handing over their sensitive information such as their credit card details and causing billions of dollars of damage [6]. Moreover, these Unicode-based homograph attacks are potentially more dangerous than regular phishing attacks as they are harder to detect visually for the user. As the popularity of IDNs rises, so does the possibility of these attacks. As the Web usage extends beyond the sole Latin script, this type of Web security risk is expected to increase significantly [1].

2.3 Problem Overview

The basic tenet of the attacks is to make the unaware target of the attack believe that she is using a legitimate Web site when in fact she is accessing a fake Web site that deceptively looks like a genuine one. This is attained by making the fake Web site address virtually indistinguishable from the real URL visually by using different characters from various alphabets such as Cyrillic and the Greek as well as ASCII alphanumeric characters (e.g., Zero stands for the letter "o"). Since the Unicode supports around 100,000 characters, the attackers can exploit resembling characters in various combinations.

We formally define the problem of homograph attacks detection as follows: *Let* $\mathcal{W} = \{w_1, w_2, \cdots, w_n\}$ *be a set of n legitimate Web sites in a DNS database such that each Web site* $w_i \in \mathcal{W}$ *has a domain name,* $DN(w) = \{c_1 c_2 \cdots c_L\}$, *of L characters derived from the standard ASCII code. Let* $\mathscr{w}$ *be a Web site with a domain name* $DN(\mathscr{w}) = \{c_1 c_2 \cdots c_L\}$ *such that at least m characters of the domain name are from* Unicode *(e.g., Cyrillic, Greek, Latin or the mix of these scripts) such that* $1 \leq m \leq L$. *We want to know if* $\mathscr{w}$ *is a legitimate or a fake Web site.*

A wide variety of approaches have been proposed to address the homograph attack. Existing approaches can be generally classified as algorithmic analysis of the characters in the URL or user-oriented security approaches. Our approach combines both algorithmic analysis of the characters and user-oriented security approaches. A suite of user-oriented security approaches that aim to draw the attention of non-ASCII Web sites browsers is discussed in [1]. These approaches include visual security indicators such as enlarging font sizes and highlighting confusing letters. Although these approaches can provide users with visual clues about the possible threats, they cannot prevent the IDN-based homograph attack. An approach based on Unicode string coloring in which each language/script is displayed uniquely in color is discussed in [4]. A recent study on domain name highlighting effectiveness concluded that it is unreliable as the lone mitigation technique [7]. Moreover, they can render genuine Web sites with mixed scripts to appear suspicious to users and useless with color blind users [9]. Punycode by Unicode Consortium [14] is the common strategy used by the existing web browsers. The idea is to convert non-ASCII characters in IDN into basic ASCII characters (a–z, 0–9). When an address is converted to Punycode, a special marker 'xn--'is used to denote that the address is in Punycode format. For example, when the IDN 'paypăl.com' with ă from the Cyrillic set is encountered, the web browser will convert the address into Punycode format as 'http://xn--paypl-tof.com/'.

Existing web browsers seem to tackle the subject of IDNs differently. Google Chrome and Internet Explorer will convert the address to Punycode if it contains mixed scripts. This may rule out legitimate mixed-script addresses, and miss any single script homograph attacks. Opera and Firefox use a whitelist of top level domains, and converts any address not in these domains to Punycode. While this helps to ensure that most of the addresses not converted to Punycode are legitimate, many legitimate sites may be converted to Punycode just because they don't belong to certain top level domains. Safari renders only problematic character sets, such as Greek and Cyrillic, as Punycode. This can lead to every legitimate Greek and Cyrillic IDN being converted to Punycode, greatly restricting the sites the user navigates to.

As user-oriented security approaches require more attentiveness from humans [6], there is a necessity for an automated analysis of the URL to prevent IDN-based mitigation attacks. An approach that extracts and verifies different terms of a URL using search engine spelling recommendation for automated phishing web site detection is discussed in [3]. A Bayesian-based approach that determines if a Unicode character in a word is be o detect whether a suspicious Unicode character in a word is visual spoofing or not is discussed in [5]. Helfrich and Dual [10] described an approach called a dual canonicalization for detecting if two encodings are homographs. The idea is based on homograph sets and deciding if two encodings are either belong to the same homograph set, or else that they belong to different homograph sets.

In spite of intense research to mitigate the problem, the IDN-based homograph attacks still occur. A recent case that highlights the homograph attack problem is the fake 'IIoydsbank.co.uk' domain complete with a high level of HTTPS and a valid TLS certificate [12]. The forged domain names easily fooled many users into trusting it as the legitimate banking website. This example shows that even a TLS certificate is just as easy to obtain a valid certificate for the forged website. Although there is a significant potential for abusing homographs, it is simply wrong to assume that all homographs are malicious or spoofing as it is commonly done in the web browsers today [5].

Moreover, the currently implemented defenses to fight against these homograph attacks are subpar. There are three main problems with the current techniques. First, the browser will still always display the page, whether the address has been converted to Punycode or not, it relies solely on the user noticing that the address is in Punycode and understanding what that means. The second problem is that none of the existing approaches gave the user a proper notification warning them of the potentially dangerous site they were about to visit. Third, none of these techniques can be sure that the site is actually a homograph attack, which can restrict the user from visiting many legitimate sites. The proposed approach aims to remedy these issues. It will achieve this by taking the site address and checking to see whether or not it is a spoof of another site. This should be an improvement over current mitigation techniques as it will actually determine between legitimate and illegitimate sites and properly notify the user when they are trying to navigate to a spoofed site. Although the existing approaches have proven successful to some degree in detecting homographs, they are not generic and not comprehensive enough.

3 Homograph Attack Mitigation Strategy

In this section, the proposed IDN-based homograph attack mitigation strategy is described. We will first discuss the proposed algorithm and then provide security analysis of the algorithm.

3.1 Homograph Attack Detection

Algorithm 1 shows the pseudo-code of the proposed strategy. The algorithm maintains a set of homograph characters $H(c)$ for each ASCII character c. For example, character ‘a’ will have $H(a)$ = {α, ά, ä, ă, Ά, …, Ă}. The domain address (URL) is the input to the algorithm and the output is whether or not the URL is spoofed.

Algorithm 1: Homograph Attack Detection (HAD)

```
INPUT: URL
OUTPUT: Warning message
BEGIN
     IF (Punycode (URL) == TRUE) THEN
         URL← Unicode (URL)   //Unicode version URL
     ENDIF
     IF (Unicode (URL) == TRUE) THEN
        URL← ASCII (URL)    //ASCII version URL
     ENDIF
     IF (ASCII (URL) == TRUE) THEN
         $\mathcal{M}$←Lookup (URL)
         IF ($|\mathcal{M}| \geq NULL$) THEN
             Message (Warning)
         ENDIF
     ENDIF
END Algorithm 1
```

The algorithm first checks the type of the URL and take the appropriate action. Specifically, if the URL address is in Punycode, the URL is converted to its corresponding Unicode type. If the URL is in Unicode format, it is converted to ASCII code. The idea is to take the non-ASCII domain address and converted to the ASCII code. In order to convert the domain address to ASCII code, the algorithm checks for visually similar characters in the domain address and convert all Unicode characters to ASCII code. For example if the character ‘ă’ was found in the IDN, it would be removed and replaced with the character ‘a’. This step would result with the original address such as www.paypăl.com being converted to the regular www.paypal.com.

The final step is to take the address generated in the second step and cross-check it in the DNS database (e.g., Google Public DNS, DNS Advantage and Norton Free DNS) to ensure its validity. The idea is that if the site is found to exist, then obviously the original IDN address (e.g., www.paypăl.com) is a homograph attack. In this case, the user is notified with a very clear and concise message that makes the danger obvious to the user. Figure 1 illustrates the proposed homographic attack mitigation technique using “www.paypăl.com”. This strategy, unlike many of the current defenses in place to mitigate homograph attacks, is designed to actively attempt to determine if an IDN is a homograph attack or not. Moreover, it can detect single, mixed and whole script attacks.

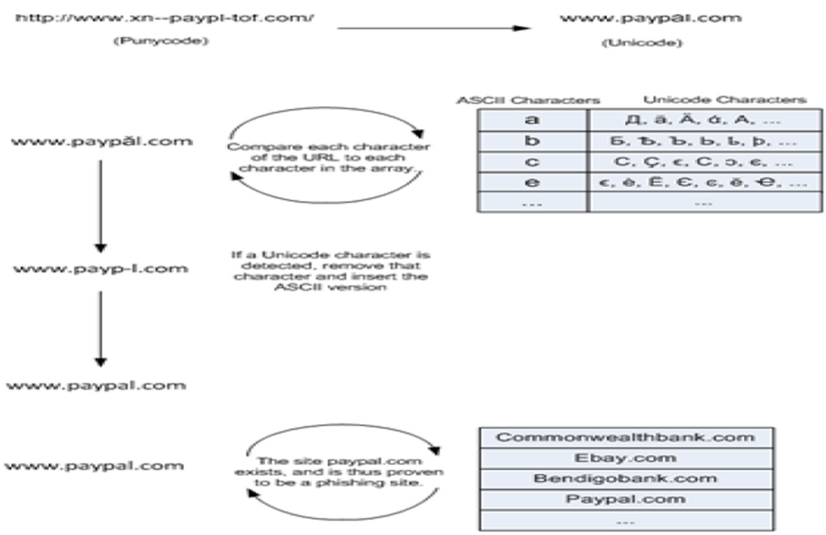


Fig. 1. Illustration of the proposed mitigation strategy.

3.2 Security Analysis

In this section, we conduct the security analysis and show that the proposed mitigation technique detects a variety of homograph attacks, such as single, mixed and whole script attacks. In the analysis, we assume that there is a set of $\mathcal{W} = \{w_1, w_2, \cdots, w_n\}$ legitimate Web sites with the ASCII domain name registered in DNS database. Also, we assume that an adversary has created a spoofed website $\mathcal{w}$ with a URL that visually appears to be a trusted site when in fact it is a malicious one.

Let $S = \{c_1c_2...c_L\}$ be a sequence of L characters representing the domain name of a Web site $\mathcal{w}$. Suppose that $\mathcal{w}$ is a phishing site impersonating a legitimate Web site $w_\mathrm{k} \in \mathcal{W}|1 \le k \le n$ in the DNS. We now show that the proposed algorithm will detect $\mathcal{w}$ as a phishing site.

Lemma 1: The proposed algorithm detects IDN-based homograph attack.

Proof: Suppose that the domain name of $\mathcal{w}$ contains a set of $s \subseteq S$ non-ASCII characters such that $1 \le |s| \le |S|$. Note that when $s = 1$, it is a single character homograph attack. In contrast, the attack is said to be a mixed homograph attack when $1 < |s| < |S|$ and s contains characters from a variety of scripts such as Greek and Latin. The whole script homograph attacks occur when $|s| = |S|$. The proposed algorithm converts the domain name of $\mathcal{w}$, regardless of the type of the domain name (i.e., Punycode or Unicode) and types of the attack (i.e., single, mixed or whole), into the corresponding ASCII code. The lookup function, using the ASCII domain name of $\mathcal{w}$, will return NULL since it does not exist in the DNS thus proving that $\mathcal{w}$ is a phishing site. ■

Lemma 2: The proposed algorithm detects ASCII-based homograph attacks.

Proof: Suppose the string S contains a set of $s \subseteq S$ visually similar ASCII characters as a Web site $w_{\text{k}} \in \mathcal{W}$ such that $1 \leq |s| \leq |S|$. An example of such case is '$w_{\text{k}} =$ Lloydsbank.com.uk' and '$\mathscr{w} =$ IIoydsbank. co. uk' where the first letter 'L' in the legitimate Web site is replaced with a capital 'i' letter 'I'. For $\mathscr{w}$ to be flagged as a phishing site, the following must hold:

$$\mathcal{M} \neq NULL \tag{1}$$

For this case, the algorithm produces $\mathcal{M} = NULL$, which indicates that $\mathscr{w}$ is not in the DNS database. Hence, the algorithm detects that $\mathscr{w}$ is a phishing site. ■

4 Implementation and Testing

In this section, we will test both the defenses currently implemented in a number of web browsers and the mitigation technique proposed in this paper. The proposed homograph attack mitigation strategy is written in Javascript and implemented into the Google Chrome browser as an add-on. The results will be compared in an attempt to analyze the effectiveness of the proposed mitigation technique.

4.1 Methods

A number of tests were devised to test how the current mitigation techniques implemented in browsers respond to IDN-based homograph attack. These tests will cover a range of different kinds of homograph attacks such as mixed-script and whole-script spoofing, and should act to test whether or not the proposed approach can successfully detect them as homograph Web sites. A legitimate address will also be used to see if this add-on can, unlike current mitigation techniques, determine if a mixed-script IDN is legitimate. Table 1 shows the test sites used in the experiments.

The **gooɡle.com** is a spoof of 'google.com', where the second 'g' has been replaced with a Latin small letter 'g' (U+0261). This is a single-script homograph attack. It uses only Latin characters to spoof the google.com site. This test will act to show how the current IDN homograph mitigation techniques in browsers treat IDNs that use an extended Latin script. The **paypӑl.com** is a spoof of 'paypal.com', where the second 'a' has been replaced by the Cyrillic 'ӑ' (U+04D1). This is an example of a mixed-script homograph attack, as it uses both Latin and Cyrillic characters. This test will act to determine how the current mitigation techniques in browsers treat IDN homograph

Table 1. Test sites used in the experiments.

Site	Unicode	Punycode
gooɡle.com	0067 006F 006F **0261** 006C 0065 002E 0063 006F 006D	xn--goole-tmc.com
paypӑl.com	0070 0061 0079 0070 **04D1** 006C 002E 0063 006F 006D	xn--paypl-tof.com
βeta.com	**03B2** 0065 0074 0061 002E 0063 006F 006D	xn--eta-rxc.com
ҽъау.com	**04BD 044A 0430 0443** 002E 0063 006F 006D	xn--80a2bt37b.com

attacks that use mixed scripts. The **βeta.com** is not a spoof as there is no beta.com. This is to represent the possibility of a legitimate site that uses mixed scripts. βeta.com uses the Greek 'β' (U+03B2). This test act to establish how the current mitigation techniques in browsers treat mixed-script IDNs that are legitimate and not homograph attacks. The **ҽъау.com** is a spoof of 'ebay.com', where the 'e' has been replaced by the Cyrillic 'ҽ' (U+04BD), the 'b' has been replaced by the Cyrillic 'ъ' (U+044A), the 'a' has been replaced by the Cyrillic 'а' (U+0430), and the 'y' has been replaced by the Cyrillic 'у' (U+0443). This is an example of a whole-script spoof, where every Latin character has been replaced by a visually similar Cyrillic character. This test will act to determine how current mitigation techniques in browsers treat whole-script homograph attacks.

We tested the Internet Explorer, Google Chrome, Mozilla Firefox, Opera, Safari and Avant. No changes were made to the browsers, they were be used with the default settings. The results of these tests will mainly be looking at whether or not the address was converted to Punycode and whether or not the browser alerted the user that they were visiting a homograph site. We used the following aspects to test each browsers:

- Converts Address to Punycode (#1)
- Makes a visual distinction (#2)
- Notifies the user that the website may be illegitimate (#3).

4.2 Results and Discussions

Table 2 shows the results of the experiments. Google Chrome converts all the addresses to Punycode, even the site βeta.com which isn't a spoofing site. It is impossible to tell legitimate IDNs from fake IDNs. Mixed IDNs are always shown as Punycode, even if they are legitimate sites. The browser makes no attempt to notify the user that the site may be unsafe. It converts the IDN based on the languages that the user has listed. By default, English is the only language listed, and the user can add more. If at least one letter in the IDN is not in the user's languages the IDN is not

Table 2. Results of the experiments.

Browser	goo**g**le.com			payp**ǎ**l.com			**β**eta.com			**ҽъау**.com		
	1	2	3	1	2	3	1	2	3	1	2	3
Chrome	√			√			√			√		
Explorer	√		√	√		√	√		√	√		√
Firefox	√			√			√			√		
Avant			√			√			√			√
Safari	√			√			√			√		
Opera	√			√			√			√		
Ours	√	√	√	√	√	√	√	√	√	√	√	√

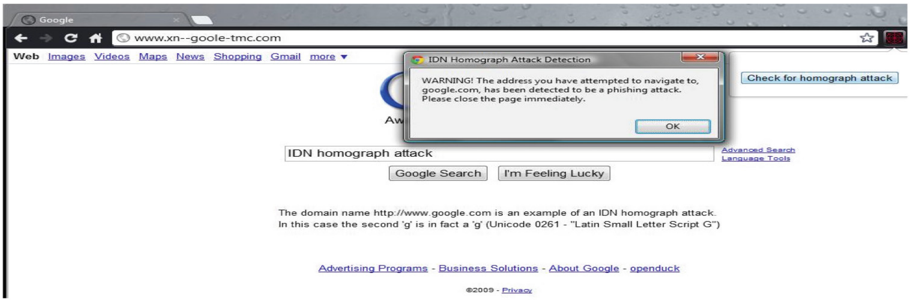
displayed. This is quite restrictive and may result in many legitimate IDNs being converted to Punycode and becoming unreadable to humans.

Internet Explorer converts all the addresses to a Punycode, even the site βeta.com which isn't a spoofing site. This makes it impossible to tell legitimate IDNs from fake IDNs. Mixed IDNs are always shown as Punycode, even if they are legitimate sites. Explorer notifies the user that the web address contains letters that are from a different language. It converts the IDN based on the languages that the user has listed. By default English is the only language listed, and the user can add more. If at least one letter in the IDN is not in the user's languages the IDN is not displayed. In addition to this, Internet Explorer notifies the user that the web address contains letters or symbols that cannot be displayed with the current language settings, and allows the user to change these settings. This is restrictive and may result in many legitimate IDNs being converted to Punycode and unreadable to humans. Mixed IDNs are always shown as Punycode, even if they are legitimate sites. It makes no attempt to notify the user that the site may be unsafe.

Mozilla Firefox loads the pages but converts addresses to Punycode, regardless of what Unicode character set or if they are mixed or single script, unless they meet the whitelist standards. A whitelist of top level domains is used, where the registrars must take care to not allow any homograph-confusable International Domain Names to be registered. Any IDNs that are not from one of these top level domains, even if they are legitimate, will be displayed in Punycode and are therefore not human readable. In addition to the standard tests, a test was done using a top level domain name other than ".com". The address paypăl.info is displayed and treated as if it was legitimate, despite it having a mixed script. This leaves Firefox users open to homograph attacks.

In the Avant browser, none of the sites were converted to Punycode, whether they were legitimate or spoof sites. The browser notifies the user that the web address contains letters or symbols that are not from the users preferred languages, but despite this it displays the site and displays the IDN in Unicode. Clicking on the notification allows the user to add languages to their preferred language list. Avant loads the page and does not convert the address to Punycode. There is no distinction between the real address and fake address. While this may lead to the user visiting phishing sites, they are also able to go to all legitimate IDNs without being restricted. This leaves Avant users open to being the victims of homograph attacks. Mixed IDNs are always shown as Punycode, even if they legitimate sites. No attempts are made to alert the user that the site may be unsafe. Safari uses a list of allowed universal character sets, which by default includes all scripts except Cherokee, Cyrillic and Greek, because these 3 scripts contain characters that are visually similar to many characters in the Latin script. While this can result in many possible homograph sites being ruled out, legitimate IDNs may also be converted to Punycode and thus become unreadable for the user.

Opera converts all mixed script sites to Punycode, even the site βeta.com which isn't a spoofing site. This results in it being impossible to tell legitimate mixed-script IDNs from fake mixed-script IDNs. The browser makes no attempt to notify the user that the site may be unsafe. It loads the pages but converts addresses to Punycode, regardless of what Unicode character set or if they are mixed or single script, unless they meet the whitelist standards. Opera uses a whitelist of top level domains, where

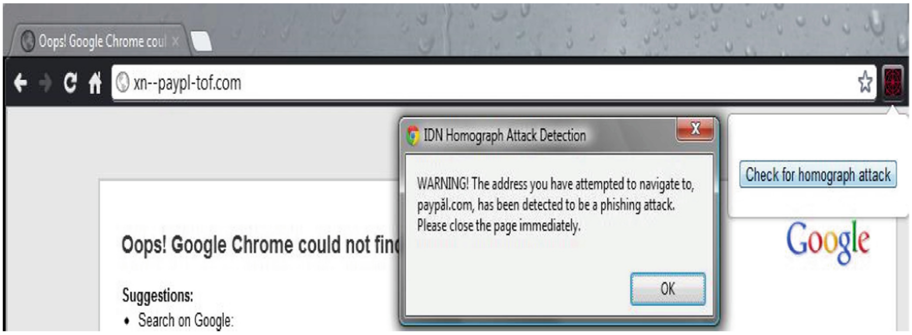


WARNING! The address you have attempted to navigate to, google.com, has been detected to be a phishing attack. Please close the page immediately.

The domain name http://www.google.com is an example of an IDN homograph attack. In this case the second 'g' is in fact a 'g' (Unicode 0261 - "Latin Small Letter Script G")

Fig. 2. Test google.com using the proposed approach.

the registrars must take care to not allow any homograph-confusable International Domain Names to be registered. It is important to note that IDNs that use Latin 1 characters, those being mostly Western European languages, are accepted and displayed by Opera. Of great concern is the fact that the whole-script spoofing site ҽъay.com is treated as legitimate by the Opera browser. This leaves Opera users open to the possibility of being the victims of homograph attacks. This is particularly concerning given how popular Opera is on mobile devices (Figs. 3, 4 and 5).



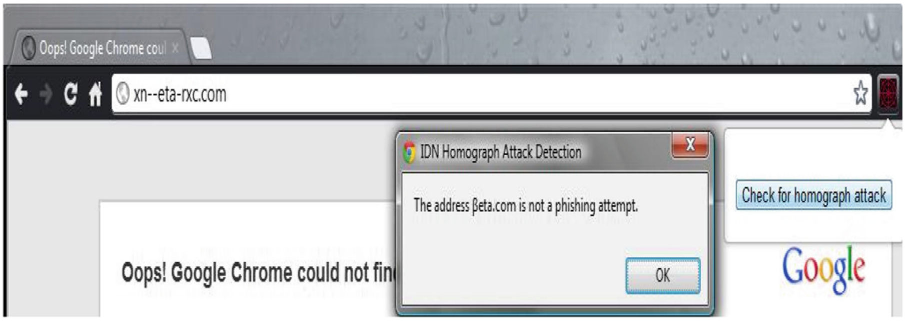
WARNING! The address you have attempted to navigate to, paypăl.com, has been detected to be a phishing attack. Please close the page immediately.

Oops! Google Chrome could not fin

Suggestions:

- Search on Google:

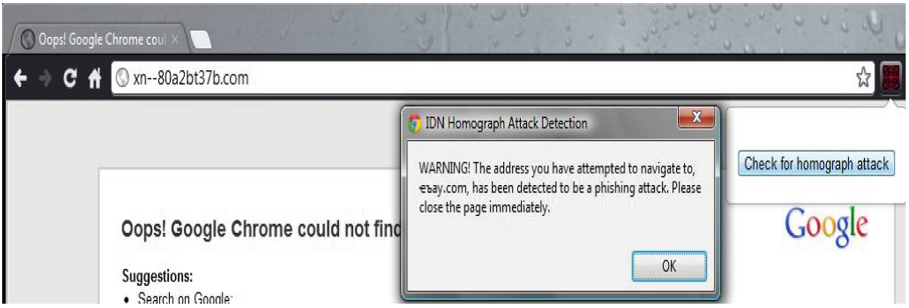
Fig. 3. Test paypăl.com using the proposed approach



The address βeta.com is not a phishing attempt.

Oops! Google Chrome could not fin

Fig. 4. Test βeta.com using the proposed approach.



WARNING! The address you have attempted to navigate to, ҽъay.com, has been detected to be a phishing attack. Please close the page immediately.

Oops! Google Chrome could not find

Suggestions:

- Search on Google

Fig. 5. Test ҽъay.com using the proposed approach

The proposed approach is implemented in Google Chrome as add-on. It was able to properly detect that the two illegitimate mixed-script sites, google.com and paypǎl.com, were homograph attacks. It then warns the user to navigate away from the site as shown in Figs. 1 and 2. The add-on was also able to properly ascertain that the site βeta.com was legitimate despite the fact that it consists of mixed scripts, unlike every other browser. It accomplished this by checking to see if there was a beta.com that the site βeta.com could possibly be spoofing. Since this was not the case it proved that the site βeta.com was legitimate. The add-on properly detected the whole-script spoofing site ҽъay.com as a homograph attack. This is something the Opera and Avant browser were unable to do. Overall the add-on proved effective, successfully determining which of the test sites were homograph attacks and which were legitimate. None of the current homograph attack mitigation techniques enabled in browsers were able to do this. Finally, unlike the other browsers, this add-on was able to properly inform the user that they had navigated to a phishing site and should immediately leave.

5 Conclusion

The introduction of IDN has enabled everyone to code a domain address in their native vernacular. At the same time, IDNs make it easier for criminals to impersonate or spoof web sites by mixing different scripts leading to IDN homograph attacks. In this paper, security risks to various Web due to non-ASCII domain names have been outlined. Generally, existing approaches expect the end users to be more aware of possible threats and proactively inform themselves not falling for the attacks. As security solutions currently in place to mitigate IDN homograph attacks are inadequate, an approach that automatically thwarts homograph attacks is proposed in this paper. The proposed IDN homograph attack mitigation strategy is implemented into the Google Chrome browser as an add-on. The effectiveness of the proposed approach was verified through tests that include single-script, mixed-script and whole-script spoofs, as well as an example of a legitimate mixed script address. Thus, this paper makes significant contribution towards secure browsing experiences for end users.

Acknowledgment. The authors wish to thank Maliha Omar. Without her help, this paper will not be possible to be completed. The authors would also like to extend their appreciation to Deakin University for partially funding this project.

References

1. Al Helou, J., Tilley, S.: Multilingual web sites: internationalized domain name homograph attacks. In: 12th IEEE International Symposium on Web Systems Evolution (WSE), pp. 89–92 (2010)
2. Roshanbin, N., Miller, J.: Finding homoglyphs - a step towards detecting unicode-based visual spoofing attacks. In: Bouguettaya, A., Hauswirth, M., Liu, L. (eds.) WISE 2011. LNCS, vol. 6997, pp. 1–14. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-24434-6_1
3. Maurer, M.-E., Höfer, L.: Sophisticated phishers make more spelling mistakes: using url similarity against phishing. In: Xiang, Y., Lopez, J., Kuo, C.-C.J., Zhou, W. (eds.) CSS 2012. LNCS, vol. 7672, pp. 414–426. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35362-8_31
4. Wenyin, L., Fu, A.Y., Deng, X.: Exposing homograph obfuscation intentions by coloring unicode strings. In: Zhang, Y., Yu, G., Bertino, E., Xu, G. (eds.) APWeb 2008. LNCS, vol. 4976, pp. 275–286. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78849-2_29
5. Qiu, B., Fang, N., Wenyin, L.: Detect visual spoofing in unicode-based text. In: 20th International Conference on Pattern Recognition (ICPR), pp. 1949–1952 (2010)
6. Abawajy, J.: User preference of cyber security awareness delivery methods. J. Behav. Inf. Technol. **33**(3), 236–247 (2014)
7. Lin, E., Greenberg, S., Trotter, E., Ma, D., Aycock, J.: Does domain highlighting help people identify phishing sites? In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 2075–2084 (2011)
8. Canova, G., Volkamer, M., Bergmann, C., Borza, R., Reinheimer, B., Stockhardt, S., Tenberg, R.: Learn to spot phishing URLs with the android nophish app. In: Bishop, M., Miloslavskaya, N., Theocharidou, M. (eds.) WISE 2015. IAICT, vol. 453, pp. 87–100. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-18500-2_8
9. Helfrich, J.N., Neff, R.: Dual canonicalization: an answer to the homograph attack, eCrime Researchers Summit (2012)
10. Baasanjav, U.B.: Linguistic diversity on the internet: Arabic, Chinese and Cyrillic script top-level domain names. Telecommun. Policy **38**(11), 961–969 (2014)
11. Hamid, I.R.A., Abawajy, J.H.: An approach for profiling phishing activities. Comput. Secur. **45**, 27–41 (2014)
12. Cluley, G.: Lloydsbank, IIoydsbank - researcher highlights the homographic phishing problem, 29 June 2015
13. Davis, M., Suignard, M.: Unicode security considerations, Unicode Technical Report #36 (2014). http://unicode.org/reports/tr36/. Accessed 10 Aug 2015

Privacy Threat Analysis of Mobile Social Network Data Publishing

Jemal H. Abawajy[1(✉)], Mohd Izuan Hafez Ninggal[2], Zaher Al Aghbari[3], Abdul Basit Darem[4], and Asma Alhashmi[4]

[1] School of Information Technology, Deakin University, Victoria, Australia
jemal@deakin.edu.au
[2] Department of Computer Science, Universiti Putra Malaysia, Putrajaya, Malaysia
mohdizuan@upm.edu.my
[3] College of Sciences, University of Sharjah, Sharjah, UAE
zaher@sharjah.ac.ae
[4] Department of Computer Science, University of Mysore, Mysore, India
basit.darem@yahoo.com

Abstract. With mobile phones becoming integral part of modern life, the popularity of mobile social networking has tremendously increased over the past few years, bringing with it many benefits but also new trepidations. In particular, privacy issues in mobile social networking has recently become a significant concern. In this paper we present our study on the privacy vulnerability of the mobile social network data publication with emphases on a re-identification and disclosure attacks. We present a new technique for uniquely identifying a targeted individual in the anonymized social network graph and empirically demonstrate the capability of the proposed approach using a very large social network datasets. The results show that the proposed approach can uniquely re-identify a target on anonymized social network data with high success rate.

Keywords: Mobile social network · Social network data publication
Privacy attack · Re-identification attacks · Disclosure attacks

1 Introduction

With hundreds of millions of avid users worldwide, social networking platforms such as Facebook and Twitter have become part and parcel of modern life. Nowadays social networks are largely used by mobile users [1]. Therefore, mobile social networking has become an indispensable source of information and communication medium for people world over. Mobile social networking provides anytime and anywhere proximity-based platform for mobile users to be connected and instantly interact with each other based on mutual interests and backgrounds. The routinization of mobile social networks has radically transformed the way people communicate, socialise and share information.

As mobile phones become indispensable in society, mobile social networks have evolved dramatically over the last few years. The growing integration of mobile social networks with other mobile services such as location-based services have significantly

X. Lin et al. (Eds.): SecureComm 2017 Workshops, LNICST 239, pp. 60–68, 2018.
https://doi.org/10.1007/978-3-319-78816-6_5

increased the amount of user information being generated and collected by the service providers. Unfortunately, mobile social networks may occasionally leak sensitive information [1] and thus privacy concerns has become a fundamental issue [2]. For example, the location information of mobile social network users can be used to track their whereabouts and it can also be disclosed to external service providers with dire consequences. Therefore, with a growing mobile social network platform users, privacy preservation of social network data has taken a centre stage in both industry and academic fields. Although mobile social network privacy has gained tremendous momentum following the recent widespread drive towards coupling mobile social network with other mobile services such as location-based services, much of the exiting work focuses on protecting the privacy of user trajectory (i.e., the protection of the location movement of a user) [3], location information privacy preservation (i.e., securing user's position as well as the time they were there) and profile matching aspects of mobile social network [5].

In this paper we address the privacy issues associated with mobile social network data publication. Although there is a wide variety of mobile social network data use cases in areas such as business, health, science and security, mobile social network user posts potentially reveal much sensitive information about them [1]. However, there is little research regarding the security and privacy concerns associated with mobile social network data collected, collated and published by service providers to enable social network data analysis. As the publication of social network data reasonably threatens user privacy, the mobile social network service providers face fundamental challenges in how to release the data they collect to the interested third party without violating the confidentiality of social networks users personal information. Currently, a variety of anonymisation techniques are used to protect the privacy of individuals in mobile social network data publishing [4].

In this paper we present our work on the privacy vulnerability of the mobile social network data publication with emphases on a re-identification and disclosure attacks. We present a new technique for uniquely identifying a targeted individual in the anonymized social network graph. The proposed approach specifically exploits neighbourhood structures of the online social networks. As friend relationships information are major privacy concerns for mobile online social networks users [7], the proposed approach specifically explores neighbourhood structures of online social networks to breach privacy of the mobile social network users. Our work complements previous work on the privacy preserving social network data publishing [9] and privacy threat analysis of social network data [12] by focusing on privacy vulnerability analysis of mobile social network data. We also focus on specific aspect of attack vector analysis namely the friendship information as this information is considered to be a serious privacy concerns for mobile online social networks users [7]. The capability of the proposed approach is empirically evaluated using a very large social network dataset. The results show that the proposed approach can uniquely re-identify a target on anonymized social network data.

The rest of the paper is structured as follows. In Sect. 2, the models used in the paper are discussed. The proposed attack and its analysis are discussed in Sects. 3 and 4 respectively. The conclusion is given in Sect. 5.

2 Privacy Threat Analysis Framework

Figure 1 shows the online mobile social network (MSN) threat analysis framework with the key actors that include MSN data source (i.e., mobile social media users), MSN data gatherers (i.e., mobile social media service provider), and MSN data explorers (i.e., third party data analysts such as researchers and adversaries).

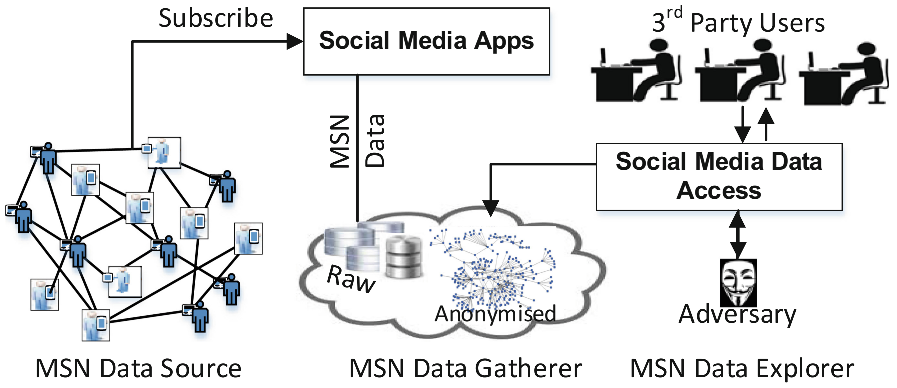


Fig. 1. High level threat analysis framework.

2.1 Mobile Social Networks Data

With the ubiquity of mobile social networks, users of the social network share terabytes of information. The mobile social network users use the social media services primarily to stay connected and interact with family members and friends. They also use the social media to find out the latest information of interest as well as share and contribute to what matters to them using built-in email or instant messaging [2]. The growing integration of mobile social networks with other services such as location-based services have significantly increased the amount of user generated information. As the result, a tremendous amount of user-generated data is collected by social network service providers. In this paper, we use an undirected and unweighted graph $G(V, E)$ to model a social network data, where $V = \{v_1, v_2, \ldots, v_n\}$ is a set of n vertices representing mobile social network users while the social links between the users is captured with a set of edges $E \subseteq V \times V$.

The third party customers have access to the published data for a variety of purposes. Although the user-generated data may include sensitive information such as user shopping habits, the social media data offers many possibilities for data analysis and business intelligence. The information is valuable for third party users such as researchers, business and government agencies to better understand interesting phenomena such as sociological and behavioural aspects of individuals or groups, measure social influence, identify the influential users in mobile social networks, and community structure detection [2]. However, as the collected data often contains sensitive information, network operators may release anonymized and sanitized versions of the complete social network graph or a subgraph to the third party users such as advertisers, marketers, sociologists, epidemiologists, and healthcare professionals.

*Definition 1 (***Anonymized Graph***):* *Let* $G(V,\ E)$ be *the original social network dataset graph. The graph* $\bar{G}\left(\bar{V},\ \bar{E}\right)$ i*s an anonymized version of the original social network graph* $G(V,\ E)$*.*

In this paper, we assume that the social network data graph $G(V,\ E)$ is sanitized into $\bar{G}\left(\bar{V},\ \bar{E}\right)$ before publishing using k-anonymization mechanism with $k = 2$.

2.2 Adversary

An adversary is assumed to have access to the published social networks data. However, unlike the third party consumers, the intent of an adversary is to re-identify certain users in the published social network data. Specifically, an adversary is interested in deriving private information such as the identity of an individual or an attribute value from the anonymized social network graph. The outcome of structural attack depends on the background knowledge that an adversary has. Although existing research with high percentage of successful re-identification commonly assume that the adversary knows large set of structural information regarding the target vertex in the anonymized social network graph, we assume that the adversary knows basic information which is friends and friend of friends.

3 Mobile Social Network Data Vulnerability Analysis

In this section, we examine a class of attack that exploits the friendship information as this information is considered to be a serious privacy concerns for mobile online social networks users [7]. We will first define some background information needed to carry out the attack.

*Definition 2 (***Privacy breach***): Given an anonymized social network data graph* $\bar{G}\left(\bar{V},\ \bar{E}\right)$, *a* privacy breach is said to have taken place when information deemed private and sensitive in the graph is disclosed to unauthorized individuals.

Mobile social network data often contains sensitive information. This data is normally provided to the third party users such as advertisers, marketers, sociologists, epidemiologists, and healthcare researchers. Normally, mobile social network users have strong believe that the mobile social network service providers keep their private information protected [4]. To ensure the privacy of the social network users, the service providers usually anonymize the data prior to publishing it for use by the third party consumers. However, maintaining the privacy of the online mobile social networks users' in published data is an increasingly important challenge facing social network operators [9, 12].

Generally, attacks in this class exploit the neighbourhood structure of a pair of connected vertices in mobile social network. Specifically, the adversary is assumed to have knowledge of neighbourhood structure of a pair of connected vertices as the background knowledge and use this knowledge to carry out the re-identification of targeted victims in anonymized mobile social network data that has been released by the social network service providers for consumption by interested third party entities.

3.1 Re-identification Attack

We know explain the procedure for the proposed re-identification attack. Let $G(V, E)$ be the social network data graph and $T \in V$ and $u \in V$ be adjacent vertices in G such that $T \neq u$. Let us assume that $T \in V$ represents the target vertex. The aim of the adversary is to re-identify target vertex (i.e., $T \in V$ from the anonymized mobile social network graph by exploiting the friendship (degree) and the neighbor information of vertex $\bar{v} \in \bar{V}$ and $\bar{u} \in \bar{V}$ where $\bar{u}$ is an adjacent vertex of a vertex $\bar{v}$ such that $\bar{v} \neq \bar{u}$. To achieve this goal, the adversary performs the following steps:

(a) Request the anonymised graph data for a vertex with similar node neighborhood information as the target vertex $T \in V$. Assume the query returns a set of vertices $\mathcal{R} \subset V$ that matches the query.
(b) Refine $\mathcal{R}$ further by comparing the link structure among every neighbours of vertices in $\mathcal{R}$. Let the output of this step be $\mathcal{D}$.
(c) Utilize the neighbourhood information of vertex u to determine vertex T in $\mathcal{D}$.

Let us explain the above procedure in detail. When the adversary queries the anonymized graph with neighborhood information as the target vertex $T \in V$, it is assumed that the adversary receives a set of matching vertices $\mathcal{R} \subset V$. It is important to note that the fewer the number of returned vertices the higher the probability that the target victim $T \in V$ could be positively and accurately re-identified. The next step is to further refine the output from the previous step. In order to improve the accuracy of the re-identification of the target vertex, the adversary then compares the link structure among every neighbours of the nodes in $\mathcal{R}$ using his background knowledge. This step is expected to further refine the original query result and produce the set of matching vertices $\mathcal{D}$. Assuming that $\mathcal{D} > 1$, the adversary then uses the background knowledge regarding the neighbourhood information of u to re-identify T in $\mathcal{D}$. The target victim could be definitely re-identified if and only if the cardinality of matching vertices is 1.

We now illustrate the vulnerability anonymized graph $\bar{G}(\bar{V}, \bar{E})$ to the procedure described above. For this we use a sample of an anonymized social network graph shown in Fig. 2. The original mobile social network data is anonymised using the k-anonymity method. Suppose we want to identify 'Ziad' in the anonymized graph which is represented by vertex 4. Note that the anonymization has converted the name of the target 'Ziad' into a number as shown in vertex 4.

In the proposed approach, the adversary exploits the friendship information of vertex $\bar{v} \in \bar{V}$ and $\bar{u} \in \bar{V}$ targeting to identify t in G where $\bar{u}$ is an adjacent vertex of a vertex $\bar{v}$ such that $\bar{v} \neq \bar{u}$. In the case of Fig. 2, when the adversary queries the graph for a vertex with 4 friends, the query matches $R = \{3, 4, 5, 6\}$ vertices in the graph. By just using the friendship information alone, the adversaries only can identify 'Ziad' with probability ¼. The adversary further refines the query using his background knowledge. Specifically, the adversary knows that of the 4 users connected to the target vertex, two of them are also connected to each other. Further, the adversary also knows that one of the neighbours is

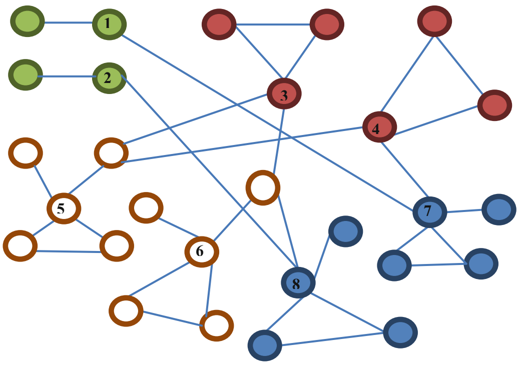


Fig. 2. A sample of anonymized mobile social network graph

connected to five users where two of them known each other. Therefore, the adversary uniquely re-identified 'Ziad' from the anonymized graph as a vertex 4.

4 Performance Analysis

In order to analyses the proposed approach's capability in terms of success rate regarding target re-identification in anonymized social network graph, we performed experimental analysis using real datasets. In this section, we analyses the performance of the proposed attack and compare it with two baseline approaches [9, 11].

4.1 Experimental Setup

We carried out the experiment using MATLAB on Pentium Dual-Core 2.50 GHz machine with 3 GB of RAM running with Windows 7 Enterprise. We used two different datasets (i.e., PolBooks, and Small-World):

- The PolBooks dataset is a network of books sold by an online store where the edges between books represent the purchase frequency of the same buyers.
- The Small-World dataset is a type of graph in which most vertices can be reached from every other vertex by a small number of hops.

The same datasets have been used in previous studies [4, 6, 8].

4.2 Result Analysis

In this section, we discuss the performance results of the proposed approach as compared to the baseline approaches. In the experiments, we used accuracy rate as the performance metric which is defined as a re-identification rate of the three approaches. The percentage represents the amount of vertex that is the dataset that are exposed to re-identification attack using the three types of graph structural information. The graph in Fig. 3 compares the accuracy rate as a function of the various datasets. Note that the three approaches use different types of social network data structural information to re-identify the target.

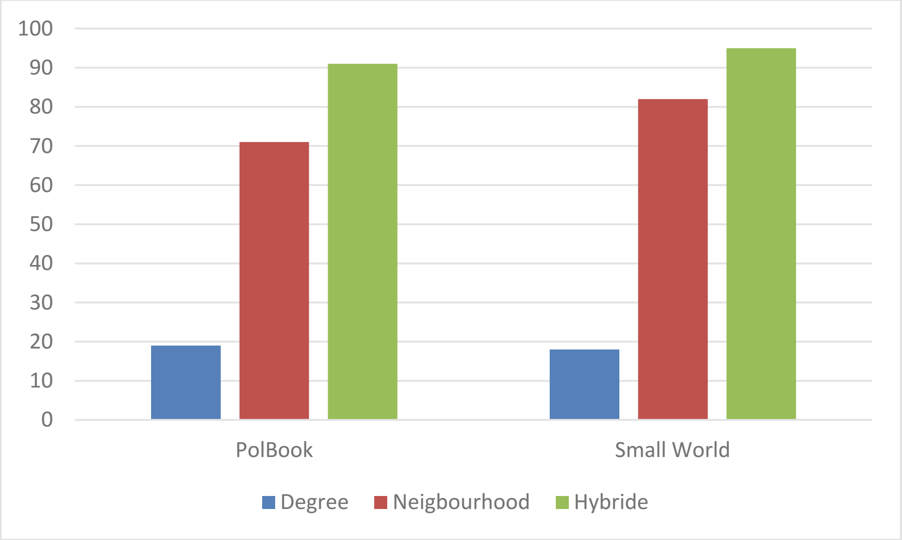


Fig. 3. The re-identification rate comparison graph

The results of the graph shown in Fig. 3 attests to the fact that the re-identification rate of the proposed approach is much higher than the baseline approaches. We note that the result shown in Fig. 3 includes only unique matching vertices based on the specific structural queries performed on the anonymized datasets. This results presented in Fig. 3 only shows the rate of vertices that definitely re-identified. From the graph shown in Fig. 3, we can see that the approach proposed in [9] has 20–30% success rate of definitely re-identifying targets in the anonymized graph datasets. In contrast, the approach proposed in [12] has higher re-deification rate as compared to the approach proposed in [9]. The experiment result shows that the approach proposed in [12] can definitely re-identified in excess of 60% of the social network users from the anonymised datasets doubling the rate of re-identification of the approach proposed in [8]. The proposed approach outperforms substantially both baseline approaches. The proposed approach can definitely re-identified in excess of 89% of the social network users from the anonymised datasets. The reason for the performance differences of the three approaches can be attributed to the structural background knowledge used by the adversary. Undeniably, the 20–30% re-identification rate is already quite in terms of the number of users who are at risk using limited background knowledge. With additional neighbourhood structural information, the approach proposed in [12] substantially increased the risk of re-identification rate to above 60%. The approach we proposed exploits both information proposed in [9, 12] and further refines them to zoom on the targeted vertices in the anonymized graph dataset. Thus combining known information and refining them can be lethal in attacking the privacy of the social network users.

5 Conclusion

There is no doubt that much of the data collected and collated by the social media service providers could assist groups working for the public interest such as sociologists and epidemiologists with new insights and possibilities for action. However, the collected data often contains sensitive information and thus must be made available to external interested parties in a responsible manner. Currently, social network operators release anonymized and sanitized versions of the complete social network graph for use by the third party users. Unfortunately, the approaches used by the social network platform providers to anonymise the data is insufficient to protect the privacy of the individuals as demonstrated in this work. In this paper we investigated the privacy vulnerability of the anonymised social network data with emphases on a re-identification and disclosure attacks. We presented three different approaches that use different background knowledge to uniquely identify a target in the anonymised social network graphs. We have shown empirically that using a variety of structural information that are readily available, the adversary can successfully re-identify targeted victim with high accuracy. Therefore, publishing social network data still raises serious concerns for individual privacy. In future work, we plan to develop privacy preserving mechanisms to safeguard the anonymised social network data publication is not vulnerable to a wide variety of re-identification and disclosure attacks.

Acknowledgement. The help of Maliha Omar is greatly appreciated. Without her support this paper will not come to its present state.

References

1. Teles, A.S., da Silva e Silva, F.J., Endler, M.: Situation-based privacy autonomous management for mobile social networks. Comput. Commun. **107**, 75–92 (2017)
2. Abawajy, J.H., Ninggal, M.I.H., Herawan, T.: Privacy preserving social network data publication. IEEE Commun. Surv. Tutor. **18**(3), 1974–1997
3. Zhang, S., Wang, G., Liu, Q., Abawajy, J.H.: A trajectory privacy-preserving scheme based on query exchange in mobile social networks. Soft Comput. 1–13 (2016)
4. Ninggal, M.I.H., Abawajy, J.H.: Utility-aware social network graph anonymization. J. Netw. Comput. Appl. **56**, 137–148 (2015)
5. Luo, E., Liu, Q., Abawajy, J.H., Wang, G.: Privacy-preserving multi-hop profile-matching protocol for proximity mobile social networks. Future Gener. Comput. Syst. **68**, 222–233 (2017)
6. Abawajy, J.H., Ninggal, M.I.H., Herawan, T.: Vertex re-identification attack using neighbourhood-pair properties. Concurr. Comput. Pract. Exp. **28**(10), 2906–2919 (2016)
7. Xiao, X., Chen, C., Sangaiah, A.K., Hu, G., Ye, R., Jiang, Y.: CenLocShare: a centralized privacy-preserving location-sharing system for mobile online social networks. Future Gener. Comput. Syst. (2017). https://doi.org/10.1016/j.future.2017.01.035
8. Ninggal, M.I.H., Abawajy, J.H.: Preserving utility in social network graph anonymization. In: The Proceedings of the 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom 2013), Melbourne, Australia, 16–18 July 2013, pp. 226–232. IEEE Computer Society (2013). ISBN 978-0-7695-5022-0

9. Liu, K., Terzi, E.: Towards identity anonymization on graphs. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, pp. 93–106. Vancouver, Canada (2008)
10. Ninggal, M.I.H., Abawajy, J.H.: Attack vector analysis and privacy-preserving social network data publishing. In: The Proceedings of the IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom 2011), Changsha, China, 16–18 November 2011, pp. 847-852. IEEE Computer Society (2013). ISBN 978-1-4577-2135-9
11. Zhou, B., Pei, J.: The k-anonymity and l-diversity approaches for privacy preservation in social networks against neighborhood attacks. Knowl. Inf. Syst. **28**(1), 47–77 (2011)
12. Ninggal, M.I.H., Abawajy, J.: Privacy threat analysis of social network data. In: Xiang, Y., Cuzzocrea, A., Hobbs, M., Zhou, W. (eds.) ICA3PP 2011. LNCS, vol. 7017, pp. 165–174. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-24669-2_16

Exploring Secure Communication in VANET Broadcasting

Muhammad Jafer, M. Arif Khan(✉), Sabih ur Rehman, and Tanveer A. Zia

School of Computing and Mathematics, Charles Sturt University,
Wagga Wagga, Australia
{mjafer,mkhan,sarehman,tzia}@csu.edu.au

Abstract. Broadcasting is a communication mechanism utilized in VANET architecture that facilitates in disseminated of public information to help reduce traffic jams/congestions. The authentic and genuine nature of public information is required to be maintained to avoid broadcasting of false information causing mass panic and hysteria. Therefore, it is of utmost importance to secure the broadcasting information so that the information cannot be altered by the intruders without compromising public nature of the information. In this paper, we have proposed a secure broadcasting architecture consisting of different layers stacked together in different formation according to operating modes. A real-time simulation model is developed in Python, while simulations are run on supercomputer for the purpose of gathering results for highway environments. We compare the results of the proposed secure highway architecture with unsecure architecture. Overall, the results show delayed propagation time due to availability of multiple information packets as well as prioritization of these information packets. However, there was no significant difference in retransmission of different information packets when compared with either different broadcasting probability or unsecure highway scenario, which indicates an effective as well as efficient secure broadcasting architecture.

Keywords: VANETs · Secure broadcasting · Network coverage
Information retransmission · Public information

1 Introduction

The revolutionary concept of connecting physical devices to internet is a step towards increasing better services and products for end user satisfaction. Among other devices such as refrigerators, televisions, smart washing machines, etc., vehicles are one of the most important devices for modern day commuters. Therefore, vehicles are at the forefront of new research in connectivity and communication [1–3]. To establish communication, On-Board Units (OBUs) are used in vehicles with most OBUs having limited radio range [4]. In order to overcome this limitation, vehicular communication adopts ad-hoc networks, known

X. Lin et al. (Eds.): SecureComm 2017 Workshops, LNICST 239, pp. 69–84, 2018.
https://doi.org/10.1007/978-3-319-78816-6_6

as Vehicular Ad-hoc Networks (VANET)s. In VANETs, communication link between vehicles change frequently making the topology dynamic and vulnerable to security risks.

There are two main types of communication supported in VANETS namely: **Vehicle-to-Vehicle** (V2V) and **Vehicle-to-Infrastructure** (V2I) communication. In general, V2V communication is established among vehicles, whereas in V2I scenario communication link is established between a vehicle and any roadside infrastructure, commonly known as Road Side Units (RSUs). Further to this, communication scenarios in VANET can also be categorised as **Point-to-Point** (P2P) and **broadcasting** (BC) [5]. P2P communication can be defined as sharing the information between two vehicles without the aid of another vehicle or fixed infrastructure. In this scenario, one vehicle acts as a source and the second vehicle acts as a destination. In BC scenario, a vehicle transmits information to all vehicles within a certain geographical area. The BC scenario used in this paper is different than the commonly used BC scenario in mobile wireless communication where a transmitter broadcasts different information for different users. In this paper, we use BC as a source vehicle broadcasting same information for multiple other vehicles.

We also classify the information to be transmitted into two categories **private** and **public** information as explained below.

Private Information: We consider information as private, transmitted using P2P communication system, if it is intended only for one single vehicle or it requires certain decryption process to extract the information from the transmitted signal. For the sake of simplicity, we assume that private information is intended only between two vehicles that resemble the P2P communication scenario defined above.

Public Information: On the other hand, public information is defined as the information available for any vehicle within the network and it does not require any decryption process to extract the information from the transmitted signal. This scenario resembles BC communication in VANETs as defined above.

Importance of transmitting authentic information, whether public or private, is very high, therefore, it is crucial to secure the information. Unsecure information specially public information can be misused and can cause mass hysteria and traffic jams. Whereas, when information is secured, it is difficult for intruders to alter the original message and hence lower the risk of creating public panic.

The focus of this paper is to investigate and propose secure broadcasting architecture for VANETs. The proposed secure broadcasting architecture facilitates in implementation of strategies that avoid tempering of information during transmission. To the best of our knowledge, there currently exists no publications related to research studies proposing secure broadcasting systems or architectures. However, there is signification research studies as well as publications in secure P2P communication. This paper builds on the lessons learnt from secure P2P communication architectures and apply these ideas in securing public information in VANET broadcasting.

Following list consists of three main contributions put forward in this paper:

- Identification and categorization of security challenges related to broadcasting in VANETs.
- Proposing of a layer based secure broadcasting architecture to counter alteration in information during broadcasting.
- Implementation of the proposed secure broadcasting architecture and collecting results related to credibility index with respect to propagation time required by an information packet to achieve network coverage.

The rest of the paper is organized as follows. Section 2 contains literature review of previous research, whereas Sect. 3 describes the system model that is used in this study. A discussion regarding proposed secure broadcasting architecture is contained in Sect. 4, while operational flow of the architecture is presented in Sect. 5. In Sect. 6, the numerical results are presented in detail. Finally, Sect. 7 concludes the paper.

2 Related Work

The main focus of this paper is to extend the security principles and techniques available in P2P communication to VANETs BC communication. Some of the major security challenges in VANET are bogus information, ID disclosure and Sybil attacks. There are a number of solutions available for these security threats in the literature such as [6–13]. However, one common challenge in the literature is that it is mainly focused for P2P mobile ad-hoc networks. In order to integrate these security features in VANET BC, we can mainly classify these feature into three groups: *Authentication*, *Anonymity* and *Availability of resources*, which is inspired by work put forward in [4,14–16].

Authentication is a process of validating both sender and associated message by receiving vehicle [14]. The validation process requires sender identification, which is defined by different properties such as location, direction, speed and owner of the vehicle. The authentication mechanism helps establishing reliability of sender's information and ultimately the mechanism facilitates in preventing Sybil attacks in VANETs. While, the process of *anonymity* dictates hiding sender information as well as encrypting this information to make it unreadable for unintended users. Sender vehicles, that are either source or relay vehicles, may be willing to share information if provided with mechanisms to avoid tracking of vehicles or sharing actual vehicle information. On the other hand, a secure system is also required to incorporate fault-tolerant design, resilient to attacks as well as survival protocols so that it remains available and operational in the presence of faults or malicious attacks [14,17]. These three distinct groups of security threats are further explored with respect to P2P and BC systems in the following sections:

2.1 Security in Point-to-Point (P2P) Communication

A Point-to-Point (P2P) communication involves at minimum two vehicles, namely source and destination. Source vehicle transmits information intended for a destination vehicle, which employs a trust mechanism to establish legitimacy of the received information. In [18], trust is based on a process called authentication that help in correctly identifying source vehicle. This authentication process consists of three different types, namely ID authentication, property authentication and location authentication. ID authentication uses unique IDs, which are either licence number or chaises number of a vehicle, for identification of a vehicle. Whereas, property authentication aid in identifying type of source, e.g. that the source is a vehicle or a traffic signal, on the other hand, location authentication identifies location of a source allowing receiving vehicle to validate received information. Authentication is an effective process of identifying source as well as validating transmitted information. However, this would compromise anonymity of a source vehicle providing convenient way of tracking as well as identifying vehicle and its passengers.

In [19], a centralized system is implemented with the help of RSUs providing encryption mechanism for all the vehicles that are registered with the system. An authentication process is also introduced by the centralized system for the purpose of validating as well as issuing certificates to registered vehicles. Source vehicles are issued encrypted certificates during transmission of information, while, these certificates are decrypted by providing public key to destination vehicles for validation of transmitted information packets. Furthermore, unique encrypted digital signature generated by the source vehicle and attached to an information packet facilitates in identifying changes in original information by a destination or relay vehicles. Any change in original causes the centralize system to either not issue or validate attached encrypted certificate. The process introduced in this study establish an authentication process without compromising anonymity. However, the process is not applicable in environments lacking RSUs as it is heavily based on a centralized system implemented through RSUs. Moreover, public nature of information in broadcasting would increase complexity of overall system due to repeated requests for issuing or validation of certification for authentication.

In [6], authentication process based on encrypted vehicle signature is used to establish authentication between a vehicle and a RSU. After successful authentication, RSU issue a short-lived anonymous certificate to the vehicle. This certificated as well as public key and signature is broadcasted by the vehicle to all the neighbouring vehicles. The broadcasted information is verified by all the neighbouring vehicles with RSU. Source vehicle in this scenario transmits encrypted information, which is decrypted using public key provided by vehicle to its neighbour. This secure system prevents external attacks by employing encrypting transmitting information as well as registration of vehicles with RSU. However, the system is dependent on availability of RSU and lack mechanism to identify internal attacks.

Encryption mechanisms used for the vehicle authentication as well as encryption purposes play a vital role in creating the secure P2P systems. Both these mechanisms help to establish P2P systems that are robust enough such that they are available to the users even under malicious attacks. For interested readers, a detailed list of literature describing such secure and robust systems based on encryption mechanisms is available at [7–10,20].

Additionally, anonymity in P2P communication facilitates in securing confidential information of vehicles such as speed, identity and location of vehicles. The methodologies used for anonymizing vehicle information in literature of P2P VANETs are based on either pseudonyms of k-anonymity principles [6–13]. In pseudonym approach, a vehicle is allotted an alias from a pool of pseudonyms by using different algorithm to achieve vehicle anonymity. Whereas in k-anonymity approach, vehicle information attributes are either suppressed or generalised to avoid identification and tracking of vehicle and its passengers.

2.2 Security in Broadcasting (BC) Communication

In BC, information is shared among all vehicles in a network, therefore, the information is public. Security aspects are relatively new in VANETs broadcasting and to the best of our knowledge, this is the first attempt to propose a secure broadcasting framework. Whereas, the three distinct security parameters of authentication, anonymity and availability of resource remain equally important for security of broadcasting. Therefore, we can extend the strategies available in P2P VANET to the security applications in BC.

The concepts and associated principles required for authentication mechanism explored in P2P communication are implementable for BC as well. Whereas, anonymity techniques based on either pseudonyms or k-anonymity principles are also effective in case of BC. However, due to public nature of information in BC, encryption and cryptographic techniques used for encryption of original message cannot be applied in their current form.

3 Generalized VANET System Model

In this section, we present a general VANET system model with $v = 1, ..., V$ vehicles in the network. These vehicles move with speed, s, of 60 to 100 km/h in the same direction on a highway that consists of multiple lanes. The vehicles are randomly distributed where they can communicate with each other using IEEE 802.11p communication protocol. IEEE 802.11p belongs to the family of IEEE 802.11p wireless protocol standards created to support mobile vehicular communication networks [21,22]. Due to availability of a large number of features in IEEE802.11p, it has become the de facto protocol for VANETs [23,24]. Among theses features, Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) and beaconing system are the two vital features that play important part in our research [25].

CSMA/CA is a packet collision avoidance process that facilitates in seamless transmission of information in a network. In this process, a vehicle, which has a desire to transmit, is required to sense the network for the purpose of establishing network usage. An immediate transmission will proceed, when there is no other transmission by any other vehicle in the network. However, a random wait time is assigned to the vehicle if network is busy. After expiry of this wait time, the vehicle will check network again and depending on the status of network, vehicle will either transmit or assign another wait time. The process of assigning wait time will continue until information is transmitted. Presence of CSMA/CA helps to avoid implementation of complex collision avoidance and detection system, which would have increased the complexity of our system many folds.

Beaconing system is another feature of IEEE802.11p that helps vehicle to maintain an up to date information regarding their neighborhood This information facilitates in accurate calculation of probability of neighborhood, P_{nc}, which is vital in calculating wait time, T_{wr}, of a information packet. P_{nc}, T_{wr} and other variables of the retransmission system are further discussed in Sect. 4.

4 Proposed Secure Broadcasting Architecture

A layer based secure broadcasting architecture has been proposed in this Section. The purpose of this proposed architecture is to identify identifying the alteration in public information during broadcasting. The proposed architecture consists of five different layers, namely anonymity, credibility, encryption/decryption, relay vehicle selection method, and transmission layer as shown in Fig. 1. These layers support different operating mode discussed in Sect. 5. A detailed discussion related to functionalities associated with these layers is explained in the following subsections:

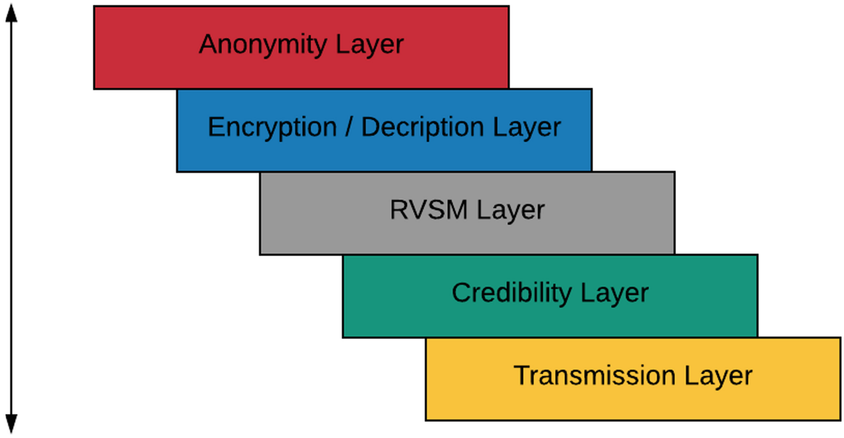


Fig. 1. Layered architecture of the proposed secure broadcasting in VANETs

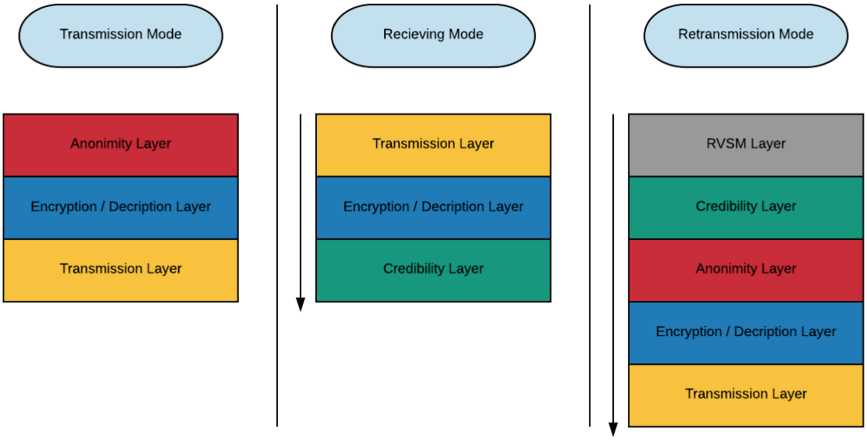


Fig. 2. Operating modes of the proposed secure broadcasting architecture

4.1 Anonymity Layer (AL)

Anonymity layer (AL) facilitates in anonymizing information for the purpose of hiding identifiable information of a vehicle. Techniques, such as shared pseudonym pool, put forward in Sect. 2 for P2P can be introduced in anonymity layer to anonymitize vehicle information. In this technique, each network in VANETs has a shared pseudonym pool consisting of unique alias that can be chosen by a vehicle to shield its identity.

4.2 Encryption/Decryption Layer (EDL)

Encryptions is one of the most effective and efficient system to secure information. Therefore, we propose encryption/decryption layer (EDL) to achieve this functionality in our model. This layer can be used to encryption actual information as well as signature of vehicles to preserve authenticity of a information packet, I_p. Due to public nature of I_p, the encryption strategies available in P2P discussed in Sect. 2, such as [8–10], can not be directly applied in VANET broadcasting.

4.3 Relay Vehicle Selection Method (RVSM) Layer

RVSM layer is required during transmission phase for the purpose of avoiding broadcasting storm. Broadcasting storm is caused by blind retransmissions to achieve network coverage, which is a process of achieving propagation of information packet, I_p, to all the vehicles in a network. RVSM layer consisting of a technique, put forward in previous research [23, 24], that assigns a wait time, T_{wr}, based on probability of neighbourhood coverage, P_{nc}, to avoid broadcasting storm. An I_p can be broadcast after the assigned T_{wr} expires. Whereas, the probability of neighbourhood coverage, P_{nc}, is determined by all the vehicles, N_{np},

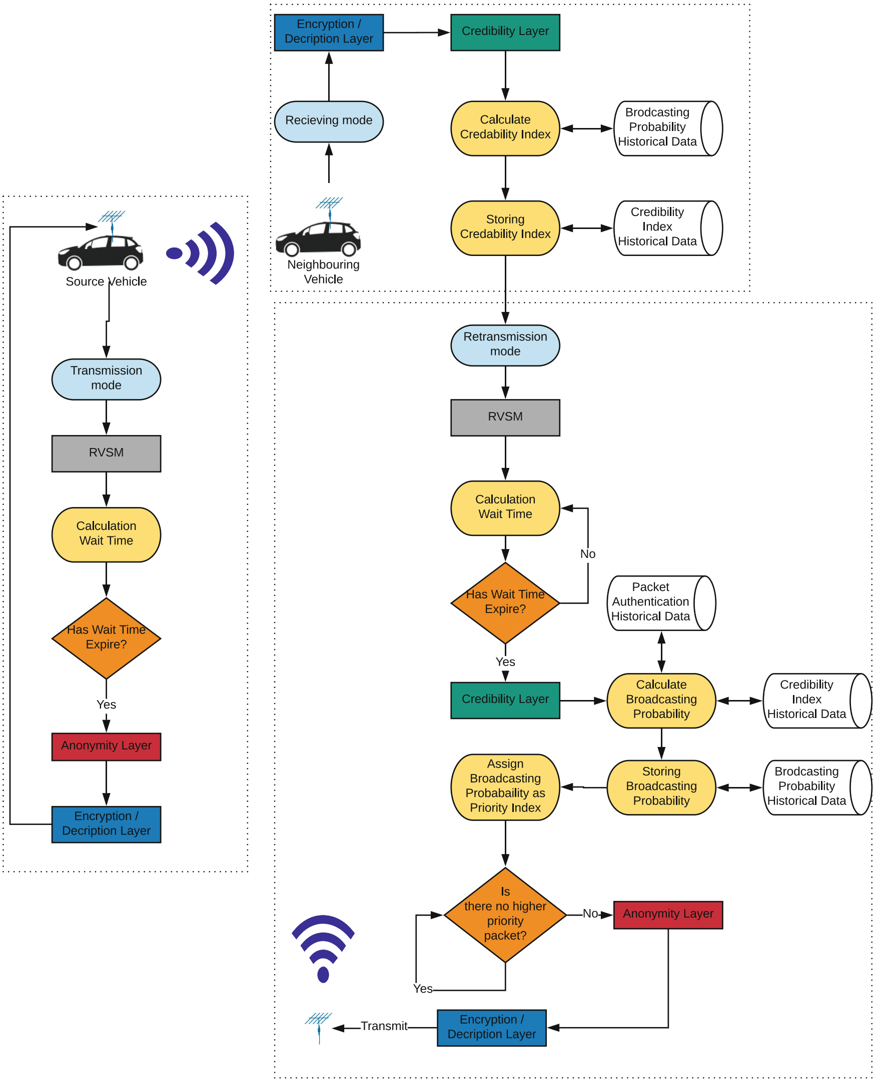

Encryption / Decription Layer
Credibility Layer
Recieving mode
Calculate Credability Index
Brodcasting Probability Historical Data
Neighbouring Vehicle
Storing Credability Index
Credibility Index Historical Data
Source Vehicle
Transmission mode
RVSM
Calculation Wait Time
Has Wait Time Expire?
Yes
Anonymity Layer
Encryption / Decription Layer
Retransmission mode
RVSM
Calculation Wait Time
No
Has Wait Time Expire?
Packet Authentication Historical Data
Yes
Credibility Layer
Calculate Broadcasting Probability
Credibility Index Historical Data
Assign Broadcasting Probabaility as Priority Index
Storing Broadcasting Probability
Brodcasting Probability Historical Data
Is there no higher priority packet?
No
Anonymity Layer
Yes
Transmit
Encryption / Decription Layer

Fig. 3. Detail explanation of different layers and transmission modes of the proposed secure broadcasting architecture

that have received this information, and all the vehicles in the neighbourhood database, N_{vh}, of that vehicle. Mathematically, P_{nc} can be defined as follows:

$$P_{nc} = \begin{cases} 0, & \text{if } N_{np} = 0 \\ 1, & \text{if } N_{vh} = 0 \\ \frac{N_{np}}{N_{vh}}, & \text{otherwise.} \end{cases} \tag{1}$$

4.4 Creditability Layer (CL)

Credibility layer establishes authenticity of an information packet, I_p, which facilitates in process of privatization during transmission. The process of establishing authenticity for a vehicle consists of computing and storing historical information related to credibility index, $\propto$, broadcasting probability, B_p, as well as authenticated packet score, P_a, of all the vehicles in its neighborhood. Credibility of a vehicle is defined by $\propto$ using historical data consisting of B_p of all the previous retransmissions. Mathematically, $\propto$ is defined as following, where B_n is the total number of historical retransmissions:

$$\propto := \begin{cases} 1, & \text{if } B_n = 0 \\ \frac{1}{B_n}\left(\sum_{i=1}^{B_n} B_{pi}\right), & \text{otherwise} \end{cases} \tag{2}$$

While, a priority value is assigned to the information packet, I_p using broadcasting probability, B_p, for the purpose of transmission. B_p relies on combination of $\propto$ and packet authentication score, which consists of average number of authentic packet received from the source vehicle of this current I_p. Formally, B_p is defined as following:

$$B_p := \frac{\propto}{P_n}\left(\sum_{i=1}^{P_n} P_{ai}\right) \tag{3}$$

where P_a is known as packet authentication score ranging between 1 and 0, while, P_n are the total number of packets received from the source vehicle. It is important to note that P_a of I_p may increase or decrease by 0.1 respectively, if another vehicle in the same vicinity either confirms or contradicts the reception of original message by the source. Whereas, if a rebuttal is transmitted by source or any other vehicle in the vicinity, one of the P_a transmitted by relay vehicle is decreased by 0.1.

4.5 Transmission Layer (TL)

Transmission layer facilitates in the propagation of information packets, I_p, in a communication network. Transmission of I_p over wireless medium is governed by IEEE802.11p protocol, however, transmission can also use other established protocols such as Wireless Access in Vehicular Environment (WAVE). We assume that a vehicle, v, transmits its information as a vector $\mathbf{x}$ such that:

$$\mathbf{x} = [x_1, x_2, ..., x_n]_{1\times N}, \tag{4}$$

where $x_1, x_2, ..., x_n$ are the coded information alphabets. The transmission vector, $\mathbf{x}$, is effected by the wireless channel fluctuations, modelled by the channel matrix, $\mathbf{H}$, and the noise vector, $\mathbf{n}$. The information signal received on a vehicle, v, can be represented by $\mathbf{y}_v$ and is given as:

$$\mathbf{y}_v = \mathbf{H}\mathbf{x}^\top + \mathbf{n}, \tag{5}$$

such that $[\mathbf{y}_v]_{N\times 1}$, $[\mathbf{H}]_{N\times N}$, $[\mathbf{n}]_{N\times 1}$ and $\mathbf{x}^\top$ represents transpose of $\mathbf{x}$. We further assume that each element of $\mathbf{H}$ is modelled as a Gaussian random variable and the noise $\mathbf{n}$ is also modelled as uniformly distributed Additive White Gaussian Noise, $AWGN$, with zero mean and unit variance. Such a model is used in most of the VANET communication scenarios such as [26–28]. Furthermore, the data rate at which each vehicle can transmit the packets is denoted by r_v and can be given as:

$$r_v = \eta \log_2 \left(1 + \frac{P_t |\mathbf{H}\mathbf{H}^*|^2}{|\mathbf{n}|^2}\right) bps, \tag{6}$$

where P_t is the transmitted power, η is the bandwidth in Hz and $(.)^*$ denotes the complex conjugate transpose of a matrix.

5 Secure Broadcasting Operating Modes

The proposed secure broadcasting architecture consists of three different operating modes, known as *transmission, receiving and retransmission* modes. These modes operate by utilize secure broadcasting layers, which are stacked together in different formation according to operating modes shown in Fig. 2. These modes are further discussed in the following sections:

5.1 Transmission Mode

A vehicle, known as source vehicle, is in transmission mode during the process of transmitting original message. The transmission mode requires a combination AL, EDL and TL. AL anonymizes source vehicle information, while, EDL helps in encrypting vehicle signature and other meta data. The encrypted information helps vehicle to identify any message(s) that are circulated with its encryption. The vehicle may identify spam messages and broadcast a rebuttal to that message if needed. This helps to safe guarding the network against spam messages and spamming vehicles.

5.2 Receiving Mode

In receiving mode, a vehicle receives an original or retransmitted information packet, I_p. This mode consists of EDL and CL. The decryption part of EDL is used to decrypt received I_p. The part of the message that is of public nature can be decrypted by this layer. While, the CL comes after EDL. During receiving mode, the CL computes and updates credibility index of transmitting vehicle based on Eq. 2.

5.3 Retransmission Mode

A vehicle is in retransmission mode when it decides to retransmit an original or retransmitted message. However, before a vehicle decides to retransmit, it has to go through an independent method run by all the vehicles in a network to establish their suitability to retransmit a message using RVSM layer. RVSM layer provides a wait time, T_{wr}, to all the information packets, I_p, that needs to be transmitted. The transmission of an I_p proceeds when T_{wr} assigned to it is expired. CL is involved after RVSM layer for the purpose of computing broadcasting probability, B_p. This probability facilitates in prioritizing all the information packets for the purpose of broadcasting. I_p with highest B_p is then forwarded to transmission layer for broadcasting over wireless medium.

Table 1. Simulation parameters

Parameters	Values
Simulation area	Variable
Frequency	5.9 GHz
Type of road	Highway with multiple lanes
Vehicle densities	5, 10, 20, 40, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500 vehicles
s	Between 60 and 100 km/h
Protocol	IEEE 802.11p
Transmission range	1000 m [29]

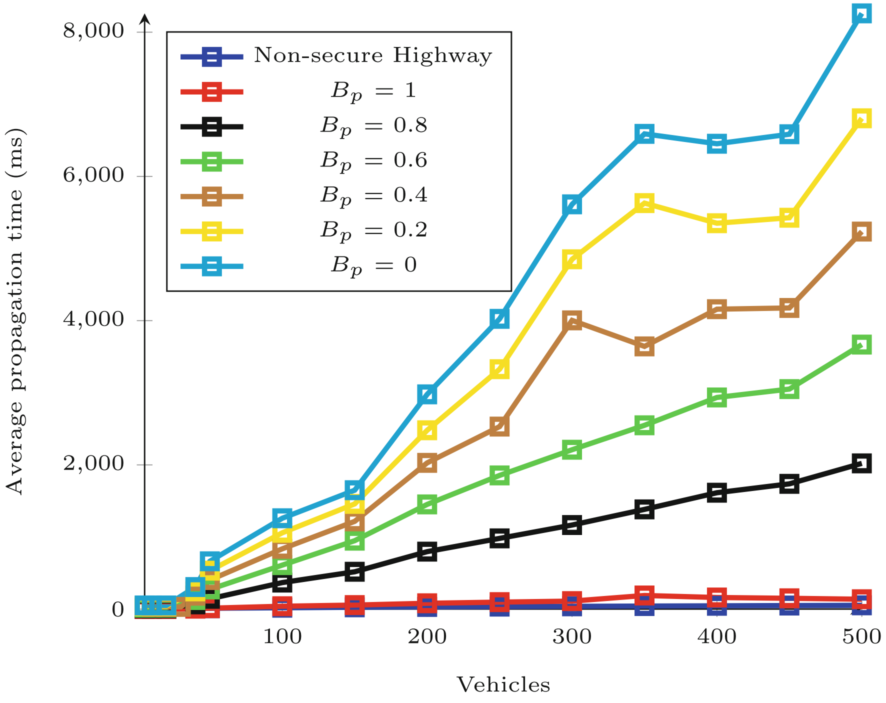


Fig. 4. Average propagation time for different broadcasting probabilities, B_p, scenarios in vehicular mobile environments for various vehicle densities.

6 Numerical Results

The secure broadcasting architecture is implemented by a real time simulation model of highway environment consisting of a priority queue model. The real time simulation model is developed in Python, while privatization of information packets in priority queue is based on Time-To-Live (TTL) and broadcasting probability, B_p. An information packet with higher value of TTL decreases its priority of retransmission as compared to lower value of TTL, on the other hand, higher values of B_p increases transmission priority of the information packet. The results related to effect of B_p on propagation time and number of transmissions are compared with a unsecure highway environment, which lacks B_p to establish priority of the information packet based on the source vehicle. There are different symbols and notations used in the simulation system, which are listed in Table 1. Furthermore, the propagation time in this section is defined as a time required for propagation of an information packets, I_p, to all the vehicles in the network.

Information packets, I_p, that consists of lower values of broadcasting probability, B_p, are transmitted after I_p with higher values of B_p are transmitted. Therefore, propagation time of an I_p is directly proportional to number of I_p with higher B_p and vehicle density. The effects of change in propagation time

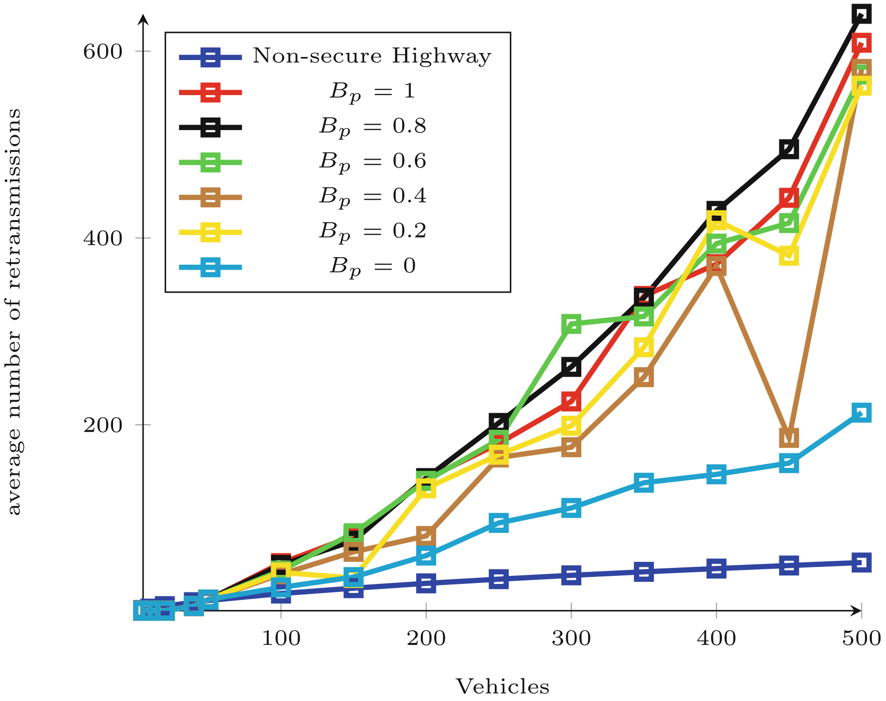


Fig. 5. Average number of retransmission in for different broadcasting probability, B_p, scenarios in vehicular mobile environments for varied densities.

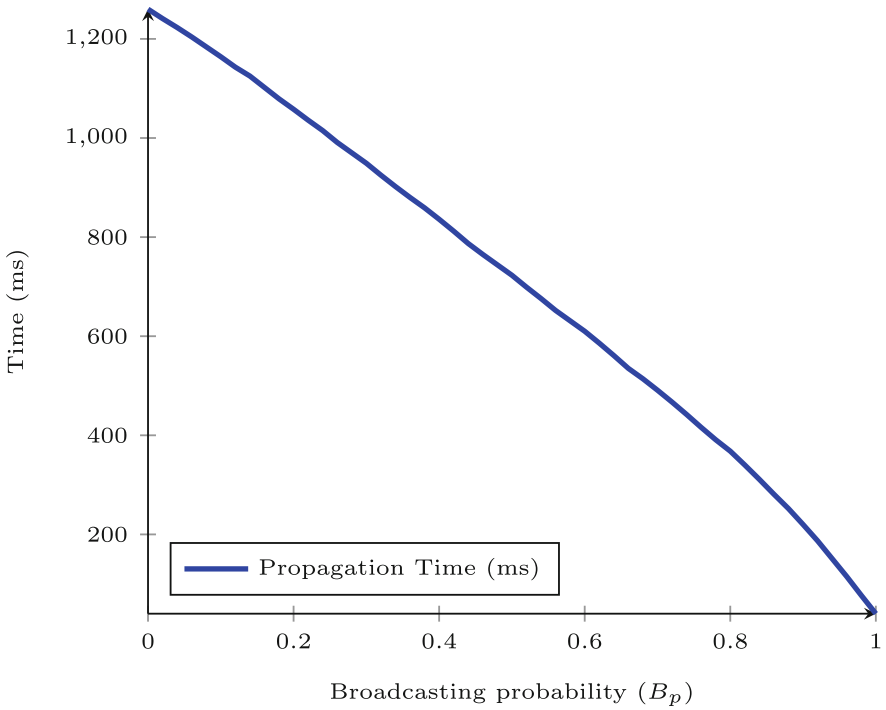


Fig. 6. Average number of retransmissions in for different broadcasting probability, B_p, scenarios in vehicular mobile environments for varied densities.

with respect to number of I_p with different B_p can be observed in Fig. 4. I_p propagation time increases with the decrease of B_p, whereas, increase of vehicle density also increases propagation time. Increase in propagation time due to vehicle density is caused by the increase in the number of vehicles needed to receive I_p in a network. On the other hand, propagation time is quite consistent for non-secure highway environment.

Number of retransmissions, N_R, is directly proportional to distribution of vehicles rather than delay in transmission. Therefore, N_R should exhibit nearly same values irrespective of the probability of retransmission. However, delay in transmission may cause changes in distribution of vehicles due to movement of vehicles over time. That is one of the reasons for different number of average retransmissions can be observed in Fig. 5 for different I_p irrespective of their broadcasting probability. Furthermore, the results in Fig. 6 present propagation time for network coverage over time in a network consisting of 100 vehicles. Theses results present the similar tendencies compared to the previous discussions regarding increase in T_wr.

The results shown in this section consist of exactly 50 I_p having values of B_p ranging from 1 to 0. Another important parameter is the number of I_p available for broadcasting at a certain time. In our simulations, the results indicated no significant effect on either propagation time or number of retransmissions for less than 50 I_p in the network. The cause of lack of signification variation during broadcasting is caused by quick transmission effect observed and analyzed in our previous work [23,24].

7 Conclusion

In this paper, we have identified and categorized security challenges related to broadcasting in VANETs. To counter these security challenges, a secure broadcasting architecture was proposed for the purpose of securing public information from intruders. The secure broadcasting architecture is layered based architecture which are stacked together in different formation according to operating modes. The network computer facility consists of super computer having a real time simulator designed in Python was used for the purpose of collecting results. These results show increase in propagation time to achieve network coverage without having any significant differences in number retransmissions when compare with unsecure highway scenario. The future work of this study is to extend this model to include dynamic readjustment of credibility index and broadcasting probability over number of time intervals for further verification of the proposed architecture.

References

1. Kumar, N., Misra, S., Rodrigues, J., Obaidat, M.: Coalition games for spatio-temporal big data in internet of vehicles environment: a comparative analysis (2015)
2. Alam, K., Saini, M., El Saddik, A.: Toward social internet of vehicles: concept, architecture, and applications. IEEE Access **3**, 343–357 (2015)
3. Gerla, M., Lee, E.-K., Pau, G., Lee, U.: Internet of vehicles: from intelligent grid to autonomous cars and vehicular clouds. In: IEEE World Forum on Internet of Things (WF-IoT), pp. 241–246, March 2014
4. ur Rehman, S., Khan, M.A., Zia, T.A., Zheng, L.: Vehicular ad-hoc networks (VANETs)-an overview and challenges. J. Wirel. Netw. Commun. **3**(3), 29–38 (2013)
5. Forouzan, A.B.: Data Communications & Networking (SIE). Tata McGraw-Hill Education, New York City (2006)
6. Choi, H.-K., Kim, I.-H., Yoo, J.-C.: Secure and efficient protocol for vehicular ad hoc network with privacy preservation. EURASIP J. Wirel. Commun. Netw. **2011**, 11 (2011)
7. Armknecht, F., Festag, A., Westhoff, D., Zeng, K.: Cross-layer privacy enhancement and non-repudiation in vehicular communication. In: 2007 ITG-GI Conference Communication in Distributed Systems (KiVS), pp. 1–12 (2007)
8. Hesham, A., Abdel-Hamid, A., El-Nasr, M.A.: A dynamic key distribution protocol for PKI-based VANETs. In: 2011 IFIP Wireless Days (WD), pp. 1–3. IEEE (2011)
9. Al Falasi, H., Barka, E.: Revocation in VANETs: a survey. In: 2011 International Conference on Innovations in Information Technology (IIT), pp. 214–219. IEEE (2011)

10. Al-Kahtani, M.S.: Survey on security attacks in vehicular ad hoc networks (VANETs). In: 2012 6th International Conference on Signal Processing and Communication Systems (ICSPCS), pp. 1–9. IEEE (2012)
11. Rivas, D.A., Barceló-Ordinas, J.M., Zapata, M.G., Morillo-Pozo, J.D.: Security on VANETs: privacy, misbehaving nodes, false information and secure data aggregation. J. Netw. Comput. Appl. **34**(6), 1942–1955 (2011). Control and Optimization over Wireless Networks
12. Djamaludin, C., Foo, E., Camtepe, S., Corke, P.: Revocation and update of trust in autonomous delay tolerant networks. Comput. Secur. **60**, 15–36 (2016)
13. Caballero-Gil, C., Molina-Gil, J., Hernández-Serrano, J., León, O., Soriano-Ibanez, M.: Providing k-anonymity and revocation in ubiquitous VANETs. Ad Hoc Netw. **36**, 482–494 (2016)
14. Engoulou, R.G., Bellaïche, M., Pierre, S., Quintero, A.: Vanet security surveys. Comput. Commun. **44**, 1–13 (2014)
15. Yadav, V., Misra, S., Afaque, M.: Security in vehicular ad hoc networks. In: Security of Self-organizing Networks: MANET, WSN, WMN, VANET, p. 227 (2010)
16. Stampoulis, A., Chai, Z.: A survey of security in vehicular networks. Project CPSC, vol. 534 (2007)
17. Qian, Y., Moayeri, N.: Design of secure and application-oriented VANETs. In: IEEE Vehicular Technology Conference on VTC Spring 2008, pp. 2794–2799 (2008)
18. Kargl, F., Ma, Z., Schoch, E.: Security engineering for VANETs. In: Proceedings of 4th Workshop on Embedded Security in Cars, pp. 15–22 (2006)
19. Papadimitratos, P., Buttyan, L., Holczer, T., Schoch, E., Freudiger, J., Raya, M., Ma, Z., Kargl, F., Kung, A., Hubaux, J.-P.: Secure vehicular communication systems: design and architecture. IEEE Commun. Mag. **46**(11), 100–109 (2008)
20. Isaac, J.T., Zeadally, S., Camara, J.S.: Security attacks and solutions for vehicular ad hoc networks. IET Commun. **4**(7), 894–903 (2010)
21. ur Rehman, S., Khan, M.A., Zia, T.A., Khokhar, R.H.: A synopsis of simulation and mobility modeling in vehicular ad-hoc networks (VANETs). IOSR J. Comput. Eng. (IOSR-JCE) **15**, 1–16 (2013). e-ISSN 2278-0661
22. Jiang, D., Delgrossi, L.: IEEE 802.11p: towards an international standard for wireless access in vehicular environments. In: IEEE Vehicular Technology Conference on VTC Spring 2008, pp. 2036–2040 (2008)
23. Jafer, M., Khan, M.A., Rehman, S.U., Zia, T.A.: Optimizing broadcasting scheme for VANETs using genetic algorithm. In: 2016 IEEE 41st Conference on Local Computer Networks Workshops (LCN Workshops), pp. 222–229, November 2016
24. Jafer, M., Khan, M.A., Rehman, S.U., Zia, T.A.: Broadcasting under highway environment in VANETs using genetic algorithm. In: 2016 IEEE 85th Vehicular Technology Conference (VTC-Spring) (VTC Workshops), June 2017
25. Saeed, R., Naemat, A., Bin Aris, A., Bin Awang, M.: Design and evaluation of lightweight IEEE 802.11p-based TDMA MAC method for road side-to-vehicle communications. In: The 12th International Conference on Advanced Communication Technology (ICACT), vol. 2, pp. 1483–1488, February 2010
26. Goldsmith, A.: Wireless Communications. Cambridge University Press, Cambridge (2005)
27. Hussain, M., Rasheed, H., Ali, N., Saqib, N.: Roadside infrastructure transmission of VANET using power line communication. In: 2017 International Conference on Communication, Computing and Digital Systems (C-CODE), pp. 139–143, March 2017

28. Lazaropoulos, A.G.: Deployment concepts for overhead high voltage broadband over power lines connections with two-hop repeater system: capacity countermeasures against aggravated topologies and high noise environments. Prog. Electromagnetics Res. **44**, 283–307 (2012)
29. Saini, M., Alelaiwi, A., Saddik, A.E.: How close are we to realizing a pragmatic vanet solution? A meta-survey. ACM Comput. Surv. (CSUR) **48**(2), 29 (2015)

Identification of Forensic Artifacts in VMWare Virtualized Computing

Cory Smith, Glenn Dietrich, and Kim-Kwang Raymond Choo(✉)

Department of Information Systems and Cyber Security,
University of Texas at San Antonio, San Antonio, TX 78249, USA
raymond.choo@fulbrightmail.org

Abstract. With popularity of virtualized computing continuing to grow, it is crucial that digital forensic knowledge keeps pace. This research sought out to identify the forensic artifacts and their locations that may be recovered from a VMware Workstation virtual machine running Windows 7 x64. Several common forensic tools were used to conduct this research, namely AccessData's Forensic Toolkit (FTK), FTK Imager, and FTK Registry Viewer. This research verified the processes required to gather digital evidence from a virtual machine disk (VMDK) file, creation of a forensic image, and mounting of evidence into these forensic tools. This research then proceeded to document recovered artifacts and their locations related to system configuration, internet usage, file creation and deletion, user administration, and more.

Keywords: Digital forensics · Forensic artifacts · Virtualization
Virtual machine · VMDK · Forensic Toolkit · FTK Registry Viewer

1 Introduction

Virtualization is often a term that you hear in relation to cloud computing. Virtualization, while it is a separate technology, is one of the most fundamental and critical components which enables versatility and scalability of cloud computing. Virtualization, as defined by VMware is "the process of creating software based representations of something rather than a physical one" [51]. These software-based representations are known as Virtual Machines (VMs). The real benefit of virtualization software is the ability to run 1-N virtual machines on a single physical server – this is done using a Hypervisor. Hypervisors are the software packages that are deployed to "virtualize" a server. These software packages turn the physical machine into a "host", which can then provide its resources to the "guests" contained on it. The hypervisor's role is to dynamically distribute the host's resources to the hosted virtual machines on an as-needed basis [5].

There are two types of hypervisors in use today [47]. Type I hypervisors are known as "Bare Metal Hypervisors", meaning that the hypervisor software is deployed right onto the physical hardware, without the use of any underlying operating systems. Due to the lack of an underlying operating system, the hypervisor is much more efficient when interacting with the host machines resources because the interaction is direct.

X. Lin et al. (Eds.): SecureComm 2017 Workshops, LNICST 239, pp. 85–103, 2018.
https://doi.org/10.1007/978-3-319-78816-6_7

Type II Hypervisors are deployed onto an already running operating system. This model requires the hypervisor to communicate with the operating system to use the host resources. Although it requires an extra step to interact with the host resources, performance delays are not noticeable [5].

For these guest virtual machines to work properly, there are several configuration files that must exist and be accessible by the virtualization software being used. These files are incredibly important for both a virtual machine to run and a digital forensics investigation. For the purposes of this research we are using a type II hypervisor in VMware Workstation Pro.

Table 1 shows the critical configuration files needed for a virtual machine to run properly.

Table 1. VMware configuration files [52]

File extension	File purpose
.log	Keeps a log of all the VM workstation activity
.nvram	Stores the state of the VMs BIOS
.vmdk	Virtual Machine Disk File; stores the contents of the VMs hard drive
.vmsd	Stores centralized metadata about VM snapshots
.vmsn	Snapshot State File; stores the running state of a VM at the time the snapshot is taken
.vmss	Suspended State File; stores the state of a suspended VM
.vmtm	Configuration file containing team metadata
.vmx	Primary configuration file for the VM, stores all the settings of the VM
.vmxf	Supplemental configuration file for VMs that are in a team

Virtualization reduces the need for hundreds or thousands of physical servers. This reduction in equipment means smaller datacenters, thus less overhead costs. The cost differential alone is enough for businesses to give serious consideration to virtualization capabilities. From power consumption, to heating and cooling cost, the savings can be extensive. Other benefits of virtualization include the ability to rapidly scale enterprise resources to meet consumer needs, test software on many different operating systems, and provide a cost-effective way to achieve fault tolerance for your enterprise services. With benefits like these, it is easy to understand why virtualization is being adopted faster than ever. The annual report from RightScale outlines cloud adoption trends from the previous year. The 2017 report surveyed 1,002 respondents and determined that 95% of organizations surveyed are experimenting with Infrastructure as a Service (IaaS). In addition, the use of multiple clouds per organization increased from 82% to 85% since 2016. In addition, 23% of enterprises with more than 1,000 employees have over 1,000 virtual machines in VMware [40]. This adoption underscores the need for the ability to perform thorough digital forensic investigations on virtualized computers.

Digital evidence is created anytime a user takes any action – criminal or not – on a computer. When the actions performed on a computer are criminal, or aid in a criminal act, digital forensics is performed to collect this evidence. The National Institute of Justice defines digital evidence as:

> "*... information stored or transmitted in binary form that may be relied on in court. It can be found on a computer hard drive, a mobile phone, a personal digital assistant (PDA), a CD, and a flash card in a digital camera, among other places. Digital evidence is commonly associated with electronic crime, or e-crime, such as child pornography or credit card fraud.*" [15].

When searching for digital evidence on a windows machine, the focus should be on gathering information pertaining to file uploads/downloads, files/folder created, opened, and removed, programs installed, executed, and removed, usage of various accounts on the machine, external device usage, and usage of the browsers by each account on the machine [30]. For the purposes of this research, we will be focusing on similar user activities as listed above.

With the popularity of virtualization, it is critical that there be documented processes and procedures on how to perform digital forensics investigations in this environment; however, it is not that simple. There are several concerns when it comes to digital forensics in a virtualized environment, several of which I will present in the final section of this paper as proposed future research topics. One of the primary concerns is the lack of documented forensic artifacts that can be recovered from a virtual machines disk file. In a traditional forensics investigation, the investigator has access to the physical media and can create images as needed for their investigation. These forensic images provide a wealth of data. This research aims to identify the forensic artifacts that can be recovered from a virtualized computer running Windows 7 by using AccessData's Forensic Toolkit to investigate the supporting files used by the VM. This research aims to strengthen the digital forensics field and associated techniques to keep up with the ever-changing technology landscape.

In the next section, we will discuss related works, their strengths and opportunities, and ultimately the driving force for this research. In Sects. 3 and 4, we will then discuss the steps taken to ensure a sound research environment, the steps taken to recover artifacts of interest and detailed findings. In the last section, we will finish with our conclusions and proposed future research efforts.

2 Related Literature

Digital forensics, in some form or another, has been around since Cliff Stoll famously investigated a mere seventy-five cent discrepancy between two accounting systems at the Lawrence Berkley National Laboratory in 1986 [45]. While only a few decades have passed, the advances in technology have been immense. As such, there has been an increased need for the ability to perform digital forensics investigations against these new technologies. Digital forensics is not an old practice, and the forensic artifacts recoverable from physical media have been well documented [14, 30]. However, emphasis on applying these processes to virtualized computing environments is limited [28, 29, 56]. Virtualization has seen a significant increase in popularity over the past few years, requiring that virtual machine forensic be researched just as heavily, if not more, than traditional physical media [16, 40].

Shavers [43] discussed the process of acquiring a forensic image from a virtual machine disk (VMDK) file and subsequently using it to create a new virtual machine. This allowed investigators to safely examine the VM and its contents. However, he did not go into detail about the specific artifacts that could be recovered, or their relevance to a digital forensics investigation.

Martini and Choo [4] proposed a six-step process for the remote programmatic collection of evidential data from virtual machines and demonstrated the utility of their process using VMware vCloud as a case study.

Cruz and Atkison [2] focused on the process of recovering a fragmented or corrupted VMDK file from a hypervisor by using a write blocker and the physical hard drive of the host. They explained the possibilities of creating a forensic image from the recovered VMDK file and using common forensic tools to analyze this image. They did not go into detail regarding the artifacts that could be recovered from this image.

While several research efforts have focused on identifying the difficulties involved with cloud forensics and the processes of performing forensic investigation on both cloud servers and client devices [9–13, 32–34], few have focused on identifying and recovering artifacts from the guest system [3, 6, 18, 49]. This research sought to answer the question, “What are the forensic artifacts and their locations that can be recovered from a VM running Windows 7 x64?”.

3 Experiment Setup

In this section, we describe the tools used, the configuration of the lab environment, and the process of seeding the VM for the investigation.

- VMware Workstation 12 Pro, version 12.1.0 build-3272444
- Windows 7 x64 ISO file for creating the VM (configured with 4 GB Memory, 2 Processors, and 80 GB of hard disk)
- AccessData’s Forensic Toolkit (FTK), version 6.0.3.5
- FTK Imager, version 3.4.2.6
- FTK Registry Viewer, version 1.8.3.0

VMware Workstation serves as the type II hypervisor for this research, allowing the creation of a VM from the Windows 7 x64 ISO file. AccessData’s Forensic Toolkit is a commonly used commercial digital forensics tool. FTK allows for quick mounting of forensic images, which can then be searched for forensic artifacts such as deleted files, configuration changes, and internet history [20]. For this research, the forensic image was created from the Virtual Machine Disk File (VMDK). FTK Imager converts the VMDK file to raw (dd) format which is needed to mount the image into FTK as evidence [26]. FTK Registry Viewer, also from AccessData, is packaged with FTK and provides the ability to view registry hives contained within the mounted evidence [37]. We will discuss in Sect. 4.1 how registry hives can be used in a forensic investigation.

The environment for this research was contained to a single physical host. The physical host ran VMware Workstation Pro, enabling me to create a virtual machine from the Windows 7 x64 ISO. The forensic tools from AccessData were also installed on the same physical host. Once the VM was configured and powered on, transactions required for subsequent investigation were then generated. This consisted of normal user activity: browsing the internet, uploading and downloading files, accessing, creating and deleting files, and creating and removing users. Transactions were conducted over a two-week period between 06122017 and 06222017 to ensure enough activity took place allowing for a thorough investigation.

To perform the investigation of the VMDK file, we first had to create a forensic image that could be mounted into FTK. To do this, FTK Imager was used, which takes a VMDK file and converts it to the raw image format needed by FTK. A forensic image is a bit-for bit replication of either an entire disk or a single partition and is like a "snapshot"; it captures the full state of the disk or partition [50]. This is done so that investigators do not modify or alter the original evidence in any way throughout their investigation (this is one of the forensic principles emphasized by McKemmish [41]). If the forensic image gets corrupted, then the image can be discarded and a new image restored. Hashing algorithms are commonly used to prove the integrity of the evidence.

Figure 1 shows that creating a forensic image with FTK Imager is as easy as pointing to the file location for the VMDK file. Figure 2 shows the completed image summary. In order to maintain proper chain of custody, this information must be recorded and maintained for the life of the evidence. Figure 3 is used to prove the integrity of the forensic image that was created. Two hashing algorithms are used to prove integrity throughout the creation of the image. First, MD5 and SHA1 hashes are calculated for the original evidence. Then, once processing has completed, additional hashes are calculated using the same algorithms as before. The comparison of these hashes can be used to validate evidence integrity during the case and potential subsequent trial [26].

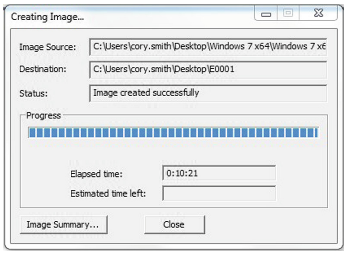
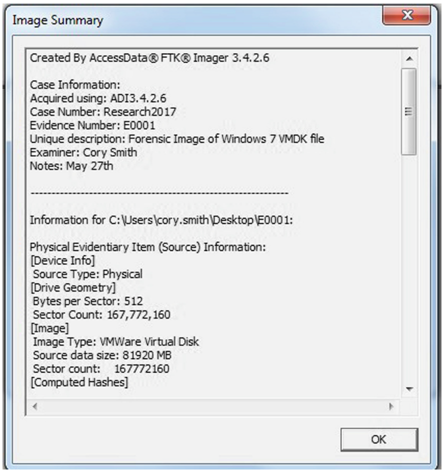


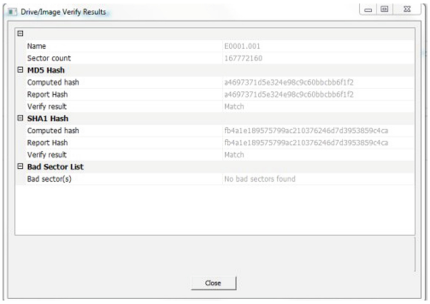
Image Source: C:\Users\cory.smith\Desktop\Windows 7 x64\Windows 7 x6
Destination: C:\Users\cory.smith\Desktop\E0001
Status: Image created successfully
Elapsed time: 0:10:21
Estimated time left:

Fig. 1. Creating an image in FTK Imager.



```
Created By AccessData® FTK® Imager 3.4.2.6

Case Information:
Acquired using: ADI3.4.2.6
Case Number: Research2017
Evidence Number: E0001
Unique description: Forensic Image of Windows 7 VMDK file
Examiner: Cory Smith
Notes: May 27th

--------------------------------------------------------------

Information for C:\Users\cory.smith\Desktop\E0001:

Physical Evidentiary Item (Source) Information:
[Device Info]
 Source Type: Physical
[Drive Geometry]
 Bytes per Sector: 512
 Sector Count: 167,772,160
[Image]
 Image Type: VMWare Virtual Disk
 Source data size: 81920 MB
 Sector count:    167772160
[Computed Hashes]
```

Fig. 2. Completed image summary in FTK Imager.



Name E0001.001
Sector count 167772160
MD5 Hash
Computed hash a4697371d5e324e98c9c60bbcbb6f1f2
Report Hash a4697371d5e324e98c9c60bbcbb6f1f2
Verify result Match
SHA1 Hash
Computed hash fb4a1e189575799ac210376246d7d3953859c4ca
Report Hash fb4a1e189575799ac210376246d7d3953859c4ca
Verify result Match
Bad Sector List
Bad sector(s) No bad sectors found

Fig. 3. FTK Imager integrity hashes.

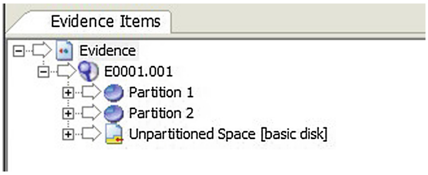


Fig. 4. Available partitions in FTK.

Once a forensic image has been created, it can easily be mounted into FTK as evidence. The FTK user guide easily outlines this process [27]. Figure 4 shows the accessible partitions in FTK, once the forensic image has been mounted as evidence.

4 Findings: Forensic Artifacts

Any time an action is taken on a computer, clues from that action are left behind, regardless of whether that action was taken by a human or a program. Digital artifacts are the pieces of information that can be gathered by recovering what is left behind. Any time a digital crime is committed, the investigation of these artifacts can provide a wealth of information as to what really happened, who did it, and the event's timeline (35). This section details the key digital artifacts that were recovered along with their respective location within the VM.

4.1 Registry Hive Artifacts

The windows registry is the authoritative source for configuration settings in the Windows operating system; every configuration change manipulates a key kept in the registry. These keys are separated out into special groupings, known as "registry hives". There are four main registry hives, as explained by Microsoft: the HKEY_-CURRENT_CONFIG, HKEY_CURRENT_USER, HKEY_LOCAL_MACHINE, and HKEY_USERS (39). Each hive maintains its own tree structure with several sub-directories containing different key value pairs, as well as its own ".log" file which maintains a history of all changes made to that registry hive [39].

HKEY_CURRENT_USER provides configuration settings for the current logged on user, and is located in "C:\Users\%USERNAME%\NTUSER.dat" [25]. This hive is critical to a forensics investigation as it maintains all the unique settings for a certain user. It also keeps track of several pieces of metadata to provide enhanced functionality to the user. By accessing the NTUSER.dat file through FTK, and opening it with FTK Registry Viewer, the investigator can retrieve critical artifacts such as URLs typed into a browser, recently accessed documents, commands typed into the Windows “Run” utility, searches from the start menu, user assist keys for application execution, and the user’s shell bags.

Within the HKEY_LOCAL_MACHINE hive, there are several sub-directories of interest; they are the Software, Security, SAM, and System folders. This hive is located in the “C:\Windows\System32\config” folder [39]. All hives present in the NTFS file system can be viewed through FTK Registry Viewer. This hive provides configurations and settings for the machine itself, and can be incredibly useful to an investigator. By interrogating the HKLM\System registry file, an investigator could recover the machine name, system time and time offset, profiles for recently connected USB drives and the shim cache. In this same directory, there is a “.alt” file, which maintains the registry key value pairs for the SYSTEM hive. By investigating the HKLM\Software registry hive, a list of installed applications was recovered. The HKLM\Security registry hive is where all system policy information is maintained, while the HKLM\SAM registry file is a collection of user credentials [1].

Per Microsoft, the HKEY_CURRENT_CONFIG contains the hardware state for the local machine and is used to compare the current configuration to the standard configuration. This can prove useful in order to determine hardware changes during a certain period. HKEY_USERS contain the standard configurations that are assigned to all new users to the system. These configurations are assigned at creation of the user [36]. An unapproved change to this key could indicate a user trying to manipulate configurations and privileges for any users created in the future.

4.2 NTFS File System Artifacts

The NTFS file system is based on a hierarchical file system, therefore, much of a digital forensics investigation is focused on the recovery of files. There are three main locations that an investigator needs to pay special attention to, namely: the Master File Table, Recycle Bin, and orphaned files.

The Master File Table relates to files like the Registry relates to configurations; it is an authoritative source for all files on the system, as well as their attributes. For every file on the NTFS file system, there is at least one entry in the MFT. This entry contains all metadata about the file such as file name, location, timestamps, permissions and content [31]. Traditionally a user assumes that deleting a file from the computer removes it permanently. In reality, deletion of the file only sets the “Active/Inactive” field to “Inactive”. Mark Stam explains the methods of extracting data from the master file table to recreate files, even if they have been marked as “Inactive” [44]. In addition to being able to recreate the file, additional metadata can be carved out of the MFT Attributes; these include the file creation time, last accessed time, as well as the last time the file was modified. There are several tools that can be used to easily parse and display the contents of the MFT, such as MFT Ripper and Analyze MFT [17, 25, 54]. Once a file has been marked “inactive” it is only a matter of time until the OS reuses the MFT entry for that file. Because of this, it can be concluded that the longer the time between deletion of files and the investigation, the greater the chances are that the file will not be able to be recovered.

Recycle Bin artifacts can be located using FTK in the “$Recycle.Bin” folder in the root directory (\$Recycle.Bin). The recycle bin is a holding place for files and folder that have been deleted by the user (Right Click > Delete). Files in this location can easily be recovered using FTK.

In the recycle bin, a folder is created for each user that has logged into the system. These folders are named using the user’s SID. It is rare that any files should exist outside of a particular user’s SID; this may represent a user attempting to hide a file on the file system since the root of the recycle bin is not viewable by any user in Windows Explorer [21]. Each time a user deletes a file, there are two entries created in their $Recycle.Bin. The first entry begins with “$I” and contains the metadata of the file: the date recycled, original file path, name of the file, and the size of the file. The second entry begins with “$R” and contains the actual data of the file. When files are deleted, the name of the file is converted to an ID string; the matching of the “$I” and “$R” entries provide the needed information to the forensic investigator.

The recycle bin artifacts allow forensic investigators to recover files that the user believes they have deleted. The findings suggested that deleted files could still be recoverable after the recycle bin had been emptied; given the investigation occurred in a timely manner. This echoed the findings of Quick and Choo [9–12], who demonstrated that data that had been removed using CCleaner and Eraser could still be recovered. Microsoft implements a wiping technique in Windows 7 for files that have been deleted from the recycle bin, this often occurs at the next restart of the machine, and indicates that the longer the time is between the deletion from the recycle bin and the investigation, the less likely it is to recover a file [21]. These recovered files can provide a wealth of information as to what the user was doing. Along with logon events, discussed in Sect. 4.4, it is simple to tie a user to the action of deleting a file. We refer the interested reader to [7] for a recent survey on Windows 7 anti-forensics approaches and countermeasures.

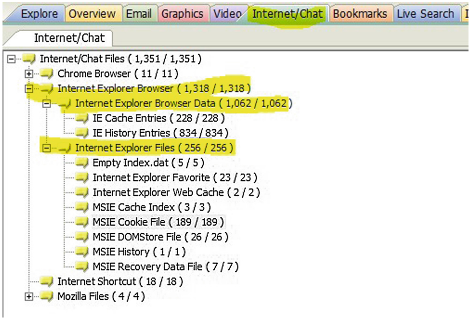
Orphaned files are a special type of file in the NTFS file system that get created when the files parent folder gets deleted from the file system. FTK provides an easy way to search through orphaned files. It is possible to recover a file that has been orphaned, given that the files MFT entry has not been entirely overwritten [21].

4.3 Web Browser Artifacts

It is no secret that we are living in a digital age, and the amount of time that we spend online is increasing. What many people do not realize is the amount of information that your computer records while you are browsing the internet [42]. When it comes to inappropriate or malicious actions, the information is stored in the same manner. Extensive browser forensics was completed throughout this research for Internet Explorer, resulting in the recovery of several key artifacts that painted the picture of what the user was doing during their online sessions.

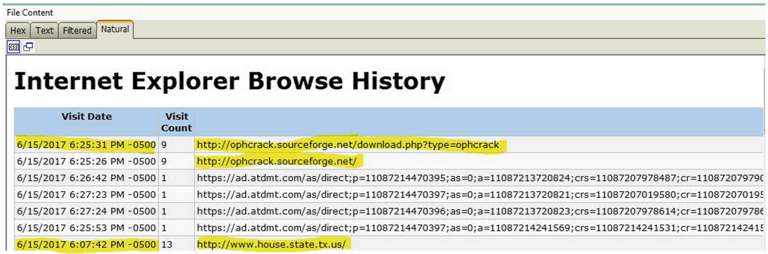
Outside of the registry, there are three major artifacts that can be recovered regarding internet usage: the browser cache, user created bookmarks, and browser cookies. The browser cache is not only a history of the sites that a user had visited, but also a cached copy of that site. The browser creates an entry in the browser cache every time a user visits a site to decrease the load time for any future visits to that site [23].

The browser cache files exist in, “C:\Users\%USERNAME%\AppData\Local\Micro-osoft\Windows\WebCache”. Using FTK, an investigator can view a listing of the sites visited as well as recreate them in the “html” tab of the tool (Figs. 5 and 6). This artifact provides a great deal of information about the active user’s activity during a certain period.



- Internet/Chat Files (1,351 / 1,351)
  - Chrome Browser (11 / 11)
  - Internet Explorer Browser (1,318 / 1,318)
    - Internet Explorer Browser Data (1,062 / 1,062)
      - IE Cache Entries (228 / 228)
      - IE History Entries (834 / 834)
    - Internet Explorer Files (256 / 256)
      - Empty Index.dat (5 / 5)
      - Internet Explorer Favorite (23 / 23)
      - Internet Explorer Web Cache (2 / 2)
      - MSIE Cache Index (3 / 3)
      - MSIE Cookie File (189 / 189)
      - MSIE DOMStore File (26 / 26)
      - MSIE History (1 / 1)
      - MSIE Recovery Data File (7 / 7)
  - Internet Shortcut (18 / 18)
  - Mozilla Files (4 / 4)

Fig. 5. Internet tab in FTK.



Internet Explorer Browse History

Visit Date	Visit Count	
6/15/2017 6:25:31 PM -0500	9	http://ophcrack.sourceforge.net/download.php?type=ophcrack
6/15/2017 6:25:26 PM -0500	9	http://ophcrack.sourceforge.net/
6/15/2017 6:26:42 PM -0500	1	https://ad.atdmt.com/as/direct;p=11087214470395;as=0;a=11087213720824;crs=11087207978487;cr=110872079790
6/15/2017 6:27:23 PM -0500	1	https://ad.atdmt.com/as/direct;p=11087214470397;as=0;a=11087213720821;crs=11087207019580;cr=110872070195
6/15/2017 6:27:24 PM -0500	1	https://ad.atdmt.com/as/direct;p=11087214470396;as=0;a=11087213720823;crs=11087207978614;cr=110872079786
6/15/2017 6:25:53 PM -0500	1	https://ad.atdmt.com/as/direct;p=11087214470397;as=0;a=11087214241569;crs=11087214241531;cr=110872142415
6/15/2017 6:07:42 PM -0500	13	http://www.house.state.tx.us/

Fig. 6. WebCache for admin user.

User created bookmarks are saved links to sites that have been visited in the past and marked for quick access. This artifact lives in, “C:\Users\%USERNAME%\Favorites”, and can provide additional information regarding past browsing history [25]. No favorites were created during this research.

Cookies are created by visited sites and attached to the user for future use. These cookies are saved in a file on the local machine and referenced any time a user visits a site that requires them. They can be used to store authentication tokens, location data, and other general user information [23]. Cookies files are stored in “C:\users\Administrator\AppData\Roaming\Microsoft\Windows\Cookies\”. They are difficult to interpret because the data is only meaningful to the site that created it. However, the existence of a cookie indicates that the user visited the referenced site it at least once, and the recovery of the cookies can provide additional information to the activities of the user.

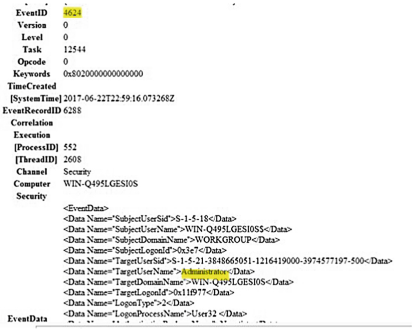
There are several artifacts of interest regarding internet usage stored in the registry; such as URL's typed by the user, form auto complete information, and browser preferences.

4.4 Windows System Log Artifacts

The windows operating system keeps thorough logs regarding what occurred on, and to, a machine. These logs provide a wealth of information to an investigator [38]. As such, they should be paid special attention to. Certain logs of interest are the Windows Event Logs and the Windows Change Logs.

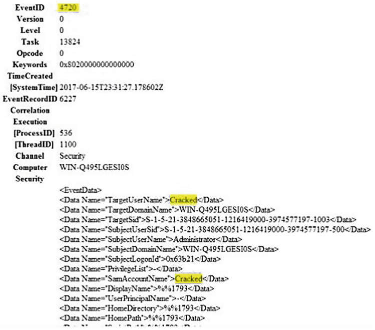
Windows event logs provide a wealth of data about what occurred on a computer. There are three main event logs on the Windows OS, namely: the Security, System, and Application. These three logs are all located in the same directory, "C:\Windows\System32\Winevt\Logs". When viewed in FTK, one would see each log with a ".evtx" file extension. The information contained in these logs are invaluable to an investigator when recreating a timeline of events.

One of the most informative logs is the security log, as it tracks all security events that occur on the computer, such as logon events, failed logons, creation of users, permission changes for users, removing users, and execution of the processes. This information paints a clear picture of what occurred while a specific user was logged in. It also records who attempted to logon to a machine. Figure 7 shows the information that gets logged when a user authenticates onto the machine. This information can be used to recreate a timeline of events. Figure 8 shows the information that is logged when a new user is created. Often attackers will create additional users on a compromised system to create alternate access methods should they ever lose access. In addition to creating new users, an attacker may also attempt to change the password for an existing account, information for this type of event can also be recovered in the Security log (Fig. 9). Examining the creation of new users can lead investigators to another account that needs to be scoped into their investigation. Figure 10 shows the information that is logged when a user is removed from the system.



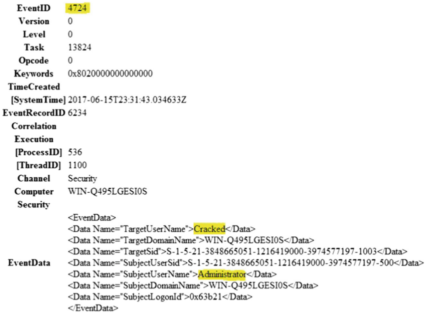
```
       EventID   4624
       Version   0
         Level   0
          Task   12544
        Opcode   0
      Keywords   0x8020000000000000
   TimeCreated
  [SystemTime]   2017-06-22T22:59:16.073268Z
 EventRecordID   6288
   Correlation
     Execution
   [ProcessID]   552
    [ThreadID]   2608
       Channel   Security
      Computer   WIN-Q495LGESI0S
      Security
                 <EventData>
                 <Data Name="SubjectUserSid">S-1-5-18</Data>
                 <Data Name="SubjectUserName">WIN-Q495LGESI0S$</Data>
                 <Data Name="SubjectDomainName">WORKGROUP</Data>
                 <Data Name="SubjectLogonId">0x3e7</Data>
                 <Data Name="TargetUserSid">S-1-5-21-3848665051-1216419000-3974577197-500</Data>
                 <Data Name="TargetUserName">Administrator</Data>
                 <Data Name="TargetDomainName">WIN-Q495LGESI0S</Data>
                 <Data Name="TargetLogonId">0x11f977</Data>
                 <Data Name="LogonType">2</Data>
     EventData   <Data Name="LogonProcessName">User32 </Data>
```

Fig. 7. Admin logon event.



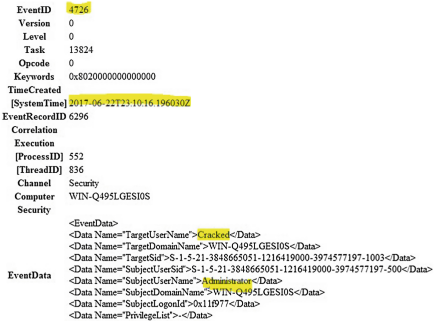
```
       EventID   4720
       Version   0
         Level   0
          Task   13824
        Opcode   0
      Keywords   0x8020000000000000
   TimeCreated
  [SystemTime]   2017-06-15T23:31:27.178602Z
 EventRecordID   6227
   Correlation
     Execution
   [ProcessID]   536
    [ThreadID]   1100
       Channel   Security
      Computer   WIN-Q495LGESI0S
      Security
                 <EventData>
                 <Data Name="TargetUserName">Cracked</Data>
                 <Data Name="TargetDomainName">WIN-Q495LGESI0S</Data>
                 <Data Name="TargetSid">S-1-5-21-3848665051-1216419000-3974577197-1003</Data>
                 <Data Name="SubjectUserSid">S-1-5-21-3848665051-1216419000-3974577197-500</Data>
                 <Data Name="SubjectUserName">Administrator</Data>
                 <Data Name="SubjectDomainName">WIN-Q495LGESI0S</Data>
                 <Data Name="SubjectLogonId">0x63b21</Data>
                 <Data Name="PrivilegeList">-</Data>
                 <Data Name="SamAccountName">Cracked</Data>
                 <Data Name="DisplayName">%%1793</Data>
                 <Data Name="UserPrincipalName">-</Data>
                 <Data Name="HomeDirectory">%%1793</Data>
                 <Data Name="HomePath">%%1793</Data>
```

Fig. 8. User creation event.



```
         EventID 4724
         Version 0
           Level 0
            Task 13824
          Opcode 0
        Keywords 0x8020000000000000
     TimeCreated
    [SystemTime] 2017-06-15T23:31:43.034633Z
   EventRecordID 6234
     Correlation
       Execution
     [ProcessID] 536
      [ThreadID] 1100
         Channel Security
        Computer WIN-Q495LGESI0S
        Security
                 <EventData>
                 <Data Name="TargetUserName">Cracked</Data>
                 <Data Name="TargetDomainName">WIN-Q495LGESI0S</Data>
                 <Data Name="TargetSid">S-1-5-21-3848665051-1216419000-3974577197-1003</Data>
       EventData <Data Name="SubjectUserSid">S-1-5-21-3848665051-1216419000-3974577197-500</Data>
                 <Data Name="SubjectUserName">Administrator</Data>
                 <Data Name="SubjectDomainName">WIN-Q495LGESI0S</Data>
                 <Data Name="SubjectLogonId">0x63b21</Data>
                 </EventData>
```

Fig. 9. Password change attempt event.



```
         EventID 4726
         Version 0
           Level 0
            Task 13824
          Opcode 0
        Keywords 0x8020000000000000
     TimeCreated
    [SystemTime] 2017-06-22T23:10:16.196030Z
   EventRecordID 6296
     Correlation
       Execution
     [ProcessID] 552
      [ThreadID] 836
         Channel Security
        Computer WIN-Q495LGESI0S
        Security
                 <EventData>
                 <Data Name="TargetUserName">Cracked</Data>
                 <Data Name="TargetDomainName">WIN-Q495LGESI0S</Data>
                 <Data Name="TargetSid">S-1-5-21-3848665051-1216419000-3974577197-1003</Data>
       EventData <Data Name="SubjectUserSid">S-1-5-21-3848665051-1216419000-3974577197-500</Data>
                 <Data Name="SubjectUserName">Administrator</Data>
                 <Data Name="SubjectDomainName">WIN-Q495LGESI0S</Data>
                 <Data Name="SubjectLogonId">0x11f977</Data>
                 <Data Name="PrivilegeList">-</Data>
```

Fig. 10. User deletion event.

Events generated by running applications are logged in the application event log. According to Microsoft, the application event log includes errors, warnings, and informational messages [22]. This information can be useful to a forensic investigator in many ways. Once a user's timeline has been established, through logon event, the investigator can then determine which applications were run during that time. Analyzing the application event log can assist in determining what that user was doing inside of each application.

The system event log maintains the starting and stopping of processes on the machine. This log is similar to the application event log as it contains errors, warnings, and informational messages pertaining to processes running on the machine. The log can be very useful to an investigator as many attacks utilize malware with known process names. The system event logs can be monitored and analyzed for these processes and identify malicious programs running on the machine [24].

Just like volume shadow copies can be restored to reverse changes to the operating system, change logs maintained by the Windows operating system can be rolled back to revert previously made changes to the file system. These artifacts include the "$LogFile" and "$UsnJrnl". Both include information regarding changes to the system. The $LogFile is much more detailed than the $UsnJrnl and is located in the root directory "\$LogFile". This file contains changes made to the file system such as creation and deletion of files and directories. The $UsnJrnl stores much less information regarding system changes than the $LogFile and is located in, "\$Extend\$UsnJrnl" off of the root directory [IR Book]. Once the file system has been mounted into FTK, as discussed earlier, the investigator can easily navigate to it and view or extract these files. There are several open source tools that can be used to intelligently analyze and display the contents of these files, such as LogFileParser [25]. Upon analysis of these artifacts, an investigator will better be able to determine which changes were made to a system over a certain time period.

4.5 Prefetch File Artifacts

Prefetch files are a critical piece of a forensic investigation. If available, they can provide investigators with a list of applications that were executed during a certain period of time. Located in "C:\Windows\Prefetch", this directory contains a ".pf" entry for every application that was executed on the machine [19]. Microsoft designed the ".pf" as a performance optimization file. The Microsoft OS tracks the first 10 s of every application's start up process and creates a ".pf" file with this information. This file is then referenced every time the application runs in order to decrease the application start time [25]. According to Luttgens, Pepe, and Mandia, the existence of a prefetch provides critical information about which programs were executed on a machine; such as name, number of executions, execution path, and when it was executed [25]. This information helps build out the attack timeline and points investigators to further evidence locations. Most importantly, even if a program has been uninstalled from the machine, the existence of a prefetch file is proof that the program existed and was utilized.

Prefetch file artifacts are crucial to a digital forensics investigation, but what if they are missing? The absence of an artifact could be an artifact in itself (e.g. signs of antiforensic activities). If the prefetch files are missing, or only exist up to a certain point, then it could indicate that the attacker was more knowledgeable than the average user. The foresight to disable the creation of prefetch files indicates a potentially skilled attacker. Disabling the creation of prefetch files modifies a key in the registry hive HKEY_LOCAL_MACHINE, this key can be viewed using FTK Registry Viewer, as discussed in Sect. 4.1. The value of the "EnablePrefetcher" registry key set to "0" which indicates that the value is disabled [16].

When available, the prefetch files can easily be recovered using FTK. Investigators can simply navigate to the Prefetch directory and view these records. There are also several open source tools that can be used to view the prefetch files and parse the data in many ways [19, 55].

4.6 LNK File Artifacts

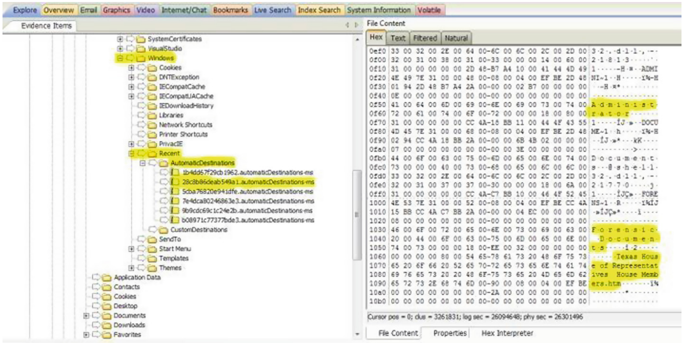
LNK files are another name for shortcut files. These types of files are the result of user action (Right Click > Create Shortcut) or program execution/install. Any time a user or program creates a shortcut, a LNK file is created in, "C:\Users\%USERNAME%\AppData\Roaming\Microsoft\Windows\Recent\".

Luttgens et al. [25] provide detailed steps for recovering the LNK files for Windows 7. Using FTK to recover the LNK files, one could recover the local path, modified, and created timestamp, as well as the file size and volume serial number.

LNK files can provide a wealth of data to forensic investigators and contribute to the re-creation of an attack timeline. These files can provide the "what," "when" and "where" of an attacker's activity while they were on a system [25].

4.7 Jump List Artifacts

Microsoft Jump Lists keep a running history of the recently used items for an application. For example, when Microsoft Word is pinned to the task bar, one could right click and choose from several of the recently opened Word documents. A user can also select to pin certain options to the jump list menu for future use. There are two types of jump lists, namely: "automatic destinations" and "custom destinations". The automatic destinations jump list is populated with recently used programs, while the custom destinations jump list is populated with the options that a user has 'pinned' to the jump list [25].



```
0ef0 33 00 32 00 2E 00 64 00-6C 00 6C 00 2C 00 2D 00 3·2·.·d·l·l·,·-·
0f00 32 00 31 00 38 00 31 00-33 00 00 00 14 00 60 00 2·1·8·1·3·····`·
0f10 31 00 00 00 00 00 2D 48-B7 A4 10 00 41 44 4D 49 1·····-H·¤··ADMI
0f20 4E 49 7E 31 00 00 48 00-08 00 04 00 EF BE 2D 48 NI~1··H·····ï¾-H
0f30 01 94 2D 48 B7 A4 2A 00-00 00 02 B7 00 00 00 00 ··-H·¤*·········
0f40 0E 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ················
0f50 41 00 64 00 6D 00 69 00-6E 00 69 00 73 00 74 00 A·d·m·i·n·i·s·t·
0f60 72 00 61 00 74 00 6F 00-72 00 00 00 18 00 80 00 r·a·t·o·r·······
0f70 31 00 00 00 00 00 CC 4A-18 BB 11 00 44 4F 43 55 1·····ÌJ·»··DOCU
0f80 4D 45 7E 31 00 00 68 00-08 00 04 00 EF BE 2D 48 ME~1··h·····ï¾-H
0f90 02 94 CC 4A 18 BB 2A 00-00 00 6B 4B 02 00 00 00 ··ÌJ·»*···kK····
0fa0 07 00 00 00 00 00 00 00-00 00 3E 00 00 00 00 00 ··········>·····
0fb0 44 00 6F 00 63 00 75 00-6D 00 65 00 6E 00 74 00 D·o·c·u·m·e·n·t·
0fc0 73 00 00 00 40 00 73 00-68 00 65 00 6C 00 6C 00 s···@·s·h·e·l·l·
0fd0 33 00 32 00 2E 00 64 00-6C 00 6C 00 2C 00 2D 00 3·2·.·d·l·l·,·-·
0fe0 32 00 31 00 37 00 37 00-30 00 00 00 18 00 6A 00 2·1·7·7·0·····j·
0ff0 31 00 00 00 00 00 CC 4A-C7 BB 10 00 46 4F 52 45 1·····ÌJÇ»··FORE
1000 4E 53 7E 31 00 00 52 00-08 00 04 00 EF BE CC 4A NS~1··R·····ï¾ÌJ
1010 15 BB CC 4A C7 BB 2A 00-00 00 04 EC 00 00 00 00 ·»ÌJÇ»*····ì····
1020 08 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ················
1030 46 00 6F 00 72 00 65 00-6E 00 73 00 69 00 63 00 F·o·r·e·n·s·i·c·
1040 20 00 44 00 6F 00 63 00-75 00 6D 00 65 00 6E 00  ·D·o·c·u·m·e·n·
1050 74 00 73 00 00 00 18 00-EE 00 32 00 00 00 00 00 t·s·····î·2·····
1060 00 00 00 00 80 00 54 65-78 61 73 20 48 6F 75 73 ······Texas Hous
1070 65 20 6F 66 20 52 65 70-72 65 73 65 6E 74 61 74 e of Representat
1080 69 76 65 73 20 20 48 6F-75 73 65 20 4D 65 6D 62 ives  House Memb
1090 65 72 73 2E 68 74 6D 00-90 00 08 00 04 00 EF BE ers.htm·······ï¾
10a0 00 00 00 00 00 00 00 00-2A 00 00 00 00 00 00 00 ········*·······
10b0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ················
```

Cursor pos = 0; clus = 3261831; log sec = 26094648; phy sec = 26301496

Fig. 11. Administrators jump list entry for "Texas House of Representatives.html".

Figure 11 shows that from a jump list entry, an investigator can uncover the user who accessed the file, the file name and the original path. Each jump list entry has a corresponding application ID, these ID's remain fairly static and can easily be looked up. In this research, a jump list entry for the network reconnaissance tool NMAP was discovered, which pointed to a file titled "Research.xml". It was determined that this file had been deleted from the file system, hence the need to interrogate the $Recycle. Bin directory. The deleted file was recovered and an analysis of this files showed that the active user launched NMAP and carried out a network scan of "scanme.Nmap.org". This is piece of the entire puzzle that allows a forensic investigator to determine the "who," "what," "when," "where," and "how" of an incident.

4.8 Installed Application Artifacts

When beginning an investigation, one thing the investigator will want to do is gather a list of all installed applications. This will provide several pieces of information that assist with tooling decisions. Certain forensic tools are used for certain types of applications and artifacts. LUTTGENS, PEPE and MANDIA provide an overview of the directories in which you can recover artifacts from installed applications; these include the default application installation directory, default application data directories, registry uninstall information, and default registry configuration data locations [25]. Using FTK, the "Uninstall" registry key was recovered. The recovered key lists the currently installed application, by expanding one of these entries an investigator can view several pieces of information about the application, such as: instance ID, help link, install source, and display name.

Such information can be used by the investigator to better select the tools they will use throughout the remainder of the investigation. The list of installed applications, combined with the timeline recovered from the Windows Security Event logs and the Prefetch file artifacts, will assist in defining what the attacker did during the refined timeline, as well as what other activity may have been taking place on that machine. For example, using FTK, it was determined that both NMAP and OphCrack had been installed and run. Nmap is a popular network reconnaissance tool used for mapping out target networks, and OphCrack is a popular password cracking tool. This provides insight into the user's activities and can point to other artifacts.

4.9 Windows Task Scheduler Artifacts

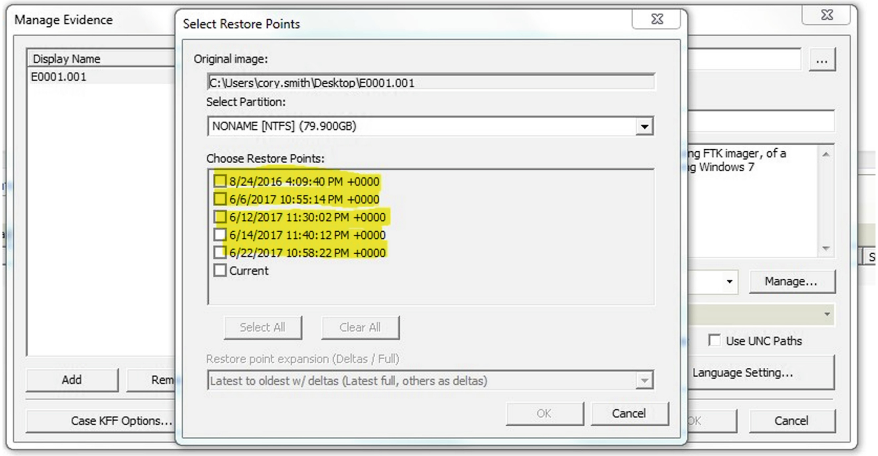
Similar to the way the UNIX Operating system schedules reoccurring tasks through the Crontab, Windows can schedule reoccurring tasks using ".job" files; this is done through the Windows Task Scheduler [48]. These tasks can be rule-based, time-based, or state-based [46]. These ".job" files can be found easily using FTK in "C:\Windows\System32\Tasks" [25].

An attacker can use scheduled tasks for many things: creating backdoors, adding and removing users, modifying accesses, or cleaning up files and directories after the attack. Recovering these files can provide information as to what was ran after the attack, what may have been scheduled in the past to enable the attack, and any jobs that have yet to run and should be prevented.

4.10 Windows Restoration Point Artifacts

Microsoft introduced the Volume Shadow Copy Service (VSS) in Windows Server 2003. This service is used to generate backups of application and operating system data, known as restoration points [53]. VSS provides the ability to generate reoccurring backups of data to protect against the potential loss of data in the future. Should a system be compromised, it can be restored from the latest shadow copy created. There are several activities that can trigger a shadow copy to be created; such as an update, program installation, or scheduled task [25].

During an investigation, an investigator may come across the remnants of a critical file or application that has since been deleted. By restoring a shadow copy from around the time the file or application was created or installed, they may be able to recover the full contents of the file. In my research, it was proven that volume shadow copy recovery with FTK was simple, allowing an investigator to interrogate the shadow copy file system as if it was the true hard disk. Figure 12 shows the number of shadow copies that were available to restore from a single VMDK file. Once restored, an investigator could then interrogate the file system for that shadow copy.



Manage Evidence

Display Name

E0001.001

Select Restore Points

Original image:

C:\Users\cory.smith\Desktop\E0001.001

Select Partition:

NONAME [NTFS] (79.900GB)

Choose Restore Points:

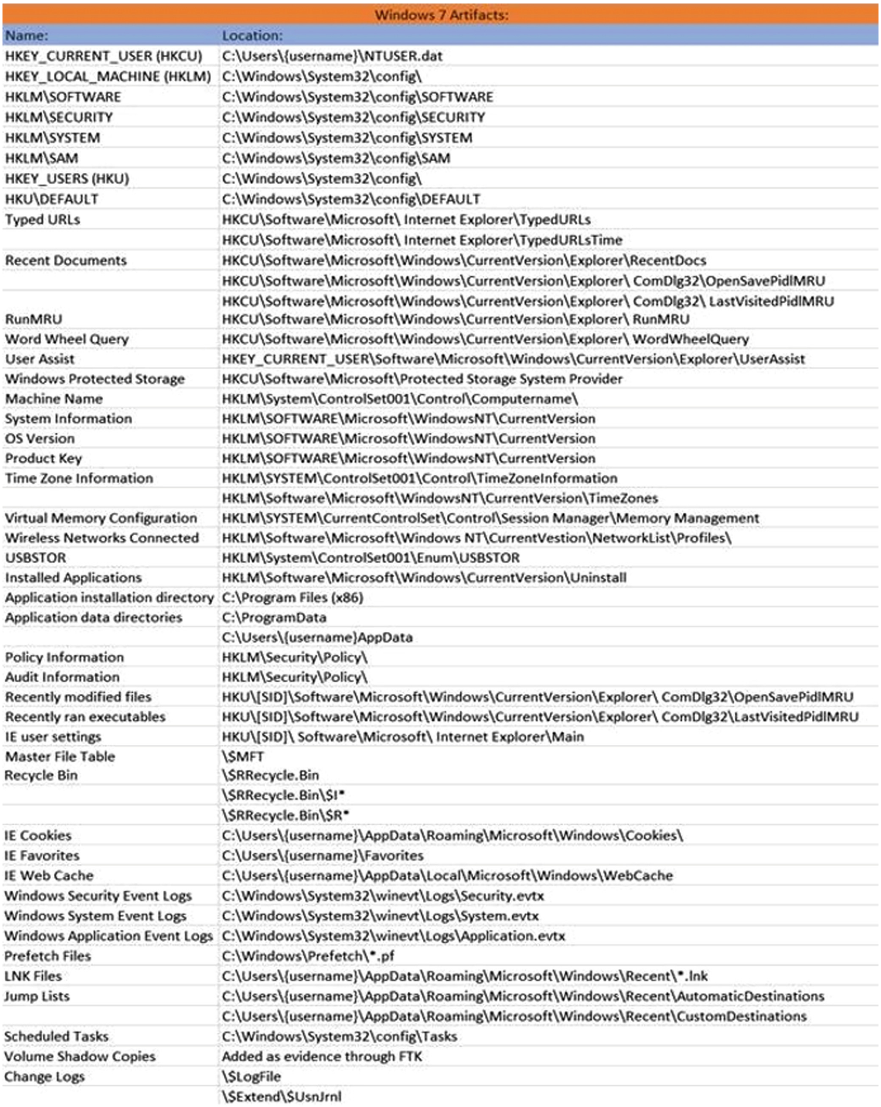
□ 8/24/2016 4:09:40 PM +0000
□ 6/6/2017 10:55:14 PM +0000
□ 6/12/2017 11:30:02 PM +0000
□ 6/14/2017 11:40:12 PM +0000
□ 6/22/2017 10:58:22 PM +0000
□ Current

Restore point expansion (Deltas / Full)

Latest to oldest w/ deltas (Latest full, others as deltas)

□ Use UNC Paths

Fig. 12. Mountable volume shadow copies from virtual machine disk file.



Windows 7 Artifacts:	
Name:	Location:
HKEY_CURRENT_USER (HKCU)	C:\Users\{username}\NTUSER.dat
HKEY_LOCAL_MACHINE (HKLM)	C:\Windows\System32\config\
HKLM\SOFTWARE	C:\Windows\System32\config\SOFTWARE
HKLM\SECURITY	C:\Windows\System32\config\SECURITY
HKLM\SYSTEM	C:\Windows\System32\config\SYSTEM
HKLM\SAM	C:\Windows\System32\config\SAM
HKEY_USERS (HKU)	C:\Windows\System32\config\
HKU\DEFAULT	C:\Windows\System32\config\DEFAULT
Typed URLs	HKCU\Software\Microsoft\ Internet Explorer\TypedURLs
	HKCU\Software\Microsoft\ Internet Explorer\TypedURLsTime
Recent Documents	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs
	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ ComDlg32\OpenSavePidlMRU
	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ ComDlg32\ LastVisitedPidlMRU
RunMRU	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ RunMRU
Word Wheel Query	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ WordWheelQuery
User Assist	HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist
Windows Protected Storage	HKCU\Software\Microsoft\Protected Storage System Provider
Machine Name	HKLM\System\ControlSet001\Control\Computername\
System Information	HKLM\SOFTWARE\Microsoft\WindowsNT\CurrentVersion
OS Version	HKLM\SOFTWARE\Microsoft\WindowsNT\CurrentVersion
Product Key	HKLM\SOFTWARE\Microsoft\WindowsNT\CurrentVersion
Time Zone Information	HKLM\SYSTEM\ControlSet001\Control\TimeZoneInformation
	HKLM\Software\Microsoft\WindowsNT\CurrentVersion\TimeZones
Virtual Memory Configuration	HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management
Wireless Networks Connected	HKLM\Software\Microsoft\Windows NT\CurrentVestion\NetworkList\Profiles\
USBSTOR	HKLM\System\ControlSet001\Enum\USBSTOR
Installed Applications	HKLM\Software\Microsoft\Windows\CurrentVersion\Uninstall
Application installation directory	C:\Program Files (x86)
Application data directories	C:\ProgramData
	C:\Users\{username}AppData
Policy Information	HKLM\Security\Policy\
Audit Information	HKLM\Security\Policy\
Recently modified files	HKU\[SID]\Software\Microsoft\Windows\CurrentVersion\Explorer\ ComDlg32\OpenSavePidlMRU
Recently ran executables	HKU\[SID]\Software\Microsoft\Windows\CurrentVersion\Explorer\ ComDlg32\LastVisitedPidlMRU
IE user settings	HKU\[SID]\ Software\Microsoft\ Internet Explorer\Main
Master File Table	\$MFT
Recycle Bin	\$RRecycle.Bin
	\$RRecycle.Bin\$I*
	\$RRecycle.Bin\$R*
IE Cookies	C:\Users\{username}\AppData\Roaming\Microsoft\Windows\Cookies\
IE Favorites	C:\Users\{username}\Favorites
IE Web Cache	C:\Users\{username}\AppData\Local\Microsoft\Windows\WebCache
Windows Security Event Logs	C:\Windows\System32\winevt\Logs\Security.evtx
Windows System Event Logs	C:\Windows\System32\winevt\Logs\System.evtx
Windows Application Event Logs	C:\Windows\System32\winevt\Logs\Application.evtx
Prefetch Files	C:\Windows\Prefetch\*.pf
LNK Files	C:\Users\{username}\AppData\Roaming\Microsoft\Windows\Recent\*.lnk
Jump Lists	C:\Users\{username}\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations
	C:\Users\{username}\AppData\Roaming\Microsoft\Windows\Recent\CustomDestinations
Scheduled Tasks	C:\Windows\System32\config\Tasks
Volume Shadow Copies	Added as evidence through FTK
Change Logs	\$LogFile
	\$Extend\$UsnJrnl

Fig. 13. Summary of Windows 7 artifacts.

5 Conclusion

The fast-paced changes in computing require a continual increase in the knowledge of digital forensics and the ability to apply this knowledge to new technologies. Virtualized computing has become the new norm, and the ability to perform a forensics investigation against a VM is critical in today's incident response process. During an investigation, those artifacts that identify who acted on a system, what they carried out, and when it was done must be recovered to develop an activity timeline. Understanding what was done, and how it was done is crucial to beginning the next steps of the incident response process. Once all the pertinent artifacts have been recovered, responders can begin developing a remediation plan as well as designing security controls to protect them from similar events in the future.

This research focused on the identification of forensic artifacts, and their locations, in virtualized computing to provide foundational knowledge to future digital forensic investigations. Specifically, this research described the process of gathering digital evidence from the virtual machine disk file, creating forensic images, and interrogating the NTFS file system. A detailed list of artifacts recovered within the timeline of this research was also presented.

Figure 13 documents forensic artifacts along with their locations that were recovered throughout this research. This is not a definitive list of artifacts that can be recovered from the Windows OS, rather a listing of artifacts that were recoverable within the strict timeline of this research, and should serve as the foundation for a digital forensics investigation against a VM running Windows 7.

As forensic tools progress, more artifacts will potentially be recovered. There is still much research that can be done pertaining to digital forensics in virtualized environments, such as the ability to recover a deleted VMDK file from the physical host to recreate a VM and provide a forensic image for investigation and the identification of forensic artifacts from virtual machines running different operating systems.

References

1. Admin. Password Recovery. Password Recovery RSS. Top Password Software, Inc., 31 May 2013. https://www.top-password.com/blog/tag/windows-samregistry-file/. Accessed 11 July 2017
2. Atkison, T., Cruz, J.C.F.: Digital Forensics on a Virtual Machine. Rep. (n.d.). http://atkison.cs.ua.edu/papers/ACMSE11_JF.pdf. Accessed 18 July 2017
3. Aziz, A.S.A., Fouad, M.M., Hassanien, A.E.: Cloud computing forensic analysis: trends and challenges. In: Hassanien, A.E., Fouad, M.M., Manaf, A.A., Zamani, M., Ahmad, R., Kacprzyk, J. (eds.) Multimedia Forensics and Security. ISRL, vol. 115, pp. 3–23. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-44270-9_1
4. Martini, B., Choo, K.-K.R.: Remote programmatic vCloud forensics: a six-step collection process and a proof of concept. In: Proceedings of 13th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2014, pp. 935–942 (2014)
5. Kleyman, B.: Hypervisor 101: Understanding the Virtualization Market. Data Center Knowledge. Penton, 03 August 2012. http://www.datacenterknowledge.com/archives/2012/08/01/hypervisor-101-a-lookhypervisor-market/. Accessed 14 June 2017
6. Birk, D., Christoph, W.: Technical Issues of Forensic Investigations in Cloud Computing Environments. Rep. (n.d.)
7. Eterovic-Soric, B., Choo, K.-K.R., Mubarak, S., Ashman, H.: Windows 7 antiforensics: a review and a novel approach. J. Forensic Sci. **62**(4), 1054–1070 (2017)

8. Esposito, C., Castiglione, A., Pop, F., Choo, K.-K.R.: Challenges of connecting edge and cloud computing: a security and forensic perspective. IEEE Cloud Comput. **4**(2), 13–17 (2017)
9. Quick, D., Choo, K.-K.R.: Digital droplets: microsoft SkyDrive forensic data remnants. Future Gener. Comput. Syst. **29**(6), 1378–1394 (2013)
10. Quick, D., Choo, K.-K.R.: Dropbox analysis: data remnants on user machines. Digit. Invest. **10**(1), 3–18 (2013)
11. Quick, D., Choo, K.-K.R.: Forensic collection of cloud storage data: does the act of collection result in changes to the data or its metadata? Digit. Invest. **10**(3), 266–277 (2013)
12. Quick, D., Choo, K.-K.R.: Google drive: forensic analysis of data remnants. J. Netw. Comput. Appl. **40**, 179–193 (2014)
13. Quick, D., Choo, K.-K.R.: Pervasive social networking forensics: intelligence and evidence from mobile device extracts. J. Netw. Comput. Appl. **86**, 24–33 (2017)
14. Dean, B.: Best Practices in Browser Forensics. IANS. IANS (n.d.). https://www.iansresearch.com/insights/reports/best-practices-in-browser-forensics. Accessed 15 June 2017
15. Digital Evidence and Forensics. National Institute of Justice, 14 April 2016. https://www.nij.gov/topics/forensics/evidence/digital/Pages/welcome.aspx. Accessed 23 July 2017
16. Disabling Prefetch. Microsoft Developer Network. Microsoft (n.d.). https://msdn.microsoft.com/en-us/library/ms940847(v=winembedded.5).aspx. Accessed 18 July 2017
17. Dkovar. Dkovar/analyzeMFT. GitHub. GitHub, Inc., 16 July 2017. https://github.com/dkovar/analyzeMFT. Accessed 13 July 2017
18. Dykstra, J., Sherman, A.T.: Understanding Issues in Cloud Forensics: Two Hypothetical Case Studies. Rep. (2011)
19. Forensic Analysis of Prefetch Files in Windows. Magnet Forensics Inc. Magnet Forensics, 6 August 2014. https://www.magnetforensics.com/computerforensics/forensic-analysis-of-prefetch-files-in-windows/. Accessed 15 July 2017
20. Forensic Toolkit (FTK). AccessData (n.d.). http://accessdata.com/products-services/forensic-toolkit-ftk. Accessed 15 July 2017
21. FTK BootCamp Windows 7 Forensics - Recycle Bin. AccessData (n.d.). http://accessdata.com/. Accessed 16 July 2017
22. How To: Access the Application Event Log. Microsoft TechNet. Microsoft (n.d.). https://technet.microsoft.com/en-us/library/ms166507(v=sql.90).aspx. Accessed 19 July 2017
23. How to Clear Cache, Cookies and History. What Is Cache, Cookies, and History and How Do You Clear Them… Content (n.d.). http://www.pgcconline.com/technicalSupport/clearCache/clearCache.html. Accessed 17 July 2017
24. How to View the System Log in Event Viewer. Microsoft TechNet. Microsoft (n.d.). https://technet.microsoft.com/en-us/library/aa996634(v=exchg.65).aspx. Accessed 19 July 2017
25. Luttgens, J., Pepe, M., Mandia, K.: Incident Response & Computer Forensics, 3rd edn. McGraw-Hill/Osborne, New York (2014)
26. Jensen, C.: FTK Imager User Guide. AccessData, Lindon, 21 March 2012
27. Jensen, C.: FTK User Guide. AccessData, Lindon, 21 January 2015
28. Choo, K.-K.R., Esposito, C., Castiglione, A.: Evidence and forensics in the cloud: challenges and future research directions. IEEE Cloud Comput. **4**(3), 14–19 (2017)
29. Choo, K.-K.R., Herman, M., Iorga, M., Martini, B.: Cloud forensics: state-of-the-art and future directions. Digit. Invest. **18**, 77–78 (2016)
30. Lee, R.: SANS Digital Forensics and Incident Response Blog. SANS Digital Forensics and Incident Response Blog | New Windows Forensics Evidence of Poster Released | SANS Institute. SANS Institute, 04 June 2015. https://digitalforensics.sans.org/blog/2015/06/04/new-windows-forensics-evidence-of-poster-released. Accessed 18 June 2017

31. Master File Table. Master File Table (Windows). Microsoft (n.d.). https://msdn.microsoft.com/en-us/library/windows/desktop/aa365230(v=vs.85).aspx. Accessed 12 July 2017
32. Cahyani, N.D.W., Martini, B., Choo, K.-K.R., Muhammad Nuh Al-Azhar, A.K.B.P.: Forensic data acquisition from cloud-of-things devices: windows smartphones as a case study. Concurr. Comput.: Pract. Exp. **29**(14) (2017)
33. Cahyani, N.D.W., Ab Rahman, N.H., Glisson, W.B., Choo, K.-K.R.: The role of mobile forensics in terrorism investigations involving the use of cloud storage service and communication apps. MONET **22**(2), 240–254 (2017)
34. Ab Rahman, N.H., Cahyani, N.D.W., Choo, K.-K.R.: Cloud incident handling and forensic-by-design: cloud storage as a case study. Concurr. Comput.: Pract. Exp. **29**(14) (2017)
35. Ab Rahman, N.H., Glisson, W.B., Yang, Y., Choo, K.-K.R.: Forensic-by-design framework for cyber-physical cloud systems. IEEE Cloud Comput. **3**(1), 50–59 (2016)
36. Predefined Keys. Predefined Keys (Windows). Microsoft (n.d.). https://msdn.microsoft.com/en-us/library/windows/desktop/ms724836(v=vs.85).aspx. Accessed 11 July 2017
37. Product Downloads. AccessData (n.d.). http://accessdata.com/product-download/registry-viewer-1.8.1.3. Accessed 14 July 2017
38. Do, Q., Martini, B., Looi, J., Wang, Y., Choo, K.-K.R.: Windows event forensic process. In: Peterson, G., Shenoi, S. (eds.) DigitalForensics 2014. IAICT, vol. 433, pp. 87–100. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44952-3_7
39. Registry Hives. Registry Hives (Windows). Microsoft (n.d.). https://msdn.microsoft.com/en-us/library/windows/desktop/ms724877(v=vs.85).aspx. Accessed 16 July 2017
40. RightScale 2017 State of the Cloud Report. Rep. RightScale, Inc (n.d.). http://assets.rightscale.com/uploads/pdfs/RightScale-2017-State-of-the-Cloud-Report.pdf?mkt_tok=eyJpIjoiTjJOaE1qTm1aRFJoTm1ZeSIsInQiOiJGQlB2WklLRWp4OFU1Mm1FS1dzRW9DOFQwaXhuT0lPYVlzcktCMmdUeEVaRk84dTlGQnFIaWNxM0k0WnNIaUgyS2ZRdGs3Nk9hUFZNeXFJVU94ZmFRdU55ZVB5NzF5WjNRQXUrbW1INlhLTUtYdEY5bmdtbFJ3VVFQbXV0YWczNCJ9. Accessed 10 June 2017
41. McKemmish, R.: What is forensic computing? Trends Issues Crime Crim. Justice **118**, 1–6 (1999)
42. Pokharel, S., Choo, K.-K.R., Liu, J.: Mobile cloud security: an adversary model for lightweight browser security. Comput. Stand. Interfaces **49**, 71–78 (2017)
43. Shavers, B.: Virtual Forensics: A Discussion of Virtual Machines Related to Forensic Analysis. Rep. Virtual Forensics (n.d.). https://www.forensicfocus.com/downloads/virtual-machines-forensics-analysis.pdf. Accessed 24 June 2017
44. Stam, M.: Lab FTK Imager: File Carving Using the MFT. 8 Bits. Techblog, 09 October 2009. http://stam.blogs.com/8bits/2009/10/lab-ftk-imager-file-carvingusing-the-mft-.html. Accessed 10 July 2017
45. Stoll, C.: The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage. Pocket, New York (2005)
46. Task Scheduler. Task Scheduler (Windows). Microsoft (n.d.). https://msdn.microsoft.com/en-us/library/windows/desktop/aa383614(v=vs.85).aspx. Accessed 14 July 2017
47. Tholeti, B.P.: Learn about Hypervisors, System Virtualization, and How It Works in a Cloud Environment. Hypervisors, Virtualization, and the Cloud, 23 September 2011. https://www.ibm.com/developerworks/cloud/library/clhypervisorcompare/. Accessed 10 June 2017
48. 2.4 .JOB File Format. [MS-TSCH]: .JOB File Format. Microsoft (n.d.). https://msdn.microsoft.com/en-us/library/cc248285.aspx. Accessed 19 July 2017
49. Urias, V.E., Young, J.W.: Hypervisor assisted forensics and incident response in the cloud. Publication no. 10.1109. IEEE (2016)
50. Vandeven, S.: Forensic Images: For Your Viewing Pleasure. Publication. SANS Institute (2014)

51. Virtualization Technology & Virtual Machine Software. VMWare. VMware, Inc., 20 July 2017. https://www.vmware.com/solutions/virtualization.html. Accessed 22 July 2017
52. VMware Workstation 5.5. What Files Make Up a Virtual Machine? VMware, Inc (n.d.). https://www.vmware.com/support/ws55/doc/ws_learning_files_in_a_vm.html. Accessed 12 June 2017
53. Volume Shadow Copy Service. Windows Server. Microsoft (n.d.). https://technet.microsoft.com/en-us/library/ee923636(v=ws.10).aspx. Accessed 15 July 2017
54. Welcome to MyKey Technology. MFT Ripper. MyKey Technology Inc (n.d.). http://mftripper.com/. Accessed 18 July 2017
55. WinPrefetchView V1.35. View the Content of Windows Prefetch (.pf) Files. Nir Sofer (n.d.). http://www.nirsoft.net/utils/win_prefetch_view.html. Accessed 16 July 2017
56. Teing, Y.-Y., Dehghantanha, A., Choo, K.-K.R., Yang, L.T.: Forensic investigation of P2P cloud storage services and backbone for IoT networks: BitTorrent Sync as a case study. Comput. Electr. Eng. **58**, 350–363 (2017)

Possible Keyloggers Without Implementing a Keyboard in Android

Itzael Jiménez Aranda(✉), Eleazar Aguirre Anaya, Raúl Acosta Bermejo,
and Ponciano Jorge Escamilla Ambrosio

Instituto Politécnico Nacional - Centro de Investigación en Computación, GAM,
07738 Mexico City, Mexico
itzaelja@gmail.com, eaguirrea@ipn.mx, {racosta,pescamilla}@cic.ipn.mx

Abstract. Like the main input way to introduce information in the majority mobile devices nowadays is the screen, it is the main source where a malware could get private information. A keylogger, in this way could obtain private information. Researches of this type of malware until this moment are focused on the Android architecture application layer, leaving aside the other layers, so a keylogger could also be implemented in another layer and only use the application layer like the insertion method. An analysis of the data flow when a key is pressed on the screen is presented, from the system call by an interruption caused by hardware, the methods involved in this flow and possible generated logs and related files, performing an experimentation procedure to extract information about the keys pressed in order to determine which points can be used to get private information without the necessity of implement a third-party keyboard.

Keywords: Keylogger · Touchlogger · Malware · Android keylogger
Touchscreen

1 Introduction

Nowadays the first option to input information to some mobile device is the screen of the device, so it is the first alternative in order to get private information. Android allows install third-party keyboards, this being something that can be harmful to the user since it can compromise user privacy [1–3]. There are many third-party keyboards that seem to be a simply keyboard with a better design or with extra features, but these could be extracting all information entered from the screen by the user [1].

A keylogger is a software able to record the keys pressed in one system. In mobile devices the key pressed is a virtual key, for that some authors call the keyloggers for the mobile devices as touchloggers. In some cases the keyloggers are used as a legitimate personal or professional IT monitoring tool, but in many cases are used to capture sensitive information, like passwords or financial information, which is then sent to third parties for criminal exploitation [4].

X. Lin et al. (Eds.): SecureComm 2017 Workshops, LNICST 239, pp. 104–112, 2018.
https://doi.org/10.1007/978-3-319-78816-6_8

The research of the keyloggers in the desktop systems is wide but in the mobile systems is the opposite. So that in the Android system it still continues without has a deep exploration. In general the researches studied how a keylogger is implemented in Android, but in all the cases is about a third-party keyboard installed in the system leaving to one side that can be possible that a keylogger can be implemented in some of the other three layers architecture of Android and just use the application layer as the insertion way to the system, an example of insertion can be a Trojan that of course it not be a keyboard. These researches make a Play Store Keyboards analysis for determine possibles keyloggers, also show the permissions requested by the Android keyloggers and the facility to store the keys pressed in some file and then send the file to some server. As well the researches give recommendations in order to make people aware in how they can avoid this kind of mobiles devices threats [1–3,5].

This work presents an Android keyboard data flow analysis with an experimentation procedure for determine which points can be used by the malware developer for implement a keylogger without the necessity of implement a third-party keyboard.

This article is organized as follow: Sect. 2 describes the research work related to the study of a keylogger on Android. Section 3 describes the problem statement. Section 4 describes the analysis done and the results obtained. And finally the Sect. 5 are conclusions and future work.

2 Related Works

As we mention in one paragraph above, much remains to be studied to the Keyloggers in the field of mobile operating systems. In [1–3] carry out a study of keyboards available in the Play Store to determine the number of possible keyloggers, the amount of requested permissions and permission types are taken into account, so these analysis are performed on the Android architecture application layer. About 80% requests Internet permission and writing memory permission. In [2] perform certain questions to mobile application developers. Why ask Internet permission to develop a third-party keyboard, was one of them and the answer was that may be necessary updates, so not because a third-party keyboard asks Internet permission means to be a keylogger, although with a possible potential to be. In [1] a study is done with the Wireshark traffic tool in the network to determine which keyboards requesting Internet permission are actually extracting information, the result was that 7.9% of the applications analyzed (11 of 139) caused network traffic when an email and password were written, although it is mentioned that the other applications could be time bombs and therefore at that time these did not show network traffic.

In [1,2,5] is demonstrated the ease of obtain information through installing a malicious third-party keyboard malicious, storing and sending information to an external server, building a keylogger, to study what types of permits require, what methods at application level are needed to develop the keyboard and so on. All of these researches are focused on the application layer of the android architecture, but can be a possibility that a keylogger is being implemented in some of the others architecture layers.

In [6] make a record of all the mobile device touch logs with a software for realice a characterization between the device and the user, they use the file */dev/input/eventX* as the way to get the logs.

In [7] studies the iOS data flow for a benign touchlogger and malign touchlogger, making a private and public framework hooking related with the iOS keyboard. The benign part is for to know if the mobile device is been using by the owner. However about the malign part is for get private information, with their method they can get the event type and the touch coordinates, so when a specific app is open the tool register the touches for relate them with the keys pressed known the coordinates and the position of the mobile device.

3 Problem Statement

As any input device, the response to an interruption made from the hardware has a flow, which passes through different stages to reach to the application that corresponds the request, so at some stage might exist some vulnerability that can be exploited if it exist, and use it in order to get private user information.

Therefore, a keylogger can be implemented not as a third-party keyboard, that is not directly in the Android application layer, since it could obtain information through some vulnerable point in the flow of data when any virtual key is pressed.

We make an analysis of the processes and files related with the touchscreen studying the touchscreen data flow when a user press the screen until the information reaches the application performing an exploratory experimentation methodology to extract information about the keys pressed in order to determine which points can be used by the malware developer for implement a keylogger without the necessity of implement a third-party keyboard, so as to get private information. This information can be useful for characterization and a detection mechanism.

4 Touch Screen's Data Flow

In [8] mention a summary of the processes in the Android touchscreen, the Fig. 1 shows the data flow. First "EventHub" reads the raw events from the "evdev" driver. After "InputReader" consumes raw events and updates internal processes statements about the position and other characteristics of each tool. Also it maintains the states of the buttons. If a physical or virtual key is pressed, "InputReader" notifies to "InputDispatcher", also "InputReader" determines whether the touch was made within the limits of the screen and if necessary it notifies to "InputDispatcher". "InputDispatcher" uses to WindowsManagerPolicy, to determine whether the event should be attended. Then "InputDispatcher" releases the event to the appropriate application which is in the application layer.

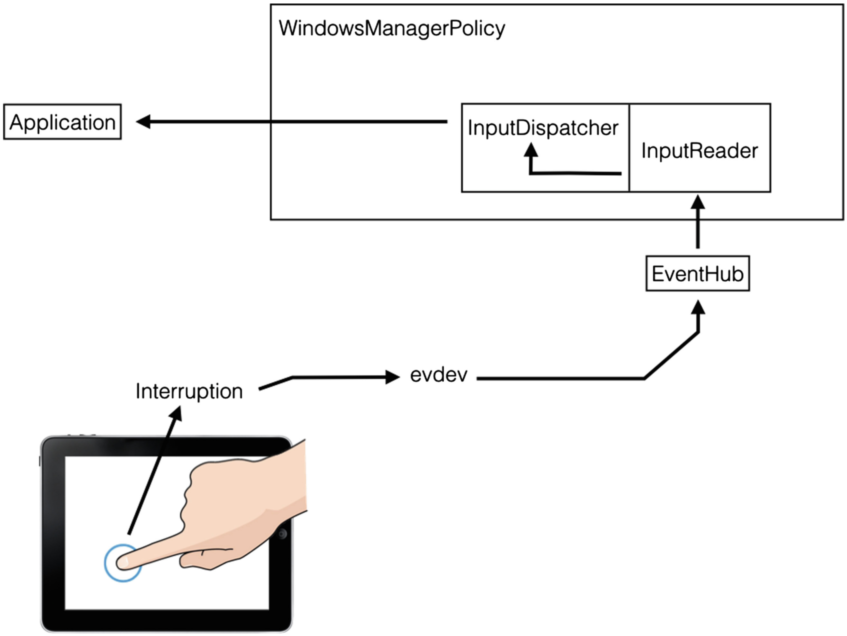


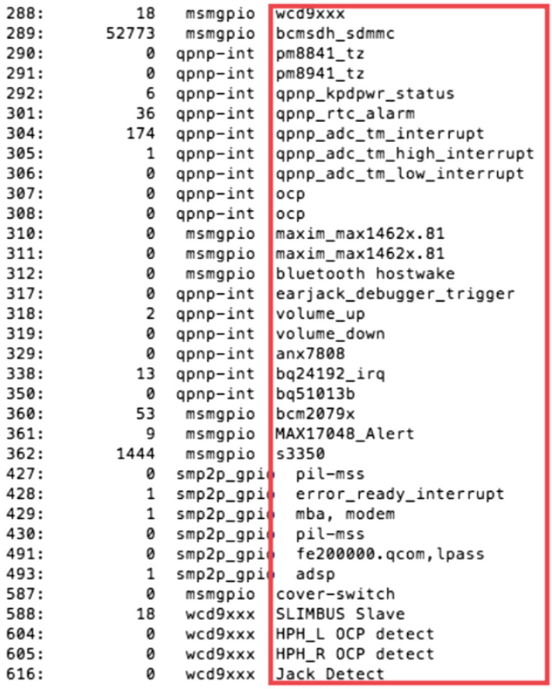
Fig. 1. Data flow when the touch screen is pressed.

For search points where is possible get data about the touchscreen flow is necessary explore the system. Considering the exploratory part is required have full access to the system, and have a native system without unnecessary modifications. For the above the device used is a LG - Nexus 5.

As the driver is the first point where the data of the touchscreen flow pass, this is the start for the exploratory methodology, for analyze its source code and experiment with it and determine if here a malware developer can get information about the keys pressed. As the Android system can be in different brand devices with different I/O devices the Android kernel have different drivers files corresponding to different I/O devices, so is mandatory know the driver which the system is using. Due the driver is loaded like a module to the kernel and it register one function using *request_irq()*, to be able to notify when the user make an interact with the hardware and handle the originated interrupts and the interruption, normally it carry the name of the module loaded in the kernel.

The */proc/interrupts* file provides information about the interrupts functions names of the system as well as the drivers interruptions functions. The Fig. 2 shows a part of how the file provides the information, highlighting the interrupts names, is necessary determine which interruptions correspond to the touchscreen as the names are not clear in a first view.

With the */system/build.prop* file and the Google Git webpage [9] is possible determine and find the system kernel and then the driver source code. So as to determine the driver used by the system we compare the name of each driver in



```
288:         18   msmgpio  wcd9xxx
289:      52773   msmgpio  bcmsdh_sdmmc
290:          0  qpnp-int  pm8841_tz
291:          0  qpnp-int  pm8941_tz
292:          6  qpnp-int  qpnp_kpdpwr_status
301:         36  qpnp-int  qpnp_rtc_alarm
304:        174  qpnp-int  qpnp_adc_tm_interrupt
305:          1  qpnp-int  qpnp_adc_tm_high_interrupt
306:          0  qpnp-int  qpnp_adc_tm_low_interrupt
307:          0  qpnp-int  ocp
308:          0  qpnp-int  ocp
310:          0   msmgpio  maxim_max1462x.81
311:          0   msmgpio  maxim_max1462x.81
312:          0   msmgpio  bluetooth hostwake
317:          0  qpnp-int  earjack_debugger_trigger
318:          2  qpnp-int  volume_up
319:          0  qpnp-int  volume_down
329:          0  qpnp-int  anx7808
338:         13  qpnp-int  bq24192_irq
350:          0  qpnp-int  bq51013b
360:         53   msmgpio  bcm2079x
361:          9   msmgpio  MAX17048_Alert
362:       1444   msmgpio  s3350
427:          0  smp2p_gpi   pil-mss
428:          1  smp2p_gpi   error_ready_interrupt
429:          1  smp2p_gpi   mba, modem
430:          0  smp2p_gpi   pil-mss
491:          0  smp2p_gpi   fe200000.qcom,lpass
493:          1  smp2p_gpi   adsp
587:          0   msmgpio  cover-switch
588:         18   wcd9xxx  SLIMBUS Slave
604:          0   wcd9xxx  HPH_L OCP detect
605:          0   wcd9xxx  HPH_R OCP detect
616:          0   wcd9xxx  Jack Detect
```

Fig. 2. Some system interrupts, marked in red their names. (Color figure online)

the specific kernel in the Google Git with the name of each interruption, but in our case it is not possible perform a relation because neither interrupts names match with the name of the drivers names in the Google Git. So it was not possible decide which driver is used by our system and explore its source code in order to find if a malware could be getting information about the keys pressed. As the driver could not be determined, the decision taken is to explore the first contacts in the user space with the touchscreen. The volatile memory is the first contact that has the process of the keyboard and it is one of the elements in the user space that interact with the touchscreen, so is made an exploration of the volatile memory because possibly it store the keys pressed. Using the adb tool information about the processes executed in the system is got. When the keyboard is being executed, is showed which process ID (PID) correspond to it and its package name, in our case is com.google.android.inputmethod.latin. Knowing the PID we can search the directory and files and analyze them.

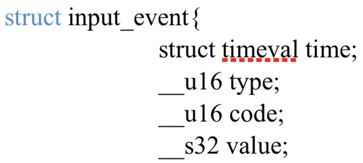
At the process directory there are several files related with the executed process, maps file provides information about the memory section assigned to the process, mem file provides information about memory held by this process, status provides information about the memory and about the process, like the name, PID and so on.

Since the */proc/PID/maps* file provides which memory sectores are assigned to the process, can be made a dump data on this sectors of memory using the mem file. Considering that in the file */proc/PID/status* provides the name of the keyboard process and the PID assigned to it, can be made a scanning of each proc directory and read the status file in order to found which of them corresponds to the keyboard and make a memory dump. A tool is developed in order to make the memory dump to the keyboard process for search the keys pressed stored in the memory, the flowchart 1 describe how the tool works. However the results show that there are not a plain text in the memory about the keys we press in the keyboard. Continuing in the user space and considering that also when the driver is loaded into the kernel, it calls the function *input_register_device()* because it needs to indicate the creation of the file */dev/input/eventX* (where X is only an integer) that corresponds to the physical device. Is determined which *eventX* file corresponds to the touchscreen exploring the system, for analyze this file and determine if it can have information about the keys pressed.

The file corresponding to the touchscreen is the *event1*, the */proc/bus/input/devices* file provides this information. Trying to read the *event1* file we notice that it always is empty and only when an event occurs it has information but only for one instant. The Fig. 4 shows a hexadecimal representation of the data when the touchscreen is pressed, that is when the event occurs.

When an interrupt occurs the kernel needs to process it and with different functions in the include*/linux/input.h*, for instance *input_event(...)*, *input_report_abs(...)* and so on, the data is put in a standard format in the */dev/input/eventX* file in order to be accessed by the user space.

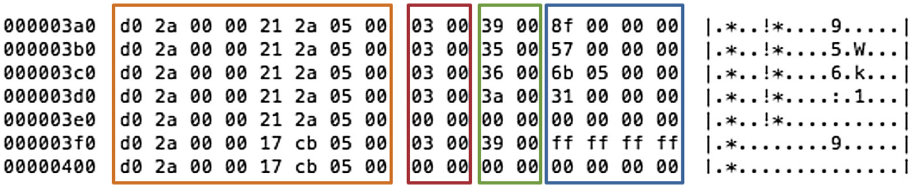
The *include/linux/input.h* provides information about the event standard format, it is a struct and has the next variables: time stamp, type, code and value [8], the Fig. 3 shows the code. Also *include/linux/input.h* file provides information about the meaning of each types and code values.



```
struct input_event{
                struct timeval time;
                __u16 type;
                __u16 code;
                __s32 value;
```

Fig. 3. Standard event format.

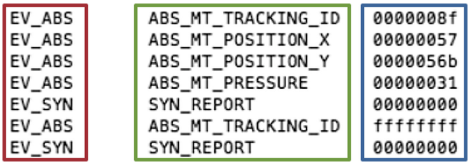
The hexadecimal data in Fig. 4 needs to be read from right to left for each hexadecimal value so as to understand them. The blue square are the values, for instance, the first value is *0000008f*, the green square are the codes, where according to */include/linux/input.h* the *0039* is the ABS_MT_TRACKING_ID which indicates the ID of the touch realized in that moment, the *0035* is the ABS_MT_POSITION_X which indicates the x coordinate of the touch, the *0036* is the ABS_MT_POSITION_Y which indicates the y coordinate of the touch, the *003a* is the ABS_MT_PRESSURE which indicates the pressure of the touch and *0000* is the SYN_REPORT which indicates the end of the report. The red square are the events type, where according to */include/linux/input.h* the *0003* means EV_ABS which indicates a touchscreen absolute event, and *0000* means EV_SYN which indicates a synchronize event. And finally the orange square indicates the timestamp.



```
000003a0  d0 2a 00 00 21 2a 05 00  03 00 39 00 8f 00 00 00  |.*..!*....9.....|
000003b0  d0 2a 00 00 21 2a 05 00  03 00 35 00 57 00 00 00  |.*..!*....5.W...|
000003c0  d0 2a 00 00 21 2a 05 00  03 00 36 00 6b 05 00 00  |.*..!*....6.k...|
000003d0  d0 2a 00 00 21 2a 05 00  03 00 3a 00 31 00 00 00  |.*..!*....:.1...|
000003e0  d0 2a 00 00 21 2a 05 00  00 00 00 00 00 00 00 00  |.*..!*..........|
000003f0  d0 2a 00 00 17 cb 05 00  03 00 39 00 ff ff ff ff  |.*........9.....|
00000400  d0 2a 00 00 17 cb 05 00  00 00 00 00 00 00 00 00  |.*..............|
```

Fig. 4. Reading the */dev/input/eventX* file with the hexdump command. Timestamps (orange square), events type (red square), codes (green square) and the values (blue square). (Color figure online)

With the command *getevent -l* the above interpretation in accord with the input.h documentation can be verify to be sure that the exploration was correct. This can be check with the Figs. 4 and 5.

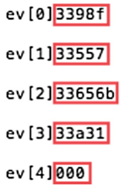


```
EV_ABS    ABS_MT_TRACKING_ID  0000008f
EV_ABS    ABS_MT_POSITION_X   00000057
EV_ABS    ABS_MT_POSITION_Y   0000056b
EV_ABS    ABS_MT_PRESSURE     00000031
EV_SYN    SYN_REPORT          00000000
EV_ABS    ABS_MT_TRACKING_ID  ffffffff
EV_SYN    SYN_REPORT          00000000
```

Fig. 5. Information showed with the getevent command. Events type (red square), codes (green square) and the values (blue square). (Color figure online)

A tool is made to experiment with this file with the purpose of get the information mentioned, because the file provides information about coordinates of the touches and can be used to determine if a key was pressed. The tool open the file */dev/input/event1*, and knowing the struct format, a same buffer struct needs to be indicate and the data can be acceded by specifying the elements in the struct in order to get the same data on the */dev/input/event1* file. The Fig. 6 shows how the software is able to take the event, key and value parameters. Due that in this file are given the coordinates, it could be used in order to make a keylogger, handling the data so as to get keys pressed.

Determine which touchscreen driver correspond to the device could be difficult because depends of different factors like the device itself and the kernel



```
ev[0]3398f

ev[1]33557

ev[2]33656b

ev[3]33a31

ev[4]000
```

Fig. 6. Extracting data from the */dev/input/eventX* file.

version, so it is difficult to be able to get information about the keys pressed as it difficult identify the current driver. Modify the kernel would allow to get information from the touchscreen's physical file and make it available to any app, like adding instructions to create a copy file without restriction permissions of the physical device file or even could be added a module in order to get data from the physical device file and put it in another file available to the apps, this options also depends in different factors like a device *rooted*, option module add enabled and user interaction. Getting information from the volatile memory also seems difficult for a malware, first because the malware needs get administrator privileges and second the data here is dynamic and the memory assigned to one process has a lot of data, make a software able to interpret this data and match some of these data with some specific characteristics and then get information will take a lot of resources and a malware doing this would make it simple to detect. However looks like that is possible extract information in the touchscreen's physical file only getting administrator privileges path due this has information about the touches coordinates.

Since there is a data flow to process the touches in the touchscreen is possible extract data about that touches, until the moment from one point, to maybe extract sensitive user information.

As we now have covered the kernel and a little the user space, the next is analyze the functions with a relation with the touchscreen and the keys pressed like *EventHub*, *InputReader*, *InputDispatcher* and so on.

5 Conclusion and Future Work

With the exploratory methodology is possible find points in the system where information about the keys pressed could be extracted, and making some experiments like examine and handling the data could be determined if the information has a relation with the keys pressed. This work shows that is possible get some information about the key pressed outside the Android application layer using the touchscreen physical file, and still without covering the full touchscreen's data flow, thus is possible that a malware is able to obtain information about the keys pressed without implement a third-part keyboard in Android.

For future work we are going to implement a tool that handle the data obtained from the *eventX* file in order to determine if is possible get user private information, we are going to analyze if is possible make some kernel modifications to provoke a information leak. Also we are going to continue analyzing the different functions related with the data flow and try to decide if there are points of information leak.

Acknowledgment. The authors are grateful to the Instituto Politcnico Nacional and the Consejo Nacional de Ciencia y Tecnologa by the support to this research.

References

1. Cho, J., Cho, G., Kim, H.: Keyboard or keylogger?: a security analysis of third-party keyboards on Android. In: 13th Annual Conference on Privacy, Security and Trust, pp. 173–176. IEEE (2015)
2. Mohsen, F., Bello-Ogunu, E., Shehab, M.: Investigating the keylogging threat in android—User perspective (Regular research paper). In: Second International Conference on Mobile and Secure Services (MobiSecServ), pp. 1–5. IEEE (2016)
3. Mohsen, F., Shehab, M.: Android keylogging threat. In: 9th International Conference on Collaborative Computing: Networking, Applications and Worksharing, pp. 545–552. IEEE (2013)
4. Kaspersky Lab.: What is a keylogger? http://www.kaspersky.com/au/internet-security-center/definitions/keylogger
5. Nasution, S.M., Purwanto, Y., Virgono, A., Ruriawan, M.F.: Modified kleptodata for spying soft-input keystroke and location based on Android mobile device. In: International Conference on Information Technology Systems and Innovation, pp. 1–5. IEEE (2015)
6. Hirabe, Y., Arakawa, Y., Yasumoto, K.: Logging all the touch operations on Android. In: Seventh International Conference on Mobile Computing and Ubiquitous Networking, pp. 93–94. IEEE (2014)
7. Damopoulos, D., Kambourakis, G., Gritzalis, S.: From keyloggers to touchloggers: take the rough with the smooth. Comput. Secur. **32**, 102–114 (2013)
8. Android open source project: Devices - Input. https://source.android.com/devices/input/index.html
9. Google Git: Git repositories on Android. https://android.googlesource.com/

A New Cyber Security Framework Towards Secure Data Communication for Unmanned Aerial Vehicle (UAV)

Md Samsul Haque(✉) and Morshed U. Chowdhury

School of Information Technology, Deakin University-Burwood Campus, Melbourne, Australia
{mshaq,morshed.chowdhury}@deakin.edu.au

Abstract. Cyber Physical Systems (CPS) like UAVs are used for mission critical tasks including military and civilian operations. Their potentiality of usage is rapidly increasing in commercial space. The need for a secure channel to wirelessly communicate and transfer message between CPS is very crucial. Key idea behind this study is to propose a novel framework that is lightweight, robust and at the same time do not compromise security and pragmatic in the jurisdictions of energy-efficient atmospheres. This paper presents an idea for a practical and efficient hierarchical architecture for UAV network using identity-based encryption. Also, proposes selective encryption technique to reduce overheads and data hiding mechanism to increase confidentiality of the message.

Keywords: Identity-based cryptography · Watermarking · Cyber security
Unmanned aerial vehicle

1 Introduction

The need for cyber security has grown with the growth and expansion of digital tools and technology. The devices we use in our everyday life are now becoming smart and connected to a global network of computers, software systems and communication links called Internet of Things (IoT). Thus, ensuring security of digital data has become a critical challenge. Traditional computer and network security approaches fails to adequately address integrity, confidentiality and availability threats for cyber physical system (CPS) and do not address a unified manner for survivability from malicious intimidations and recoverability from attacks [1]. Recent years, research has explored towards vulnerability of CPS, particularly for Unmanned Aerial Vehicle and ground control systems, but little research has been done in secure trust creation, communication and message transfer. In this paper, we surveyed the available literature and defined a secure framework to enhance security to cyber physical system communication for UAV network. The rest of this paper is organized as follows: Sect. 2 describes background information, and security threat to UAV, Sect. 3 reviews the existing literature, Sect. 4 proposes a solution for the problem, Sect. 5 briefly analyses the performance and security for the proposed framework and finally Sect. 6 concludes and discusses future work.

X. Lin et al. (Eds.): SecureComm 2017 Workshops, LNICST 239, pp. 113–122, 2018.
https://doi.org/10.1007/978-3-319-78816-6_9

2 Background

Cyber physical systems (CPS) are autonomous systems which are the convergence of communication, computing and control systems [2]. There are several uses of CPS, which includes smart grid system, oil and gas distribution networks, advanced communication systems, UAV and smart ground vehicles. UAVs are cyber physical systems that can be controlled remotely from a ground control station or can fly autonomously using on-board computers based on pre-programmed flight plans. They are also intelligent system, able to communicate with its controller and return payload data, capable of automatically take corrective action or automatic decision making during an event [3]. The main elements of an UAV are control elements, wireless and satellite communication link, sensors and actuators. UAVs are resource constraint device. They use batteries for power, however Top Flight Technologies [4] has designed a hybrid gas-electric aircraft that uses both batteries and gasoline, significantly improving its performance.

UAVs were mainly used in defense operational environment but nowadays, they are ubiquitous and their uses are rapidly expanding in commercial, scientific, recreational and other applications. They are used as a major tool [5] for law enforcement agencies, shippers, aerial photographers, farmers, humanitarian agencies, and more. Giant companies such as Amazon, Google are planning to use UAVs for goods and services delivery [6]. The FAA forecast [7] estimates that by 2020 there will be 7 million of unmanned aerial vehicle occupying United States airspace.

With the increase in UAV usage potential risks and security threats also starts to arise. UAVs are potentially easier to hack as because they are designed to have a quick and easy setup and often uses unencrypted communication and data transfer with many ports are still open. Moreover, the unique configuration such as open state of the sensors, wireless network, serially safety structure, etc. makes these devices highly exposed technical systems. In recent years, research has explored cyber security threats to the UAV that are used for defense industry, but little research has been done to explore what additional cyber threats are for the use of commercially available UAVs. Also, much of the security technology and processes are currently being developed without doing a proper threat analysis. Because of utilizing unsecure devices [8] could result in unauthorized disclosure of classified information.

2.1 Cyber Security Threats on UAV

Threats on CPS goes beyond attacking the individual system components. By using a multi-vector attack a skilled attacker exploits the weaknesses of individual components and the combined effect however, may be catastrophic. Security threats on UAV can be on the onboard flight controller and ground control system, sensor, actuator, wireless data link and routing infrastructure. Determining the nature of the vulnerability the attacks can be categorized into three groups: hardware, wireless and sensor spoofing attack [9]. Hardware attack is where attacker has access to the UAV autopilot components directly. In wireless attack the attacks are carried out through one of the wireless communication channels and sensor spoofing attack, is carried out by injecting or passing false data by the miscreant through the on-board GPS channels. In this paper our focus is on the wireless attack to secure wireless data communication channels. An attacker can carry out such attacks from a far distance while the UAV is being operated. The most significant threat of wireless attacks is the fact that an attacker can gain full control of the UAV if the communication protocol is known, and can break the encryption of the communication channel. Successful attack requires breach of at least one of the information security objectives: confidentiality, integrity or availability [10].

Example, of an attack to UAV is deliberately jamming communication link while filming of an Australian triathlon with an UAV. The operator lost complete control over the vehicle, believes that an attacker using a "channel hop" attack intentionally interfered with his operation, causing it to crash into one of the athletes [11]. Another most recent and controversial incidents was that the Iranian forces claimed possession of an RQ 170 Sentinel. One of the theory described that Iranian forces jammed the satellite communication of the UAV and GPS functionality which make it easy to attack the GPS system by sensor spoofing attack [12].

3 Related Literature

Research in communication security is a continuous process. The complex nature in UAV has driven to the domain of new security research. Much of the research has been accomplished on capability, reliability and efficiency of the system in terms of time and power [13].

A hierarchical architecture for wireless sensor network (WSN) based on the Boneh-Franklin algorithm proposed in paper [14]. The author presents a hierarchical key management scheme based on the basic Boneh-Franklin and Diffie-Hellman (DH) algorithms to solve large energy consumption in communication and computation. Identity based hierarchical Key Management Scheme in Tactical Mobile Ad Hoc Networks proposed in paper [15]. Authors offered a technique of key management in distributed hierarchal network. The nodes of hierarchy can get their keys updated either from a threshold sibling or from their parents. The technique of dynamic node selection formulated as a stochastic problem and the proposed scheme can select the best nodes to be used considering their security conditions and energy states.

Cryptography and Steganography are used with enhanced security module in paper [16]. Authors used symmetric encryption algorithm called Advanced Encryption Standard (AES) and image based steganography. A part of the encrypted message is hidden into an image and the unhidden part of the encrypted message will be converted into two secret keys. To decrypt the message one need keys for Cryptography and Steganography, two extra keys and the reverse process of the key generation. The limitations of this paper are that the length of the input and output sequences for the Advanced Encryption Standard (AES) and the proposed framework is a flat network where all users has similar access to data. Paper [17], proposes an approach for securing transmitted message over communication network. It uses symmetric encryption algorithm AES and text based steganography to provide an extra layer of security. The AES provides the initial confidentiality of the secret data and then the encrypted data are represented in binary and then hidden is textual carrier. The AES encryption algorithm uses 256 bits' key for extra security against brute force attacks. This paper is also lack of providing forward and backward security as it is designed for a flat network system.

The security approach described in paper [18, 19] presents a solution for an agent-based model for cyber physical systems by using hierarchical access. Hierarchy is implemented through a public key cryptosystem with divided private key and steganography. The steganalysis and cryptanalysis provides a higher level of security to the original data. Although the proposed approach brings a new perspective for the security of agent-based cyber-physical systems but it lacks implementing the approach to any specific application domain. Different security threats for UAVs System are analyzed and a cyber-security threat model has been proposed in Paper [10] The proposed model help designers and users of the UAV systems to understand cyber-security threat profile of the system and address various system vulnerabilities, identify high priority threats, and select mitigation techniques for these threats. They have also tried to evaluate risk generation by different vulnerabilities to the UAVs system. Although various security threats to a UAV system is analyzed and a cyber-security threat model showing possible attack paths has been proposed on this paper but it is not clear which threats might affect the UAV systems most.

Traditional information security mechanism such as cryptography, intrusion detection method or steganography alone is not sufficient to protect UAV system. More specifically these techniques do not consider the compatibility of the sensor, actuator, communication link measurements of the physical and control mechanisms of UAV, which has a massive importance for the security of cyber physical systems like UAV. Also, typical communication security mechanisms often increase communication latency to unacceptable levels, specifically for real-time systems. UAVs are complex by nature and need to have embedded security functionalities and the security solutions. Because complex infrastructures have different objectives and assumptions concerning what needs to be protected, and have specific applications that are not originally designed for a general IT environment. Therefore, it is necessary to develop unique security solutions for different application and infrastructures to fill the gap.

4 Proposed Solution

Security research for CPS varies depending on the application domain of the system. UAVs are typically resource constrained in terms of computation, communication, energy and storage. So, the Security solutions for UAV data communication must be robust, efficient, and satisfy the real-time requirements. At the same time, it must be lightweight without affecting performance. In this paper, we are proposing a framework to achieve a collective system lightweightness, that does not compromise security and efficiency of the system. We are partly inspired by the concept presented in the research work in paper [20] that proposed the lightweight security enforcement in Cyber-Physical Systems. Our contributions to the knowledge are as follows:

- Distribution of Computing Overheads: Our proposed structure provides lightweightness and security by offloading computationally expensive workloads from resource constrained devices to powerful equipment. Leveraging the architecture of the underlying system and constructing a multilevel structure we can achieve such a framework.

- System Lightweightness: To achieve system lightweightness we are proposing to use a lightweight cryptographic primitive and using selective data encryption technique to attain better system performance and increase efficiency in message transfer without hampering security.
- Obscuring transmitted data and digital data right management: Stenography or data watermarking technique increase the confidentiality during data transfer and integrity of the stored data.

The motivation of the proposed framework is to provide balance between UAVs regarding resource consumption and security by creating a robust and new security architecture.

4.1 Distribution of Computing Overheads

Flying Ad-Hoc Network (FANET), is a new form of network family that can perform their task without human intervention which can complete their job without human intervention [21]. In FANET, the UAVs become node. It consists of two parts, ad-hoc network and one or more access point like a satellite or ground base station (BS). The UAV-to-BS communication, the connection is created with an infrastructure like a ground base or satellite to transfer the data. UAVs are comprised of sensors that use wireless networks for data communication. Wireless sensor network (WSN) facilitate the interaction between base station and the UAV These networks are exposed and unguarded. So, potential interception or eavesdropping can cause security concerns and it is possible for a potential adversary to snoop or fabricate the transmitted information. Also, these sensors are restricted in terms of bandwidth, energy, computing power, storage, and memory. These constrained resources nature make it impractical for WSNs to deploy traditional security schemes to transmit data between UAVs. Moreover next-generation UAVs will use more and more mortification sensors and actuators that will be dynamic and long lasting. Therefore, we are proposing a multilevel hierarchical system for data transfer that distributes computing overhead in FANET and at the same time ensures the independence and security of the sub-networks. Figure 1, shows a hierarchical architecture of UAV network for overhead distribution.

Hierarchical system is organized into a cluster or groups. Each cluster performs the operations in specific areas, with a cluster head (CH) elected for every cluster routinely and dynamically. The approach in hierarchical system is different than a classical flat network system in which all cluster members have the same access rights. In hierarchical network systems access to information brings a new level of security to the system. Information can only be viewed by those who have access to it. The CH is superior to ordinary sensor nodes. They have more computational ability, storage, memory, and energy and battery power. Cluster heads performs tasks such as aggregating information from the ordinary cluster members, processing data within the cluster, forwarding the data to base station and leading the cluster to the destination [22].

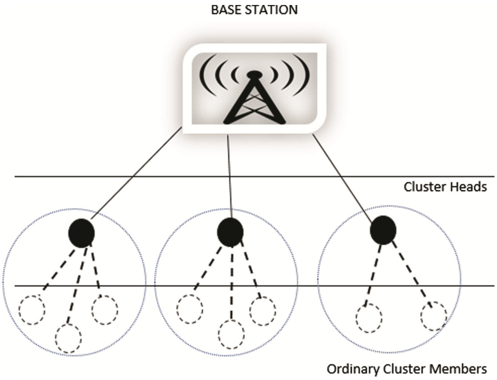


Fig. 1. Hierarchical UAV network architecture for overhead distribution

4.2 System Lightweightness

Embedded systems like UAV, suffer from limited resources in different areas including hardware, energy consumption, and bandwidth usage. This leads to the design and implementation of a framework that uses security primitives to reduce these overheads. Lightweight cryptographic primitives are preferred security methods over generic designs for constrained resources implementations. Cryptography algorithms scramble the secret data in such a way that it is unreadable by a third party. They are generally classified into asymmetric and symmetric key encryption algorithms. With Symmetric key cryptography, only one key is used for both encryption and decryption, which makes it suitable for securing stored data. On the other hand, asymmetric encryption, also known as public-key encryption, a public-key is used to encrypt data. The receiver uses a private key to decrypt the message. Public key cryptosystem (PKC) provides the most effective mechanisms for establishing security services including authentication, non-repudiation, integrity, confidentiality and digital signature. Asymmetric algorithms are much slower than symmetric ones and it is common practice to use both primitives in practical implementation. While symmetric primitives will process the heavy payloads, asymmetric primitives can be used to distribute symmetric keys securely. As Cryptographic key management is fundamental part of network security, in this paper, we are proposing Identity (ID)-based cryptography, using bilinear pairing over elliptic curve cryptography (ECC). The motivation is to provide a balance between resource consumption and security strength in hierarchical ad-hoc network.

4.3 Identity-Based Encryption (IBE)

Shamir [23] proposed the idea of the IBE scheme which uses unique ID of the device as its public key. In FANET, base stations act like a private key generator (PKG) and ID of a node can be assigned at the pre-deployment phase from base station to ensure uniqueness. The three obvious advantages [24] of IBE over conventional PKC are, firstly, IBE removes the need for certificates. Hence, we do not need certificate distribution and verification which save communication and computation overheads for the resource constrained UAVs. Secondly, IBE enables noninteractive key agreement between UAV nodes and finally, any type of string can be a public key in IBE which does not exist with conventional PKC. Bilinear pairing is the integral part of Identity based key management scheme, which allows non-interactive key distribution between a pair of cluster nodes. Bilinear pairing operations are based on elliptic curves with given parameter. Elliptic curve discrete logarithm problem is more difficult to break than the factorization and discrete logarithm problem. Hence, the security strength of ECC is much stronger and complex than other public key cryptosystems. Also, its encrypted message size is very small as well, which implies lower bandwidth, power, and computational requirements [25]. We will not provide the details implementation of IBE in this paper but mathematical background and encryption and decryption process will be used form paper [14, 24, 26].

4.4 Selective Data Encryption

The fundamental of selective encryption algorithms is to encrypt some certain portions of the messages with less overheads. It is a very useful method for the different data formats such as text, image, audio and video. It can reduce the overhead on data encryption/decryption process, and improve the efficiency of the network without negotiating the security of the system. In our framework, we are proposing using a probabilistic selective encryption approach where a sender node includes proper uncertainty in the process of message encryption, so only the delegated recipient can decrypt the ciphertext and other unauthorized nodes have no knowledge of the transmitted messages. The concept and implementation of selective encryption will be used from paper [27, 28]. Authors of both papers proposed a selective encryption algorithm which is probabilistic in nature and is faster compare to toss a coin method where each alternate word has encrypted.

4.5 Obscuring Transmitted Data and Digital Data Right Management

Data hiding is the mechanism of securely embedding information to some cover medium and in the best case nobody can see that both parties are communicating in secret. A secret message can be plaintext, an image, audio, video, ciphertext, or anything which can be represented in the form of a bit string. Different applications have their unique security requirements for example, some applications may require a larger secret message to be hidden inside data, while others require absolute invisibility of the secret message. Steganography algorithms traditionally hide secret message by using an overt communications channel to carry the secret data and digital watermarking applies data hiding for digital rights management and data authentication. A digital watermark is a digital signal or pattern inserted into a message that provide data confidentiality. For our work, we refer steganography as being the general data hiding technique and digital watermarking as a specific instance of steganography.

Encryption ensures confidentiality but it does not provide data integrity. An attacker still can record packets without knowing what is inside the packets, and replay them. If the impostor record the whole data stream and re-transmits all split packets, then the recipient would recognize a valid data stream and act accordingly. Again, even if the attacker views the cover file where the information is hidden within, there shall be no clue that there is any hidden data under the cover. In this way, the individual won't endeavor to decipher the data. In this paper, we proposed to use text based watermarking [17, 29] using Word Shift Coding Protocol (WSCP), that hides the secret data in the spaces between the words of the carrier text. Watermarking is also suitable for some tasks which encryption cannot such as copyright marking. Embedding encrypted copyright information within the contents of the file itself can prevent it being easily identified and removed.

5 Performance and Security Analysis

Proposed security framework improves the performance of resource constraint CPS in the following dimensions [30]:

- Flexibility: Addition and removal of new nodes are very flexible by allowing only new nodes, BS and CH to be involved in node addition and BS for node revocation, keeping other nodes free from overheads.
- Storage: It decreases storage requirements, saving memory to store keys and increases scalability of the system.
- Communication: Less communication in key distribution, therefore decreasing energy consumption. Because of reduced network traffic communication overhead decreases and increases systems lifespan.
- Efficiency: Uses less computing power to generate keys using fast and efficient encryption mechanism. Simple, short, and effective private key used to extract the secret message.

The framework also provides a secure communication mechanism in terms of:

Data embedding: The objective of data embedding is to safeguard the message against the adversary so the opponent cannot perceive the existence of message inside the cover object. Protection vanishes after decryption. Therefore, after encryption, watermarking technique embeds hidden copyright protection information to digital information being transmitted. These two techniques are complementary rather than overlapping.

Forward and backward secrecy: New nodes cannot detect previous messages because of periodic cluster head alteration of secret keys related to each cluster. Key reinforcement assures that keys related to the lower level nodes of hierarchy are also updated. Hence, the enemy can only get information in a certain region for a limited time span although the private keys are exposed.

Resilience against node capture attack: Compromised nodes are nodes that are manipulated by the adversary. Communication links between non-captured nodes are protected by using discrete logarithmic problem imposed by bilinear pairing. The key reinforcement mechanism assures that security limitations of key pre-distribution technique are kept limited within the cluster.

6 Conclusion

Security must be built into the applications themselves for an embedded system like UAV. In this paper, we proposed a system that ensures data security and confidentiality by tailoring traditional information security solutions. We have proposed a hierarchical structure for key distribution and information sharing to ensure confidentiality and increase the overall security of the system. The main benefit of our framework is that it provides network flexibility by allowing nodes to serve as cluster heads periodically and dynamically. Then the ordinary cluster nodes use IBE to create trust and negotiate keys with CH. Because of resource constrained nature of UAV, instead of using IBE, nodes use selective encryption techniques for message transfer. One part of the message will be send using selective encryption and other part will be sent using steganography. Future work will involve developing applicable techniques for UAV domain and conducting extensive security testing. The framework will be validated under controlled and reproducible environment. We will simulate the security framework in a testbed environment using OMNeT++, an object oriented modular discrete event-based network simulation framework mainly focused on the modeling of dynamic nature of ad-hoc communication networks.

References

1. Burmester, M., Magkos, E., Chrissikopoulos, V.: Modeling security in cyber–physical systems. Int. J. Crit. Infrastruct. Prot. **5**(3), 118–126 (2012)
2. Dini, G., Tiloca, M.: A Simulation Tool for Evaluating Attack Impact in Cyber Physical Systems. In: Hodicky, J. (ed.) MESAS. LNCS, vol. 8906, pp. 77–94. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13823-7_8
3. Austin, R.: Unmanned Aircraft Systems: UAVS Design, Development and Deployment. Wiley, Hoboken (2011)
4. Airborg™ H8 10 K with top flight hybrid-power system (2017). http://www.tflighttech.com/products/airborg-h8-10k-with-top-flight-hybrid-power-system.html. Accessed July 2017
5. Snell, B.: McAfee labs 2017 threats predictions: “Dronejacking” places threats in the sky (2016). https://www.mcafee.com/au/resources/reports/rp-threats-predictions-2017.pdf
6. Rani, C., Modares, H., Sriram, R., Mikulski, D., Lewis, F.L.: Security of unmanned aerial vehicle systems against cyber-physical attacks. J. Def. Model. Simul. Appl. Methodol. Technol. **13**(3), 331–342 (2015)
7. FAA releases 2016 to 2036 aerospace forecast (2016). https://www.faa.gov/news/updates/?newsId=85227
8. Mansfield, K., Eveleigh, T., Holzer, T.H., Sarkani, S.: Unmanned aerial vehicle smart device ground control station cyber security threat model. In: 2013 IEEE International Conference on Technologies for Homeland Security (HST), pp. 722–728. IEEE (2013)
9. Kim, A., Wampler, B., Goppert, J., Hwang, I., Aldridge, H.: Cyber attack vulnerabilities analysis for unmanned aerial vehicles. In: Infotech@ Aerospace, pp. 1–30 (2012)
10. Javaid, A.Y., Sun, W., Devabhaktuni, V.K., Alam, M.: Cyber security threat analysis and modeling of an unmanned aerial vehicle system. In: 2012 IEEE Conference on Technologies for Homeland Security (HST), pp. 585–590. IEEE (2012)
11. Gallagher, S.: Triathlete injured by “hacked” camera drone (2014). https://arstechnica.com/security/2014/04/triathlete-injured-by-hacked-camera-drone/. Accessed June 2017

12. Hartmann, K., Steup, C.: The vulnerability of UAVs to cyber attacks-an approach to the risk assessment. In: 2013 5th International Conference on Cyber Conflict (CyCon), pp. 1–23. IEEE (2013)
13. Javaid, A.Y.: Cyber security threat analysis and attack simulation for unmanned aerial vehicle network. University of Toledo (2015)
14. Hu, S.: A hierarchical key management scheme for wireless sensor networks based on identity-based encryption. In: 2015 IEEE International Conference on Computer and Communications (ICCC), pp. 384–389. IEEE (2015)
15. Yu, F.R., Tang, H., Mason, P.C., Wang, F.: A hierarchical identity based key management scheme in tactical mobile ad hoc networks. IEEE Trans. Netw. Serv. Manage. **7**(4), 258–267 (2010)
16. Sarmah, D.K., Bajpai, N.: Proposed system for data hiding using cryptography and steganography. Int. J. Comput. Appl. **8**(9), 7–10 (2010)
17. Altigani, A., Barry, B.: A hybrid approach to secure transmitted messages using advanced encryption standard (AES) and word shift coding protocol. In: 2013 International Conference on Computing, Electrical and Electronics Engineering (ICCEEE), pp. 134–139. IEEE (2013)
18. Vegh, L., Miclea, L.: A new approach towards increased security in cyber-physical systems. In: Systems, Signals and Image Processing (IWSSIP), pp. 175–178: IEEE (2014)
19. Vegh, L., Miclea, L.: Enhancing security in cyber-physical systems through cryptographic and steganographic techniques. In: 2014 IEEE International Conference on Automation, Quality and Testing, Robotics, pp. 1–6. IEEE (2014)
20. Yang, Y., Lu, J., Choo, K.-K.R., Liu, J.K.: On lightweight security enforcement in cyber-physical systems. In: Güneysu, T., Leander, G., Moradi, A. (eds.) LightSec 2015. LNCS, vol. 9542, pp. 97–112. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29078-2_6
21. Bekmezci, I., Sahingoz, O.K., Temel, Ş.: Flying ad-hoc networks (FANETs): a survey. Ad Hoc Netw. **11**(3), 1254–1270 (2013)
22. Faquih, A., Kadam, P., Saquib, Z.: Cryptographic techniques for wireless sensor networks: a survey. In: 2015 IEEE Bombay Section Symposium (IBSS), pp. 1–6. IEEE (2015)
23. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985). https://doi.org/10.1007/3-540-39568-7_5
24. Fang, Y., Zhu, X., Zhang, Y.: Securing resource-constrained wireless ad hoc networks. IEEE Wirel. Commun. **16**(2), 24–30 (2009)
25. Zhang, L., Tang, S., Luo, H.: Elliptic curve cryptography-based authentication with identity protection for smart grids. PLoS ONE **11**(3), e0151253 (2016)
26. Kodali, R.K., Chougule, S.K.: Hierarchical key agreement protocol for wireless sensor networks. Int. J. Recent Trends Eng. Technol. **9**(1), 25 (2013)
27. Oh, J.-Y., Yang, D.-I., Chon, K.-H.: A selective encryption algorithm based on AES for medical information. Healthc. Inf. Res. **16**(1), 22–29 (2010)
28. Ren, Y., Boukerche, A., Mokdad, L.: Performance analysis of a selective encryption algorithm for wireless ad hoc networks. In: 2011 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1038–1043. IEEE (2011)
29. Almuhammadi, S., Al-Shaaby, A.: A survey on recent approaches combining cryptography and steganography. Comput. Sci. Inf. Technol. **7**(3), 63–74 (2017)
30. Sahingoz, O.K.: Large scale wireless sensor networks with multi-level dynamic key management scheme. J. Syst. Architect. **59**(9), 801–807 (2013)

Securing Healthcare Data Using Biometric Authentication

Sharmin Jahan[1], Mozammel Chowdhury[2](✉), Rafiqul Islam[2],
and Junaid Chaudhry[3]

[1] Department of Biochemistry and Molecular Biology,
Jahangirnagar University, Dhaka, Bangladesh
sharmin.biochemist@yahoo.com

[2] School of Computing & Mathematics, Charles Sturt University,
Bathurst, Australia
{mochowdhury,mislam}@csu.edu.au

[3] Security Research Institute, Edith Cowan University,
Joondalup, WA 6027, Australia
j.chaudhry@ecu.edu.au

Abstract. Preservation of privacy and security of healthcare data is very important in the electronic healthcare domain. Unauthorized access or attacks by hackers can breach or damage sensitive data of patients' health records that may lead to disclosure of patient's privacy or may slowdown the system. Hence, it is very crucial to provide and enforce privacy and security of clinical data using a secure authentication system. Biometric-based access control over healthcare data can provide the necessary security and privacy. In recent years, biometric technologies have gained traction in health care applications. This paper proposes a biometric authentication scheme to preserve privacy and security in healthcare systems. In this work, we have employed biometric fingerprint as a trait for user authentication and monitoring access to the healthcare systems.

Keywords: Privacy preservation · Security · Clinical data · EHR

1 Introduction

With the advancement of sophisticated information and communication technology (ICT), both patients and healthcare professionals can access, store and share clinical information electronically in an efficient and easier manner [1]. The emergence of ICT enabled electronic healthcare systems are very compelling for the health industry due to the many advantages. Electronic healthcare systems such as eHealth or tele-health improves the quality of healthcare by making Patients Health Information (PHI) easily accessible, improving efficiency, and reducing the cost of health service delivery. Patients rarely get to spend much time with their physicians face-to-face. Recent advances in Electronic Health Record (EHR) technology have significantly increased the amount of clinical data consisting of medical documents and patient health records, that are electronically available and accessible [2].

X. Lin et al. (Eds.): SecureComm 2017 Workshops, LNICST 239, pp. 123–132, 2018.
https://doi.org/10.1007/978-3-319-78816-6_10

Despite many benefits, electronic healthcare systems still face a number of privacy and security challenges [3]. Due to the sensitive nature of medical information and healthcare records, issues of integrity, security, privacy, and confidentiality are significant concerns in this domain. Since medical information is usually associated to individuals, privacy and security must be effectively addressed and ensured to protect patient's health data. This is explicitly stated by the privacy regulations [4–6] to protect the electronic health data that the healthcare institutions maintain about their patients. HIPAA [4] standards for electronic health care information in the USA, address the physical security and confidentiality of patient identifiable electronic health care records. Security and privacy risks involved with archiving and retrieving patient records in the healthcare systems has increased the need for a reliable user authentication scheme to the protection and privacy of medical records in the healthcare domain.

Over the recent years, biometric technologies, such as fingerprint, iris recognition and hand geometry, have gained traction in electronic healthcare applications. Biometrics technology is capable to mitigate the security problems in the electronic healthcare systems by providing reliable and secure user authentication compared to the traditional approaches. Traditional authentication approaches based on user signatures, password, PINs, tokens and access cards are not appropriate in the electronic healthcare systems due to the possibility of being lost, stolen, forgotten, manipulated or misplaced. In general, traditional authentication methods are not based on inherent individual attributes [7]. On the other hand, biometrics is a security mechanism that assigns a unique identity to an individual according to some physiological (fingerprint or face) or behavioral characteristics (voice or gait) [8]. Therefore, biometrics based approaches are more reliable and capable than traditional authentication methods of distinguishing between an authorized person and an imposter. Biometric traits cannot be lost or forgotten; they are difficult to duplicate, share, or distribute. Moreover, it requires the presence of the person being authenticated; it is difficult to forge and unlikely for a user to repudiate [9]. Biometrics offers a sense of security and convenience both to patients and physicians alike. In order to stay ahead of the emerging security threats posed by electronic healthcare systems, healthcare organizations are moving from traditional approaches to the utilization of biometrics technology.

In this paper, we present a biometrics framework based on fingerprint with the potential to extend current data privacy protection and identity verification systems in the context of electronic healthcare systems. In addition, this paper highlights the applications of biometrics in addressing some of the security and privacy challenges in electronic healthcare systems. The remainder of the paper is organized as follows. Section 2 presents an overview of biometrics technology and its applications currently available for healthcare security. In Sect. 3, a description of the proposed approach is provided. We outline the main results and discussions in Sect. 4. Finally, conclusions are documented in Sect. 5.

2 Biometrics Technology in Electronic Healthcare

Biometrics technology serves to identify and authenticate individuals in many security applications based on the physiological, chemical, or behavioral attributes of the individual [10] replacing current password, PIN or token based systems. Like many other application domain, its relevance is increasing in healthcare systems. Using biometrics, the patient, and only the patient, is able to control access to their electronic medical data. With biometric-driven patient control, medical records are no longer held exclusively by providers, but instead shared with authorised providers on an as-needed basis, under the direction of the patient. The privacy and security of medical data is assured via biometric-based verification of authorised individuals, and care is improved through the real-time sharing of centralised medical data that facilitates medical decisions by doctors. Biometrics technology in the healthcare domain can be capable to:

- combat fraud and abuse in health care entitlements programmes;
- protect and help in the management of confidential medical records;
- identify patients; and
- secure medical facilities and equipment.

Biometrics technology can protect the privacy and confidentiality of medical records by means of authentication of both patients and healthcare providers. It can emulate the current well accepted system whereby a patient authenticates herself when seeking treatment or visiting a doctor's office for consultation. A typical scenario consists of a patient telling his or her name to a receptionist and then signing a release form. In order to meet the guidelines of the HIPAA regulations, both health professionals and patients must be given access to medical records. Taking into account the requirements of both patients and health professionals, biometric authentication is able to meet the privacy requirements.

Biometric-based applications are chiefly intended to solve two main categories of problems: solutions that secure against nefarious individuals via intelligence background checks or law enforcement database checks (1:N searches); and solutions that protect a transaction and its associated data by verifying the identity of the individual performing the transaction (1:1 verifications). Healthcare biometric solutions fit in the second category, a type of commercial transaction that requires a 1:1 verification. In the healthcare domain, biometric verification can be used at the following principal access control points of a patient-centric solution:

- Patient verification on login
- Patient verification upon appointment arrival
- Provider verification on login.

Several organizations have employed biometrics for securing their electronic medical records (EMRs) that use modalities such as the fingerprint and iris [11–13]. Researchers have recently studied to employ new types of biometric traits for identification such as, heart rate variability (HRV) [14], interpulse interval (IPI) [15], features of electrocardiogram (ECG) [16] and photoplethysmogram (PPG) [17]. They have proposed approaches using HRV or IPIs as biometric characteristics to generate identity for authentication and encryption [13, 14, 18, 19]. Several data encryption schemes have been proposed based on ECG [16, 20, 21], PPG [17] and multiple physiological signals [22]. Clancy et al. [23] suggested that fingerprint can be used to generate keys for cryptosystems in electronic healthcare platform.

In Europe, the use of biometrics in the health area is still scarce. Presently most applications are restricted to access control and limited to fingerprints and iris scans, but several pilot projects have been initiated to widen the scope. Danish Biometrics, for example, is developing a biometric recognition solution for secure log-on procedures for doctors and nurses at the Copenhagen Hospital. In Germany, where a nationwide eHealth infrastructure is being introduced, doctors will be able to digitally sign prescriptions using fingerprints. In an Italian health care location, a biometric system controls the access to the surgical rooms [24]. In Texas, USA, a biometric and smart card-based program to address recipient and provider fraud in the Medicaid system has been in operation since 2004. The Medicaid Integrity Pilot, or MIP, was initially designed to evaluate the performance and acceptance of fingerprint and smart card technologies for recipient authentication at the point of service [25].

The Australian Methadone program uses iris recognition technology in support of the treatment of citizens addicted to heroin. The Methadone program registers patients in an iris recognition system to detect duplicate enrolees and to enable authentication for clients unable coherently or consistently to claim an identity. Personal information including biometric data, name, permitted dosage, last dosage, and next scheduled dosage are included in the database. A similar pilot program for the controlled distribution of methadone has been deployed in the Netherlands using fingerprint technology and smart cards. One can envision similar uses of biometrics to automate and control distribution of vaccines during epidemics [26].

South African government launched a pilot fingerprint identification program for government employees in response to growing health care concerns due mainly to the HIV/AIDS epidemic. In the past, individuals tended to steal identification cards to receive health care benefits, and many enrolled under multiple identities to receive additional, or replicate, services. The system was designed to detect multiple users at enrolment and to verify a user's identity when they seek services. The system allows patients and medical providers to interact with various settings of care, including hospitals and pharmacies, to prevent fraud and to manage the administration of health care benefits. This system can also track health histories to monitor the overall costs associated with disease treatment [25].

3 Proposed Biometric Authentication System

This section demonstrates the architecture of the proposed biometric authentication system to control access to an electronic healthcare system. A fingerprint biometric scheme can either verify or identify of an individual based on his or her fingerprint as trait. In verification approach, it verifies the authenticity of one person by his fingerprint. In identification approach, it establishes the person's identity among those enrolled in a database. Without knowledge of the person's identity, the fingerprint identification system tries to match his fingerprint(s) with those in the whole fingerprint database. Among all other biometric traits, fingerprint biometrics occupies an important and a very special place in the field of health security due to its uniqueness and availability. The diagram of a proposed biometric system is shown in Fig. 1.

3.1 Fingerprint Acquisition

The first stage of the fingerprint authentication process is to capture a digital image of the fingerprint pattern using a sensor. The captured image is called a live scan. This live scan is digitally processed to create a biometric template (a collection of extracted features) which is stored and used for matching. Many technologies have been used including optical, capacitive, RF, thermal, piezo resistive, ultrasonic, piezoelectric [27].

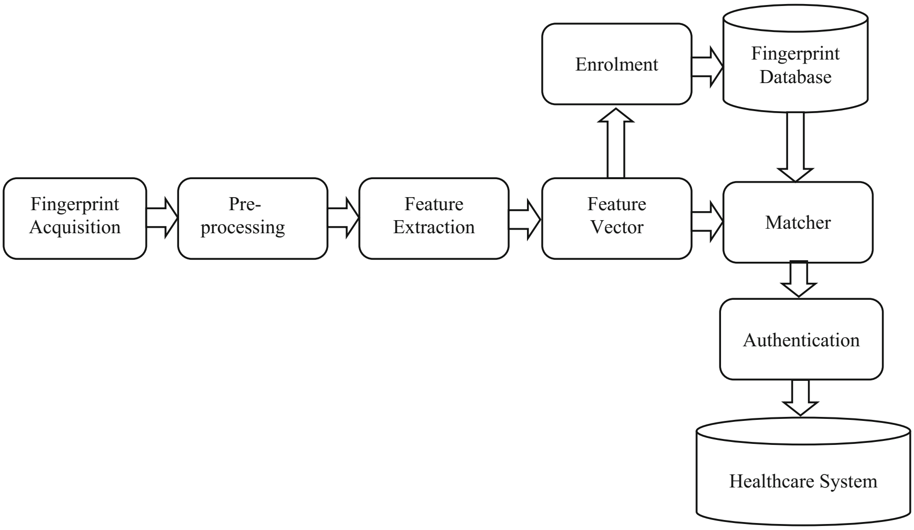


Fig. 1. Architecture of the proposed biometric authentication scheme for electronic healthcare.

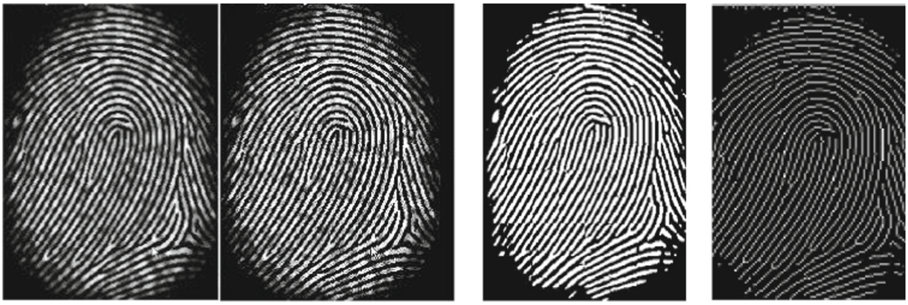
3.2 Pre-processing

Pre-processing is required to enhance the quality of an image by filtering and removing unnecessary noises because the captured images may be of poor quality. This process removes the noises in the images and enhance them for better features extraction. For image filtering we employ a fuzzy filtering technique [28] (Fig. 2).

3.3 Features Extraction

Based on the features used, fingerprint verification methods can be classified into two categories: minutiae based or texture based. The minutiae based fingerprint verification systems have shown high accuracy [29]. The texture based methods use the entire fingerprint image or local texture around minutiae points [30]. In this paper, we employ the minutie features for fingerprint identification.

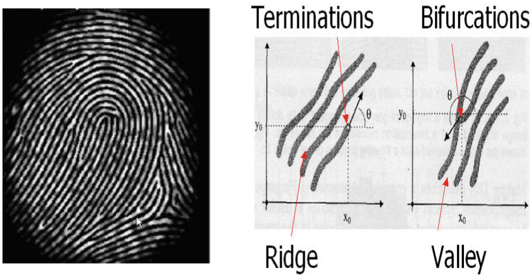
Minutiae are some specific points in a fingerprint, these are the small details in a fingerprint that are most important for fingerprint recognition. There are three major types of minutiae features: the ridge ending, the bifurcation, and the dot (also called short ridge) (Fig. 3). The ridge ending is the spot where a ridge ends. A bifurcation is



(a) Original image (b) Filtered image (c) Binarized image (d) Thinned image

Fig. 2. Pre-processing steps in fingerprint identification.

the spot where a ridge splits into two ridges. Spots are those fingerprint ridges that are significantly shorter than other ridges [31]. Minutia keypoints are searched over a enhanced, binarized and thinned version of the input fingerprint image. The local orientation for each minutia keypoint is obtained from the estimated orientation field. To extract the minutiae set, the open FVS library is used [32].



(a) A fingerprint (b) Minutia

Fig. 3. A fingerprint image and its minutia.

3.4 Matching

A fingerprint matching module computes a match score between two fingerprints, which should be high for fingerprints from the same finger and low for those from different fingers. Most fingerprint-matching algorithms adopt one of four approaches: image correlation, phase matching, skeleton matching, and minutiae matching. Minutiae-based representation is commonly used, primarily because minutiae-based fingerprint matching is more reliable and acceptable by the forensic experts and other security professional and its representation is storage efficient.

In this paper, we employ the Matching Score Matrix (MSM) algorithm [33] to compare the minutiae features extracted from the test fingerprints with features of database fingerprints to verify a person. The Matching Score Matrix algorithm is able to reduce the number of matching comparisons in linear search. The main idea of the algorithm is that the similarity (called the matching score) between any pair of the finger templates is calculated in advance, and then the order of the comparison with the input image is decided according to the matching scores.

4 Experimental Evaluation

In this section, we evaluate the performance of our proposed approach and compare with other similar techniques reported in this work. To demonstrate the effectiveness of our algorithm, we perform experiment using several standard fingerprint datasets. Experiments are carried out on a computer with 2.8 GHz Intel Core i7 processor. The algorithm has been implemented using Visual C++.

4.1 Datasets

The performance of the proposed fingerprint biometric verification scheme has been evaluated on FVC2002 fingerprint databases [33]. FVC2002 project has four different databases: DB1, DB2, DB3 and DB4. Each database has 110 fingers with 8 impressions per finger (110 × 8 = 880 fingerprints in all); The fingers are split into set A (100 fingers − evaluation set) and set B (10 fingers − training set). During a session, fingers were alternatively dried and moistened. Some characteristics of these two databases are summarized in Table 1.

Table 1. Description of FVC 2002 databases.

Database	Sensor type	Image size	Set A (Testing) (fingers × images)	Set B (fingers × images)	Resolution
DB1	Optical sensor	388 × 374	100 × 8	10 × 8	500 dpi
DB2	Optical sensor	296 × 560	100 × 8	10 × 8	569 dpi
DB3	Capacitive sensor	300 × 300	100 × 8	10 × 8	500 dpi
DB4	Synthetic	288 × 384	100 × 8	10 × 8	About 500 dpi

4.2 Results

To justify the performance of the proposed scheme, we evaluate three statistical measures: False Match Rate (FMR), False Non-Match Rate (FNMR) and Equal Error Rate (EER). The FMR is the rate at which the system incorrectly matches or accepts imposter fingerprint inputs (also known as False Acceptance Rate or FAR). FNMR is the rate at which inputs of genuine fingerprint are incorrectly rejected by the system (also referred as False Rejection Rate or FRR). EER is the rate at which both FMR and FNMR are equal. EER determines the threshold values for FMR and FNMR of the biometric system. When the rates are equal (FMR = FNMR), the common value is referred to as the equal error rate. The value indicates that the proportion of false acceptances is equal to the proportion of false rejections. The lower the EER value, the higher the accuracy of the biometric system. The results for EER using different fingerprint databases are reported in Table 2. We compare the accuracy of our method with another standard method [35]. Our evaluation results confirm that the number of matching processes is reduced and the error rate for identification is reduced by the proposed method. From the experimental results, we can see that the proposed method is superior to the conventional minutiae-based one for all the databases. Even though the performances for FVC 2002 DB3 and DB4 are lower than those for FVC 2002 DB1 and DB2.

Table 2. EER comparisons of two matching methods on FVC databases.

Database	FISiA [35]	Proposed method
DB1	1.0%	0.26
DB2	0.89%	0.19
DB3	1.7%	0.87
DB4	2.3%	1.2

5 Conclusion

Biometrics can be incorporated in a wide-range of health care applications. Driven by the desires of healthcare authorities to offer better healthcare services at a low cost, electronic healthcare has revolutionized the healthcare industry. However, while electronic healthcare system comes with numerous advantages that improve health services, it still suffers from security and privacy issues in handling health information. eHealth security issues are mainly centered around user authentication, data integrity, data confidentiality, and patient privacy protection. Biometrics technology addresses the above security problems by providing reliable and secure user authentication compared to the traditional approaches. This research offers a comprehensive biometrics authentication scheme in order to protect unauthorised access to the healthcare system and preserve its privacy and security.

References

1. Jahan, S., Chowdhury, M.M.H.: Assessment of present health status in Bangladesh and the applicability of e-health in healthcare services: a survey of patients' expectation toward e-health. World J. Comput. Appl. Technol. **2**(6), 121–124 (2014)
2. Martínez, S., Sánchez, D., Valls, A.: A semantic framework to protect the privacy of electronic health records with non-numerical attributes. J. Biomed. Inform. **46**, 294–303 (2013)
3. Fernández-Alemán, J.L., et al.: Security and privacy in electronic health records: a systematic literature review. J. Biomed. Inform. **46**, 541–562 (2013)

4. HIPAA 1996: US Department of Health & Human Services. https://www.hhs.gov/sites/default/files/privacysummary.pdf
5. EU Directive 95. http://www.dataprotection.ie/docs/EU_Directive_95/46/EC_Chapter1/92.htm
6. The Department of Health, Australian Government. PCEHR: Personally Controlled Electronic Health Record System Operator: Annual Report 2012–2013
7. Chandra, A., Durand, R., Weaver, S.: The uses and potential of biometrics in health care: are consumers and providers ready for it? Int. J. Pharm. Healthc. Mark. **2**(1), 22–34 (2008)
8. Chowdhury, M., Islam, R., Gao, J.: Robust ear biometric recognition using neural network. In: IEEE Conference on Industrial Electronics & Applications, ICIEA 2017, Siem Reap, Cambodia (2017)
9. Zhang, D., Campbell, J.P., Maltoni, D., Bolle, R.M.: Special issue on biometric systems. IEEE Trans. Syst. Man Cybern. Part C **35**(3), 273–275 (2005)
10. Wayman, J.L., Jain, A.K., Maltoni, D., Maio, D.: Biometric Systems: Technology. Design and Performance Evaluation. Springer, London (2005). https://doi.org/10.1007/b138151
11. A4 Health Systems: A4 Health Systems Electronic Medical Record Solutions. http://www.a4healthsystems.com/
12. BCBSRI: Blue Cross Blue Shield of Rhode Island. https://www.bcbsri.com
13. University of South Alabama Health System. http://www.southalabama.edu/usahealthsystem/
14. Bao, S.D., Zhang, Y.T., Shen, L.F.: Physiological signal based entity authentication for body area sensor networks and mobile healthcare systems. In: 27th Annual International Conference of the Engineering in Medicine and Biology Society, IEEEEMBS 2005, pp. 2455–2458 (2005)
15. Poon, C.C.Y., Zhang, Y.-T., Bao, S.-D.: A novel biometrics method to secure wireless body area sensor networks for telemedicine and mhealth. IEEE Commun. Mag. **44**(4), 73–81 (2006)
16. Venkatasubramanian, K.K., Banerjee, A., Gupta, S.K.S.: ECG-based key agreement in body sensor networks. In: INFOCOM Workshops 2008, pp. 1–6. IEEE (2008)
17. Venkatasubramanian, K.K., Banerjee, A., Gupta, S.: Plethysmogram-based secure inter-sensor communication in body area networks. In Military Communications Conference, MILCOM 2008, pp. 1–7. IEEE (2008)
18. Bao, S.D., Poon, C.C.Y., Shen, L.F., Zhang, Y.T.: Using the timing information of heartbeats as an entity identifier to secure body sensor network. IEEE Trans. Inf. Technol. Biomed. **12**(6), 772–779 (2008)
19. Bao, S.D., Shen, L.F., Zhang, Y.T.: A novel key distribution of body area networks for telemedicine. In: 2004 IEEE International Workshop on Biomedical Circuits and Systems, pp. 1–17–20a (2004)
20. Bui, F.M., Hatzinakos, D.: Biometric methods for secure communications in body sensor networks: resource-efficient key management and signal-level data scrambling. EURASIP J. Adv. Signal Process. **13**, 3142–3156 (2008)
21. Challa, N., Cam, H., Sikri, M.: Secure and efficient data transmission over body sensor and wireless networks. EURASIP J. Wirel. Commun. Netw. **3**, 707–710 (2008)
22. Cherukuri, S., Venkatasubramanian, K.K., Gupta, S.K.S.: BioSec: a biometric based approach for securing communication in wireless networks of biosensors implanted in the human body. In: Proceedings of the 2003 International Conference on Parallel Processing Workshops, pp. 432–439 (2003)
23. Clancy, T.C., Kiyavash, N., Lin, D.J.: Secure smartcard-based fingerprint authentication. In: Proceedings of the 2003 ACM SIGMM Workshop on Biometrics Methods and Applications. ACM, Berkley (2003)

24. Biohealth Newsletter, vol. 5, December 2007. http://biohealth.gsf.de
25. Marohn, D.: Biometrics in healthcare. Biometr. Technol. Today **14**, 9–11 (2006)
26. DOH: Review of Methadone treatment in Australia. http://www.health.gov.au/internet/main/publishing.nsf/content/phd-illicit-review-of-methadone-treatment
27. Jain, A.K., Feng, J., Nandakum, K.: Fingerprint matching. IEEE Comput. Mag. **43**, 36–44 (2010)
28. Chowdhury, M., Gao, J., Islam, R.: Fuzzy logic based filtering for image de-noising. In: IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2016, Vancouver, Canada, pp. 2372–2376
29. Jain, A.K., Hong, L., Bolle, R.: On-line fingerprint verification. IEEE Trans. Pattern Anal. Mach. Intell. **19**, 302–314 (1997). ISSN 0162-8828
30. Chikkerur, S., Pankanti, S., Jea, A., Ratha, N., Bolle, R.: Fingerprint representation using localized texture features. In: Proceedings of ICPR 2006, August 2006, pp. 521–524. IEEE Computer Society, Hong Kong (2006). ISSN 1051-4651
31. Zhao, F., Tang, X.: Preprocessing and postprocessing for skeleton-based fingerprint minutiae extraction. Pattern Recognit. **40**(4), 1270–1281 (2007)
32. FVS 2003: Fingerprint Verification System. http://fvs.sourceforge.net
33. Maeda, T., Matsushita, M., Sasakawa, K.: Identification algorithm using a matching score matrix. IEICE Trans. Inf. Syst. **1**(7), 819–824 (2001)
34. FVC 2002 Fingerprint Database. http://bias.csr.unibo.it/fvc2002/
35. Zhou, R., Zhong, D., Han, J.: Fingerprint Identification Using SIFT-Based Minutia Descriptors and Improved All Descriptor-Pair Matching. Sensors **13**, 3142–3156 (2013). https://doi.org/10.3390/s130303142

Fast and Robust Biometric Authentication Scheme Using Human Ear

Mozammel Chowdhury[1], Rafiqul Islam[1](✉), and Junbin Gao[2]

[1] School of Computing and Mathematics, Charles Sturt University,
Sydney, Australia
{mochowdhury, mislam}@csu.edu.au

[2] Discipline of Business Analytics, The University of Sydney Business School,
Sydney, Australia
junbin.gao@sydney.edu.au

Abstract. Biometric authentication using human ear is a recent trend in security applications including access control, user recognition, surveillance, forensic, and border security systems. This paper aims to propose a fast and robust authentication scheme using ear biometric. In this work, a fast technique based on the AdaBoost algorithm is used to detect the ear of the user from profile images. An efficient stereo matching algorithm is used to match the user's ear data (probe) to the previously enrolled (stored) ear data in a gallery database for verification and recognition. Correspondences are established between extracted features of the probe and gallery image sequences. The performance of the recognition approach is evaluated on different standard ear datasets and compared with other techniques. Experimental results suggest the superiority of the proposed approach over other popular techniques reported in this work.

Keywords: Biometric authentication · Access control · Ear recognition

1 Introduction

Biometric identification and authentication has been gaining popularity for providing safety and security in many applications such as, access control, surveillance system, visa processing, national IDs, border checking, law enforcement applications and so on. Biometric system is a technique that relies on the unique biometric characteristics of individuals to verify or recognize the user for secure access to a system [1]. A biometric system may operate in one or both two modes: authentication and identification. In authentication mode, one-to-one matching is performed to compare a user's biometric data to a specific pattern of the claimed identity enrolled in the system earlier. In identification or recognition process, one-to-many matching is done to identify a user's biometric by comparing it against every identity patterns stored in a large database. The traditional methods for user authentication or identification have deficiencies that restrict their applicability in security systems. The properties used in the traditional authentication methods can be forgotten, disclosed, lost or stolen. Biometric characteristics on the other hand, are unique and not duplicable or transferable. Therefore, biometric trait based security systems have been proven superior to traditional ID based systems [2].

X. Lin et al. (Eds.): SecureComm 2017 Workshops, LNICST 239, pp. 133–144, 2018.
https://doi.org/10.1007/978-3-319-78816-6_11

Most of the biometric systems use traits such as, fingerprint, face, facial components, palm print, hand geometry, iris, retina, gait and voice [3]. In recent years, the use of human ear as a biometric trait is a promising trend in the research community. The ear is quite attractive biometric candidate because, the shape of the ear is unique to individuals and generally unaffected by changing facial expressions, anxiety, use of cosmetics or eye glasses and aging [4]. Moreover, several ear features such as smaller in size, co-location with face, and relatively less change in shape due to aging has made it very popular among biometric communities. An ear also has reduced spatial resolution and uniform distribution of colour. However, due to its complex geometrical shape and often being obscured by hair, ear-ring, head-cover and the similar, developing fast, accurate and robust ear based biometric systems is still very challenging [5, 6].

A typical automatic ear-based biometric system consists of the following steps: detection (or segmentation) of the ear, normalization and enhancement, feature extraction and matching (recognition or verification). Ear detection refers to the localization of the ear shape in a facial profile image. After detection or segmentation, the ear region can be normalized (in orientation or in size) and enhanced to make it simple for further operations such as feature extraction and matching processes. Since the other processing steps like feature extraction, recognition or verification depend on accurate detection of the ear, this stage is crucial in biometric system.

This paper proposes a robust and efficient ear based biometric system using AdaBoost based ear detection, local features extraction and stereo matching based recognition algorithms. Correspondence matching is crucial for meaningful comparisons of two images. The importance of good correspondences is even greater in the case of ear recognition. Standard systems often align the ears or a few other features, using translation, or similarity transformations. However, these can still result in significant misalignments in the ear region. To handle this situation, we use stereo matching. This allows for arbitrary, one-to-one continuous transformations between images, along with possible occlusions, while maintaining an epipolar constraint. In matching correspondences between scan lines in two images, a stereo matching cost is optimized, which reflects how well the two images match. Consequently, we can use the stereo matching cost as a measure of similarity between two ear images (probe and gallery image). Although, stereo matching algorithms have been used in face recognition earlier [20, 21], we are the first to use stereo matching approach for ear recognition. The proposed system does not require training or extraction of the ear contour and hence reduces the computational cost compared to other existing methods. Hence, the low computation time renders its suitability to employ it in real time applications. The obtained ear recognition results can be combined with other biometric modalities such as facial features to develop a more robust and accurate recognition system.

The rest of the paper is organized as follows. In Sect. 2, we have discussed the related works on ear based biometric recognition. The proposed scheme is presented in Sect. 3. Experimental results are reported and discussed in Sect. 4. Finally, Sect. 5 concludes the paper.

2 Related Works

Ear based biometric authentication system is considered as one of the most promising solutions for secure systems. Due to many practical applications, there is currently an increasing demand of biometric technology in the industry. According to the surveys [2–4], most of the proposed ear based recognition approaches use either PCA (Principal Component Analysis) or the ICP algorithm for matching [6–9].

Yaqubi et al. propose a system employing edge features taken over multiple positions and orientations [10]. The extracted features are classified using an SVN and a kNN with recognition accuracy of 96.5%.

Islam et al. [11] find local surface patches (LSP) to select features for their system. PCA is then used to find the most descriptive features in the LSP. The feature extractor repeats selecting LSP until the desired number of features is found. The algorithm is evaluated using UND ear database. They obtain a recognition rate of 93.5%. However, the approach has not been tested with pose variation and different scaling.

Wang et al. [12] employ different feature vectors in their method using seven moment invariants. The feature vectors are used as the input for a back propagation neural network which is trained to classify the moment invariant feature sets.

Gutierrez et al. [13] divide the detected ear regions into three equally sized segments. The upper segment shows the helix, the middle one shows the concha and the lower part shows the lobule. Each of these sub images is decomposed by wavelet transform and then fed into a modular neural network (MNN).

Alaraj et al. [14] use PCA in their work for feature representation. The approach use a multilayer feed forward neural network for classification of the PCA based feature components. They have reported a rank-1 performance of 96%.

3 Proposed Approach

The proposed biometric authentication scheme based on human ear is consisted of the following stages: (i) Acquisition of profile face images, (ii) Ear data extraction and normalization, (iii) Refinement, (iv) Features extraction, (v) Feature matching, and (vi) Recognition/Authentication. The architecture of the proposed ear biometric system is depicted in Fig. 1.

3.1 Ear Detection and Normalization

Ear detection consists of extracting the position of the ear in a facial profile image. Different automatic ear detection methods have been published in recent years. In this work, the ear region is detected on profile face images using the AdaBoost based detector [18]. The motivation behind the selection of this detector is that it possesses high accuracy and speed. After detecting the ear region, the corresponding ear data is then extracted. To ensure the whole ear is extracted, we expand the detected ear regions by an additional 20 pixels around each direction. The extracted ear data varies in dimensions. Therefore, we normalize the extracted ear shape with uniform dimension of 160 by 140.

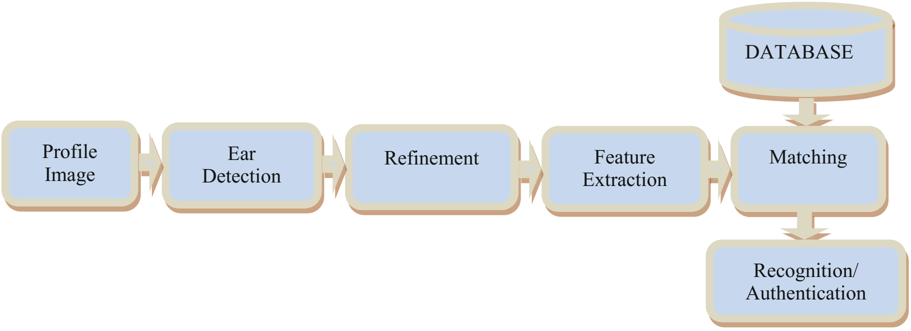


Fig. 1. Architecture of the proposed authentication system using ear biometric.

3.2 Pre-processing of Ear Data

Once the human ear is detected, we employ a fuzzy filter [19] to remove all the spikes and holes from the extracted ear region. We choose this filter because it has the advantage of both median and average filtering and possesses high accuracy and speed. This filter employs fuzzy rules for deciding the gray level of the pixels within a window in the image.

3.3 Features Extraction

One of the crucial tasks in biometric ear recognition is the features extraction. Different types of features commonly used in ear recognition include: intensity and shape features, Fourier descriptors, wavelet-based (i.e. Gabor) features or SIFT points [25]. The extracted ear features are used for matching with the one stored in the gallery database. In this work, we use local edge features since they are invariant to pose variation, occlusion and illumination changes.

Edge or gradient histogram corresponds to the spatial distribution of the edge features in the image. The gradient of an image $f(x, y)$ can be expressed by,

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} \tag{1}$$

where, $G_x = \frac{\partial f}{\partial x}$ is the gradient in x direction, and

$G_y = \frac{\partial f}{\partial y}$ is the gradient in y direction.

The gradient direction can be calculated by the formula:

$$\theta = \tan^{-1}\left[\frac{G_y}{G_x}\right] \tag{2}$$

We use Canny edge detector to extract the edge features from the ear data. The gradient values are normalized to [0, 255].

3.4 Feature Matching and Recognition

The extracted ear features of a user (probe data) are matched with the specific ear data (gallery data) stored in the gallery database built off-line. Matching can be performed based on the error of registering between the two data sets, more specifically, two clouds of points. In this work, we use a stereo algorithm [22] to compute the degree of similarity, which is quite fast and efficient. The stereo algorithm compares two ear data (probe and gallery) and computes the degree of similarity between the probe image and the gallery image and identifies the user's ear that produces the best matching score. Prior to stereo matching, we need to estimate the epipolar geometry. The probe and gallery images are rectified and the similarity score is computed by computing the stereo matching cost of every row of the rectified images.

3.5 Epipolar Geometry and Rectification

The rectification allows the use of epipolar geometry environment where the epipolar lines are horizontal i.e., parallel to the lines of the image sequences [23]. In epipolar geometry, any point lying on an epipolar line in the reference image (i.e., probe image) corresponds to a point lying on the same epipolar line in the target image (i.e., gallery image). After rectification of the two ear images, the matched points have necessarily the same coordinate in the both images. Therefore, in case of searching for corresponding points in two ear images, it is only necessary to search in the same epipolar line, reducing a 2D search space to 1D. In order to achieve rectification, we adopt the algorithm proposed by Fusiello et al. [24].

3.6 Stereo Algorithm and Matching Costs

We employ a robust and fast stereo algorithm for matching correspondences between the probe and gallery images based on fuzzy correlation measure. The aim of matching correspondences is to compute the measure of similarity or matching cost for identification of the user's ear. To determine the correspondences between two images, we match the windows of pixels on the same epipolar lines in the reference (probe) and target (gallery) image. In our method, we assume that the pixels surrounded by a window possess approximately equal disparity. Thus, the matching cost C for a pixel (x, y) in the probe image is estimated by taking a window of pixels centered at (x, y) in the probe image, and placing a similar window of pixels centered at (x + d, y) in the gallery image and computing the difference between these two windows using a fuzzy correlation measure given by the following Eq. (3). Here, d is a searching range over the same epipolar line in the gallery image.

$$C(x,y,d) = \frac{\sum_{x,y \in W} F(x,y)|I_P(x,y) \times I_G(x+d,y)|}{\sqrt{\sum_{x,y \in W} F(x,y) I_P^2(x,y) \times \sum_{x,y \in W} F(x,y) I_G^2(x+d,y)}} \quad (3)$$

where $I_P(x, y)$ and $I_G(x, y)$ are the intensities of the pixels at position (x, y) in the probe and gallery images, respectively; and W is a square window. F(x, y) is the fuzzy measure corresponding to the pixel at position (x, y), has Gaussian distribution which is proportional to fuzzy membership function:

$$F(x, y) = \exp(-\frac{\left|I_P(x, y) - I_G(x + d, y)\right|^2}{2\sigma^2}) \tag{4}$$

where, σ is the standard deviation of all pixels within the window.

The matching cost C(x, y) for every pixel (x, y) can be computed by the winner-take all strategy such that,

$$C(x,\ y)\ =\ \arg\max C(x,\ y,\ d) \tag{5}$$

3.7 Final Matching Cost and Recognition

In order to authenticate a user, the matching is performed between the probe ear image and the enrolled gallery pattern. For recognition process, a number of iteration is accomplished for matching the probe image with the stored gallery images. When we match a probe image to a gallery image using our proposed stereo algorithm, we obtain different window costs. We pick the best matching scores and estimate a normalized (average) matching cost for every pair of the probe and the gallery images, by using the following equation:

$$C(I_P, I_G) = \frac{\sum_{i=1}^{n} C(I_{P,i}, I_{G,i})}{\sum_{i=1}^{n} \left|I_{P,i}\right| + \left|I_{G,i}\right|} \tag{6}$$

where C is the normalized matching cost for the image pair: the probe and a gallery ear image. Thus, we compute normalized costs for all pair of images by comparing the probe with all gallery images. We then identify the gallery ear image that provides best similarity measure given by,

$$\text{Similarity, } S\ =\ \max\left\{C_n(I_{\mathrm{P}},\ I_{\mathrm{G}})\right\} \tag{7}$$

where, C_n refers to the normalized cost of n^{th} image pair (the probe and the gallery image), n = 1 … N; and N denotes the total number of images considered in the gallery. The best match is considered for identification when, S > T. Here, T is a predetermined threshold and is set to 0.65 by empirical evaluation.

4 Experimental Evaluation

In this section, we evaluate the performance of our proposed algorithm and compare with other similar techniques reported in this work. To demonstrate the effectiveness of our algorithm, we perform experiment using several real images and standard ear datasets as well. Experiments are carried out on a computer with 2.8 GHz Intel Core i7 processor. The algorithm has been implemented using Visual C++.

4.1 Datasets

In this experiment, we use three different standard datasets of ear images, prepared by the University of Notre Dame (UND) [15], the University of Science and technology in Beijing (USTB) [16], and the Indian Institute of Technology, Delhi (IITD) [17]. The UND dataset includes 942 images of 302 human subjects, the USTB database contains 308 images of 77 subjects, and the IIITD database includes 421 images of 121 subjects. The detailed features of the ear databases are summarized in Table 1. Figure 2 shows some sample images of these ear databases.

Table 1. Features of the ear datasets.

Dataset	Total images	Individuals	Additional features
UND-F	942	302	3D and corresponding 2D profile images from 302 human subjects including some partially occluded images, captured in 2003 and 2004
USTB	308	77	Images were captured from 77 human subjects in 4 different sessions between November 2003 and January 2004
IITD	421	121	3 images were taken per subject in an indoor environment, collected between October 2006 and June 2007

4.2 Results

The authentication process has been successfully evaluated with 100% accuracy using the real image sequences. To evaluate the recognition performance, the algorithm is tested with three standard ear datasets: UND, USTB and IITD. In this work, we use one image for every subject from each dataset as a probe image while the remaining one image is used as the gallery image. Figures 3 and 4 demonstrate ear detection and local feature extraction process, respectively. The performance of our proposed recognition scheme is evaluated using the stereo matching algorithm which is fast and efficient for user identification. The recognition performance of our proposed method is compared with other existing similar methods such as, support vector machine (SVM) [10], AdaBoost [11], neural network (NN) [12], modular neural network (MNN) [13], and NN with principal component analysis (NN + PCA) [14]. The comparisons for different methods are reported in the Fig. 5, which clearly indicates the superiority of our

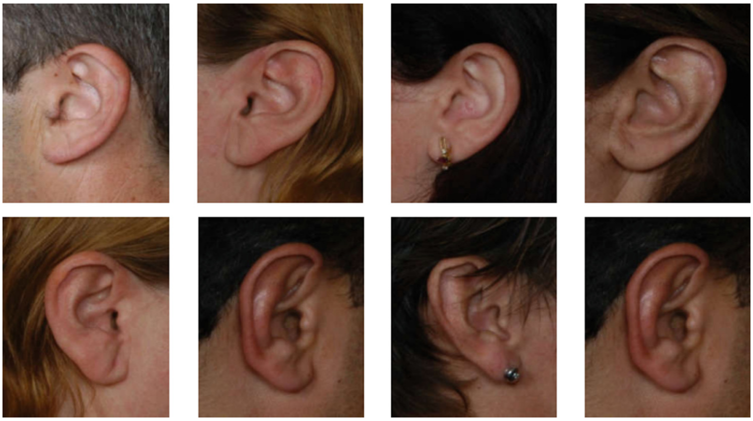


Fig. 2. Example of profile images with ear of different shapes: left ears (top), and right ears (bottom).

proposed method. We also test our algorithm using different datasets and experimental evaluation indicates that our approach provides better performance for the UND database with a recognition rate of 98.96%, as shown in Fig. 6. The proposed algorithm achieves a very low false positive rate (FPR) which is 0.25%. Figure 7 presents a comparison of FPRs for different methods.

We compare the computation time of our recognition algorithm with other methods. A Visual C++ implementation of our algorithm requires around 0.39 s to extract the local edge features from a probe ear image, and the average time to match a probe-gallery pair in the recognition process is 0.18 s on the UND dataset. Table 2 summarizes the comparison for matching time of the recognition approach of this paper with others on the UND database. Matching times are computed on different machines

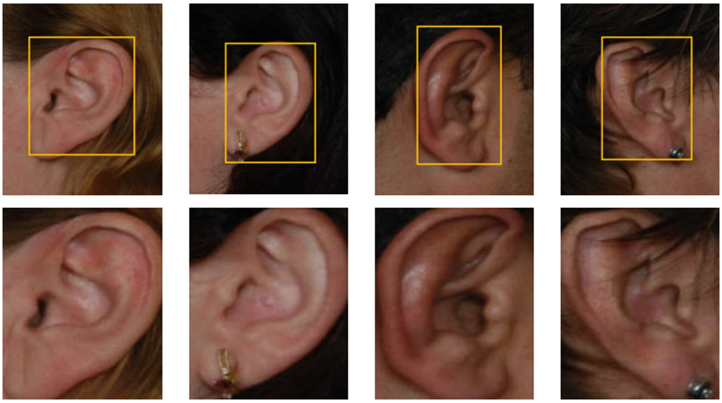


Fig. 3. Detection process: detected ear shape (top) and extracted ear (bottom).

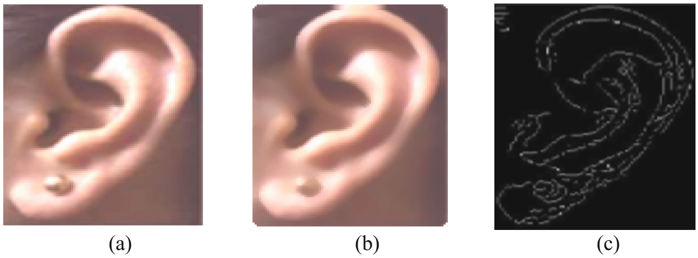


Fig. 4. Features extraction: (a) initial ear shape, (b) refined ear data, and (c) extracted edge features.

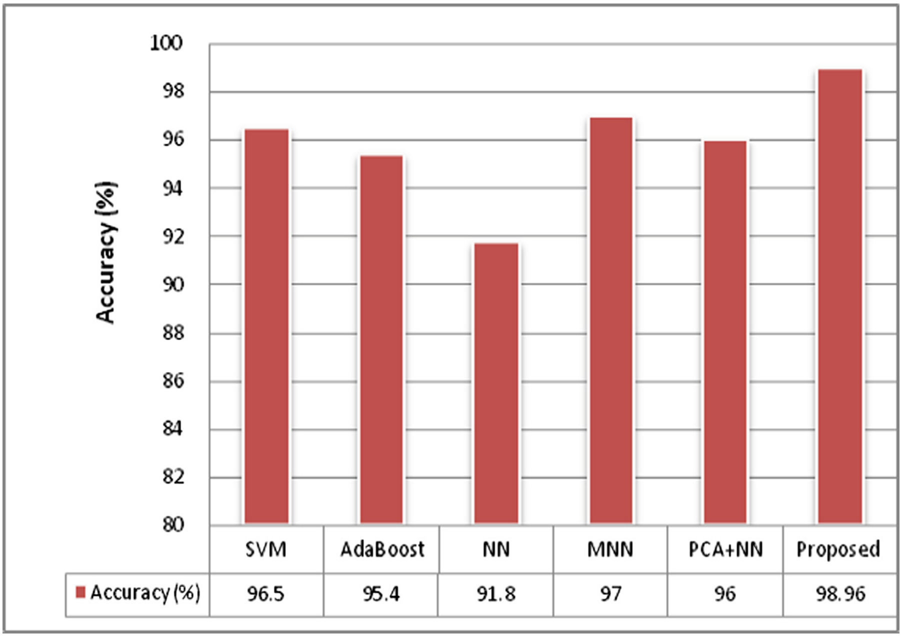


Fig. 5. Comparison for recognition accuracy with different methods.

in different approaches. Results show that our proposed matching algorithm can achieve superior performance with significant reduction of computation time compared to other methods. Empirically we find that a window of size 3 × 3 pixels and a range value of +5 (d) for searching correspondence pixels are good choices for better results.

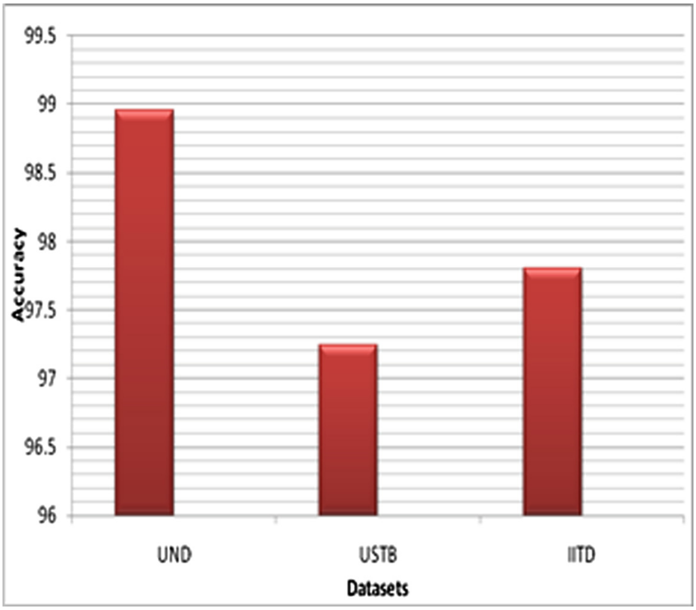


Fig. 6. Recognition accuracy on different datasets.

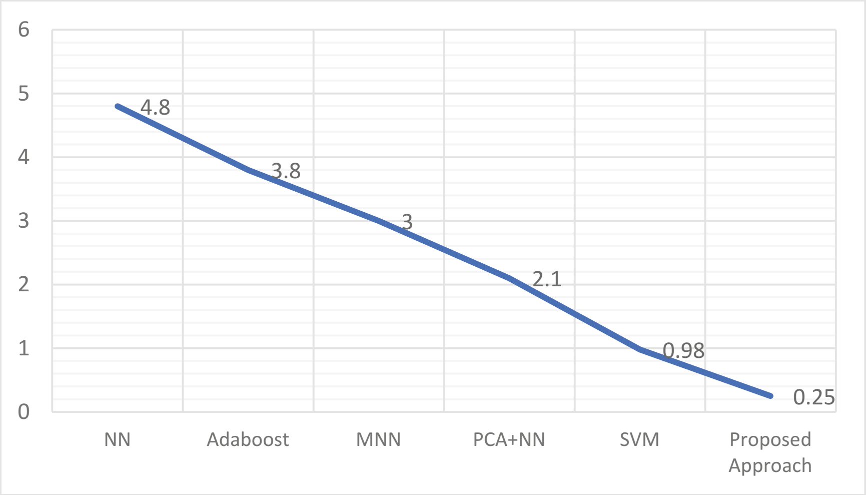


Fig. 7. Comparison of false positive rates for different methods.

Table 2. Computational time for matching a pair of probe-gallery ear image with different methods on UND dataset.

Method	Feature extraction time	Average matching time
Our method (Visual C++)	0.39 s	0.18 s
Islam et al. [11] (MATLAB)	22.2 s	2.28 s
Chen and Bhanu [26] (C++)	N/A	1.1 s

5 Conclusion

This paper proposes a robust and efficient human identification based on ear biometric trait employing a hybrid neural network. Our system could be capable to cope with pose variations, occlusion and illumination changes. The effectiveness of the proposed algorithms has been tested with different standard datasets. Experimental evaluation confirms that our proposed method achieves superior performance comparable to other existing similar methods. Our next target is to extend the algorithm by combining it with other biometric modalities such as facial features to develop a more robust, secure and accurate human recognition system. We believe that this proposed method will be useful for many real-time applications where very fast processing time is important.

References

1. Chowdhury, M., Gao, J., Islam, R.: Biometric authentication using facial recognition. In: Deng, R., Weng, J., Ren, K., Yegneswaran, V. (eds.) SecureComm 2016. LNICST, vol. 198, pp. 287–295. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59608-2_16
2. Marqués, I., Graña, M.: Image security and biometrics: a review. In: Corchado, E., Snášel, V., Abraham, A., Woźniak, M., Graña, M., Cho, S.-B. (eds.) HAIS 2012. LNCS (LNAI), vol. 7209, pp. 436–447. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28931-6_42
3. Jain, A., Kumar, A.: Biometric recognition: an overview. In: Mordini, E., Tzovaras, D. (eds.) The International Library of Ethics, Law and Technology, vol. 11, pp. 49–79. Springer, Heidelberg (2012). https://doi.org/10.1007/978-94-007-3892-8_3
4. Islam, S.M.S., Bennamoun, M., Owens, R., Davies, R.: A review of recent advances in 3D ear and expression invariant face biometrics. ACM Comput. Surv. **44**(3), 14:1–14:34 (2012)
5. Islam, S.M.S., Davies, R., Bennamoun, M., Owens, R.A., Mian, A.S.: Multibiometric human recognition using 3D ear and face features. Pattern Recogn. **46**(3), 613–627 (2013)
6. Choras, M.: Ear biometrics based on geometrical feature extraction. Electron. Lett. Comput. Vis. Image Anal. **5**, 84–95 (2005)
7. Yuizono, T., Wang, Y., Satoh, K., Nakayama, S.: Study on individual recognition for ear images by using genetic local search. In: Proceedings of Congress on Evolutionary Computation, pp. 237–242 (2002)
8. Hurley, D.J., Nixon, M.S., Carter, J.N.: Force field feature extraction for ear biometrics. Comput. Vis. Image Underst. **98**(3), 491–512 (2005)
9. Yan, P., Bowyer, K.W.: Biometric recognition using 3D ear shape. IEEE Trans. PAMI **29**(8), 1297–1308 (2007)

10. Yaqubi, M., Faez, K., Motamed, S.: Ear recognition using features inspired by visual cortex and support vector machine technique. In: International Conference on Computer and Communication Engineering (ICCCE), pp. 533–537 (2008)
11. Islam, S., Davies, R., Bennamoun, M., Mian, A.: Efficient detection and recognition of 3D ears. Int. J. Comput. Vis. **95**, 52–73 (2011)
12. Wang, X., Xia, H., Wang, Z.: The research of ear identification based on improved algorithm of moment invariants. In: Third International Conference on Information and Computing (ICIC), p. 58 (2010)
13. Gutierrez, L., Melin, P., Lopez, M.: Modular neural network integrator for human recognition from ear images. In: The 2010 International Joint Conference on Neural Networks (IJCNN) (2010)
14. Alaraj, M., Hou, J., Fukami, T.: A neural network based human identification framework using ear images. In: TENCON (2010)
15. UND (2005) Database. http://www.nd.edu/cvrl/CVRL/DataSets.html
16. USTB (2002) Database. http://www.en.ustb.edu.cn/resb/
17. IIT Delhi ear database. http://www4.comp.polyu.edu.hk/∼csajaykr/IITD/Database_Ear.htm
18. Liu, H., Liu, D.: Improving adaboost ear detection with skin-color model and multi-template matching. In: 3rd IEEE ICCSIT, vol. 8, pp. 106–109 (2010)
19. Chowdhury, M., Gao, J., Islam, R.: Fuzzy logic based filtering for image de-noising. In: IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2016), Vancouver, Canada (2016)
20. Castillo, C.D., Jacobs, D.W.: Using stereo matching for 2D face recognition across pose. In: Proceedings IEEE International Conference Computer Vision and Pattern Recognition (2007)
21. Ashraf, A.B., Lucey, S., Chen, T.: Learning patch correspondences for improved viewpoint invariant face recognition. In: Proceedings IEEE International Conference Computer Vision and Pattern Recognition, June 2008
22. Chowdhury, M., Gao, J., Islam, R.: Fast stereo matching with fuzzy correlation. In: IEEE Conference on Industrial Electronics and Applications (ICIEA 2016), Hefei, China (2016)
23. Chowdhury, M., Bhuiyan, M.A.: Fast window based stereo matching for 3D scene reconstruction. Int. Arab J. Inf. Technol. **10**(3), 209–214 (2013)
24. Fusiello, A., Trucco, E., Verri, A.: A compact algorithm for rectification of stereo pairs. Mach. Vis. Appl. **12**, 16–22 (2000)
25. Kumar, R., Selvam, P., Rao, K.N.: Pattern extraction methods for ear biometrics: a survey. In: Proceedings World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), Coimbatore, India, pp. 1657–1660 (2009)
26. Chen, H., Bhanu, B.: Human ear recognition in 3D. IEEE Trans. PAMI **29**(4), 718–737 (2007)

WebAD^2: A Cascading Model Based on Machine Learning for Web Attacks Detection

Ying Lin(✉) and Bo Li

School of Computer Science and Engineering,
Beihang University, Beijing, China
{linying,libo}@act.buaa.edu.cn

Abstract. Anomalies in network are complicated and fast-changing, which pose serious threats to network security. In an intrusion detection system (IDS), achieving high detection rate and low false alarm rate is an essential requirement. Furthermore, faced with the explosive growth of network data, rapid recognition counts for as much as accuracy. In this paper, we propose a two-stage cascading model, named WebAD^2, for detecting web attacks. WebAD^2 applies machine learning techniques to detect anomalous behaviors. However, unlike traditional approaches, WebAD^2 divided machine learning process into two stages. In the first stage, partial but key features are selected for training and detecting to accelerate the detection speed. The intermediate results are passed to the second stage and all features are applied to refine the detection results, therefore improve the accuracy of the model. We conduct comprehensive experiments to evaluate the effectiveness and efficiency of WebAD^2. The results show that WebAD^2 could significantly improve the model efficiency without sacrificing the detection accuracy. The processing speed is reduced up to more than 70% on average, with an accuracy decrease less than 1%. What's more, the performance results on NSL-KDD also verify that WebAD^2 could be universal to detect network flow traffics.

Keywords: Web attack · Anomaly detection · Machine learning
Cascading model · URI analysis

1 Introduction

With the prevalence of web applications, web attacks increase with a tremendous speed and has become the top threat of Internet [1,2]. To counteract web attacks, Intrusion Detection Systems equipped with web analyzing functions emerge and play a more and more significant role in network security [3–6]. Intrusion detection systems search for malicious activities or policy violations by monitoring network traffic or host activities. In general, intrusion detection techniques could be divided into two types: misuse detection and anomaly detection. Misuse detection requires keeping the dictionary of attacks up to date, and are totally blind

X. Lin et al. (Eds.): SecureComm 2017 Workshops, LNICST 239, pp. 145–165, 2018.
https://doi.org/10.1007/978-3-319-78816-6_12

to zero day attacks. Anomaly detection could identify unknown attacks, however suffer from higher false positives rate compared with misuse detection techniques [7,8]. Recently, with the development and maturity of machine learning approaches, anomaly detection attracts great attention from academy and gains more and more application in industry.

With the explosion of network data, nowadays intrusion detection systems are facing with huger challenge than five years ago. An effective IDS should be capable of detecting anomaly with high accuracy and keep up with the continuously changing of network traffic. Real-time monitoring is of essential importance for IDS [9]. Simple anomaly might snowball into substantial harm because of the complex structure of network. Most available researches in anomaly detection are based on previous offline benchmark datasets which are old and out of date. Many approaches put great emphasis on algorithm optimization to achieve higher accuracy regardless of the expenses of space and time complexities. The reality is that these approaches could not be directly put into use because of bad performance especially when dealing with massive network traffic. One should keep in mind that detecting malicious behaviors efficiently and precisely is an essential function for Intrusion Detection Systems.

Another problem for anomaly detection is the lack of a representative accessible network traffic dataset due to the variety of networks, traffic profiles and attack types. Many researches still use the darpa'98 and kdd'99 cup as the benchmark datasets, which suffer from the problems discussed by McHugh [10]. And they may not be a perfect representative of existing real networks.

To address the above issues, we propose a two-stage cascading model, WebAD^2, to detect anomalies in web logs. WebAD^2 relies on machine-learning based classifiers to differentiate anomaly web traffics from normal traffics. Through our previous work on URI analysis [11] we found that it is possible to identify the majority of anomaly behaviors just using several high-quality features. Inspired by this observation, we design a cascading model to balance the efficiency and accuracy in the detection phase. We used two benchmarking datasets to evaluate WebAD^2. One is 440 GB Web Log data from CNCERT, which contains more than 11 attack types and 2,000,000 samples. Another is NSL-KDD, a dataset suggested to solve some of the inherent problems of kdd'99. Our main contributions are summarized as follow.

1. A two-stage cascading model is designed to balance the detection accuracy and efficiency especially when facing with massive network traffic. Partial features are selected in the first stage to reduce detection time. The intermediate results are passed to the second stage and all features are used to refine the detection results and improve the accuracy of our model. We choose the confidence score as a indicator to determine whether the web traffic data handled in the first stage should be passed to the second stage for further inspection or not.
2. These features using in first stage should be self-tuned to the actual conditions of different dataset. We also present a formula to compute the balance score between time and accuracy and select the best combination.

3. Experiments are conducted to evaluate how detection accuracy and time consumption changes with the varying of feature counts. The results convince that it is reasonable to detect most anomaly traffics with several features in a short time. These high cost-efficient features are selected as the seed features in the first stage of our model.
4. We evaluate WebAD^2 through comprehensive experiments on two real-life datasets: CNCERT web logs and NSL-KDD. By comparing the results of several classifiers, we found that our two-stage model can improve the modeling efficiency without sacrificing the accuracy.

The remainder paper is organized as follows. Section 2 presents the related work in the domain of anomaly detection and URL analysis. In Sect. 3, we explain the features processing approaches and algorithms. Section 4 describes the details of detection model and the approach to select high cost-efficient features. Then the experiment results of model in CNCERT web logs and NSL-KDD are illustrated in Sect. 5, and contrasted with the results of basic model. Section 6 concludes this paper and outlines future work.

2 Related Work

A Internet security report in Q1 2017 from [12] Akamai Technologies[1] reveals that SQLI, LFI (Local File Include), and XSS accounted for 93% of observed web application attacks. And Web application attacks increased nearly 35% compared with Q1 2016. That indicates the malicious attackers will not budge, and the number of Web attack will continue to grow.

Misuse detection and anomaly detection are two widely-used techniques in nowadays intrusion detection systems [3,4]. The core technique in misuse detection is pattern matching, which identifies "bad guys" by matching current records with known attack patterns. However, misuse detection suffers from the invisibility of unknown attacks such as 0 day vulnerabilities. Anomaly detection approaches is intrinsically different from misuse detection. A normal behavior model is established by learning upon benign network traffic. In the detection phase, the normal model is used as a baseline to identify outliers. With the renaissance of artificial intelligence and machine learning, anomaly detection became one of the hottest research area in network security. Generally speaking, anomaly detection could be classified into three categories: statistics-based methods, data mining-based methods and machine learning-based methods [13,14].

Statistics-based methods aim at discovering distribution information through statistical techniques. Some research works focus on the statistical analysis in URI, G.V. [15,16]. Gaussian distribution and Markov model are applied to analyze attribute length, attribute character distribution, structural inference, token finder, attribute presence or absence and attribute order. However, the computation only depending on the weight will lead to low detection rate. Study in

[1] An American content delivery network (CDN) and cloud services provider.

[4] is based on the analysis of URL entropy. Split URL strings into tokens with delimiters and calculate its entropy. Entropy can be viewed as a summary of a string. The key is that normal URL is much simpler than abnormal, which can find some complicated anomalies but also may neglect some anomaly whose entropy is small.

Data mining-based methods have various kinds. Research [17] achieves high detection rate relying on a combination of association rule and fuzzy set theory. [18] recognizes traffics using anomalous entropy to reduce false alarm rate with random forest; [19] applies KNN, Bayes Network and Random Forest to classify traffic, and clustering was employed to recognize the unknown applications. Methods in [20,21] aggregate the traffic flow by density-based clustering and sub-space clustering, and detect anomaly by ranking.

Machine learning-based methods mainly consist of SVM, Markov model, PCA, etc. Study in [22] relies on SVM and random forest to improve classification accuracy and reduce the runtime. [23] achieves low false alarm rate with the kernel function of Gaussian Radial Basis Function. Fan [24] proposes an ensemble approach which can effectively identify web-based attacks using hidden Markov models with different parameters. [25] applies PCA for traffic anomaly detection to improve precision, which is sensitive to noise though.

Feature selection technique is also the important research field of anomaly detection. Iglesias and Zseby [26] propose a multi-stage feature selection method using filters and stepwise for network traffic based anomaly detection. [27] has focused on many existing feature selection techniques to remove irrelevant features from NSL-KDD dataset to develop a robust classifier that will be computationally efficient and effective. Feature selection aims to increase the efficiency of machine learning. WebAD^2 could further improves the efficiency after feature selection.

Most existing anomaly detection algorithms focus on improving prediction ability. Nevertheless, in the face of the challenge of a large amount of data to predict, computing time and space consumption are important factor to be optimized. Consequently, in this paper, we apply a cascading model to improve the modeling efficiency without sacrificing the accuracy. Many previous algorithms are employed and get a great promotion in our model. And this model could be employed to improve the performance of existing system such as distributed system or parallel system. They have the same goal to accelerate processing without any conflict.

3 Features and Preprocessing

3.1 Data Model

Feature extraction and feature selection can eliminate strong correlated, redundant and irrelevant features to reduce data redundancy and storage consumption. It also provides a superb way for better and easier understanding of dataset. Many investigators have explored on feature selection. For better performance, we take advantage of previous studies to select the feature of our dataset of web log. Many analyses about web logs regard URL or URI as an important property. Owning to its universality and intelligibility in log records, the most of features we used are based on the URL or URI. A method of token is employed to deal with URI without any query in this paper. We classify all features into 4 categories: Number-related Features, Length-related Features, Character -related Features, Structure-based Features, as Table 1 shown. The main features will be introduced later.

Table 1. All features

ID	Name	Type	Introduction
Number-related features			
1	Num_digit	Integer	Number of digit in URI string
2	Num_letter	Integer	Number of letter in URI string
3	Num_punctuation	Integer	number of punctuation in URI string
4	Num_token	Integer	Number of token in URI string
5	Num_parameter	Integer	Number of parameter in URI string
Length-related features			
6	Length_URI	Integer	Length of URL
7	Length_max_token	Integer	The max length of token
8	Length_min_token	Integer	The min length of token
9	Length_min_parameter	Integer	The min length of parameter
10	Length_max_parameter	Integer	The max length of parameter
11	Length_cookie	Integer	Length of cookie
12	Length_post	Integer	Length of post
13	Length_referer	Integer	Length of referer
Character-related features			
14	Character_cookie	Integer	Anomaly probability in cookie character
15	Character_parameter	Integer	Anomaly probability in parameter character
16	Character_URI	Integer	Anomaly probability in URI character
17	Character_post	Integer	Anomaly probability in post character
Structure-based Features			
18	Relative entropy	Float	Relative entropy of URI string
19	Depth	Integer	The URI path depth
20	Token	Integer	Anomaly probability of token

Each record in the Web Log are processed and converted into a associated vector $\mathbf{V} = [\mathbf{v_1}, \mathbf{v_2}, ...; \mathbf{v_k}]$, where v_i represents the value of i^{th} feature, and k is the total number of features we used. The classifiers of machine learning are applied to categorize traffic. Records are sperated into two categories: normal data are labeled as 0, while abnormal data are labeled as 1. Relative values of several features are depicted in Fig. 1. For simplicity of exposition, all values are standardized between $0-1$. It's clear to see that, the Depth, Digit, Letter, Token of attack are higher than access, while Relative Entropy is contrary. The details of processing for some important feature are described as follow.

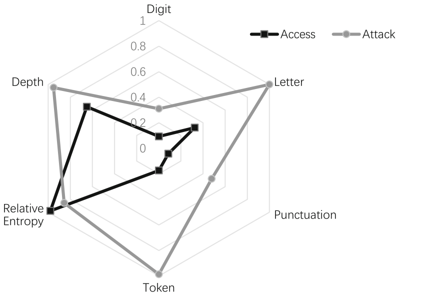


Fig. 1. Rader map of relative value of features

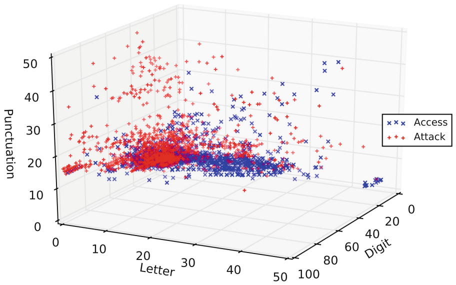


Fig. 2. The frequency of letter, digit, punctuation in URI

3.2 URI Token

URI is regarded as an important feature in many previous studies on HTTP web logs, and they usually use the parameters of URI query. However, there are some URIs without query in our dataset. We benefit from the method that Naive Bayes handles the problem of spam emails and it considers each word in path is equal. In this paper, every URI path is divided into some tokens. Naive Bayes are employed to calculate the abnormal probability of a URI path. Finally simplify the probability value to 0 or 1. During the actual operation, the URI path sring is segmented by delimiters. For example, URI "www.example/show_shiji/id/46485.php" is convert to token set [*'www'*, *'example'*, *'show'*, *'shiji'*, *'id'*, *'php'*]. For easier analysis, pure digital and single letter are ignored.

Statistical analysis of tokens frequency contributes to anomaly detection. For the frequency of these tokens appears differently in normal activities and abnormal. We can observe in Fig. 1, the anomaly value of *Token* is higher than normal. For example, in SQL injection, the "select" and "from" is easy to observe. We can simply regard an URI string as a Token set. In many existing researches, features in language are expressed by N-gram, usually "1-Gramm" and "2-Gramm", using Markov model to describe the structure information with higher prediction ability. However, the structure information displayed in the URI is not really obvious. We use Naive Bayes method to determine the abnormal or normal probability of a URI string, which is faster than Markov model in computation, and shows better prediction performance also.

Every token is regarded as independent in Naive Bayes. The particular arithmetics are described as follow.

1. Split URI string to get the Token list by recognizing delimiter, such as []=?@|${}.
2. Filter out the pure individual letters and numbers of token. We can get *Num_token* in this step also.
3. Union these token lists to a token set.
4. According to the labeled data, calculate the prior probability $P(Y)$ and the conditional probability $P(X_i|Y)$ of every token in normal and abnormal respectively.
5. Calculate the normal and abnormal posteriori probability of every URI path string as:

$$P(Y|X) = \frac{P(Y)\prod_{i=1}^{d} P(X_i|Y)}{P(X)}. \tag{1}$$

6. Simplify the normal and abnormal probability of every path string as the values of feature *Token*. $Ft = 1$ means that the URI is more possibly abnormal than normal:

$$Ft = \begin{cases} 0, & \text{if } P(Y=0|X) > P(Y=1|X) \\ 1, & \text{if } P(Y=0|X) \le P(Y=1|X) \end{cases}. \tag{2}$$

3.3 Relative Entropy

Entropy is quoted in statistics and information theory discipline to represent the uncertainty of an event. Threepak and Watcharapupong [4] inspect web attacking scripts usually have more sophisticated request patterns than legitimate ones. Web Log file is one of the important sources to obtain evidence of the attack. And in URI, web attack requests may contain more repeated contents than normal requests.

We imitate the same splitting method as URI token for simplification, and calculate the entropy of URI path string referring to existing studies [4,18]:

$$E_t = \frac{1}{\lambda}\sum_{i=1}^{N} p_i \left[\log(\lambda) - \log(p_i)\right] \tag{3}$$

$$E_{\max} = \frac{1}{\lambda}\sum_{i=1}^{\lambda} 1\times\left[\log(\lambda) - \log(1)\right] = \log(\lambda) \tag{4}$$

$$E_{rel} = \frac{E_t}{E_{\max}}. \tag{5}$$

where an URI, contains λ tokens with n distinct ones $(n < \lambda)$, p_i $(i = 1$ to $n)$ denotes the frequency that the i^{th} word appears. The relative entropy (E_{rel}) is formulated for simplification and normalization in formula (5). The maximum entropy ($E_{\max}$), is the entropy of URL with all tokens occur only once, formulated in formula (4).

In general, abnormal attacks are more sophisticated than the normal requests. The farther to 1 the relative entropy value is, the more sophisticated the URL request is. This method eliminates some complicated unusual URL, but it does not work on these anomalies with high entropy.

3.4 Character

According to the observation, the character distribution varies in different websites. Nevertheless, characters appear in normal activities are generally steady, mostly human-readable and only printable. A normal character set could be built by statistics of characters in normal URI strings. There are also some attack character groups like '..', '.fg./' only in abnormal URI request instead of normal. The value of *Character_URI* is to present the probability of anomaly with character. The Character of normal URI request is 0 and abnormal is 1 respectively.

If there are characters of an URI path outside the normal character set, the value is 1, otherwise is 0. The method can also be applied to detect the anomaly of other string as well as URI request. Details of method are described as followed.

1. Learning phase: we aim to get the normal character set C in this stage. C represents initialized an empty set. For every URI string in normal URI, $C = C \cup S_i$, where S_i denotes all characters of i^{th} URI string, $i = 1$ to M and M denotes the number of normal activities;
2. Predicting phase: for every URI string, $\forall c \in S_i$, if $c \in C$ the probability value of URI character is 1; else, is 0. If there are characters like ' ./ ', ' /. ', ' .. ', ' ** ', ' */ ', ' select% ', '<script>', the value of URI character is 1;

It is interesting to notice that the approach for character also can be adopted to detect the anomaly of other string, such as *Character_cookie*, *Character_parameter*, *Character_post* in Table 3, not only URI path string.

3.5 Digit, Letter and Punctuation

There are some rules and syntaxes in natural language. The frequency of the characters is different in English such as "E" is the most common. So we have a hypothesis that this phenomenon is also possible in the distribution of URI string. We analyze the statistical result of the normal and abnormal character frequency and find that the attack and normal access frequency distribution are different in characters. For clear understanding and easy computation, all characters are classified into three categories: digit, letter and punctuation. As Fig. 1 showing, the relative frequencies of digit, letter and punctuation are distinct in normal and abnormal. Because of the complexity of anomaly, some attack contents could be contained in the URI quest. What's more, there may be more specific punctuations in abnormal attack. Such as one attack URI quest:

/article/youbianjc/you?seq=../../../../../../../../etc/passw

The frequencies of punctuation '/' and '.' are higher than normal quest. As Fig. 2 illustrates, there are more punctuations and digits in anomaly than normal access, so we add *Num_digit*, *Num_letter*, *Num_punctuation* to the feature set, whose contributions are verified positive in the experiments.

3.6 Depth

The URI path depth of a normal site is usually 3 or so commonly. Because the deeper link is, the more difficult it is for user to obtain the needed information quickly, and for search engine to crawl the web. We simply identify '/' to determine the depth, and analyze its distribution. The depth of normal access are concentrated in smaller value than anomaly, which is painted in Fig. 1. For example, there are many of '/' in an anomaly URI request:

/fgs/index.php?s=/article/show/id/{*$*{*@phpinfo()*}}

4 Cascading Model Based on Machine Learning

This paper aims to maintain high accuracy as well as efficiency. The key is to build a two-stage model, which could complete anomaly detection work in two phases efficiently and effectively.

4.1 Motivation

Under the great pressure of massive data, it makes no sense to pursue only on high accuracy and ignore the time consumption, especially in the circumstance of real-time detection. Hence to minimize the time in a acceptable range of accuracy is one of the important studies in anomaly detection. While now many researches focus on the improvement in accuracy of algorithm, ignoring the time consumption. We conduct the optimization in the detection model instead algorithm.

Different features have different contributions to the classifier. Feature selection and feature Extraction can reduce the number of features, which are an important for machine learning to improve efficiency and avoid curse of dimensionality. Removing irrelevant or redundant features or components brings obvious benefits in terms of computational resources, such as reducing resource consumption for processing, storing and transmitting data, improving the prediction performance of the classifiers, and providing a better understanding of data. After Feature selection, for the same objective, we further assume that using several features with high contribution could detect most of the anomalies in a shorter time than using all of features. And for feature Extraction, such like PCA, WebAD^2 can also used to speed up the performance. The ranking of eigenvalues is of significance to select the best components for quick recognition in the first step, which could enhance the time performance of PCA. A statistics experiment is conducted on our dataset of web log. Figure 3 reveals the variation of accuracy and time on average in our web log dataset. It is observed that with the increase of feature counts, the growth of accuracy levels off gradually, but the increase of time is still clearly. When feature numbers go over 4, the accuracy starts to be in slow growth.

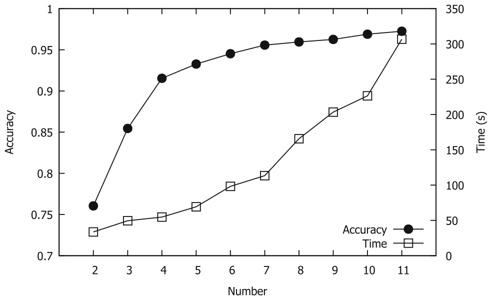


Fig. 3. Variation of accuracy and time with feature number increasing

4.2 Cascadinng Model

We draw on cascading model to hierarchically settle the conflict of efficiency and accuracy. The first stage of WebAD^2 is designed to reduce the cost of time and space. Several high cost-efficient features are picked out according to the practical situation. The second stage aims at increasing the accuracy, trained with all features to ensure the performance of detection. Figure 4 illustrates the structure of two-stage model. As can be seen from the left half, during the process of training, all labeled samples in learning set are preprocessed into the feature vectors. The first model adopts the feature vectors with several cost-efficient features, while the second employs all features to refine the result. The right half displays predicting process. Samples in the testing set are preprocessed into an associated vector only involving partial cost-efficient features. Then classifiers in the first model determine the category in the light of the confidence score. These samples exceeding the threshold θ of confidence score are identified by the first stage. The others whose confidence score are dissatisfactory stream down to second stage, and determined the category with all features. Normal data are labeled as 0, while abnormal data are labeled as 1.

The first level model constructs with the several high cost-efficient features, which is ideal for fast data collection in both machine learning and predicting stages. In weblog dataset, we select four most cost-efficient features. The features in first stage are self-tuned according to the actual requirement, and also can be captured with the method in Sect. 4.3. In predicting set, unlabeled samples are preprocessed into feature vectors, only consisting of these cost-efficient features. Then classifiers in the first model calculate the confidence scores, which present the anomaly and normal probability of these vectors, and every classifier can be assigned with different threshold. This threshold value is to determine the destination of one sample. There are two outputs of every classifier: (1) If its confidence score is below the threshold θ, this sample must stream to the second stage. (2) If its probability exceeds θ, the classification process will suspend, and this sample is identified as the corresponding label. As for the value of θ, in practice, first we can set a small threshold, and increase it gradually until reaching the acceptable accuracy.

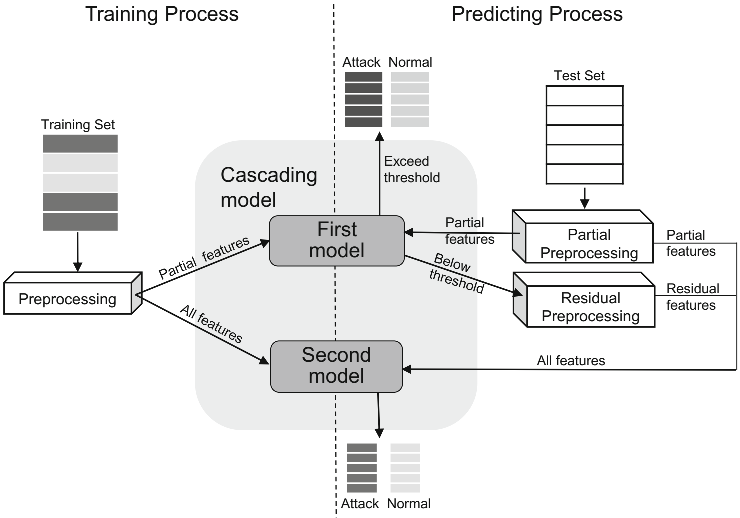


Fig. 4. The structure of two-stage model WebAD^2

Any classifier can be employed to learn and predict samples, such as Decision Trees, Random Forest, Logistic Regression, Adaboost, and Support Vector Machines. Every classifier can be assigned different threshold. The confidence score is a indicator to determine whether the sample should be passed to the second stage for further inspection or not. Sample is labeled as the anomaly if confidence score exceeds the threshold. The threshold makes a noticeable difference on the performance of second stage by impacting the number streaming down from the first stage. The higher threshold is, the more samples the second stage will process. More samples will increase the time and storage consumption, resulting in a lower efficiency. Nevertheless, a lower threshold may retain more samples in the first stage, which make the classification result suspect. When very large volumes of data need to be processed, the setting of threshold is crucial to balance the accuracy and efficiency.

The second level model collects more information from all the features of these samples difficult identified in first stage. Any classifier can be employed to learn and predict samples, the same as the first stage. When training, for better understanding of dataset, all training samples are exploited. The second stage only predicts these data streaming down from the first stage, whose vectors are added with the residual features for higher accuracy and recognition rate. Although the residual features may be generally time-consuming to calculate and possibly make less contribution than the highest cost-efficient features.

For most samples are retained in the first stage, the reduction in the number of samples results in the decrease of prediction time. What's more, the threshold set in the first stage makes a crucial difference on the samples number in second stage, which then impacts the time of second step. We set a small value of θ, and increase it to satisfy an acceptable accuracy. Especially, suppose that the threshold θ is set as 1.0, which means all the sample will stream down to the second stage. Whereas, it is futile, even counterproductive, as it has cost plenty of time in the first stage.

4.3 Combination Selection of Features

This paper addresses a method to select partial features for the first stage model, which is self-adaptive to select the highest cost-efficient features for distinct datasets involving totally different features.

We understand that several features in the first stage make a great difference in the promotion of performance. Another problem to be solve is how to pick out the highest cost-efficient features. It is an apparent fact that different combinations of features lead to different performance results. We create a situation to study how they behave in the same environment, where 2 million of identical web log data with different combinations of features need to be detected by Decision Tree. It takes 4.71 s for one combination consisting of *Num_digit*, *Num_letter*, *Num_punctuation* and *Depth* to get the accuracy of 91.17% in the second stage. Another combination is composed of *Depth*, *Token* and *Relative Entropy*, whose accuracy reaches up to 94.24% but in 21.55 s. This distinction is reasonable that higher accuracy are probably at the cost of longer time.

So we should as far as possible reduce time consumption at a prerequisite of satisfying accuracy. There are a lot of research on feature selection for anomaly detection, ranking the traffic features according to their contribution. Research [26] proposes a multi-stage feature selection method using filters and stepwise regression wrappers. There are two ways to select features for the first stage. If your original data have employed the feature selection or feature extraction, the features combination in first stage can be selected on the basis of the ranking information. Another method, as formula (6), (7), is to calculate the combination score (CS) to balance accuracy and time consumption. These formulas is derived from F-Measure, as formula (8).

$$T' = 1 - \frac{T}{T_{\max}} \tag{6}$$

$$CS = \frac{1}{\frac{1}{\lambda F} + \frac{1}{(1-\lambda)T'}} = \frac{\lambda(1-\lambda)FT'}{\lambda F + (1-\lambda)T'} \tag{7}$$

$$F = \frac{2}{\frac{1}{P} + \frac{1}{R}} = \frac{2PR}{P+R} \tag{8}$$

Where F denotes the F-score, T denotes the time of detection. T_{max} is the maximum time and T' is the normalization of T. λ $(0 < \lambda < 1)$ is adopted to adjust the weights of F and T, often set as 0.5. The F-score is a measurement to evaluate accuracy in statistical analysis of binary classification. We also take advantage of F-score as the evaluation standard of the performance of our model in experiments. Combination selection method will rank the balance scores of different combination of features and select the highest value after the setting of features number in the first stage. This method does not take the internal information of dataset into consider like correlation and redundancy of features. It only believes the results as important, which is straightforward but effective and more universal in different IDSs.

5 Experiments and Analysis

In this section, validation experiments are performed to verify the effectiveness of classification in WebAD^2 compared with a basic model. Basic model refers to one-stage model proceeding all features in one time. We apply the metrics of precision, recall, FPR and F-score to evaluate the performance of models. All experiments are run in a machine with Intel Core i5-4570, 2 GB memory and 3.20 GHz CPU under Windows 10.

5.1 Dataset

Due to the variety of networks, traffic profiles and attack types, the representativeness of any dataset for intrusion detection is circumscribed. The network research community still lacks of a representative accessible network traffic dataset. Many of the published researches in anomaly detection still apply the darpa'98 and kdd'99 cup. However, because of the lack of public datasets for network-based IDSs, KDD dataset still suffers from some of the problems discussed by McHugh [10] and may not be a perfect representative of existing real networks.

In light of this, we verify the performance of WebAD^2 on two datasets. First, we use a dataset of web log data from The National Computer Network

Table 2. Attack type in web log dataset

Type	Introduction
SQLI	SQL injection attack
XSS	Cross-site scripting
CODE	Arbitrary code execution vulnerability
COLLECTOR	A malicious content acquisition
SCANNER	Malicious scanning
FILEI	Access to sensitive directory/file
RLFI	Remote/local file contains
OS_COMMAND	Arbitrary command execution
WEBSHELL	Webshell access, contains PHP DDOS
SPECIAL	Particularity of the attack
OTHERS	Others

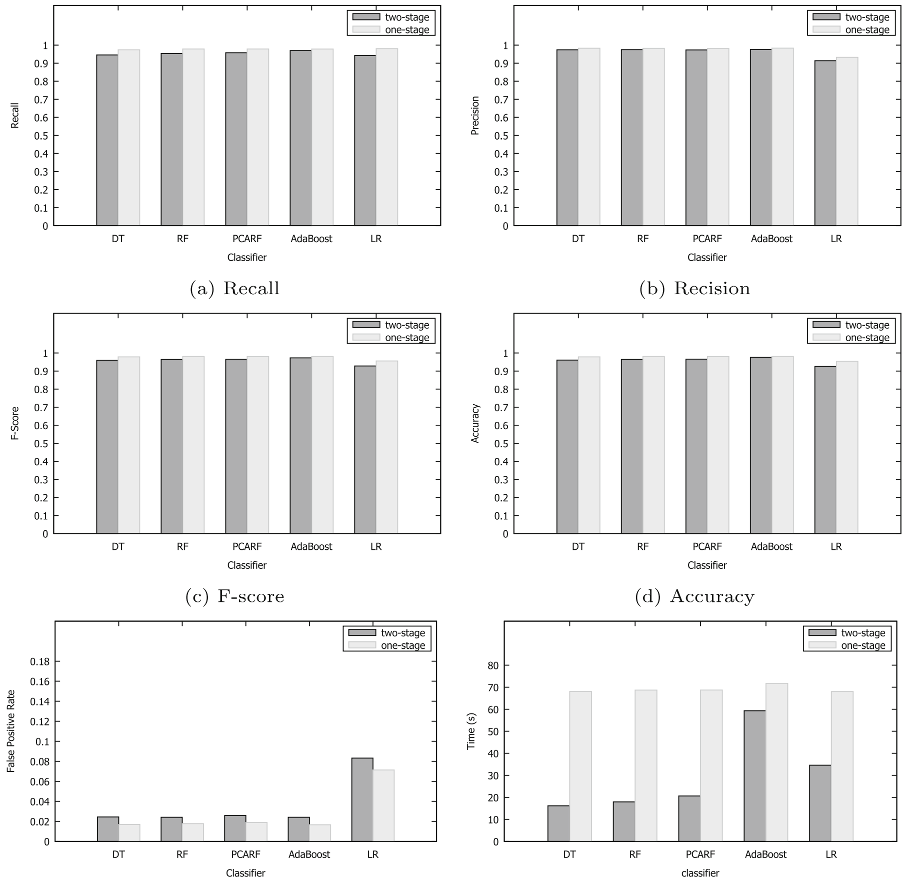
Emergency Response Technical Team/Coordination Center of China (known as CNCERT or CNCERT/CC). The data information is actual and reliable from large-scale web sites. As shown in the Table 2, this dataset contains more than 11 attack types such like SQLI, XSS, DDos. And there are more anomalous samples in the web log dataset compared with KDD CUP99, that half are anomalous samples. This 440 GB dataset consists of 2,000,000 records selected randomly. Both normal and abnormal records are 1,000,000. What's more important, the data are labeled, making great contribution to the learning of classification. Normal records are labeled as 0, while abnormal records are 1.

5.2 Experiment on Web Log Dataset

First in CNCERT web log dataset, we conduct experiments about the performance of two-stage WebAD^2 and one-stage basic model to compare the accuracy and time consumption. The threshold θ of confidence score is set as 0.8. In Fig. 5, we depict the distinction of two different models using the same dataset and setting. The time consumption including preprocessing and detecting significantly decrease in WebAD^2. For example, in the classification Decision Tree, when 400,000 data to be predicted, WebAD^2 spends 16.18 s while basic model needs 68.12 s, reducing more than 76%. The total accuracy of two stage reaches to 0.9627 and basic model is 0.9717, basically maintaining a high accuracy in a short period of time. So when facing challenge of the large data processing and real-time prediction, WebAD^2 has such an enormous advantage for rapid identifying. Random Forest (RF) and PCA Random Forest (PCARF) are based on decision tree, so they almost achieve the same performance. Random Forest constitutes of many decision tree, which could improve accuracy and reduce false alarm rate without over fitting. PCA Random Forest using the top components form principal component analysis (PCA) in the first stage. The one with best effectiveness is AdaBoost classification, but it has a lower efficiency, because of almost half samples stream down to second stage with the limit of confidence score. Logistic Regression (LR) also can improve the performance. Both classifications in two-level cascading model have remarkable advantage in recognition speed, meanwhile keep a good accuracy.

We take five-fold cross-validation for classification evaluation. In each validation experiment, the number of training dataset and test dataset are in a proportion of 4:1. Cross-validation process is repeated 5 times in total, and the average of all results from the folds is consider as a single estimation. Train all classifiers using train set and employ them to detect the anomalies in the test set. In the first stage, we only select *Num_digit*, *Num_letter*, *Num_punctuation* and *Depth* for classifier learning. Results are shown in Table 3.

We specify the procedure now. First of all, the combination selection of features in the first stage proceeds to select the highest cost-efficient in dataset. The combination of *Num_digit*, *Num_letter*, *Num_punctuation* and *Depth* reaches the best balance of accuracy and time consumption ultimately. We employ multiple classifier in the experiments to obtain the most suitable one. Each classifier can set different threshold. The threshold plays a noticeable role on the samples



(a) Recall

(b) Recision

(c) F-score

(d) Accuracy

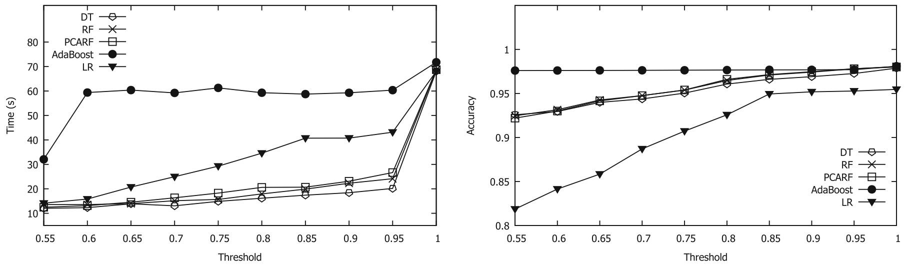
(e) False Positive Rate

(f) Time

Fig. 5. Performance in dataset of web log

number of second stage to impact the time and accuracy. Figure 6 illustrates the accuracy and time cost with the change of threshold. The higher threshold is, the more samples the second stage will process. Especially, when the threshold is set as 1.0, all the sample will stream down to the second stage. You can set a lower threshold, and constantly augment the threshold, until the correct rate in an acceptable range.

The second part of table shows the effectiveness of second stage in WebAD^2. More informations remain to be gathered for refined detection. The values in second stage are computed only in the samples streaming down from the first stage. Multiple classifiers are evaluated with five-fold cross-validation and AdaBoost also perform best. It is apparent and reasonable that the samples in second stage are hard to recognized. But the results remain higher accuracy, precision and recall, because all the features are employed in this stage. And the last part of Table 3 sums up the performance of two stage.



(a) Time

(b) Accuracy

Fig. 6. Performance in dataset of web log

Table 3. The classifier result in web log dataset

Classifier		Accuracy	Precision	Recall	FPR	F-score	Number
First stage	Decision Tree	0.96269	0.97934	0.94609	0.02036	0.96243	337104
	Random Forest	0.96890	0.98180	0.95584	**0.01790**	0.96864	332136
	PCA Random Forest	0.97039	0.98020	0.96021	0.01942	0.97010	322831
	AdaBoost	**0.97877**	**0.98627**	**0.98419**	0.03513	**0.98523**	160782
	Logistic Regression	0.88511	0.96603	0.88529	0.11555	0.92390	144767
Second stage	Decision Tree	0.94977	0.94835	0.94533	0.04623	0.94684	62896
	Random Forest	0.94454	0.94288	0.94334	0.05432	0.94311	67864
	PCA Random Forest	0.94775	0.94680	0.94857	0.05306	0.94768	77169
	AdaBoost	**0.97514**	**0.96919**	0.96002	**0.01662**	**0.96458**	239218
	Logistic Regression	0.94877	0.88428	**0.97553**	0.06482	0.92767	255233
WebAD^2	Decision Tree	0.96066	0.97447	0.94597	0.02443	0.96001	400000
	Random Forest	0.96477	0.97520	0.95372	0.02408	0.96434	400000
	PCA Random Forest	0.96602	0.97375	0.95796	0.02591	0.96579	400000
	AdaBoost	**0.97660**	**0.97605**	**0.96973**	**0.02406**	**0.97288**	400000
	Logistic Regression	0.92573	0.91387	0.94287	0.08318	0.92814	400000

From the Fig. 5 we can see that, all classifiers improve the modeling efficiency without sacrificing the accuracy. AdaBoost has the best effectiveness performance but costs more time than others. Because the majority of samples stream down to the second stage, which increases the time consumption, as the last column in Table 3 shown. Synthetically, Decision Tree, Random Forest and PCA Random Forest are the best choice for high accuracy and less time.

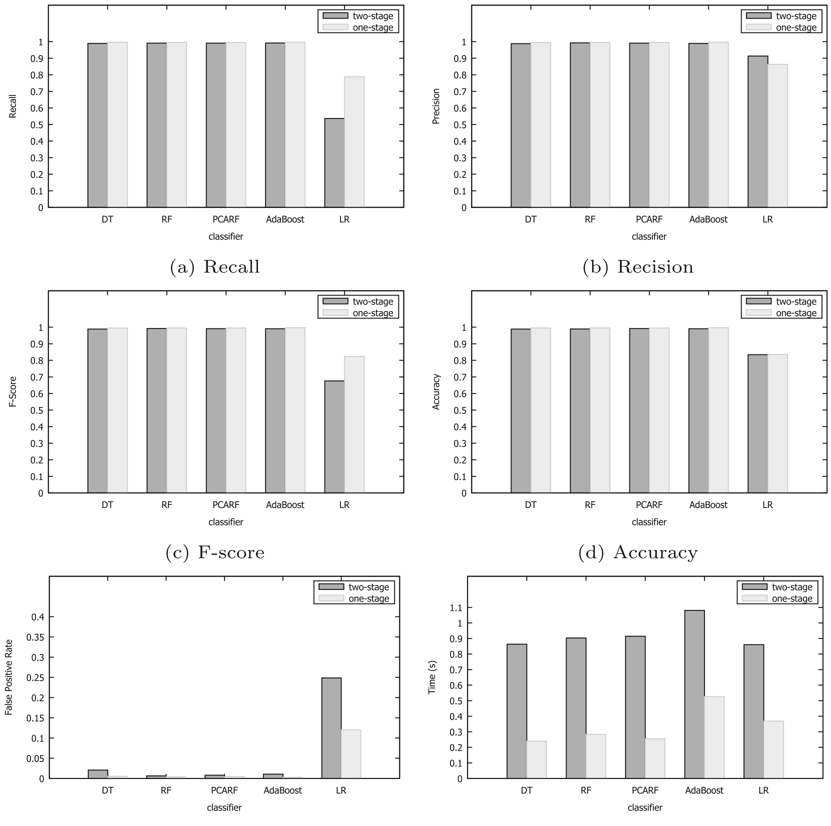
5.3 Experiments on NSL-KDD

Another set of experiments is requisite to indicate the universality of two-stage cascading model in different dataset. NSL-KDD is employed as an effective benchmark dataset to evaluate the intrusion detection methods. First, for there are some features in character form and classifiers only can predict the sample in numeric form, we preprocess the data turning them into numeric values. The number of the values of this features are constant. For example, one feature protocol type contains 4 different value: (1) ICMP; (2) TCP; (3) UDP; (4) others, so that can be marked as the sequence number correspondingly.

Table 4. The classifier result in NSL-KDD

Classifier		Accuracy	Precision	Recall	FPR	F-score	Number
First stage	Decision Tree	0.99006	0.98915	0.99080	0.01066	0.98997	29357
	Random Forest	**0.99462**	**0.99521**	**0.99395**	**0.00472**	**0.99458**	28878
	PCA Random Forest	0.99387	0.99388	0.99382	0.00608	0.99385	28473
	AdaBoost	0.99126	0.98898	0.99225	0.00960	0.99061	26018
	Logistic Regression	0.88173	1.00000	0.00063	0.00000	0.00127	11591
Second stage	Decision Tree	0.87719	0.89000	0.86829	0.86829	0.87901	346
	Random Forest	0.91158	0.91607	0.88631	0.06743	0.90094	825
	PCA Random Forest	0.94001	0.93069	0.92916	0.05185	0.92993	1230
	AdaBoost	**0.98724**	**0.99299**	**0.98903**	**0.01716**	**0.99101**	3685
	Logistic Regression	0.80385	0.85784	0.87950	0.40779	0.40779	18112
WebAD^2	Decision Tree	0.98875	0.98799	0.98937	0.02066	0.98868	29703
	Random Forest	0.98932	**0.99301**	0.99096	0.00646	**0.99198**	29703
	PCA Random Forest	**0.99164**	0.99126	0.99114	**0.00797**	0.99120	29703
	AdaBoost	0.99076	0.98948	**0.99185**	0.01054	0.99066	29703
	Logistic Regression	0.83424	0.91332	0.53654	0.24866	0.67597	29703

The results of experiments are showed in Table 4. In the classifier PCA Random Forest, the accuracy of in WebAD^2 is up to 0.99164, and the basic model is 0.99526. The performance in NSL-KDD is more obvious than web log dataset. WebAD^2 spends 0.25494 s to classify almost all anomalies and normal accesses, only 27.9% of 0.9144 s the basic model costs. These trials robustly demonstrate that two-level cascading model can greatly improves the modeling efficiency without sacrificing the accuracy as the Fig. 7 depicts. The accuracy of all the classifiers are close to 1, with time falling by about one third. The performance in NSL-KDD is even better than in the web log dataset, which indicates the universality of two-level cascading model.



(a) Recall

(b) Recision

(c) F-score

(d) Accuracy

(e) False Positive Rate

(f) Time

Fig. 7. Performance in dataset of NSL-KDD

To summarize the above experiments, we can get that:

1. The first stage can identify most of anomalies in a short time. It manages the balance of time and accuracy, and ensures reliability through the setting of suited threshold. The first stage makes a significant contribution to the reduction of time in cascading model;

2. The second stage can reach a higher accuracy exploiting all features but need more time. But due to it only dispose a small quantity of samples streaming from the first stage, its total time is less than the basic model. The excellent contribution of second stage is to improve the accuracy;
3. The second stage adopts more features than first stage, so that shows better power to accurately distinguish normal traffics and abnormal traffics. But note that this model relays on the first stage model to classify most of the samples no matter what classifier is used.

6 Conclusions

In this paper, we propose a two-stage anomaly detection model, named WebAD^2, to identify web attacks and anomalies. Our studies reveal that about 85% attacks or anomalies could be identified by exploiting only a small set of features. Therefore, in the first stage, we select partial but key features to differentiate anomalies from normal web logs and gain significant performance boost. In the second stage, all features are exploited to finer-tune the detection results and achieve a satisfactory detection accuracy. WebAD^2 also could be employed to improve the performance of existing system such as distributed system or parallel system to further accelerate the processing. This paper also puts forward a feature selection method to choose cost-efficient features in the first stage, which ensures that an appropriate balance between accuracy and detection efficiency could be maintained.

The experimental results show that WebAD^2 can greatly reduce the time consumption without sacrificing anomaly detection accuracy. Besides, WebAD^2 could deal with massive web logs, and still meets the demand of real-time detection. In future work, we will apply clustering algorithms to further classify the anomalies and obtain fine-grained detection results.

References

1. Prokhorenko, V., Choo, K.K.R., Ashman, H.: Context-oriented web application protection model. Elsevier Science Inc. (2016)
2. Prokhorenko, V., Choo, K.K.R., Ashman, H.: Intent-based extensible real-time php supervision framework. IEEE Trans. Inf. Forensics Secur. **11**(10), 2215–2226 (2016)
3. Kruegel, C., Vigna, G., Robertson, W.: A multi-model approach to the detection of web-based attacks. Comput. Netw. **48**(5), 717–738 (2005)
4. Threepak, T., Watcharapupong, A.: Web attack detection using entropy-based analysis. In: The International Conference on Information Networking 2014 (ICOIN 2014), pp. 244–247. IEEE (2014)
5. Peng, J., Choo, K.K.R., Ashman, H.: User profiling in intrusion detection: a review. J. Netw. Comput. Appl. **72**, 14–27 (2016)
6. Osanaiye, O., Cai, H., Choo, K.K.R., Dehghantanha, A., Xu, Z., Dlodlo, M.: Ensemble-based multi-filter feature selection method for ddos detection in cloud computing. Eurasip J. Wirel. Commun. Netw. **2016**(1), 130 (2016)

7. Bhuyan, M.H., Bhattacharyya, D.K., Kalita, J.K.: Network anomaly detection: methods, systems and tools. IEEE Commun. Surv. Tutor. **16**(1), 303–336 (2014)
8. Nadiammai, G., Hemalatha, M.: Effective approach toward intrusion detection system using data mining techniques. Egypt. Inform. J. **15**(1), 37–50 (2014)
9. Osanaiye, O., Choo, K.K.R., Dlodlo, M.: Distributed denial of service (DDoS) resilience in cloud: review and conceptual cloud DDoS mitigation framework. J. Netw. Comput. Appl. **67**(C), 147–165 (2016)
10. McHugh, J.: Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. ACM Trans. Inf. Syst. Secur. (TISSEC) **3**(4), 262–294 (2000)
11. Zhang, S., Li, B., Li, J., Zhang, M., Chen, Y.: A novel anomaly detection approach for mitigating web-based attacks against clouds. In: 2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing (CSCloud), pp. 289–294. IEEE (2015)
12. Akamai: Q1 2017 state of the internet/security report. Technical report, Akamai Technologies, Inc (2017). https://www.akamai.com/us/en/about/our-thinking/state-of-the-internet-report/global-state-of-the-internet-security-ddos-attack-reports.jsp
13. Hu, W., Hu, W., Maybank, S.: Adaboost-based algorithm for network intrusion detection. IEEE Trans. Syst. Man Cybern. Part B (Cybern.) **38**(2), 577–583 (2008)
14. Hu, W., Gao, J., Wang, Y., Wu, O., Maybank, S.: Online adaboost-based parameterized methods for dynamic distributed network intrusion detection. IEEE Trans. Cybern. **44**(1), 66–82 (2014)
15. Kruegel, C., Vigna, G.: Anomaly detection of web-based attacks. In: Proceedings of the 10th ACM Conference on Computer and Communications Security, pp. 251–261. ACM (2003)
16. Robertson, W.K., Vigna, G., Krgel, C., Kemmerer, R.A.: Using generalization and characterization techniques in the anomaly-based detection of web attacks. In: Network and Distributed System Security Symposium, NDSS 2006, San Diego, California, USA (2006)
17. Mabu, S., Chen, C., Lu, N., Shimada, K., Hirasawa, K.: An intrusion-detection model based on fuzzy class-association-rule mining using genetic network programming. IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.) **41**(1), 130–139 (2011)
18. Yao, D., Yin, M., Luo, J., Zhang, S.: Network anomaly detection using random forests and entropy of traffic features. In: 2012 Fourth International Conference on Multimedia Information Networking and Security, pp. 926–929. IEEE (2012)
19. Zhang, J., Chen, X., Xiang, Y., Zhou, W., Wu, J.: Robust network traffic classification. IEEE/ACM Trans. Netw. **23**(4), 1257–1270 (2015)
20. Casas, P., Mazel, J., Owezarski, P.: Unsupervised network intrusion detection systems: detecting the unknown without knowledge. Comput. Commun. **35**(7), 772–783 (2012)
21. Owezarski, P.: A near real-time algorithm for autonomous identification and characterization of honeypot attacks. In: Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security, pp. 531–542. ACM (2015)
22. Hasan, M.A.M., Nasser, M., Pal, B., Ahmad, S.: Support vector machine and random forest modeling for intrusion detection system (IDS). J. Intell. Learn. Syst. Appl. **6**(1), 45 (2014)
23. Bhavsar, Y.B., Waghmare, K.C.: Intrusion detection system using data mining technique: support vector machine. Int. J. Emerg. Technol. Adv. Eng. **3**(3), 581–586 (2013)

24. Fan, W.K.G.: An adaptive anomaly detection of web-based attacks. In: 2012 7th International Conference on Computer Science and Education (ICCSE), pp. 690–694. IEEE (2012)
25. Casas, P., Vaton, S., Fillatre, L., Nikiforov, I.: Optimal volume anomaly detection and isolation in large-scale IP networks using coarse-grained measurements. Comput. Netw. **54**(11), 1750–1766 (2010)
26. Iglesias, F., Zseby, T.: Analysis of network traffic features for anomaly detection. Mach. Learn. **101**(1–3), 59–84 (2015)
27. Hota, H.S., Shrivas, A.K.: Decision tree techniques applied on NSL-KDD data and its comparison with various feature selection techniques. In: Kumar Kundu, M., Mohapatra, D.P., Konar, A., Chakraborty, A. (eds.) Advanced Computing, Networking and Informatics- Volume 1. SIST, vol. 27, pp. 205–211. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07353-8_24

Human Factors, Self-awareness and Intervention Approaches in Cyber Security When Using Mobile Devices and Social Networks

Ken Eustace[1](✉), Rafiqul Islam[1], Philip Tsang[2], and Geoff Fellows[1]

[1] Cyber Security Research Group, Charles Sturt University, Boorooma Street,
Wagga Wagga, NSW 2678, Australia
{keustace,mislam,gfellows}@csu.edu.au
[2] Web Consortium Education Foundation, Hong Kong, China

Abstract. This paper will describe three case studies on the human factors, in personal and public safety and cyber security from the Asia Pacific region (APAC). A deeper consideration of human factors, the impact of "Internet of Things" and cyber security education about the behaviour and actions that can be taken by individuals is at the foundation of public safety and cyber security. The growth of disruption by cyber criminals - especially when using small devices and applications to interact with large social networks is a cause for concern. This is part of the evolving development of a cyber-physical world. The paper presents three case studies and proposes a *Self-awareness and Intervention Model* for public safety and security by increasing and *maintaining* the awareness, understanding and preparedness of cyber security measures by the individual when using mobile device applications to participate in large social systems and concludes by highlighting the importance of including the human factors and message framing alongside the cyber security measures in place.

Keywords: Android Application Security · Context-based behavior
Human factors · Internet of Things · Intervention · Message framing strategies
People with special needs · Public safety and security · Self-awareness
Vulnerability monitoring · Wireless access surveys

1 Introduction

There are several cyber security perspectives operating on ICT security projects and issues around APAC. On June 21, 2013, the Australian Government announced that cyber security will be one of the top priorities as part of 15 new Strategic Research Priorities for Australia with a focus on safeguarding personal security using partnerships and with responsibilities within the Asia Pacific region and the wider global context.

Guiding cyber security research activity and underpinning the role that Cyber Security research will play as part of those strategic research priorities, is Australia's Cyber Security Strategy [1].

X. Lin et al. (Eds.): SecureComm 2017 Workshops, LNICST 239, pp. 166–181, 2018.
https://doi.org/10.1007/978-3-319-78816-6_13

1.1 Improving Cyber Security in the Asia Pacific Region

Research projects involving cyber security are influenced by the transformations and growth of economies in APAC and should seek to identify ways to improve cyber security for individuals, organizations, businesses and scale up to include government, national infrastructure and even bilateral agreements with Asian countries.

Turban et al. [2] stated that CISCO expects that the average global network speeds will double by 2018 with the 5G networks at the top end of data transfer speeds, being offset by the amount of traffic due to mobile devices and social media interactions. This in turn requires new procedures to lower risks and increase security on mobile devices, cloud services and in social media sites.

Prieto [3] found that regional IP traffic for APAC will grow three-fold at 67.8 exabytes/month by 2020 and that 71% of total IP traffic will originate with non-PC devices including tablets, smartphones, and televisions, compared to 47% in 2015. At the same time by 2020, smartphones will generate 30% of total IP traffic, with PC's total IP traffic contribution to fall to 29%. With global networks supporting 16.3 billion devices in 2015 then going up to 26.3 billion devices by 2020, the growth due to the Internet of Things (IoT) that will create changes and more traffic increases. This will be partly due to the growth in video surveillance, smart meters, health monitoring and wearable applications.

According to Wilkinson [4], most cybercrime legal structures and cyber maturity [28] throughout the APAC region are not adequate to combat issues such as data and identity theft, child exploitation and ransom-based attack and that this is compounded with limited international support between the countries that face the burden of rampant digital criminal activity.

Online deception in social media, as one example, has more to do with human behaviours and education rather than use of intrusion detection and prevention technologies. According to Tsikerdekis and Zeadally [5], online deception in social media involves the cyber threat actor being:

> *"Unknown and invisible, ready to exploit the unwary and uninformed and seeking financial gain or reputation damage"*

There is a research interest in safeguarding the personal security and identity management of people using smartphones with particular concern for senior citizens and those with disabilities and in regional Australia, using social media and mobile applications. By building and safeguarding the cyber security awareness and behaviour for all citizens, the resulting improvement would trickle up to improve the other Cyber Security issues in the Asia Pacific region.

There is much concern about cyber security for seniors and people with special needs as the global trend shows that more seniors than ever before will be accessing technology: at home, on the move with smartphones and small devices or in aged care facilities, according to Harvie, Eustace and Burmeister [6–8]. This trend is likely to be encouraged as service providers use electronic interaction in order to save costs and more seniors and people with special needs expect online access to services.

2 Case Studies

The human factors associated with public safety and security are complex so we use three case studies to describe issues surrounding vulnerable people, application security and the ease and use of urban wireless access points.

The *Vulnerable People* case is about safeguarding the personal security and identity management of vulnerable people with using social media and mobile applications. The case study examines what strategies and behaviours can be taken to safeguard the independence, privacy, security and identity of people with disabilities, seniors or those in aged care.

The *Android Application Security* case changes the perspective and examines the security issues surrounding the popularity and growth of Android applications on smartphones and deals more specifically with application security on Android devices.

The final case is about *Wireless Access Points* and features the 2015 Wi-Fi Air, Sea and Land Survey in Hong Kong. The focus of this study is on the value of taking action on monitoring, understanding and educating about the adjacent wireless environment and its risks and vulnerabilities.

2.1 The Vulnerable People Case Study

The support and training in social media and mobile applications given to seniors and people with disabilities must include awareness and understanding of cyber threat actors and other cyber security issues. Phahlamohlaka's [9] interest on information warfare also included aspects of other research work on human factors and the socio-economic challenges for rural development as regional communities change with the increased use and wireless technology, connectivity and information [19]. What was happening in regional and rural South Africa may have similar effect regional and rural Australia as broadband and wireless cell phone networks expand by 2020 as part of a National Broadband Network (NBN). Those seniors and people with disabilities living in regional and rural areas face double jeopardy and may be part of the *Cyber Security Awareness Divide* as suggested by Connolly et al. [10].

Once we have in place the cybersecurity awareness and behaviours for seniors and people with disabilities then follows a program of wider education and strategies for building awareness and training needs of personal ICT security for all by using the phases for a cyberattack as a template or model for behavioural response and action in each phase. Developing a *Self-awareness and intervention Model* and maintaining the awareness, understanding and preparedness of cyber security measures by the individual begin with prevention measures. Then the *Detection* phase in cyber security can act as the *guardian* for the individual.

By adapting a simplified cyber-security model, after Stallings and Brown [11] based upon the common cyber security phases of Prevention + Detection + Response + Recovery, then each individual can follow a sequence of context-based behaviors to safeguard their data and identity during the four main phases and develop a *Personal Cyber Security Awareness and Intervention Model (PDR^2)* that is based on a set of context-based strategies and behaviours to strengthen each link in the cyber security chain as shown in Fig. 1.

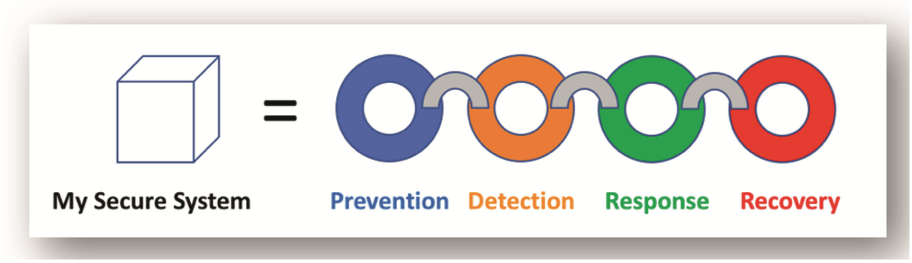


Fig. 1. PDR^2 is a Personal Cyber Security Model of Self-awareness and Intervention that represents a 'white box' full of cyber security strategies and behaviours.

While so many businesses and organizations concentrate on the *Prevention* aspect by setting up firewall and protection zones around a wired and wireless network or a password protection schema based on regular password changes every 30 or 60 days and use of advice on password 'strength (weak, medium, strong). In a recent discussion with colleagues in Hong Kong the benefits of the 'Great Firewall of China' was on the agenda.

At the same time, ethics and security go hand in hand to protect the privacy of the individual and assist with identity management, particularly with using social media technology. Indeed, identity management is also a concern for groups of people and the organization as whole. Additional safeguards also exist such as the different privilege levels that are given to people who can access the network.

However, a lot of ongoing protection can tend to be passive after a while as there is a tendency to just *setup and leave* the protection controls in place, even though all the protection measures are warranted, one is left to question how much of the individual, group or organization focus in on the more continuously active *Detection* phase of the model?

In business, government and large organizations like a university, the cyber-security effort is on a much larger scale than the home or office network and is targeted towards the protection of intellectual property (IP) and corporate data stores. In the latter case, so much of the new or cutting edge IP development is done by postgraduates enrolled in a course at the Doctorate or Research Masters levels.

Similarly, if we examine the *Response* phase of the model, then a lot of global support is available via the efforts done by CERT. The Computer Emergency Response Team (CERT) offers to members a coordinated, active, up to date response to vulnerabilities on the network by releasing regular 'alerts' to problems as they emerge. Such quick alerts to vulnerabilities due to new malware or security holes in operating system and application software updates are quite effective in maintaining alertness in the behaviour of individuals. Getting the right balance is vital and shaped by the amount of time, resources and budget that an individual can apply or endure, while also affected by the overuse of protection and detection that will see the user experience and benefits reduced as security levels are increased. The result by going too far may a fortress attitude so there is a need to balance the detection measures so that they act as the effective guardian or the sentinel on the watch acting in the best interests of the individual, the group or the organization.

The use of good detection methods will provide the data to support the decisions made by the individual, the group or the organization and act as a feedback loop to the protection and response phases of the model. The perception with business and organizations may be that the stakeholders only take notice when an incident occurs, acting as bystanders at an accident or fire scene.

Such behaviours may be compared to the behaviour of individual, groups or organizations involved during fire or bomb scare drills. The behaviours will differ from the drill when the threat is real. Such training is a drill and the practiced behaviours may not resemble what happens in the real situation. It is proposed that the detection phase as the 'guardian' will provide the data, information and experiences needed for the other prevention and response phases in the personal cyber security model.

De Bruijn and Janssen [12] suggest that our daily dependence on ICT has created the *cyberphysical* society, and that the need and demand is now greater to understand the complex and varied aspects of cyber security. They also propose the use of an evidence-based message framing strategy is needed to *frame cyber security*, similar to the use of the three case studies in this paper. Six communication strategies were identified by De Bruijn & Janssen as providing a way to frame or explain cybersecurity (See Table 1).

Table 1. The communication strategies that can be used to explain cybersecurity to the community.

	Six communication strategies by De Bruijn and Janssen [12]
1	*Do not exacerbate or worsen cybersecurity*
2	*Make it clear who the villains are*
3	*Give cybersecurity a face by putting heroes in the sunlight*
4	*Connect cybersecurity to values other than security alone*
5	*Personalise the message for easy recognition*
6	*Connect to other tangible and clear issues*

However, communication strategies 2 and 3 presents some risk. In *making it clear who the villains are*, we risk promotion and copying by others, while in *putting heroes in the sunlight*, we risk making them targets for attack. By educating and communicating the benefits of learning and using a simple personal cyber security model to each individual in a community, each person including seniors and those with disabilities, can exercise the personal cyber security to maintain or change to a set of behaviours in their own personal cyber-security model.

The issues around privacy and identity management exist in the use of social media technology (mobile apps), so all of us need to be aware and to learn how to apply our own personal version or instance of the simple personal cyber security model. Without implementing a personal cyber security model or bothering to learn what behaviours are appropriate then the growth of cyber-security vulnerabilities can continue and Recovery times and the risks increase. This is where cyber security experts get involved in education and policy making through use of ideas like De Bruijn & Janssen's ideas [12] about the use of better communication by education and message framing strategies that help understanding by removing ambiguities.

Such a model should also be context-based and be developed as a user-centred model where each individual takes ownership of his or her own personal cyber security model. If the user-centred approach together with good message framing techniques can be made to work in tandem for those disadvantaged people in the community then it should be horizontally and vertically scalable to other people, groups, organizations using social media in our cyber physical society.

Some disadvantaged people and those in aged care have wearable devices for medical, rehabilitation or fitness purposes. This calls into question the security and privacy risks wireless protocols for medical devices (pacemaker, defibrillator) even though safeguarded by legislation and manufacturer *third party* protocols. The growth of wearable bands for fitness designed to use social media and cloud storage of personal fitness data so the user can download and view, also includes GPS data for tracking. The risks in information sending by the user coupled with storage and retrieval may mean the data can be read without anyone knowing. Just who owns the personal health data?

The use of trusted parties and privacy policy needs to be included with all wearable devices before the advent of whole body networks (WBN) using linked wearable devices such has a fitness band, a pacemaker and device a retinal implant. If all the individual personal cyber security models were in place and connected then each node in a WBN may re-enforce the next by working as a cyber-security mesh in practice.

2.2 The Android Application Security Case Study

Turban et al. [2] described how a botnet of Android phones was used to send large volumes of spam via Yahoo e-mail servers - using SMS as the "command and control channel". The era of the Smartphone has arrived and these devices permeate into every aspect of our daily lives, the importance of establishing effective security measures becomes increasingly important. In this literature, we analyse the current security systems in place, their evident shortcomings and the proven potential malicious or unsolicited behaviour. The streamlined nature of the application marketplace and rigid security implementation combined with a general lack of awareness and comprehension of security implications amongst end users is a legitimate cause for concern. There is a critical need for a revised security strategy surrounding the Android application framework and a number of proposed solutions are examined.

Stammberger [13] summed up the ubiquitous trend of smart devices in one simple statement:

> "*PCs are no longer the dominant form of computing*"

Preito [20] supports the rise of the smart devices with the CISCO data that showed that by 2020 then smartphones IP traffic will exceed that of PCs. As far back as 2008, mobile broadband connections had increased and exceeded that of fixed broadband subscribers and by the end of 2009 there were an estimated total of 4.6 billion cellular subscriptions worldwide according to the U.S. Department of Homeland Security [14]. Since then the popularity of smart phones has risen exponentially along with the popularity of mobile phone applications.

On Android-based devices, applications are predominantly downloaded and installed via the Android Market although other markets do exist such as Amazon's Appstore. On July 14 2011, it was reported by Nickinson [15] that the Android Market had surpassed 250,000 applications and over 6 billion downloads. Enck et al. [16] suggested that the:

> "*low barriers [for developers] to bring applications to market, and even lower barriers for users to obtain and use them*"

has undoubtedly promoted this uptake. Indeed, for a small fee, developers are able to freely produce and distribute applications in what is widely regarded as the "*unmoderated Android market*" according to Vidas et al. [17].

Dissimilar to desktop Operating Systems, applications on Android are treated as "*mutually untrusting, potentially malicious principals*" – as described by Felt et al. [18] requiring specific permission to be granted once only during install-time. End users are shown a page just before installation listing the applications requested permissions at which time they may accept all permissions and proceed or decline cancel the installation completely. Permissions are displayed in three layers; categories, specific permissions and a hidden details dialog. A survey conducted by Felt et al. [19] concluded that the categories were so broad that they caused users to over-estimate the implications resulting in "*a negative impact on the amount of attention that users pay to [specific] permissions*".

More worrying perhaps was the fact that 42% of the lab study respondents were completely oblivious to the existence of such permissions with one user saying "I don't ever pay attention. I just accept and download it" Felt et al. [19]. This trend is possibly facilitated by the "*streamlined*" nature of the current market. However, end user complacencies are not the only area lacking. Even for a technically minded individual the vague permission descriptions fail to provide sufficient information for effective decision making. One such respondent with a small amount of experience as an Android developer commented (Felt et al. [19]):

> "*I've done some programming but I don't know all the permissions. … I just don't know if the permissions are so fine grained that they make texting a special permission that you have to add*"

Overall, the current mechanisms in place are severely inadequate and simply too rigid. Essentially, when installing an application of undetermined integrity, a user must decide, or more fittingly take a somewhat educated guess, whether or not the permissions being requested are appropriate based on a single page of largely misunderstood permission descriptions and implications. In some cases, user reviews can assist users in making a decision but that requires actions and knowledge on behalf of other users and is by no means a satisfactory solution.

The potential for, and evidence of, unscrupulous application behaviour is all too real and unfortunately, largely underestimated. Stammberger [13] warns of a "*Dangerfield Paradox*" where the continuing abundance of PC-based attacks diverts attention away from devices "*despite the inevitability, importance, and difficulty of solving*" this arising issue. In fact, the issue has largely arrived already, with the same survey revealing that of the 269 respondents:

> "*65% report that attacks against their smart devices already require the regular attention of their IT staff, or will start requiring it this year. In fact, 23% of organizations surveyed already repel device attacks at least once monthly, while 10% must do so on a daily basis*" (Stammberger [13])

It also reveals the growing uncertainty amongst users with 77% expressing some level of concern about current mobile phone security. This concern is well-founded and justified by several other research articles which investigate the various exploitable aspects of the Android application framework as it currently stands. Areas of primary focus are the misuse and collection of sensitive and personal information, application permission re-delegation which undermines user-granted permissions themselves and the penetration of advertising and analytic libraries.

Enck et al. [16] developed a decompiler to extract 21 million lines of code from the top 50 applications across each of the 22 application categories for analysis (as of September 1, 2010). This comprehensive examination "*uncovered pervasive use/misuse of personal/phone identifiers, and deep penetration of advertising and analytics networks*". Such networks were found to be integrated into over half of the applications studied. An alternate taint tracking method, TaintDroid, was used to report similar findings. TaintDroid,

> "*automatically labels (taints) data from privacy-sensitive sources and transitively applies labels as sensitive data propagates through program variables, files, and inter-process messages*" Enck et al. [20]

and records this data as it attempts to leave the system. Although a significantly smaller sample size of just 30 popular applications, "*68 instances of potential misuse of users' private information across 20 applications*" [20] were detected. The penetration of advertising networks was also apparent with half the applications attempting to report the user's location to remote advertising servers, occasionally with additional private data such as IMEI or phone numbers.

Both studies reported "*an overwhelming concern for misuse of privacy sensitive information such as phone identifiers and location information*" (Enck et al. [16]). They revealed that phone identifiers are being transmitted and used for a whole range of purposes from "*cookie-esque*" tracking to account numbers and were frequently leaked in plain text and "finger-printed" on remote servers (Enck et al. [16]). It was also reported that IMEI numbers are often tied to personally-identifiable information discrediting "*the common belief that the IMEI to phone owner mapping is not visible outside the cellular network*" (Enck et al. [16]).

The probing of permissions was also a suspicious and widespread occurrence. This activity was found to be instigated from not only advertising and analytical libraries, but also from some developer toolkits which can lead to dangerous functionalities appearing in "*well-known*" branded applications. For example, the "*CBS Sports Pro Football*" application was found to exhibit permission probing behaviour whilst "*USA TODAY*" and "*FOX News*" programs were found to access IMEI data due to the developer toolkits used (Enck et al. [16]).

From all this we can derive one common theme that, as pointed out by Enck et al. [20],

> "*resolving the tension between the fun and utility of running third-party mobile applications and the privacy risks they pose is a critical challenge for smartphone platforms*".

Enck et al. [20], goes on to list the major challenges of protecting sensitive data on a Smartphone. Firstly, smartphones have limited resources restricting the *"use of heavyweight information tracking systems*" (Enck et al. [20]). The interactivity between applications on a device also present difficulties for monitoring systems to be able "*distinguish multiple information types, which requires additional computation and storage*" (Enck et al. [20]). Also, the dynamic nature of context-based information can be hard to identify. Enck et al. [20] points out that "*for example, geographic locations are pairs of floating point numbers that frequently change and are hard to predict*".

Enck et al. [20] claims TaintDroid to "*provide a novel, efficient, system-wide, multiple-marking, taint tracking design by combining multiple granularities of information tracking*" but is quick to point out

> "*like similar information-flow tracking systems, a fundamental limitation of TaintDroid is that it can be circumvented through leaks via implicit flows*". Enck et al. [20]

This of course being a sign of malicious behaviour itself and potentially detected via other means such as static analysis. An observed, major obstacle, especially for a real-time monitoring system such as TaintDroid is keeping performance overhead to a minimal and acceptable level. TaintDroid claims"*only 14% performance overhead on a CPU-bound micro-benchmark*" (Enck et al. [20]) and although this is painted in a positive light, would still appear to be quite a footprint when considering that this would be consistently monitoring in the background.

Other possibilities were briefly but critically analysed by Felt et al. [18]. Mandatory Access Control (MAC) systems propose a hierarchy of integrity and confidentiality levels where the flow of data between different levels is restricted (Felt et al. [18]). Although, sound in theory, the Android framework is too complex and applications would transcend various levels simultaneously thus resulting in a confusion of the policies and a possible deadlock between the requester and deputy.

Stack Inspection has the advantage of being able monitor "*confused-deputy*" attacks during run-time but also comes with several limitations. These are specified by Felt et al. [18] as the Stack Inspection being "*dependent on the runtime for correctness*" and having to "*be re-implemented repeatedly for a system with multiple types of runtimes*". History-Based Access Control is similar in operation, relying on runtime mechanisms, and "*reduces the permissions of trusted code after any interaction with untrusted code*" (Felt et al. [18]).

It is obvious that Android and its associated marketplace and applications have reached a level of ubiquitous that has attracted, and will continue to attract, unscrupulous individuals and activities. The explosion and uptake of this technology has left security behind and appropriate measures and standards need to be put in place to prevent the inevitable scamming and malicious attacks. This issue needs to be addressed sooner rather than later. Due to the complex and mutually untrusting relationship between the Android OS and its third-party applications, this solution will need a multi-faceted approach and poses some challenging obstacles. The end result will likely need to be a collation of a number of proposed solutions, possibly the combination run-time monitoring, static analysis and application certification.

2.3 The Wireless Access Points Case Study

This final case study describes the Wi-Fi Air, Sea and land Survey of Hong Kong SAR. It presents the rationale and the main findings of a quantitative study of a longitudinal Wi-Fi security survey in Hong Kong, China (Fig. 2). The authors have been conducting one of the world's most comprehensive longitudinal Wi-Fi surveys in HK since 2002 (Tsang et al. [21]; Tsang and Eustace [22]). This survey has looked at the different ways that one can visualise Wi-Fi Access Point data, to see how it is distributed within a city or an area and to draw conclusions about its use by means of mapping. Air and Sea data was collected by helicopter using Laptops and a Raspberry Pi. This was complemented by land survey data collected by foot (Android smartphone) and by car (war driving).

Fig. 2. 2015 Wi-Fi Helicopter Survey Meeting Place: Clipper Room, Peninsula Hotel, Kowloon 1:45 pm 12 April 2015. The helicopter and team of researchers and students involved in the 2015 Wi-Fi Survey in Hong Kong.

The findings will be of interest to security experts and ICT educators in general. In 2015, we are already seeing the move from IEEE 802.11n on both the 2.4 GHz and the optional 5 GHz bands to the enhanced IEEE 802.11ac (5G Wi-Fi), making the 2015 Wi-Fi Air, Sea and Land survey valuable as a baseline study for the growth in use of 5G Wi-Fi. The industry partners in this project included HK Technology Exchange Limited, HK Technology Association, Web Consortium Education Foundation & Heliservices Limited and the objectives of the 2015 Wi-Fi Air, Sea and Land Survey of Hong Kong are shown in Table 2.

Table 2. Wi-Fi air, sea and land survey of Hong Kong objectives

	Wi-Fi air, sea and land survey of Hong Kong objectives
1	*Setting a new world record for a Wi-Fi survey*
2	*Surveying the latest trend in usage and security awareness in Wi-Fi and wireless communications deployment*
3	*Connecting with business sponsors on the use of secure Wi-Fi access points*
4	*Provide hands-on authentic work experience conducting a Wi-Fi Survey*
5	*Formulating practical educational values from a Wi-Fi experiment*
6	*Providing leadership in wireless LAN survey methodology*

A variety of network technologies such as Bluetooth, Wi-Fi, WiMAX, ZigBee, NFC and RFID technologies exist in Hong Kong, providing some unique cyber security risks and issues concerning the communications medium. Data transfer occurs via radio waves and so the wireless environment has several points of attack including the wireless device, the access point (AP) and the transmission medium for the radio waves.

Fong and Wong [23] conducted a questionnaire survey of 207 participants on Hong Kong Wi-Fi Adoption and Security in 2014 and suggested that while Wi-Fi is easy and convenient for, knowledge gaps exist in using and setting up a Wi-Fi service. Vulnerabilities will exist if the network is not secured. With the on-going development of the "Internet of Things", Wi-Fi cyber security measures and education programs should be expanded along with the more secure use of location-based NFC mobile payment services.

Raspberry Pi and Android Phone performance was better on the land survey than the use of two configurations of laptops with high gain antenna in the air and sea survey by helicopter. Table 3 the results that display some concern for the human factors about security and privacy issues. Such vulnerability monitoring surveys must be held on a regular basis.

Table 3. The main results from the 2015 Wi-Fi air, sea and land data collection environment in Hong Kong

Air and sea survey	Land survey
3 data sets (air) and 4 data sets (sea) of SSIDs were combined with duplicated MAC addresses filtered out *7133 Access Points (AP) detected* *90% of APs used 802.11n* *70% of APs used WPA/WPA2 security and 30% were open (a privacy concern in restaurants and hotels without guest authentication)* *3G/4G APs from mobile devices (Wi-Fi modem or mobile phone hotspots were discovered (MACs and SSIDs)*	Data sets were combined from Raspberry P1 2 and Android smart phone with WiGLE to collect the SSIDs 31 350 APs collected 97% used 802.11;n 75% of APs used WPA/WPA2 security and 25% were open and several open SSIDs were from wireless printers (another privacy concern) 2% of SSIDs came from mobile phone Wi-Fi hotspots. Xiaomi, Samsung and Apple were most common brands

If we use 2014 as a reference year prior to the Wi-Fi survey, Symantec [24] observed that that the cyber security threats in 2014 of concern included that 17% of all Android apps (nearly one million total) in 2014 were actually malware in disguise and that 70% of social media scams were manually shared. The number of data breaches increased 23% in 2014 E-crime and malware via ransom-ware attacks grew 113% in 2014. Table 4 below shows the relative black market value of stolen access or information in 2014 as trading continues on the dark Web and elsewhere.

Table 4. Value of information sold on black market (Symantec [24])

Item	2014 value $US
1,000 Stolen Email Addresses	$0.50 to $10
Credit Card Details	$0.50 to $20
Scans of Real Passports	$1 to $2
Stolen Gaming Accounts	$10 to $15
Custom Malware	$12 to $3500
1,000 Social Network Followers	$2 to $12
Stolen Cloud Accounts)	$7 to $8
1 Million Verified Email Spam Mail-outs	$70 to $150
Registered and Activated Russian Mobile SIM Card	$100

Knowing how cybercriminals are threatening security is the first step to securing information. From social media vulnerabilities to acts of digital extortion, it can be suggested there are threats that now extend the existing wireless network threats. Up to five types of attackers have been identified in network security:

1. the *script kiddy* (an unskilled individual who uses scripts developed by others),
2. the *knowledgeable enthusiast*,
3. the *criminal hacker*,
4. the *idealist* with a cause,
5. black-budget *funded sovereign-state employee*.

The latter attacked is the most dangerous and is usually drawn from the ranks of the previous four types. This attacker has the best motivation, the time, and the funds to breach any system via the internet and others systems with brief physical access using USB ports or other ports such as thunderbolt with purpose-built devices. Any system that attaches to the internet or is taken outside is vulnerable. Even charging a laptop at an airport charging station gives an access point for an attack.

3 Conclusion and Discussion

The human factors associated with public safety and security were explored using three case studies from the Asia Pacific region to describe issues surrounding vulnerable people, application security and the ease and use of urban wireless access points.

The *Vulnerable People, Android Application Security and Wireless Access Points* case studies bring forth the human factors at work and some essential cyber security strategies and behaviours as well as a need to know and understand the adjacent wireless environment by all Internet users and organisations. The development of a Personal Cyber Security Model of Self-awareness and Intervention that represents a 'white box' full of cyber security strategies and behaviours, requires localised community education and training programs that will build trust and respect for others, especially in using wearable devices, health monitors and in all social media and online gaming communities. Such heightened and maintained awareness will help users to have discretion when sharing data and stop data leakage from poor user behaviour.

However these essential human efforts must also couple tightly with the evolving cyber security strategies and policies. The Android application security wireless access points studies showed the need for sandboxing, run-time monitoring, static analysis and application certification and at the same time a needed to know and secure local Wi-Fi access points. The use of encryption and firewalls, password and query statement strength as well as biometrics and other procedures for verification and authentication is only the start. The volume and change of threat tactics as network traffic increase via the Internet of Things, will also lead to an increased diversity of devices and applications, sensors and devices using the ZigBee protocol networks, requiring security measures beyond the Wi-Fi protocol networks as well as malware, denial of service attacks, phishing awareness and detection.

Cybercriminals are targeting social networks, smart devices and mobile applications via wireless networks. Smartphones, tablets and even television sets - all need stronger defences or authentication processes for the control of remote access and connections to cloud services. Turban et al. [2] suggested that some minimum security defences for mobile devices already exist (Table 5).

Table 5. Cyber security minimum defences for mobile phones and devices.

Minimum Defences
Authentication Voice and fingerprint biometric integrated with the operating system/device interface.
Malware Detection. Constant rogue App monitoring at the main App stores to detect and destroy malicious apps.
Loss or theft of Device. Mobile *kill switch* or remote erase capability option is available from smartphone platform

The way that social network sites grow into large social systems has increased, so too has the need for more personal action in regard to personal cyber security. Starting with improved and context-based individual and group behaviours, the personal cyber security model is one way that the individuals may safeguard and standardise their own cyber security safety and goes beyond being a strategy model to scaffolding user behaviour. As information warfare and e-crime around the world is increasing along with large scale surveillance measures to stop the leakage of intellectual property and classified information, each individual must also install, upgrade of refine their own cyber security at the personal level.

The authors propose that the role of Detection in cyber security can act as the guardian for the individual, by adapting a simplified cyber-security personal cyber security model [11] then each individual can follow a sequence of context-based behaviours to safeguard their data and identity. Safety in numbers is a true idiom when an individual connects online then each node in the social network is also affected and the security levels ‘cascade’ as the community grows and helps to lower the paranoia and concern of also being part of a larger social system. Fellows et al. [25] discussed how ICT security techniques and human factors can work together to lower any concern that the individual may have in being an active member of a large social system. They suggested that the personal cyber security model will work best if the importance of human factors is considered alongside with the technical ICT security strategies.

The use of encryption of data and SSID passwords are among the technical measures, that must be accompanied by the use of discretion by the individual when sharing information and stopping any leakage, as other members of the social network may share with other networks who would not be part of the original intention to share. Recent problems that famous people have had with photos being circulated on Facebook and Twitter are examples of this kind of low level leakage, while Snowden’s leakage of leaking classified documents about U.S. surveillance programs (Shoichet [26]) has shown the consequences when the lack of trust, respect and affection as human factors, is scaled upwards and spread rapidly out of control over social networks.

In conclusion, they suggested that the human factors surrounding the core issues of the leakage problem go beyond all cyber security strategies in place and just boil down to how well each individual is treated by others, by both ‘near’ and ‘far’ neighbours in their multiple social systems. Tsikerdekis and Zeadally [5] suggested that deception prevention begin with harder and standardized user registration and credential verification. Such steps are needed when using social media and safeguarding personal data with wearable devices in a whole-body networking future. By including the human factors and user behaviour in developing cyber security policy and practices with the latest technology for authentication such mobile biometrics, remote erase capability and encryption, can we hope to prevent compromise and deception.

Just as weak security on smart devices are a serious threat, either from intentional or unintentional attacks, so too are people without a high level of self-awareness and intervention control over the security of their near and far network connections. Understanding the human factors and development of human capital play an important role here, requiring mandatory measures for all. Warren [27] stated that human capital is related to situational awareness and incident reaction (personal cyber security model) but also requires development of culture over time as well as the retention and recruitment process of human resources.

In addition, a set of core security technology skills are required. According to Warren [27], the Australian Signals Directorate (ASD) stated that 85% of all targeted attacks could be prevented by four simple mitigation strategies:

1. Application Whitelisting (selected applications, DLLs only)
2. Patching Application updates
3. Patching Operating System updates
4. Restricting Admin privileges based on user duties.

As the network of nodes (device and social networks) for each individual grows into a complex system, giant components will form as the hub or router with access to attributes or properties of other connected nodes (edges). Each node then contributes to develop their own cyber maturity from the bottom up all the way to build and measure the national cyber maturity, as described by Feakin et al. [28] for cyber security policy the Asia-Pacific Region.

Each device node or human actor must have its security attributes at the top level. So then network security of such a complex system may be simplified as a whole and have better external protection if each of its nodes and actors have high levels of internal security, self-awareness and intervention mechanisms - whether a smart sensor or a human. We need to focus on personal cyber security/identity awareness and behaviours for each citizen. These personal cyber security/identity awareness and behaviours then operate at the individual level each time that a smart citizen interacts via more access points to a myriad of IoT small devices such as Raspberry Pi and Arduino sensor projects come online.

A *Personal Cyber Security Model of Self-awareness and Intervention* can improve defences from the constant security threats by building a mesh of individual secure nodes (human actors), each armed with improved security behaviours and strategies that cluster and build secure networks (defensive nodes) against all cybercrime. Such actionable wisdom is a creative process, providing an operational frame of reference to the cyber security agenda.

Cyber security researchers everywhere also need a special ‘dark ethics’ policy to defend against making an error of logic, in giving any rights that are forfeited by the behaviour of the cracker or cybercriminal.

References

1. Australia’s Cyber Security Strategy. https://cybersecuritystrategy.dpmc.gov.au/assets/img/PMC-Cyber-Strategy.pdf
2. Turban, E., Volonino, L., Wood, G.R.: Information Technology for Management: Digital Strategies for Insight, Action and Sustainable Performance. Wiley, Hoboken (2015)
3. Prieto, R.: Cisco VNI Predicts Near-Tripling of IP Traffic. https://newsroom.cisco.com/press-release-content?type=webcontent&articleId=1771211
4. Wilkinson, M.: Cyber Security Challenges facing Australia in the Asian-Pacific Region. http://thinkspace.csu.edu.au/itc571securitychallengesinapac/
5. Tsikerdekis, M., Zeadally, S.: Online deception in social media. Commun. ACM **57**(9), 72–80 (2014)
6. Eustace, K., Burmeister, O.: Ethics and governance of ICT-based social engagement in institutional aged care. In: Seventh Australian Institute of Computer Ethics Conference (AiCE) (2013)
7. Harvie, G., Burmeister, O., Eustace, K.: Bringing the oldest-old into the digital age: overcoming challenges of mobility, literacy and learning. In: 13th National Conference of Emerging Researchers in Ageing. http://www.era.edu.au/ERA+2014
8. Harvie, G., Eustace, K., Burmeister, O.K.: Assistive technology devices for the oldest-old: maintaining independence for the fourth age. In: Kreps, D., Fletcher, G., Griffiths, M. (eds.) HCC 2016. IAICT, vol. 474, pp. 25–33. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44805-3_3
9. Phahlamohlaka, J.: Defence, peace, safety, security and information warfare research, In: SCM Seminar Series. Charles Sturt University, Wagga Wagga, NSW, Australia (2012)

10. Connolly, C., Maurushat, A., Vaile, D., van Dijk, P.: An overview of international cyber-security awareness raising and educational initiatives. In: International Cyber-Security Awareness, Sydney (2011). Site: acma.gov.au
11. Stallings, W., Brown, L.: Computer Security: Principles and Practice. Pearson, Upper Saddle River (2012)
12. De Bruijn, H., Janssen, M.: Building cybersecurity awareness: the need for evidence-based framing strategies. Gov. Inf. Q. **34**(1), 1–7 (2017)
13. Stammberger, K.: Mobile & Smart Device Security Survey 2010: Concern Grows as Vulnerable Devices Proliferate, Smartphones are the Tip of the Iceberg. Mocana Corporation (2010)
14. U.S. Department of Homeland Security: Technical Information Paper-TIP-10-105-01 Cyber Threats to Mobile Devices, Washington (2010)
15. Nickinson, P.: Android market now has more than a quarter-million applications (2011). http://www.androidcentral.com/android-market-now-has-more-quarter-million-applications
16. Enck, W., Octeau, D., McDaniel, P., Chaudhuri, S.: A study of Android application security. In: 20th USENIX Security Symposium (2011)
17. Vidas, T., Votipka, D., Christin, N.: All your droids are belong to us: a survey of current android attacks. In: Proceedings of the 5th USENIX Conference on Offensive Technologies. USENIX Association, Berkeley (2011)
18. Felt, A.P., Wang, H.J., Moschuk, A., Hanna, S., Chin, E.: Permission re-delegation: attacks and defenses. In: 20th USENIX Security Symposium (2011)
19. Felt, A.P., Ha, E., Egelman, S., Haney, A., Chin, E., Wagner, D.: Android permissions: user attention, comprehension, and behaviour, Electrical Engineering and Computer Sciences, University of California at Berkeley (2012)
20. Enck, W., Gilbert, P., Chun, B., Cox, L., Jung, J., McDaniel, P.: TaintDroid: an information-flow tracking system for realtime privacy. In: 9th USENIX Symposium on Operating Systems Design and Implementation (2010)
21. Tsang, P., Kwok, P., Kwong, R., White, B., Fox, R.: Innovation in ICT teaching: a longitudinal case study of Wi-Fi in Hong Kong. Int. J. Innov. Learn. **10**(1), 85–101 (2011)
22. Tsang, P., Eustace, K.: Educational and social implications from a longitudinal Wi-Fi security study (2002–2014). In: Proceedings of the International Conference on Information Technologies, InfoTech 2014, Bulgaria (2014)
23. Fong, K., Wong, S.: Hong Kong Wi-Fi adoption and security survey 2014. Comput. Inf. Sci. **8**(1) (2015)
24. Symantec: 2015 Internet Security Threat Report. https://www4.symantec.com/mktginfo/whitepaper/ISTR/21347932_GA-internet-security-threat-report-volume-20-2015-social_v2.pdf
25. Fellows, G., McAfee, M., Eustace, K.: The role of human factors in the ICT security of large social systems, Las Vegas USA (2013, Unpublished paper)
26. Shoichet, C.E.: Is Snowden worth the risk? Latin America weighs pros and cons. http://www.cnn.com/2013/07/11/world/americas/latin-america-snowden-asylum
27. Warren, M.: Keynote Address, ICT Higher Degree Research Symposium 2016. Charles Sturt University. https://www.asd.gov.au/infosec/mitigationstrategies.htm
28. Feakin, T., Woodall, J., Nevill, L.: Cyber maturity in the Asia-Pacific Region 2015. ASPI. https://www.aspi.org.au/publications/cyber-maturity-in-the-asia-pacific-region-2015/Cyber-Maturity-2015.pdf

Hiding Fast Flux Botnet in Plain Email Sight

Zhi Wang, Meilin Qin, Mengqi Chen, and Chunfu Jia(✉)

Nankai University, Tianjin, China
cfjia@nankai.edu.cn

Abstract. Fast flux and domain flux are widely used as evading techniques to conceal botnet C&C server. But nowadays, more and more machine learning schemes are introduced to recognize and detect fluxing botnet automatically and effectively. In this paper, we propose a novel fluxing scheme to hide C&C server in plain email sight. Email flux tries to blend in with normal email communication. With the excellent reputation of email servers, the malicious activity is more likely to get lost in the normal email crowd. Therefore, DNS-based botnet detection schemes are difficult to detect the email flux botnet. Comparing to the cost of registering a public IP address or a domain, the cost of registering an email account is much less, and email account reveals less geolocation information. And we introduce asymmetric encryption strategy to fortify DGA, preventing adversaries from taking down the botnet by registering email account before bot master. We also discuss possible countermeasures in the future to mitigate email flux.

Keywords: Fast flux · Domain flux · Botnet
Command and control channel · Evasion technique

1 Introduction

Botnet is a network of compromised computers, known as bots or zombies, that could be remotely controlled by an attacker in the Internet, so-called botmaster. Currently, botnets are the main platform for attackers to carry out large scale cyber crimes, such as sending spams, phishing, launching distributed denial of service (DDoS) attacks. According to Symantec report, in 2016 there are 98.6 million hosts controlled in botnets, which is an increase of 6.7 million over last year [1]. There were at least 255,065 unique phishing attacks worldwide, which represents an increase of over 10% from the 230,280 attacks identified in 2015 [2]. The number of DDoS attacks per day ranged from 131 to 904 in the second quarter in 2017 [3]. Hence, botnet is one of the most significant threats to the Internet.

To build a complete botnet, a stealthy command and control (C&C) channel must be built between the botmaster and bots, through which the botmaster can send commands to all bots. Due to host-based detection such as reverse

X. Lin et al. (Eds.): SecureComm 2017 Workshops, LNICST 239, pp. 182–197, 2018.
https://doi.org/10.1007/978-3-319-78816-6_14

engineering is hard, most defenders focus on detecting the C&C channel, trying to cut off the communication and shut down the botnet. Therefore, the botmaster will make every effort to conceal the C&C channel to decrease the risk of detection. For example, [4,5] exploit social network to construct botnet, [6] uses email protocol as C&C channel and [7] hides the commands in SMS message.

To make the C&C channels more stealthy, there are many evading techniques such as fast flux and domain flux. With fast flux, the bots would query a certain domain that is mapped onto a set of IP addresses that change frequently [8]. However, fast flux uses only one single domain name, which will lead to a single point of failure. In domain flux, the botmaster associates one or more IP with several domains to avoid being easily blocked by blacklisting.

Although the fast flux and domain flux techniques can hide botnet C&C server behind a set of IP addresses or randomly generated domain names, the defenders can also identify the botnets through DNS traffic analysis. The fast flux and domain flux rely on DNS service, and there are some significant difference between fluxing botnet DNS traffic and normal DNS traffic, such as the number of different IP addresses resolved from the same domain, the length of each packet and so on. Many machine learning models are trained to recognize suspicious fluxing DNS communication automatically. Email is not a kind of IP-based C&C delivery, thus the email sent by bot will have similar features in DNS traffic with normal users. We propose an email flux method that can bypass the existing machine learning detection techniques against fast flux and domain flux.

We summarize the contributions of this paper as follows:

– We present the email flux, which is a different from fast flux and domain flux. It applies randomly generated email addresses to establish the fluxing C&C channel so that email flux could evade traditional machine learning methods using DNS traffic analysis.
– We enhance the traditional domain generation algorithm(DGA) used in domain flux, preventing adversaries from controlling the automatically generated email accounts in advance.
– We analyze the traffic, cost and reputation of email flux and prove its availability and concealment.
– We discuss the possible countermeasures in the future for mitigating email flux and other possible new fluxing channels.

2 Related Works

Botnet is a network of compromised computers, known as bots or zombies, that could be instructed by a controller in the Internet, so-called botmaster. Botnets can be used to perform DDoS attack, steal data and send spam. In the face of potential attacks, the Intrusion Detection System (IDS) is an important defensive mechanism to defend against these possible attacks.

The common types of IDS techniques include: signature based detection, anomaly detection, artificial neural network (ANN) based IDS and fuzzy logic based IDS [9]. The signature based detection can detect the malicious traffic by using a set of rules and known signature attack stored in a knowledge database [10]. However, the disadvantage is that it could not detect unknown new attacks. The anomaly detection could detect abnormal system behavior and malicious traffic, which needs to be specified a baseline by the security researcher. The ANN based IDS detection utilizes ANN as a pattern recognition technique. The fuzzy logic based on rule can detect the intrusion behavior of the traffic [9].

The core of the botnet is its C&C, many attackers use fast flux and domain flux methods to hide their C&C channels [11,12]. By exploiting fast flux technique, the botmaster hides the real IP addresses that belong to his C&C servers. Each bot can use the same domain name to connect with C&C servers, while the IP addresses resolved are constantly changing.

There are many approaches to detect fast flux such as active or passive DNS traffic monitoring. [13] uses a combination of passive DNS monitoring and active DNS probing to detect botnets, which based on a cluster analysis of the features obtained from the payload of DNS-messages and uses active probing analysis. [14] is based on large-scale passive analysis of DNS traffic generated by hundreds of local recursive DNS (RDNS) servers located in different networks and scattered across several different geographical locations, to detect and track malicious flux networks. They clarify four characteristics of flux domain names: (1) short TTL; (2) high frequency of change of the set of resolved IPs returned at each query; (3) large overall set of resolved IPs acquired by querying the same domain; (4) the resolved IPs are scattered across many different networks. Then they utilize these features to filter flux domains. Even though the fast flux seems to be a fine evading technique, it has a single-point-of-failure problem. If a security researcher discovers a botnet's domain name, he will blacklist this domain name and block the communication of botnet.

To avoid this issue, attackers utilize domain flux method to hide C&C servers of botnet such as Tropig [8] and Conficker [15]. By using domain flux technique, the botmasters can frequently change domain names mapped to a single IP address. Each bot generate a list of domain names by running the same DGA and then tries to connect to the domain names in the list until the success of finding C&C servers. Generally, the inputs (or seeds) of the DGA are the current date information and some numeric parameters. Unlike fast flux, domain flux is more resilient to avoiding take-down attempts. More specifically, even if the current domain is blocked, the botmaster only need to register the next domain to control his botnet again.

Because the domain names change frequently, blacklisting domain name is not effective. In order to detect domain flux, many approaches have been proposed. [16] uses DNS query data and analyzes the network and zone features of domains to build a dynamic reputation system. [17] monitors DNS traffic and presents 15 behavioral features used in the identification of malicious domains. [18] detects malware-related domains based on DNS resolution patterns by monitoring DNS traffic from the upper DNS hierarchy. [19] preserves the privacy of the users of the network and only uses the DNS replies to detect domain flux.

[20] proposes a combination of clustering and classification algorithm that relies on the similarity in characteristic distribution of domain names to remove noise and group similar domains. However, it can only detect centralized botnet, not P2P botnet.

The biggest difference between email flux and traditional flux is that email flux uses email as C&C channel. In order to make the botnet hard to be shut down, the fast flux and domain flux botnets sacrifice their concealment for robustness. Whether it is fast flux or domain flux will generates anomaly DNS traffic, and the existing detection methods for fast flux and domain flux can be simply summarized by monitoring DNS traffic. In contrast, email flux do not rely on IP-based C&C delivery. Although the emails sent by bot have a few differences to normal email communication, it is almost the same from the point of DNS monitoring. Thus the existing DNS-based detection method cannot catch the email flux.

There are literatures that select other channels for C&C communication. [21] utilizes the URL shortening service. The botmaster hides the IP address of C&C server into URLs, and change URL into automatically generated alias. However, it is still an IP-based C&C delivery. Visiting websites that do not exist will result in Name Error DNS responses, which is suspicious and has been the target for many detection methods like [22]. Besides, the algorithm that generates alias is similar to DGA, so that the whole botnet may be taken over by the defenders through pre-calculating and registering the alias [8]. We imporved the DGA and use push mechanism to send commands directly to each bot in case of being controlled by the defenders.

[7] selects SMS as C&C channel. Due to SMS message has to be sent by one phone number which is in use by the owner of the compromised phone, it is easy to be aware of. Email account does not combined with the compromised host, and the botmaster can register and allocate one email account to each bot. [6] firstly presents the feasibility of email-based botnet, we have made some development on its basis. [6] does not present a complete botnet, it just proves the feasibility that one bot can execute the command embed in email. It demonstrates the difficulty for the defenders to crack the encoded commands from the point of view of cryptography, however, the defenders can block the suspicious email account without knowing the content. We propose a more specific and practical email-based botnet, and introduce flux method to improve the robustness and resilience of botnet.

3 Designing of Email Flux

We propose an email flux method to hide C&C channels. Our design derives from traditional domain flux, but there are numerous differences between email flux and traditional flux method. First of all, the C&C channel is different. Domain flux is IP-based, thus a bot will generate anomaly DNS traffic, while email flux is similar to normal email communication. Second, in traditional domain flux, bots request commands from the generated domains, which is called pull mode.

In our work, the commands are sent directly from botmaster to bot email account, which is called push mode. Third, the email that embedded with commands is stored in email server. Standing in the DNS perspective, the communication is set between the compromised host and an email server, whose reputation is much higher than the domain used in domain flux.

The process of a C&C communication is simple but efficient. The botmaster embeds commands in emails with the asymmetric encryption algorithm and sends the private key to bots directly. A bot extracts the command by its decryption key and responds to it. The response will be sent to email accounts which are automatically generated and physically controlled by botmaster. Due to the bots just send their response to those email accounts rather than get commands from here, the botnet is impossible to be controlled by the defenders.

The email flux botnet uses a email addresses generation algorithm that generates a set of random email addresses composed of alphabet letters and digits. The inputs of the algorithm will be the current date information and a customized string. That is to say, there are 2 parameters which can determine the output email address. The date will be automatically changed while the string is determined by botmaster. The botmaster can change the generated email address list at any time through sending a new customized string to bots. For each round, such as a day, week, or the month, the email flux botnet generates k (e.g., 1000) different email addresses through the algorithm.

The botmaster and each bot have to share an email addresses generation algorithm, therefore, the botmaster and each bot will independently generate the same lists of email addresses periodically. The botmaster can also ask bots to change the customized string in order to get a new list of addresses. The botmaster periodically registers certain email addresses in the list in advance. Then, the bot contacts email addresses in the list in order until one succeeds–the botmaster will reply the response to show its validation.

Email flux can be simply classified into two stages: registering the email addresses to email providers and connecting to these email addresses via email. Essentially, email flux refers to periodically changing and registering the email addresses to bypass detection and blacklisting. There are many high reputation email providers we can choose, such as gmail, outlook, yahoo, and so on. Hence, in order to improve the resilience of email flux against take-down attempts, we can frequently change the email service provider.

3.1 Registration Stage

The botmaster has to register the email accounts at first and then check the response sent by bots. The botmaster needs to execute an email addresses generation algorithm to obtain k email addresses. The input of the algorithm will be the current date and a customized string. After acquiring a list of email addresses, the botmaster will select several addresses on the top of the list and register them. In general, at the beginning of each day's communication, the botmaster can only receive message in the first email account on the list.

However, as Fig. 1 shows, this address may be blocked by defenders because of a large number of suspicious email communication.

Here, we take registering outlook mailbox as an example to explain the registration process in detail. Step one, log in to www.outlook.com. Step two, fill in contact information and user information. Step three, enter an email address generated by the algorithm. If this email address has been registered, the system will prompt you to re-enter a new one. Step four, input validation information, to confirm that the account created is a real person. Other mailboxes may need to be verified by SMS.

Finally, the botmaster will receive a notification from the mail service provider if the registration is successful. Then, newly registered email address is available for email flux.

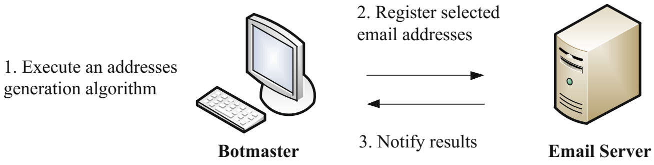

1. Execute an addresses generation algorithm
2. Register selected email addresses
3. Notify results
Botmaster
Email Server

Fig. 1. Email flux registration

3.2 Connection Stage

Each bot will periodically try to connect with email addresses to send message, as shown in Fig. 2. First, each bot independently executes an email addresses generation algorithm to get k email addresses. The input to the algorithm is also the current date a customized string. That is to say, the inputs of bot and botmaster are exactly the same, to make sure the lists of email addresses are the same as what botmaster generates.

Next, the bot attempts to send message to the email addresses in the list in order until one succeeds. These email accounts play the role of C&C servers because each bot will contact them. After sending its response, the bot will receive two kinds of response: a confirmed message and undelivered message. When the bot receives a confirmed message, it indicates that bot has successfully connected to the C&C server. Due to the botmaster physically controls the C&C servers, he can send email to bots to indicate he has received its response. The confirmed messages are also be encrypted.

When the bot receive a undelivered message, it means the bot attempted to contact an email address that had not yet been registered by the botmaster or had been blocked by the email provider. In this case, the bot needs to connect to the next email address in the list. If these email addresses all failed, bot will contact the email addresses hard-coded in its configuration file.

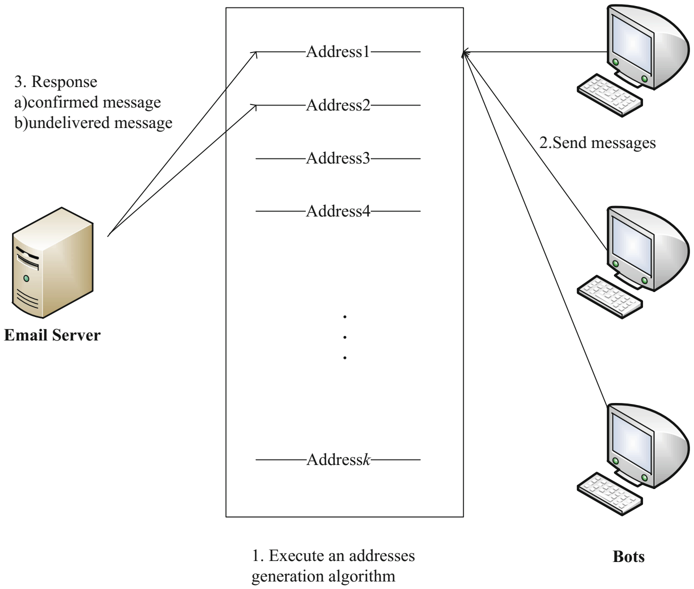


Fig. 2. Email flux connection

3.3 Improvement of Algorithm

There is a weakness in traditional domain flux that use DGA to generate domain names. Once the defenders know the DGA algorithm through reverse engineering, they can forecast and pre-register the next round of domain lists. [8] shows how the defenders take control of a botnet through forecast and pre-register the automatically generated domains. In order to make sure the communication can be carried out successfully, in our work, the botmaster should register a number of email accounts in advance and then pass the customized string he uses to bots. Since the customized string is one of the parameters that determine the generated email addresses, in this case, even though the defenders caputre the bot and know the email address generation algorithm, they still cannot preempt the email address on the top of the list. Those email accounts pre-registered by botmaster should belong to different email operators so that it is impossible to block them all in a short time. That gives the botmaster enough time to judge whether the defenders know the email address list and deal with it.

4 Analysis

In this section, we use quantified data to analyze the feature, or advantage, of email flux in detail. We set each list generated by bot with k=1000 email addresses. The botmaster registers top 5 email addresses in the list. The email service providers in the experiment we select are Gmail, Outlook, Sina, Foxmail and 163 email. First, we describe the traffic of email flux, and then we discuss the reputation of email. Finally, we evaluate the costs of botmaster managing a email flux botnet.

4.1 Traffic

In order to combat spam, email providers limit the amount of mails that each user can send. These limits restrict the number of messages sent per day and the number of recipients per message. After a user reaching the limits, he can't send new messages for up to 24 hours. However, they can still receive incoming email. As shown in Table 1, there are some major limitations set by popular email providers.

Table 1. Daily sending limit

Email provider	Message sent per day
Gmail	500
Outlook	100
163	50
Foxmail	50
Sina	30

From the Table 1, we draw a conclusion that the number of messages sent by each bot per day should not exceed the minimum 30. Thus, we set the number of messages to n=20 in our experiment. We assume the botnet consists of 5,000 bots, thus the total volume of emails per day is T= 100,000.

As shown in Fig. 3, we get the total number of global emails in June 2017 from https://talosintelligence.com. There are totally 59,209 available email server in the world. Thus, each email server will receive an average of more than 6 million emails every day. Supposing that all the bots use the same email server, the percentage of malicious email is only about 1.6%. If the defender use traditional detection method to locate bot members, they can only find the email server as the C&C server of the botnet because each bot member sends message to the mail server. If the defender blocks the detected email server, 98.4% innocent user will be implicated. However, the bot will use different email service providers and servers for communication. The effect of shutting down the email server will be even worse. Thus, email flux is feasible, and traditional detection and blocking method is useless for our email flux botnet.

4.2 Reputation

Reputation is an important factor in botnet detection. There are certain detection methods based on the reputation of domain [16,23,24]. Each domain has a

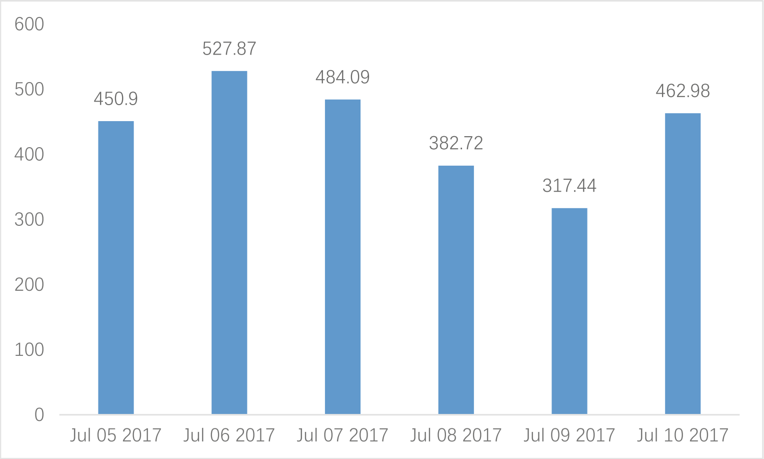


Fig. 3. Total number of global emails (billions)

reputation score upon registration. We select email as C&C channel, the email account itself does not have reputation based on the traditional reputation computing algorithm. However, the email provider does. Table 2 is the global rank of major email providers we get from Alexa. It shows that the reputation of the email providers that we use in the email flux botnet is extremely high. Thus they are not easy to cause suspicion, which indicates the superiority of email flux compared with other domain flux.

Table 2. Alexa rank of email providers

Site	Global rank	Rank in country (CN)
163.com	375	64
Sina.com	4,568	375
Outlook.com	5,014	9,088
Gmail.com	11,564	15,985
Foxmail.com	32,936	3,089

4.3 Costs

To compare with the costs of domain flux, we collect the price of registering a domain as follows:

As shown in Table 3, the money spend on registering domains are very high. Besides, the process of registering a new domain is cumbersome. The registrant has to fill in their personal information such as real name, phone number and identify card number. After submission, the relevant department will take a phone call for verification. Thus, domain registration not only cost a lot but also hard to fake. Traditional domain flux needs numerous of new domains every day, which may bring a heavy burden for botmaster.

On the contrary, the email registration is far more convenient. First of all, there is no fee for email registration. Then, the email account does not bind with personal information except for phone number, and there is no verification from email providers. Due to it is easy to register an email address, the price of buying a email account is low. Thus, the botmaster can acquire email accounts by manual registration on his own, automatically registration through certain program or buying online.

5 Potential Countermeasures

In this paper, we classify detection of email flux botnets into three types: hosts, DNS traffic and email providers. At hosts, security researchers attempt to detect and analyze malware by monitoring system statues. However, malware can use complex and advanced technology to conceal itself and increase the difficulties of analysis. Network monitors usually monitor DNS traffic to detect botnet. Since email services are very popular and have heavy usage volume, it is unlikely to be noticed. Besides, all email flux traffic is encrypted by email service providers automatically, and we also enhance the encryption to make it difficult for defenders to investigate it.

5.1 Detection in Hosts

If security researchers can detect malware for botnets on hosts, then they will know email addresses generation algorithms or addresses lists through a series of analyzing. The security researchers may distribute these email addresses to email provider for blacklisting. However, as email addresses are just a fraction of large set of email accounts used for actual communicating, blacklisting techniques is

Table 3. The cost of registering domains.

Types	Captions	First year (CNY)	Renewal (CNY)
.com	Global registration volume first	60	78
.net	The most popular domain name	65	78
.cn	The most popular for Chinese people	35	68
.top	To show one's personality	9	34
.cc	Competitive domain name	38	60
.org	Trusted domain name	70	78
.shop	For e-commerce	49	188
.me	For personal use	28	160

ineffective in countering such email fluxing. Reverse engineering of bot executables is a time-consuming process, and during this time the botmaster may send commands to the bot to change the algorithm. Also, there are many evasion techniques make reverse engineering difficult to be implemented. For example, malware authors can utilize emulation technology to obfuscate malware [25]. They also can use code protection tools to protect malicious code.

5.2 Detection in DNS Traffic

Analyzing IP Addresses. Fast flux detection schemes typically analyze the IP address diversity via monitoring DNS traffic. [26] analyzes traffic characteristics and introduces dynamic whitelisting to differentiate between FFSN and CDN. [27] develops a automated identification of fast flux domains by IP address diversity and flux-agent. Email flux is different from fast flux and do not need to frequently change IP addresses. Therefore, email flux can not be detected by analyzing IP address diversity.

Analyzing Group Activities. There are some methods that focus on group activities for DNS requests. [28,29] based on features extracted from groups of domains, which has to consider a problem of how to group these domains. The authors chose *random* groups of domains to overcome this problem. But there is no rigorous way to test and verify the validity of these hypotheses. [30] also considered the history of suspicious domain group activities, at the same time, they still analyzed suspicious failures in DNS traffic. In email flux, the bot do not generate a large of domains. Therefore, the above detection methods based on DNS group activities are invalid for our botnet.

Analyzing Failures Resolutions. Many domains generated by DGA need to be resolved via DNS, but the botmaster usually pre-registers only a small part of domains. Thus it will result in failure resolutions traffic by queries of bots. [22] presented a technique to efficiently analyze streams of unsuccessful domain name resolutions to automatically identify DGA-based botnet by using a combination of clustering and classification algorithms. Such failures domain resolutions also called Non-Existent Domain (NXDomains). [31] utilized the failures around successful DNS queries and the entropy of the domains belonging to such queries, for detecting the botnet. [32] also proposed a light-weight anomaly detection approach based on failed DNS queries, with a novel tool DNS failure graphs. The graphs captures the interactions between hosts and failed domain names. One of methods in [33] is identifying randomly name failed DNS requests. These detection approaches mainly analyze failures resolutions, which is not applicable in our case. Since email flux generates the email addresses instead of domain names, they can not detect it.

Analyzing Individual Domains. [34] presented a DGA classifier to classify individual domains. They used two basic linguistic features named meaningful characters ratio and n-gram normality score to tell DGA and non-DGA-generated domains. [35] also focus on detecting domains on a per-domain basis. They leveraged a random forest classifier to classify single domains. Similarly, because email flux do not generate domain names, such methods also can not detect it.

5.3 Detection in Email Providers

Limiting Registrations. In email service, there must be two email accounts in one communication process. The IP addresses of each host used to access domain is automatically assigned when connecting to the Internet. However, the botmaster have to register an email account first and then use it. Although we have improved the traditional DGA algorithm that do not need numerous new email accounts every day, the botnet still needs a large number of email accounts according to its size. Such a large scale is almost impossible for manual registration, thus the defenders only need to prevent automatic registration.

Nowadays, registering an email account only need to identify a common letter-based CAPTCHA. Obviously, they are insufficient because there have been many ways to crack it [36–38]. There are many improved forms of CAPTCHA and the email providers can update it.

Besides, if an email address is comprise of a series of random letters and is not 'human-pronounceable', it is probably automatically registered for malicious intent. The email provider can set more limitation for them. For example, require them to fill out their detailed personal information.

Restricting Newly Registered Accounts. The email flux botnet requires the botmaster to register several newly generated email addresses and put them into use every day. Generally speaking, the newly registered email account do not receive a lot of emails because few people know its address. Thus, the email providers can increase the security level of newly registered accounts, for example, filter out all the emails whose contents or addresses are comprise of a series of random letters and are not 'human-pronounceable'.

Some email providers, such as QQ, prohibit the newly registered email accounts from using third-party clients to send and receive emails. It is a good idea because the ability of dealing with the communication in a large-scale botnet is beyond the scope of human being, and a small-scale botnet can only result in limited impact.

Broadening the Detection Focus. As email service providers are mostly private enterprises, they need to pay special attention to the privacy of their customers. Thus, the cooperation between the defenders and email service providers are limited and only the email service providers can come into contact with the email content. As shown in Table 4, it is true that the spam is the biggest threat in email field. But in fact, the email service providers only focus on checking if an email is a spam with machine learning algorithms. That is to say, if the botmaster can make the emails embedded with commands different from spam, e.g. the frequency of sending, the format and length, it is highly possible to run the botnet well.

Therefore, the email service provider should broaden their detection on spam detection. For instance, if one email account often sends or receives emails with the same content, it is probably controlled by a member of a botnet. Besides, the cooperation with defenders should be strengthened within the range of privacy protection.

Table 4. The percentage of legitimate emails and spams

Email type	Average volume (billions)	Percentage
Legitimate	63.79	14.45%
Spam	65	85.55%

6 Conclusion

As far as we known, we first proposed the email flux botnet. The bot can automatically generate a list of random email addresses for covert email-based C&C communication. The email flux botnet can obviously bypass traditional DNS-based detection methods against fast flux and domain flux, because email flux exploits the communication with good reputation email servers to build stealthy botnet C&C channel without suspicious DNS traffic. And we enhance the traditional DGA algorithm used in domain flux, preventing adversaries from taking down or taking over botnet by registering C&C email account in advance. We discuss the potential countermeasures in the future to mitigate the threat of email flux botnet.

Acknowledgements. The authors acknowledge the National Natural Science Foundation of China (Grant: 61300242, 61772291), the Tianjin Research Program of Application Foundation and Advanced Technology (Grant: 15JCQNJC41500, 17JCZDJC30500), the Open Project Foundation of Information Security Evaluation Center of Civil Aviation, Civil Aviation University of China (Grant: CAAC-ISECCA-201701, CAAC-ISECCA-201702), Project supported by the National Key Basic Research Program of China (Grant: 2013CB834204).

References

1. Symantec: internet security threat report for 2016 (2017). https://www.symantec.com/zh/cn/security-center/threat-report?inid=globalnav_scflyout_istr
2. APWG: global phishing survey for 2016 (2017). https://apwg.org/apwg-news-center/APWG-News/
3. Kaspersky: DDoS attacks in Q2 2017 (2017). https://securelist.com/ddos-attacks-in-q2-2017/79241/
4. Kartaltepe, E.J., Morales, J.A., Xu, S., Sandhu, R.: Social network-based botnet command-and-control: emerging threats and countermeasures. In: Zhou, J., Yung, M. (eds.) ACNS 2010. LNCS, vol. 6123, pp. 511–528. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13708-2_30
5. Yin, T., Zhang, Y., Li, S.: DR-SNBot: a social network-based botnet with strong destroy-resistance. In: IEEE International Conference on Networking, Architecture, and Storage, pp. 191–199 (2014)
6. Singh, K., Srivastava, A., Giffin, J., Lee, W.: Evaluating email's feasibility for botnet command and control. In: IEEE International Conference on Dependable Systems and Networks with Ftcs and DCC, pp. 376–385. IEEE, Anchorage, June 2008
7. Zeng, Y., Shin, K.G., Hu, X.: Design of SMS commanded-and-controlled and P2P-structured mobile botnets. In: Proceedings of the Fifth ACM Conference on Security and Privacy in Wireless and Mobile Networks, WISEC 2012, pp. 137–148, ACM, New York (2012)
8. Stone-Gross, B., Cova, M., Cavallaro, L., Gilbert, B., Szydlowski, M., Kemmerer, R., Kruegel, C., Vigna, G.: Your botnet is my botnet: analysis of a botnet takeover. In: ACM Conference on Computer and Communications Security, CCS 2009, Chicago, Illinois, USA, , pp. 635–647, November 2009
9. Iqbal, S., Kiah, M.L.M., Dhaghighi, B., Hussain, M., Khan, S., Khan, M.K., Choo, K.-K.R.: On cloud security attacks: a taxonomy and intrusion detection and prevention as a service. J. Netw. Comput. Appl. **74**, 98–120 (2016)
10. Osanaiye, O., Choo, K.-K.R., Dlodlo, M.: Distributed denial of service (DDoS) resilience in cloud: review and conceptual cloud ddos mitigation framework. J. Netw. Comput. Appl. **67**, 147–165 (2016)
11. Ollmann, G.: Botnet communication topologies. Retrieved September, vol. 30, p. 9 (2009)
12. Salusky, W., Danford, R.: Know your enemy: fast-flux service networks. Honeynet Proj., pp. 1–24 (2007)
13. Pomorova, O., Savenko, O., Lysenko, S., Kryshchuk, A., Bobrovnikova, K.: Anti-evasion technique for the botnets detection based on the passive DNS monitoring and active DNS probing. In: Gaj, P., Kwiecień, A., Stera, P. (eds.) CN 2016. CCIS, vol. 608, pp. 83–95. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39207-3_8
14. Perdisci, R., Corona, I., Giacinto, G.: Early detection of malicious flux networks via large-scale passive dns traffic analysis. IEEE Trans. Dependable Secure Comput. **9**, 714–726 (2012)
15. Porras, P., Di, H., Yegneswaran, V.: A foray into conficker's logic and Rendezvous points. In: USENIX Conference on Large-Scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More, p. 7 (2009)

16. Antonakakis, M., Perdisci, R., Dagon, D., Lee, W., Feamster, N.: Building a dynamic reputation system for DNS. In: Proceedings of the 19th USENIX Conference on Security, USENIX Security 2010, p. 18. USENIX Association, Berkeley (2010)
17. Bilge, L., Kirda, E., Kruegel, C., Balduzzi, M.: Exposure: Finding malicious domains using passive dns analysis. In: Network and Distributed System Security Symposium, NDSS 2011, San Diego, California, USA, February 2011
18. Antonakakis, M., Perdisci, R., Lee, W., Nikolaos Vasiloglou, I., Dagon, D.: Detecting malware domains at the upper DNS hierarchy. In: USENIX Conference on Security, p. 27 (2011)
19. Guerid, H., Mittig, K., Serhrouchni, A.: Privacy-preserving domain-flux botnet detection in a large scale network. In: International Conference on Communication Systems and Networks, pp. 1–9 (2013)
20. Nguyen, T.-D., CAO, T.-D., Nguyen, L.-G.: DGA botnet detection using collaborative filtering and density-based clustering. In: Proceedings of the Sixth International Symposium on Information and Communication Technology, SoICT 2015, pp. 203–209. ACM, New York (2015)
21. Lee S., Kim, J.: Fluxing botnet command and control channels with URL shortening services. Elsevier Science Publishers B. V. (2013)
22. Antonakakis, M., Perdisci, R., Nadji, Y., Vasiloglou, N., Abu-Nimeh, S., Lee, W., Dagon, D.: From throw-away traffic to bots: detecting the rise of DGA-based malware. In: USENIX Conference on Security Symposium, p. 24 (2011)
23. Yahyazadeh, M., Abadi, M.: BotGrab: a negative reputation system for botnet detection. Comput. Electr. Eng. **41**(6), 68–85 (2015)
24. Sharifnya, R., Abadi, M.: A novel reputation system to detect dga-based botnets. In: International Econference on Computer and Knowledge Engineering, pp. 417–423 (2013)
25. Sharif, M., Lanzi, A., Giffin, J., Lee, W.: Automatic reverse engineering of malware emulators. In: 2009 30th IEEE Symposium on Security and Privacy, pp. 94–109, May 2009
26. Campbell, S., Chan, S., R. Lee, J.: Detection of fast flux service networks. In: Australasian Information Security Conference, pp. 57–66 (2011)
27. Holz, T., Gorecki, C., Rieck, K., Freiling, F.C.: Measuring and detecting fast-flux service networks. In: Network and Distributed System Security Symposium, NDSS 2008, San Diego, California, USA, pp. 487–492, February 2008
28. Yadav, S., Reddy, A.K.K., Reddy, A.L., Ranjan, S.: Detecting algorithmically generated malicious domain names. In: ACM SIGCOMM Conference on Internet Measurement 2010, Melbourne, Australia, pp. 48–61, November 2010
29. Yadav, S., Reddy, A.K.K., Reddy, A.L.N., Ranjan, S.: Detecting algorithmically generated domain-flux attacks with dns traffic analysis. IEEE/ACM Trans. Netw. **20**(5), 1663–1677 (2012)
30. Sharifnya, R., Abadi, M.: Dfbotkiller: domain-flux botnet detection based on the history of group activities and failures in DNS traffic. Digit. Invest. **12**(12), 15–26 (2015)
31. Yadav, S., Reddy, A.L.N.: Winning with DNS failures: strategies for faster botnet detection. In: Rajarajan, M., Piper, F., Wang, H., Kesidis, G. (eds.) SecureComm 2011. LNICST, vol. 96, pp. 446–459. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31909-9_26
32. Jiang, N., Cao, J., Jin, Y., Li, L.E., Zhang, Z.L.: Identifying suspicious activities through DNS failure graph analysis. In: The 18th IEEE International Conference on Network Protocols, pp. 144–153, October 2010

33. Gavrilut, D.T., Popoiu, G., Benchea, R.: Identifying DGA-based botnets using network anomaly detection. In: 2016 18th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), pp. 292–299, September 2016
34. Schiavoni, S., Maggi, F., Cavallaro, L., Zanero, S.: Phoenix: DGA-based botnet tracking and intelligence. In: Dietrich, S. (ed.) DIMVA 2014. LNCS, vol. 8550, pp. 192–211. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08509-8_11
35. Anderson, H.S., Woodbridge, J., Filar, B.: DeepDGA: adversarially-tuned domain generation and detection. In: ACM Workshop on Artificial Intelligence and Security, pp. 13–21 (2016)
36. Golle, P.: Machine learning attacks against the Asirra CAPTCHA. In: ACM Conference on Computer and Communications Security, CCS 2008, Alexandria, Virginia, USA, pp. 535–542, October 2008
37. Yan, J., El Ahmad, A.S.: A low-cost attack on a microsoft CAPTCHA. In: Proceedings of the 15th ACM Conference on Computer and Communications Security, CCS 2008, pp. 543–554. ACM, New York (2008)
38. Zhu, B.B., Yan, J., Li, Q., Yang, C., Liu, J., Xu, N., Yi, M., Cai, K.: Attacks and design of image recognition CAPTCHAS. In: Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010, pp. 187–200. ACM, New York (2010)

Cyber Security Decision Support for Remediation in Automated Computer Network Defence

Maxwell Dondo(✉)

Defence Research and Development Canada, Ottawa, ON K1A 0Z4, Canada
maxwell.dondo@drdc-rddc.gc.ca

Abstract. In making important cyber security course of action (COA) decisions, experts mostly use their knowledge and experience to collate and synthesise information from multiple and sometimes conflicting sources such as the continually evolving cyber security tools. Such a decision making process is resource intensive and could result in inconsistencies from experts' subjective interpretations of how to address the network's security risks. The push towards automated computer network defence (CND) systems requires autonomous decision making and recommendation approaches for network security remediation. In this work, we present such a novel approach through a TOPSIS-based multi-attribute decision making COA selection technique. Our model uses a survey of experts to show that human experts' decisions are indeed inconsistent, even when they are provided with the same information. We then present our decision making approach that is based on considering multiple COA selection factors in an operational environment and implementing a multi-objective selection method that provides network defenders with the best actionable COAs for an automated CND system. Our results show consistency that is unmatched by human experts.

Keywords: Course of action · Vulnerability · Patching
Attack graph · Remediation · Decision-making

1 Introduction

Computer networks supporting modern business processes or missions are becoming increasingly complex. Unfortunately, that complexity means more effort is required to determine and address its vulnerabilities to maintain network security. As a result, defenders have increasingly relied on automation and recommendation tools to assist them in providing the information necessary to implement effective network defence [1,2]. As explained in Sect. 2, some of the automation tools such as MulVAL simplify the network defence task by presenting to the defender all the possible ways that attackers could use to reach certain goals on the defended network [3]. The defender must use this information to determine the set of defensive activities to prevent or make it difficult for the

X. Lin et al. (Eds.): SecureComm 2017 Workshops, LNICST 239, pp. 198–217, 2018.
https://doi.org/10.1007/978-3-319-78816-6_15

attacker to reach those goals. These defensive activities, such as patching vulnerable software or reconfigurations through firewall rule changes are the cyber COAs that are the subject of this study [4,5]. From the options presented to them, defenders must select the COAs that maximally improve the security of the network given finite resources, plausibility of remediation methods and the need to maintain business continuity.

Although making expedient COA selections is a hard problem, it is required. Network security tools and the defenders' expertise provide a holistic understanding of the security posture of the defended network. But, explicit information about the best COAs that maximally improve the network's security and maintain business continuity is not readily available to defenders. Considering the multiple factors that need to be taken into account to select such COAs, the reliance on human expertise can be resource-intensive and can lead to inconsistent results due to the difficulty in making multi-factor decisions inherent in human operators [6,7]. Automation and stand-alone selection tools have been touted as obvious solutions for such limitations [2]. But the context-aware methodologies they need to support consistent and repeatable COA selections are unavailable. In this work, we provide such a methodology. Our approach selects the best actionable network security COAs for implementation given finite remediation resources while minimising disruptions to business processes. We also show the inconsistency of human operators even when they are presented with the same information.

As explained in Sect. 2, existing tools such as Altiris [8] or Redseal [9] currently support COA selection. But, they are not designed to incorporate operational data in their decision making process. Human operators must apply their operational knowledge and experience to information from such tools, as well as other contextual data, to complete the remediation picture and make the necessary COA selection decisions. Multiple experts often make such security decisions, which they have been shown to be mostly incapable of delivering with the consistency expected in CND [6,7]. Thus, we argue that operators need a consistent autonomous methodology such as ours to select the best actionable COAs in an operational environment.

In our approach, which we present in Sect. 3, we first determine factors (attributes) that affect COA selection in the operational environment [10]. Then we formulate the COA selection problem as a multi-attribute decision-making (MADM) problem (a problem that depends on multiple attributes) based on these factors. To solve it, we chose the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) [2,11]. Among many other possible MADM techniques, TOPSIS is well suited for our problem definition since it has been widely and successfully used in solving MADM problems similar to ours [1,2,11].

We applied our model to COA data that we generated using an arbitrarily simulated operational environment on an in-house tool, the inteGrated ENd to End deciSIon Support (GENESIS) [12]. Separately, we used a survey of experts to analyse the effectiveness of using human network security operators in COA selections. Our experts, drawn from knowledgeable colleagues with a minimum of ten years cyber security mitigations experience each, were elicited for their COA selection recommendations in the same simulated environment. We then compared the experts' selections with those from our methodology.

As described in Sect. 4, the survey results validated the difficulty of getting consistent selections among the different experts given the multiple factors they need to consider. This is a known human limitation that was also reported by Miller *et al.* [7] and Kim *et al.* [2]. Although we did not have a way to validate the selections of our model, which we leave for possible future work, it produced self-consistent results that agree with the original multiple objectives of our approach. We present our conclusions in Sect. 5.

2 Courses of Action

Computer network security COAs are the remediation activities required to improve the security of a network. In typical networks, which are usually made up of many interconnected assets, these COAs can be numerous and difficult to determine. Fortunately, attack graph-based algorithms, such as MulVAL [3], simplify that task by determining all the possible paths that an attacker can take to achieve certain goals on a vulnerable network.

An attack graph shows how an attacker could link together network configuration and vulnerability information to achieve their goals [3]. A typical attack graph, showing how an attacker could reach their goal, is illustrated in Fig. 1.

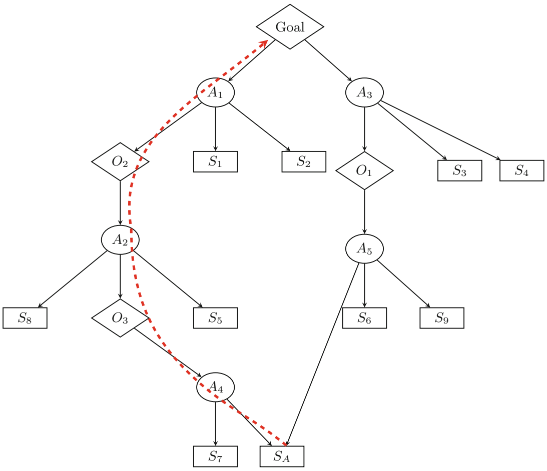


Fig. 1. An example of an attack graph. The dashed line illustrates one of the two possible paths from S_A to the goal node (adapted from [10]).

The attack graph in Fig. 1 is made up of three types of nodes. The rectangular SINKs, oval ANDs and diamond ORs represent facts, logical conjunctions and logical disjunctions respectively. For example, A_2 is only true if A_4, S_5 and S_8 are all true. As illustrated by the dashed line, an attacker located at S_A could reach the *Goal* node through the following logical path: $S_A \rightarrow A_4 \rightarrow O_3 \rightarrow A_2 \rightarrow O_2 \rightarrow A_1 \rightarrow$ Goal. However, that path is not possible if, for example, SINK node S_5 is removed from the attack graph.

Such a removal of a SINK node defines a COA set that we will consider in this work. We represent it as C_1 $[S_5]$ for the first COA set in a set of other COA sets. Other possible COA sets for the dashed path are C_2 $[S_1]$, C_3 $[S_3, S_5]$, C_4 $[S_4, S_5]$ and C_5 $[S_3, S_4, S_5]$. The five COA sets collectively constitute one possible set of COA sets. Our work focuses on determining which one of these five is the best actionable COA set to implement in an operational environment. Further reading on attack graphs can be found in the literature [3–5].

2.1 Characterising COAs

In each COA set, there can be SINK nodes of different types, each type representing a weakness on the network that can be exploited by an attacker. Examples of these types are the existence of software vulnerabilities (*vulExists*), the existence of logical connectivity between two network entities (*hacl*) or the existence of some network service such as email (*networkServiceInfo*). Although there are many possible SINK types (ARMOUR [13], an automated CND architecture, for example, defines nine types), we simplify our work by focusing on the above three. These three are the most common types in COA sets [4,10,12]. This simplification does not affect the generalisation of the problem at hand, and we therefore defer the inclusion of other SINK types to possible future work.

2.2 COA Selection Factors

In an operational environment, network defenders are presented with many COAs to consider. To select a COA, they must consider the different technical and operational factors that characterise its remediation activities. Examples of such factors are the SINK type (e.g. patching an existing software vulnerability) or the availability of technical resources. Our work focuses on preferentially selecting COAs based on these factors. A summary list of the COA factors (attributes) is shown in Table 1. The factors, which were introduced in [10], are listed with their associated ranges of possible numerical scores as used later in our analysis.

From the table, the first three factors represent the SINK type. For example, the factor *Service change* represents the presence of a network service, such as web service, whose mere existence could be exploited by an attacker. The next factor represents the impact on missions or business processes if the COA set is implemented. However, it may be practically infeasible to remove some SINK nodes. We represent this impediment by the fifth factor. Remediation is usually facilitated by using tools such as patching software or scripts. The availability of such tools is represented by the sixth factor.

Table 1. COA selection factors and numerical scores.

Factor i	Description
1. Vulnerability to patch $[0, 20]$	Corresponds to the *vulExists*$(\cdots)$ node in an attack graph, and represents the number of vulnerabilities in a COA set
2. Firewall change $[0, 10]$	Corresponds to the *hacl*$(\cdots)$ node from an attack graph. Often configured in the firewall, it represents a change in the communication rules between two hosts
3. Service change $[0, 10]$	Corresponding to an attack graph's *networkServiceInfo*$(\cdots)$ SINK node. It represents changes to the network services
4. Impact to missions $[0, 10]$	This attribute represents the impact on missions if the COA set is implemented
5. Patch impossible $[0, 1]$	This attribute represents the feasibility of implementing a patch, even if patch exists
6. No remediation tool $[0, 1]$	This attribute represents the existence of a remediation tool
7. Resource limitation $[0, 1]$	This attribute represents the shortage of resources to implement the COA set
8. COA cost $[0, 50]$	This is a predetermined COA remediation cost. This attribute represents the cost assigned to the COA set [4,5]
9. Security benefit $[0, 1]$	This attribute represents the percentage of the attack graph that is eliminated by the implementation of the COA set

The next factor represents the shortage of resources to implement the COAs. The COA cost attribute assigns a relative numerical measure representing the total remediation costs. In addition, the effort required in removing some SINK nodes may be higher than others and the network defenders might not have enough resources to ensure the complete removal of all COA set nodes. The final factor represents the security benefit obtained if the COA is implemented. For our security benefit, we use a rank measure developed by Sawilla and Ou [4,14]. It represents the importance of a graph vertex to an attacker. The security benefit comes from the fraction of these vertices that is removed (through remediation) to prevent an attacker from reaching their goal. An ideal rank elimination is 1 as opposed to an undesirable value of 0.

Based on our research, we found the nine factors presented in Table 1 to be vital for COA selection in an operational environment. However, it is possible that there are other factors that we may have missed. Our approach can be extended to include an extended set of factors if necessary.

2.3 COA Selection Challenges

Most network defenders have the knowledge and experience to make remediation decisions based on considering known remediation factors and selecting the best actionable COAs. For each factor, the selection objective is either to maximise or minimise it. For example, from the factors in Table 1, the remediation objective is to minimise mission impacts (Factor 4) and maximise the security benefits (Factor 9).

Humans can effectively handle one objective at a time. But, when multiple objectives are to be simultaneously considered, research has shown that the consistency and reliability of such selection decisions become questionable [6,7]. This limitation has raised the need for selection automation and recommendation methodologies to assist the human operator, a capability gap that our approach aims to fill.

2.4 Related Work on COA Selection

There are some commercial COA selection tools such as patch management systems (e.g. Altiris [8,15]) or reconfiguration management systems (e.g. Redseal [9]). However, they are not designed to incorporate operational context information that is important for selecting the best COAs while maintaining business continuity. The limited information they provide leads to inconsistent subjective decisions by human operators [2,7], a limitation that we aim to address in our work.

COA selection approaches by researchers such as Sawilla *et al.* use the attack graph context [4,5,14]. Their approaches include assumed cost measures representing the limitation of resources as well as a measure of the security benefit obtained by making a particular selection. However, their approaches do not explicitly include operational context or mission-related factors. Their selections do not address cases where missions or business continuity could be impacted by the implementation of the COAs on operational networks. In addition to utilising the attack graph and resource limitation concepts used by Sawilla *et al.*, our work includes other operational factors in deciding the COAs to select for implementation.

Other researchers have focused their selection methodologies on COA remediation costs [16] or network risk [17]. The former uses lowest cost COAs to recommend graph cuts. The latter selects COAs based on the risks and costs determined from the vulnerability and host importance in the attack paths. However, both approaches do not address operational impacts which are important in defence operational networks. But, we find the use of host importance measures by Hong *et al.* [17] relevant in providing operational context to defended networks. So, we borrowed that concept and applied it to mission impacts in our work.

The work by Kim *et al.* [2] focuses on security event prioritisation, the remediation of which is the same as the COAs we are focusing on in this work. What is important about their work is a prioritisation approach that includes host importance measures as one of their deciding factors. They also take into consideration the asset criticality based on the mission that the asset is part of. We consider these factors to be important in COA selection and include them in our work. Another important aspect of their approach is the use of a modified TOPSIS technique to determine their final prioritisation. The modified approach allowed them to avoid changes in prioritisation based on different input scores. It also allowed them to compare results across different calculation runs. In addition to TOPSIS's wide acceptance and success in solving MADM problems [1,18–21], its application by Kim *et al.* shows that it is well suited to solve our selection problem. We therefore adopt that approach and similarly incorporate missions and impact data.

3 Our TOPSIS Approach

3.1 TOPSIS

TOPSIS, which was proposed by Hwang and Yoon, is a MADM methodology that selects the best alternative in a multi-attribute problem [11]. The idea is centered on the premise that the best alternative should have the shortest geometric distance from a hypothetical positive ideal solution (the zenith) and longest geometric distance from a hypothetical negative ideal solution (the nadir).

Consider a problem to make prioritised selections from m alternatives $\mathrm{C_i} : \ \mathrm{i} = 1, \cdots, \mathrm{m}$. Each alternative $\mathrm{C_i}$ is characterised by n factors such that the score for the ith factor of the jth alternative is x_{ij}. These alternative scores are represented by the decision making matrix shown in Table 2. The weights w_i represent the overriding preferences of one factor over others.

Table 2. The decision matrix.

	Factor 1	Factor 2	$\cdots$	Factor n
C1	x_{11}	x_{12}	$\cdots$	x_{1n}
C2	x_{21}	x_{22}	$\cdots$	x_{2n}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Cm	x_{m1}	x_{m2}	$\cdots$	x_{mn}
Weights	w_1	w_2	$\cdots$	w_n

To determine the relative closeness of the alternatives from the zenith, TOPSIS's first step is to normalise the decision matrix shown in Table 2 as follows:

$$z_{ij} = \frac{x_{ij}}{\sqrt{\sum_{j=1}^{m} x_{ij}^2}} \tag{1}$$

The normalised decision matrix is then multiplied by the weights for each factor to give the weighted normalised decision matrix such that $v_{ij} = w_i z_{ij}\ \forall i, j : i = 1, \cdots, m,\ j = 1, \cdots, n$.

The zenith $A^+ = \{v_1^+, \cdots, v_m^+\}$ is made up of the best values for each criterion and the nadir $A^- = \{v_1^-, \cdots, v_m^-\}$ is made up of the worst values of each criterion. For example, v_1^+ is the highest value for a maximisation objective on Factor 1, and v_1^- is the lowest value. Similarly v_2^+ is the lowest value for a minimisation objective on Factor 2, and v_2^- is the highest value.

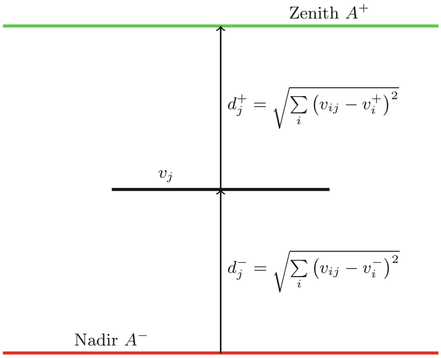

Zenith A^+
$d_j^+ = \sqrt{\sum_i (v_{ij} - v_i^+)^2}$
v_j
$d_j^- = \sqrt{\sum_i (v_{ij} - v_i^-)^2}$
Nadir A^-

Fig. 2. A simplified illustration of the TOPSIS approach.

These concepts are illustrated in Fig. 2. Given the Euclidean distances shown in the figure, the relative closeness score t^+ for each alternative to the zenith is calculated from

$$t_j^+ = \frac{\left[\sum_i \left(v_{ij} - v_i^-\right)^2\right]^{\frac{1}{2}}}{\left[\sum_i \left(v_{ij} - v_i^+\right)^2\right]^{\frac{1}{2}} + \left[\sum_i \left(v_{ij} - v_i^-\right)^2\right]^{\frac{1}{2}}} \tag{2}$$

This means that selection alternative a is better than b if and only if $t_a^+ > t_b^+$, and indistinguishable if $t_a^+ = t_b^+$.

3.2 Our Approach

Our model uses TOPSIS to analyse the multiple factors and their corresponding objectives so as to select the best actionable COAs for the given factors. We first populate the decision matrix shown in Table 2 with scores from the COA sets that we need to choose from by assigning values to each factor for all the COA alternatives $\mathrm{C_i}$ under consideration.

The first three factors are simple counts of the number of SINKs of each type in the COA set. For example, if there are 2 vulnerabilities in the COA set, then the score for the first factor would be 2. Since a network can only be as secure as its least secure assets, the missions impact values are determined by the highest mission impact score of the asset, or set of assets, whose SINK nodes are associated with a COA set [7]. That means, the mission impact for $\mathrm{C_i}$ would be the highest mission impact on hosts identified by the vulnerable nodes identified in the COA set. For example, if the COA set points to SINKs on hypothetical nodes 7, 8, and 9 (see Fig. 1), then the mission score for that COA set is the maximum score for the missions supported by those three nodes. Although the mission impact score ranges in $[0, 10]$, we used discrete values of 0, 1, 5, 8 and 10 representing *None* (N), *Low* (L), *Medium* (M), *High* (H) and *Very High* (VH) respectively. Such assignments correspond to those used in operational networks [2,10].

Operators assign scores for the next three factors from known remediation impediments. The cost and rank scores are assigned by GENESIS based on Sawilla's algorithms [4,14]. To simplify our problem, we assumed that all the scores would be automatically assigned by an autonomous remediation module that aggregates network security data. For military networks, we further assumed that missions data would be readily associated with each asset on the network. In keeping with organisational policy or prevailing security risks, operators can assign relative weights to the factors so that selection preference can be given to some factors over others. For our work, we assume that all factors have equal weights although our model can handle varying them to represent operational preferences.

Before applying TOPSIS, we slightly modified it by changing the zenith and nadir vectors. In their work, Kim *et al.* [2] noted that new input values can change the zenith and/or nadir. Such changes require the recalculation of t^+, which could result in changes to the selection alternatives. They avoided this problem by fixing the values of the zenith and nadir to the maximum (for maximisation) or minimum (for minimisation) possible scores for the zenith and the opposite for the nadir. We use this technique in our work to allow for selection comparisons across multiple sets of COA sets.

We then apply TOPSIS to our decision matrix. Our model, which simultaneously combines all selection objectives, calculates the values of t^+ for each COA set alternative. We then rank the COA sets based on the their relative scores t^+. The COA set with the highest score becomes the first choice for implementation. We tested our model on a simulated experimental network.

4 Experimental Results

4.1 Test Data

For test data, we used COA sets that we generated using an existing virtualised lab network prototype, GENESIS, shown in Fig. 3. The network consists of fully configured virtual hosts running real operating systems and servers with real vulnerabilities. To emulate an operational network, it is made up of three zones, the demilitarised zone (DMZ), the security and corporate zones. Despite its small size, the COAs generated from it are the same as those generated in a large enterprise network, except that the latter would have a significantly larger number of COAs.

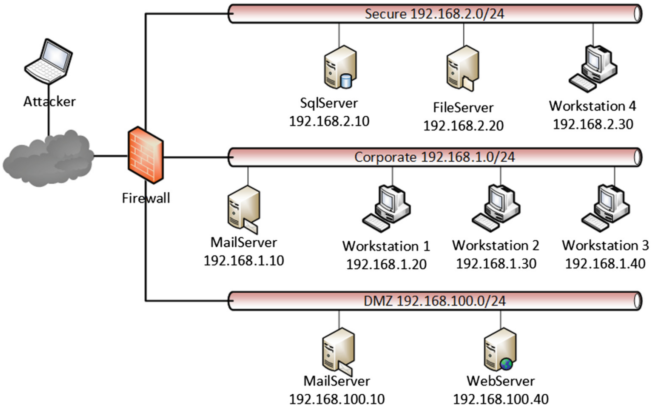


Fig. 3. The test network in a virtual environment (adapted from [4]).

We simulate an attacker, located on the Internet, targeting any of the assets on the network. We also assumed that an attacker could target any asset on the network from any other network host, a reasonable assumption given that attackers can launch multi-step attacks from any other node. This setup allowed us to generate more data for our testing than we could have achieved otherwise. We then arbitrarily assigned attackers and targets on the network and used MulVAL to generate attack graphs for each attacker-goal combination [3–5]. For each combination, we generated COA sets by repeatedly relaxing the remediation budget limits using Sawilla's algorithm [4]. This approach enabled us to generate 120 unique sets of COA sets for our experiments.

Table 3. A set of COA sets generated using MulVAL.

COA set	Nodes in COA set
C_1	[99]
C_2	[99,114]
C_3	[99,114,163]
C_4	[99,114,21,163]
C_5	[99,114,21,29,163]
C_6	[99,114,121]

A typical set of COA sets generated this way is shown in Table 3. The table shows 6 COA sets C_i for $i = 1, \cdots, 6$. When MulVAL [3] generates the attack graph, it assigns identification numbers to each SINK node in the COA set (see Sect. 2). These node numbers are represented in square brackets in Table 3. For example, the COA set C_1 requires the removal of SINK node 99, which may be patching a software vulnerability. We will use the above set of COA sets in the examples of our results later on.

4.2 Survey of Experts

In order to analyse the selections of human experts under given scenarios represented by our selection attributes, we carried out a survey of cyber security experts drawn from experienced colleagues. We presented them with the network shown in Fig. 3, whose assets had been arbitrarily assigned to missions. We assumed that this network is able to support different missions that could be impacted differently by remediation, a reasonable assumption for a typical operational network.

To limit the time spent on the survey, we arbitrarily selected 50 sets of COA sets for the survey. Through a custom survey application, we tasked the experts with completing the survey three independent times. During the survey, they were repeatedly presented with a set of COA sets similar to the one in Table 3 together with selection rationales as represented by the values of selection attributes. The experts were then expected to use their knowledge and experience on those rationales to select the best and second best COA sets to implement to improve network security. For example, from Table 3, an expert could select C_5 and C_2 as the best and second best COA sets. We recorded the survey results for further analysis.

4.3 Decision Making with TOPSIS

TOPSIS Consistency

We used attribute scores to populate the TOPSIS decision matrix (see Table 2). Then we determined the relative closeness score t^+ for each COA set, and selected the set with the highest value of t^+ as the best alternative. We also ranked the rest of the alternatives based on the COA sets' scores. For example, in one scenario, the rankings of the set of COA sets in Table 3 were, from best to worst, C_3, C_4, C_2, C_6, C_5 and C_1. However, before we discuss the selections in detail, we analyse the validity and consistency of our TOPSIS approach for the given attributes.

To analyse TOPSIS's consistency and repeatability, we determined how well its solutions satisfied the multiple objectives reflected in the decision matrix. For each objective, we determined how changes to other attributes affect the overall TOPSIS score, and therefore the resulting selections. If TOPSIS is consistent, we would expect the variation in scores to show a monotonic increasing curve for maximisation and decreasing for minimisation objectives. We demonstrated these variations using three arbitrarily selected, but representative, scenarios.

Scenario 1

Using the set of COA sets shown in Table 3, we first consider the variations of TOPSIS scores on COA set C_6. We varied the missions impact scores on C_6 (equivalent to mission impacts on an asset on node 121 for example) while keeping all the other attribute scores constant. The variation of the TOPSIS scores on COA set C_6 are shown as blue diamond shapes in Fig. 4. The C_6 graph (blue diamonds) shows that as the impact level of the COA set C_6 was increased from N to VH, the TOPSIS score showed a monotonic decreasing trend. This is an expected result, since an increase in the impact should translate into a less favourable alternative, and therefore a lower TOPSIS score.

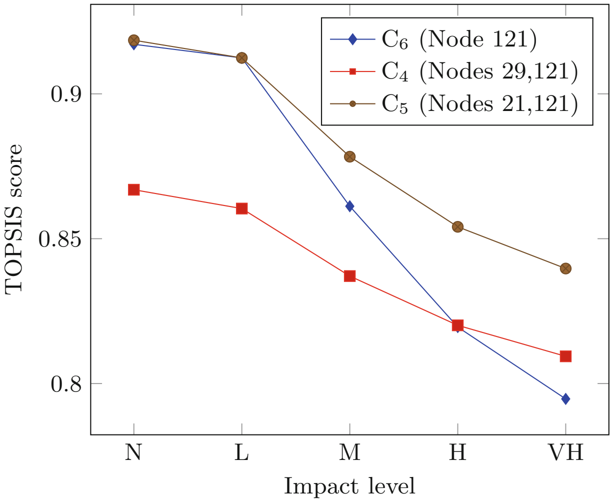


Fig. 4. TOPSIS scores for variations in impact levels for COA sets C_5 and C_6 due to changes in impact levels on nodes 21, 29 and 121.

Scenario 2

In this experiment, we also performed similar analysis on COA sets C_4 and C_5. We simultaneously varied the mission impact scores for C_4, C_5 and C_6 (equivalent to simultaneous mission impacts on assets at nodes 21, 29 and 121). The variations in TOPSIS scores for COAs C_4 and C_5 are respectively shown on red squares and brown circles graphs in Fig. 4.

Similar to the C_6 (blue diamond) graph, these two graphs are monotonically decreasing with worsening impact scores as would be expected. The curves also show that as the impact scores for COA sets C_4 and C_5 were changed simultaneously by the same value, the resulting TOPSIS scores maintained the superiority of C_5's score over C_4's. Due to the nonlinear nature of t^+, the gap between the curves' scores is not always maintained, although the relative ranking is.

Scenario 3
In the third and final scenario, we analysed the TOPSIS scores on COA set C_6 with variations in impact and rank scores on that COA set. The variations are shown in Fig. 5. For each graph, the monotonically decreasing shapes are the same as in Fig. 4–a trend that is expected for the same reasons. For each impact level on C_6, the TOPSIS scores are also monotonically increasing with increasing rank values. These results are also what we expected since the objective on this attribute is to maximise the security benefit score.

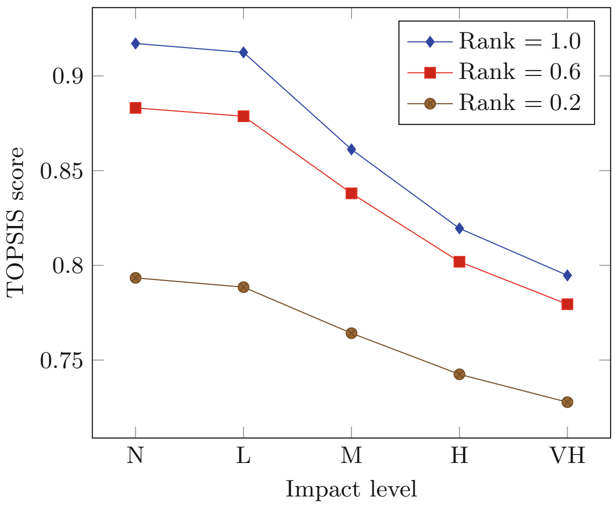


Fig. 5. Variations in TOPSIS scores for set C_6 with variations on impact and security benefit scores for the same COA set.

The rest of the attributes showed trends pursuant with their decision matrix objectives. This led us to conclude that although we do not have a way to validate these comparisons at this time, they are repeatable and self consistent and, most importantly, meet the objectives for which they were intended.

TOPSIS Selection Examples

Having determined the consistency of TOPSIS, we performed a number of experiments to show how well it can handle COA selections in an operational environment. We aimed to demonstrate this by showing the different selection options as the values for the COA attributes were changed. The summary of a typical selection ranking using our algorithm is shown in Table 4.

The table is divided into three parts represented by circled letters. Each part demonstrates an important aspect of our TOPSIS approach. The first part (a) of the table shows ranking variations as the impact values for node C_6 are changed. The second part (b) shows similar ranking variations for simultaneous impact changes for nodes C_5 and C_6, representing similar impacts on nodes 29 and 121.

Table 4. TOPSIS selections for COAs on nodes with varying impacts. Column VH* also simulates an unavailable patch.

COA	Selections									
	ⓐ Impacts for C_6 nodes					ⓑ Impacts for C_5 and C_6 nodes				
set	N	L	M	H	VH	L	M	H	VH	ⓒ VH*
C_1	6	6	6	5	5	6	6	5	4	3
C_2	4	4	3	3	3	4	4	3	3	2
C_3	2	2	1	1	1	2	2	1	1	✗
C_4	3	3	2	2	2	3	3	2	2	1
C_5	5	5	5	4	4	5	5	6	6	5
C_6	1	1	4	6	6	1	1	4	5	4

The third part (c) shows the same ranking selections as in the VH impact of the second part of the table, but with no patch for a vulnerability in COA set C_3.

Impacts on C_6 *Nodes*
The first part of Table 4 shows that when there was no (N) impact to missions, COA set C_6 was the best choice. The same selection was taken at low (L) impact, although C_3 was selected for higher impacts. These selection variations were a result of the changes in TOPSIS scores. The different selections confirm score changes with variations to mission impacts as illustrated in Figs. 4 and 5. In this case the higher mission impact levels for C_6 gave it a lower TOPSIS score, making it a less favourable alternative and C_3 the best choice. The unchanging selections for some impact levels (e.g. from M to VH for C_3) were due to insufficient relative score variations resulting from changes in impact levels in COA set C_6.

Impacts for C_5 *and* C_6 *Nodes*
We obtained similar results when we simultaneously changed the impact scores for C_5 and C_6 (equivalent to changing impact scores on node 121). The ranking trend was the same for the lowest impacts (N and L), and mostly the same for high impacts, but significantly different for medium impact (note that the ranking for no impact is the same as in the previous case). This difference was due to insufficient TOPSIS score changes, resulting from changes in mission impacts from low (L) to medium (M), to allow C_3 (second choice) to be selected instead.

No Patch for C_3
Finally, using the same scores for the VH impact scores in the previous part, we simulated an unavailable patch for C_3. The COA set C_3 was eliminated from the ranking and COA set C_4 became the best option.

These selection experiments show the capability of our algorithm to select the best actionable COAs satisfying its multiple objectives. This is the consistency and repeatability we expected. The invariancy of the selections observed in some cases are a result of small changes in the TOPSIS score that were not high enough to trigger selection changes. To further study our algorithm's performance, we analyse surveyed experts' selections and compare them with those from our approach.

4.4 Experts' Selections

As discussed earlier, we asked three experts, A, B, and C, to make preferential selections from 50 sets of COA sets. We asked each expert to complete the survey three times. Each instance of the experts' survey attempt is represented by the number of that attempt. For example, the first, second and third survey attempts by expert A are represented as A_1, A_2, and A_3 respectively. Our objective was to analyse the degree to which our experts' selections agreed with each other and with our TOPSIS approach.

To analyse the degree of agreement, we define an agreement factor s as the ratio of selection agreements m to the total number of sets of COA sets n under consideration [10]. Thus

$$s = \frac{m}{n} \tag{3}$$

Similar to correlation measures, we consider agreement factors close to the perfect agreement $s = 1$ as very strong and those close to no agreement ($s = 0$) as very weak.

Table 5. Comparisons of experts' selection agreements.

		Expert selections								
		A_1	A_2	A_3	B_1	B_2	B_3	C_1	C_2	C_3
Experts' selections	A_1	–	74%	72%	46%	52%	46%	68%	66%	68%
	A_2		–	84%	54%	58%	52%	68%	66%	66%
	A_3			–	54%	58%	52%	70%	72%	72%
	B_1				–	72%	72%	56%	58%	56%
	B_2					–	78%	58%	56%	58%
	B_3						–	62%	60%	62%
	C_1							–	88%	88%
	C_2								–	92%
	C_3									–
TOPSIS		64%	68%	74%	60%	54%	58%	84%	94%	84%
All		44%								
TOPSIS		32%								

The results from the comparisons of the three experts' selections are summarised in Table 5. In the table, each expert's three selections are compared against the other experts'. For example, expert A's first survey selections A_1, were compared against their second A_2 and third A_3 selections, as well as those performed by B and C. The third row from the bottom shows the agreements of each expert with our TOPSIS approach. The next row shows the simultaneous agreement levels of all the experts. The simultaneous agreements of all experts and our TOPSIS approach is shown in the last row.

The self-consistency of over 70% among the experts is strong for human experts in a multi-attribute problem. The experts showed their highest self-consistency between the second and third selections. This is an expected result since the experts would have been more familiar with the alternatives during the final two survey attempts than during the first.

The agreements among different experts is not as strong as the experts' self-consistency. The highest agreement was between A's and C's selections, at about 70%. In total, all experts' selections are in simultaneous agreement in 44% of the cases, which is low. Such results underscore the need for a consistent approach to prioritise the COAs in an environment that could be manned by many experts or for an application to autonomous defence modules in automated CND.

With all the inconsistencies in the experts' selections, it is difficult, if not impossible, to determine if the selections are correct. So, we further analysed the selections against the systematic TOPSIS rankings that we have just determined to be consistent in its selections (although we have no way to validate its accuracy at this time). As shown in Table 5, expert C has the highest agreement with the TOPSIS ranking. However, all experts' selections are in simultaneous agreement with the consistent TOPSIS selections in only 32% of the cases, implying a collective expert consistency of 32%. This is not a surprising result since the experts could choose COAs that are off the maximal selection (provided by TOPSIS) by varying levels of magnitude.

An example of these selection inconsistencies are shown in Table 6. In the table, we show the first two selections by each expert and compare them to the TOPSIS selection (T). We also show the TOPSIS rankings for selection comparisons. In this example, the experts' selections at low impact levels were all consistent. However, at a high impact level, the inconsistency is apparent. While A and C were self-consistent, C's selection did not match the TOPSIS selection as A did. B's selections did not match each other nor TOPSIS's, again reinforcing the need for a methodology, such as demonstrated by our approach, that could either be used in automated CND or provide consistent remediation support to network security operators.

Table 6. Analyst and TOPSIS selections for COAs on nodes with varying impacts.

	Selections													
COA	Low impact							High impact						
set	A_1	A_2	B_1	B_2	C_1	C_2	T	A_1	A_2	B_1	B_2	C_1	C_2	T
C_1							6							5
C_2							4							3
C_3							2	✓	✓					✓
C_4							3				✓			2
C_5							5					✓	✓	6
C_6	✓	✓	✓	✓	✓	✓	✓			✓				5

4.5 Discussion and Possible Future Work

Our work showed the inconsistency in the selections by human experts even when they are presented with the same information. In practice, this is not unexpected as it reflects each expert's security preferences, which are based on their knowledge and experience. Unfortunately, such inconsistencies may result in errors when the network can least afford them. In addition, the inconsistencies make it hard to model and incorporate COA selections into automated CND systems such as ARMOUR. This reinforces the need for consistent systematic approaches such as ours, which can be integrated with automated CND systems.

In the absence of a ground truth, it is not possible to validate the solutions from our approach. But, the approach is self-consistent and its selections and rankings are based on systematic measures that represent a combination of multiple objectives that reflect the security requirements in an operational environment. We therefore argue that, given TOPSIS's high success rates in solving MADM problems [1,2], its solution is a close representation of the multiple objectives we originally identified in our decision matrix. Compared to the inconsistent manual process by human operators, our approach is a good candidate for applications in autonomous network security modules in automated CND.

There are two main possible applications of our work. The first application is for autonomous COA selection decision making in automated CND systems. The approach would get security context data from the defended network environment and aggregate it with operational data to determine selection measures that would help the system to select the best actionable COA under the given conditions. The second possible application is to train cyber security experts in making consistent selection decisions. The system could be used to compare its selections against experts' and the results used to train operators or identify areas needing improvement. It could also be used to identify and correct operators' subjective selection biases in both the selection factors and the operators.

One deficiency from our approach is that it is difficult to quantify a selection miss to determine how far it is from the correct result. The multi-factor score difference could be so minor to be insignificant or so big that it could be a show stopper. All our approach does is to determine a measure of closeness to an ideal solution that meets our objective. We recommend future work to look into variance measures that reflect how far a selection is from a perceived ideal one.

Although our study is based on a limited number of factors that we determined using our network security expertise, it has produced promising results showing the relative ranking of COAs. While our approach is supposed to work with a broader set of factors than the ones we used in our work, we did not test it as it was not part of our study and we do not know how scalable that expansion would be. We therefore recommend future work to study possible additional factors that could influence COA selection decisions. The study could use consensus-based rating techniques, such as the Delphi method [22], with security experts to determine and prioritise those factors. Factor prioritisation weights can then be assigned accordingly (see Sect. 3).

To improve the accuracy and comparison of our algorithm, we could investigate the aggregation of experts' decisions instead of analysing them as individual decisions. That way, we can compare our algorithm against the consensus decisions of our experts. Methodologies such as the Delphi method [22] could also be used to determine experts' consensus selections. We recommend further studies on whether such consensus selections could be considered as the ground truth against which to compare our algorithm.

Our work selected to use the TOPSIS approach based on its reputation in solving MADM problems similar to ours. However, other techniques such as simple additive weighting (SAW), and analytical hierarchy process (AHP), for example, could be considered as possible solution candidates as well. In addition to consistency and repeatability, the techniques can then be evaluated based on other measures such as simplicity of use, time, and understandability for example. We leave such investigations to possible future work.

Our work was carried out on a simulated lab network with real vulnerabilities. Practical networks are more complicated than the GENESIS network that we used. We did not have data to test our approach with such complex networks, so it is unclear how our approach would perform under such conditions. We expect the number of COA sets to be significantly large, requiring multiple sizeable runs of our algorithm. This could take more time than in the small network used in our study. Such additional time could impact decision making in automated systems where consistent results are needed promptly. We therefore recommend future research to investigate the impact of applying our algorithm on large operational networks.

5 Conclusions

In this work we have shown how inconsistent human operators can be when asked to make course of action (COA) selections even if they are provided with the same information. This is undesirable for automated computer network defence (CND), which requires consistent and repeatable COA selections based on identified contextual and security information. To correct this inherent weakness in human decision making, we have developed a multi-attribute decision-making (MADM) algorithm to select actionable COAs for the effective security of a defended network. For its decisions, our algorithm uses network security, operational and contextual factors that we believe to be the most important for COA selection and prioritisation.

We have shown our approach produces repeatable and consistent selections based on quantifiable security measures from the network and the operational environment. Our solution is based on the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) technique that has found significant successes in solving MADM problems resembling ours. Therefore, in the absence of a ground truth to validate our approach, we argue that TOPSIS's repeatable and consistent solution of our MADM COA selection problem as characterised by the multiple selection objectives, will effectively provide network security that meets those goals. Our results' repeatability and self consistency have been shown to outperform human experts', making our model a good candidate for automated CND applications that require consistent and reliable solutions based on the security environment.

Our approach could also contribute to the efficient utilisation of resources in an operational environment. The low levels of simultaneous agreements among experts show that sole reliance on human expertise could contribute to resource wasting as experts would need to expend more time to resolve their selection disagreements before implementing remediation measures. Such time could be best utilised in performing other security tasks if the technique we report in this work is exploited into an operational tool.

References

1. Kim, A., Kang, M.H.: Determining asset criticality for cyber defense. Naval Research Laboratory. Technical report NRL/MR/5540-11-9350 (2011)
2. Kim, A., Kang, M.H., Luo, J.Z., Velasquez, A.: A framework for event prioritization in cyber network defense. Naval Research Laboratory. Technical report NRL/MR/5540-14-9541 (2014)
3. Ou, X., Govindavajhala, S., Appel, A.W.: MulVAL: a logic-based network security analyzer. In: USENIX Security (2005)
4. Sawilla, R., Burrell, C.: Course of action recommendations for practical network defence. Defence Research and Development Canada. Technical Report DRDC Ottawa TM 2009–130 (2009)
5. Sawilla, R., Skillicorn, D.: Partial cuts in attack graphs for cost effective network defense. In: IEEE International Conference on Technologies for Homeland Security, HST 2012, pp. 291–297 (2012)
6. Miller, G.A.: The magical number seven, plus or minus two: some limits on our capacity for processing information. Psychol. Rev. **101**, 343 (1994)
7. Miller, S., Appleby, S., Garibaldi, J.M., Aickelin, U.: Towards a more systematic approach to secure systems design and analysis. Int. J. Secure Softw. Eng. **4**(1), 11–30 (2013)
8. Symantec: IT analytics 7.1 for altiris it management suite from symantec. Symantec, Technical report (2013)
9. RedSeal Networks. Security target. https://www.redseal.net
10. Dondo, M.: A neural network approach for cyber security course of action selection. Defence Research and Development Canada, Technical report DRDC-RDDC-2016-R269 (2016)
11. Hwang, C.-L., Yoon, K.: Multiple Attribute Decision Making: Methods and Applications a State-of-the-Art Survey. Springer Science & Business Media, Heidelberg (2012). https://doi.org/10.1007/978-3-642-48318-9
12. McKenzie, C.: GENESIS: integrated end-to-end decision support for computer network defence, design and architecture document. Defence Research and Development Canada. Technical report DRDC Ottawa CR 2011–009 (2011)
13. Sawilla, R.E., Wiemer, D.J.: Automated computer network defence technology demonstration project (ARMOUR TDP): concept of operations, architecture, and integration framework. In: 2011 IEEE International Conference on Technologies for Homeland Security (HST), pp. 167–172, November 2011

14. Sawilla, R.E., Ou, X.: Identifying critical attack assets in dependency attack graphs. In: Jajodia, S., Lopez, J. (eds.) ESORICS 2008. LNCS, vol. 5283, pp. 18–34. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88313-5_2
15. Symantec. Symantec patch management solution powered by altiris technology. http://www.symantec.com/products
16. Alhomidi, M., Reed, M.: Finding the minimum cut set in attack graphs using genetic algorithms. In: 2013 International Conference on Computer Applications Technology. ICCAT 2013, pp. 1–6, January 2013
17. Hong, J., Kim, D.S., Haqiq, A.: What vulnerability do we need to patch first? In: 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks. DSN 2014, pp. 684–689, June 2014
18. Chakraborty, S., Yeh, C.-H.: A simulation based comparative study of normalization procedures in multiattribute decision making. In: Proceedings of the 6th Conference on Artificial Intelligence: Knowledge Engineering and Data Bases (2007)
19. Boran, F.E., Genç, S., Kurt, M., Akay, D.: A multi-criteria intuitionistic fuzzy group decision making for supplier selection with TOPSIS method. Expert Syst. Appl. **36**(8), 11363–11368 (2009)
20. Safari, H., Khanmohammadi, E., Hafezamini, A., Ahangari, S.S.: A new technique for multi criteria decision making based on modified similarity method. Middle-East J. Sci. Res. **14**(5), 712–719 (2013)
21. Velasquez, M., Hester, P.T.: An analysis of multi-criteria decision making methods. Int. J. Oper. Res. **10**, 56–66 (2013)
22. Linstone, H.A., Turoff, M.: The Delphi Method. Addison-Wesley, Reading (2002)

Situational Crime Prevention and the Mitigation of Cloud Computing Threats

Chaz Vidal[1] and Kim-Kwang Raymond Choo[2,1](✉)

[1] School of Information Technology and Mathematical Sciences,
University of South Australia, Adelaide, SA 5095, Australia
raymond.choo@fulbrightmail.org
[2] Department of Information Systems and Cyber Security,
University of Texas at San Antonio, San Antonio, TX 78249-0631, USA

Abstract. Security is a key challenge in the deployment and broader acceptance of cloud computing services, and existing research efforts include evaluating the effectiveness of various security solutions such as security policy implementations and technological solutions. Attacks on cloud environment may be considered from the criminological perspective, and crime theories be used to protect the cloud. This paper introduces a conceptual cloud security model utilizing the concept of situational crime prevention (SCP). Using SCP techniques, it may be possible to design process and technology-based steps to modifying the cloud computing environment to make it less attractive to crime.

Keywords: Situational crime prevention · Cloud security
Crime opportunity theories

1 Introduction

The use of cloud computing has become ubiquitous in recent years. Cloud computing comes in many forms such as easily configurable servers (e.g. those from Amazon Web Services and other cloud service providers) and online file storage services (e.g. Dropbox). Consumers with access to these technological resources then have the ability to use the resources in the way they need, such as building virtual servers for application development or web serving or online internet based backups. Most of these uses are generally non malicious, but with the use of technology does comes with it an inherent risk as overall security remains a prime concern certainly for cloud service providers (CSPs) and those who use cloud services.

Security is a major impediment to the overall uptake of cloud computing and there have been a number of security incidents that involved the use of cloud services in high profile criminal activities, which in turn highlights the need for enhanced security, privacy and forensic capabilities (Hiller and Russell 2013; Quick et al. 2013). Such incidents may also be considered cybercrime if they are in violation of existing legislation at the jurisdiction the incidents occurred or where the victim is located. Cloud computing infrastructure can then be protected using a combination of specific technology-based solutions (Vidal and Choo 2015; Osanaiye et al. 2016; Poh et al. 2017).

X. Lin et al. (Eds.): SecureComm 2017 Workshops, LNICST 239, pp. 218–233, 2018.
https://doi.org/10.1007/978-3-319-78816-6_16

A large number of strategies to manage and enhance security within cloud computing environments have also been proposed in the literature (Ab Rahman and Choo 2015; Iqbal et al. 2016). However, we approach this from a different perspective. Specifically, we posit that a more effective approach is to combine existing mitigation strategies using the lens of a crime prevention theory, i.e. situational crime prevention (SCP) in this paper. In the next section, we present background information.

2 Cloud Security

Cloud computing has arguably come of age. It has progressed from a collection of web-based services to a clearly defined computing strategy, one that is used by both commercial consumers and large enterprize customers alike. The National Institute of Standards and Technology (NIST), for example, describes cloud computing as a model for enabling network access to configurable computing resources quickly with minimal interaction from own service providers (Mell and Grance 2011).

NIST also ascribes five characteristics of a cloud service model, namely: on-demand self-service, seamless network access, resource pooling, rapid elasticity and measured services. The service models include Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). In recent times, other service models have also been suggested in the literature such as Security as a Service (Varadharajan and Tupakula 2014), Collaboration-as-a-Service and Network-as-a-Service (Gu et al. 2013).

NIST further defines how these service models are deployed, namely: a private cloud, a community cloud, a public cloud or a hybrid cloud.

The technology for cloud computing has matured and have become widely accepted (Khan et al. 2012). However, despite these improvements, there are difficulties that have been recognized as barriers to wider acceptance. One particular aspect of cloud computing that has become more problematic is ensuring adequate security is implemented both for its potential users and by CSPs (Lokhande and Shelke 2013).

Despite its maturity, cloud computing is still vulnerable to security issues including cyber attacks and misuse and abuse of the cloud computing infrastructure. Some of these attacks are more difficult to carry out in nature such as extracting private information from different virtual machines (VMs) that share the same cloud computing resources. Other attacks are more traditional such as Distributed Denial of Service (DDoS) on known public clouds (Dawoud et al. 2010). Cloud computing is also vulnerable to misuse and abuse from its own users such as using cloud resources to host malware or contraband or illegal material (Choo 2010; Julidotter and Choo 2015; Rogers 2012).

Because of the potential for misuse by the criminal element, the onus not only is on the cloud users to protect and educate themselves on cloud usage but also on CSPs to establish protection mechanisms (Antonopoulos and Gillam 2010).

For CSPs, a number of strategies can be employed to ensure that adequate protection of the cloud service. As an on organization, CSPs can apply Information Security standards to their services (AS/NZS 2006; Ab Rahman and Choo 2015). These standards will allow the CSPs to identify the threats and risks associated with the delivery of the cloud service and formulate specific controls to mitigate these risks.

Identifying cloud computing risks have been an easier job for CSPs because of the availability of industry-based groups and their work in showing the top threats to cloud computing. The Cloud Security Alliance, for example, over the past few years have shown where CSPs should concentrate on to mitigate the threats to their cloud services (Cloud Security Alliance 2016).

3 Cybercrime in the Cloud

Since the advent of computers and their availability for most everyone, cybercrime has been steadily on the rise. There have been several attempts to describe cybercrime and there appears to be some difficulty in providing a universally accepted definition (Hunton 2011). In recent years, cybercrime has been used to described technology related criminal acts perpetrated through the Internet but at the same time, there are instances when cybercrime covers more than just criminal acts but also includes undesirable or offensive behavior.

With the many definitions in use today, it is important to focus on an agreed to framework to describe cybercrime and cyber-criminals. Australia's National Cybercrime Working Group working under Australia's Attorney-General Department (2013) produced a definition to cybercrime which describes cybercrime in two aspects:

- Crimes directed at computers or other information communications technologies (ICTs) (such as hacking and denial of service attacks), and
- Crimes where computers or ICTs are an integral part of an offence (such as online fraud, identity theft and the distribution of child exploitation material).

The first point describes crime that is targeted directly at networked and computer environments which cloud computing infrastructure is inherently based on. The second one describes the commission of traditional physical crimes utilizing the available technology today which indicates illegal usage of cloud computing resources can very well be classified as cybercrime.

These types of activities over the past few years have escalated and we can show how business has suffered. In 2013, for example, the InfoSec Institute (Paganini 2013) gathered existing research on the costs of cybercrime and showed that this was rising. In the United States alone, each cybercrime incident costs on average $12 million which was up over 78% from 4 years ago.

Verizon (2015) published a report on data breaches from cybercrime episodes and the underlying cause and overall cost of such data breaches. They came up with a cost per record model, which indicated how much a set of records stolen from organizations could be used and monetized for fraud. The cost goes up from as low as $392,000 to $200 million with an expected average of $8.8 million per 100 million records lost through data breaches.

For small business, the impact of cybercrime can also be felt. Industry security organizations like TrendLabs (2015) reported that even the smallest of business can fall prey to cybercriminals simply because these businesses also store information that these criminals need and want. Personal identification information such as addresses, social security numbers and banking account numbers and security PINs and credit card numbers are still stored by these small business and are an attractive target for determined attackers. With a larger percentage of small business moving to the cloud, cybercriminals are sure to follow suit.

A survey performed by the consulting firm PricewaterhouseCoopers (PwC) in cooperation with the United States Computer Emergency Response Team (US-CERT) and the US Secret Service in 2014 on US business reported that 34% of respondents showed an increase in cyber security incidents year on year (PWC 2014). Self reported losses were approximately $415,000 on average and these were caused by cybercrime activities such as malware, phishing, network interruption, spyware, and denial of service attacks. In many cases, businesses point the blame at attackers from outside the organization. However, there was still a large percentage that pointed to malicious insiders as the cause of a number of security incidents.

In Australia, the problem of cyber security is not just confined to large business but small businesses are especially vulnerable. This is not surprising, as small businesses may find the use of cloud computing particularly cost effective and allow them to compete with larger enterprises on a level playing field. The use of cloud computing too has become the focus of a determined criminal element. While there have been fewer attacks against CSPs, there are still risks associated with the use of cloud computing, such as the following:

- Authentication Issues that could be exploited to allow access to data by unauthorized personnel (Abdollahifar 2013).
- Denial of Service (DoS) or Distributed Denial of Service (DDoS) attacks that will cause clients to lose access to required services (Archer and Boehm 2009).
- The coopting of cloud services for criminal activities such as utilizing cloud resources to run malware and botnet networks (Osanaiye et al. 2016; Ouedraogo and Mouratidis 2013).
- CSPs complicit in criminal activity such as allowing the storage of copyrighted material (Duncan et al. 2012).
- Physical attacks on data centers containing cloud computing infrastructure and other insider activities can cause data breaches are lead to illegal remote access to data (Greenberg et al. 2008).
- Data stored in the cloud could also be vulnerable through vulnerabilities, say in software components (e.g. flawed implementation of encryption), or external attacks via phishing and man in the middle attacks (Hooper et al. 2013).
- Other attacks also involve skipping attacks on the cloud infrastructure itself and targets the client devices used to access the network resource such as compromising the client access via key loggers or web session hijacking (Ghorbani et al. 2010), or seeking to circumvent security solutions (e.g. SSL/TLS validations) on client devices (D'Orazio and Choo 2017; D'Orazio et al. 2017).

4 Cybercrime and Situational Crime Prevention

More and more business are looking to use cloud computing to gain competitive advantages and to cut costs in delivering their own services. Consumers are also in the same path to using cloud computing resources for different reasons, such as ease of use and the availability of computing resources for a fraction of the cost of buying it themselves. Because of this, cyber criminals are attempting to exploit cloud computing weaknesses to reach their targets, which can be for financial gains, to gain competitive advantages, for national security related matters (e.g. by state-sponsored actors), or illegal content (Paganini 2013).

Examples of attacks on cloud based services include:

1. DDoS based attacks against cloud services such as the Xbox and Playstation networks (Sawers 2015).
2. DNS based attacks that caused access problems to IaaS provider Rackspace (O'Connor 2014).
3. Hijacking of existing IaaS servers provided by Amazon for BitCoin mining processes (Litke and Stewart 2014).

Reports suggest that the number of incidences of cybercrime, especially against cloud services is poised to rise (PWC 2014). More recently in 2017, Microsoft (2017) reported that:

> *the frequency and sophistication of attacks on cloud-based accounts are accelerating. The Identity Security and Protection team has seen a 300 percent increase in user accounts attacked over the past year. A large majority of these compromises are the result of weak, guessable passwords and poor password management, followed by targeted phishing attacks and breaches of third-party services* (Microsoft 2017, p. 3)

As such, it would be beneficial to understand how these crimes can occur. One way to begin understanding the roots of these types of crimes is to use crime science. Crime science seeks to explain how crime transpires (Hartel et al. 2010). This is somewhat different from criminology that seeks to frame the crime in terms of the criminal's behaviors and motivations. Crime science utilizes conceptual frameworks to explain the actual incidents of crime, the "how" if you will, and not the actual criminal; or the "who" and "why". In this manner, crime science is a problem solving approach to crime that is outcome specific.

One such approach in use in crime science is the crime opportunity theory which, at its core, suggests that opportunity "plays a role in all crime" (Felson and Clarke 1998). Felson and Clarke's (1998) research was the basis of formulating this approach, which was a departure from the prevailing theories of crime at the time which focused on the reducing criminal propensities instead of reducing opportunities for crime.

There are three main aspects to the crime opportunity theory. The first is called the Routine Activity Theory (RAT), which tries to frame that crime is more likely to occur when there is the occurrence of three aspects which are a possible criminal or offender, a likely or highly valued target and the absence of a capable guardian against the crime to act.

The second is the Crime Pattern Theory (CPT), which makes the case for how criminals and targets are located at any given time. Using three main concepts, namely: nodes, paths and edges, this theory attempts to describe crime in a way that suggests movements.

- Nodes designate where people or targets come and go.
- Paths between nodes indicate how and where people travel.
- Edges refer to areas where people congregated to live, work or enjoy recreation.

CPT uses these concepts to map out likely incidences of crime within these areas.

The third is the Rational Choice Theory and it introduces the concept that criminals make very specific and, to their own mind, rational and logical choices when enacting a crime. This theory tries to frame how a criminal offender makes choices when enacting short-term criminal goals and in so doing, tries to understand how the criminal choice occurs within a particular motive and specific opportunities.

Taken together, these aspects frame crime as occurring in many levels. From the larger level of society; which the Routine Activity describes, to the local area which CPT maps out; to the individual criminal, whose actions are governed by rational choice. Ensuring that opportunities for crime are reduced at all these levels changes the instances of crime.

These theories then are used to structure a preventative approach to crime – situational crime prevention (Clarke 1997) as well as a refinement to this initial approach (Cornish and Clarke 2003). This approach to crime prevention introduces specific changes in the management and environment where a crime occurs. Clarke (1997) describes situational prevention to include three aspects to reduce the opportunities for crime. That it is directed at highly specific forms of crime, involve the management of the environment of the crime and making the commission of the crime itself less rewarding and more risky and difficult as well as less excusable for offenders.

Clarke then showed SCP procedures and case studies in which such procedures were utilized and explained that "[s]ituational crime prevention then involves the development of techniques to prevent, constrain or disrupt criminal activity" (Clarke 1997).

In another approach to crime prevention, Cornish (1994) used concepts from the Rational Choice Theory to propose an approach to preventing crime through disrupting an offenders approach to crime, their "crime script" and ensuring that this natural flow of the crime is interrupted at various points. He posited that crime follows a series of steps or that criminals follow a "script" of some kind such that crimes occur according to this script. If this crime script is disrupted at any point, then there is to be the expected change in the behavior and the prevention of the crime itself. This was demonstrated in research performed by Smith (2014) that showed how disrupting the natural flow of a recruitment process for criminal organizations aids in the minimization of criminal behavior.

Using concepts of RAT, it is also suggested that modifying any of the three aspects (motivated offender, suitable target, and capable guardianship) can prevent crime. Hollis-Peel and Welsh (2014) tested this theory to show how guardianship can be measured and used to prevent crime in maintaining home security.

Utilizing the crime opportunity theories and SCP, various research has been made to adapt these theories to cybercrime in general and specific cybercrime in particular.

Although created and developed primarily for physical crimes, the crime opportunity theories and their associated aspects have been adapted to understand the incidence of cybercrime.

When it comes to cyber security and the mitigation of threats against computing infrastructure, so called cyber threats, solutions have either come from a technological or a process or policy specific area. These security solutions concentrate on identifying and mitigating these cyber threats through employing security controls such as employing new processes (Goodman et al. 2008), or using technology to mitigate identified weaknesses in infrastructure (Christie 2011). A holistic way to prevent cybercrime specifically or crime in general is to employ SCP models.

SCP is a technique that has emerged from the crime opportunity theories co-developed by Clarke (1983). Clarke says that in order to reduce crime, there must be changes to the environment of a crime to reduce the opportunities for a crime to occur and that “the pivotal point of situational crime prevention theory is that the criminal’s pseudo-rational decision is a function of the perceived net benefits. If crime prevention measures do not adequately increase perceived costs and decreased perceived benefits, rational choice theory argues that the crime will not occur” (Clarke 1997).

Some research suggests that cybercrime is a different category of crime and that these crimes cannot be easily explained by the prevailing crime theories (Yar 2005). However, other research, such as from Beebe and Rao (2005), have taken the SCP theories and extend them to apply to the growing problem of information system security. In their research, they used SCP techniques and suggested a theoretical model that can be applied to an online environment (Beebe and Rao 2005). They suggested that in the model to look at the reduction of anticipated benefits for cybercriminals in engaging in cybercrime although they did not offer any concrete steps or activities to reducing these benefits.

Other aspects of crime opportunity theory have also been used to map into specific cases of cybercrime. Pratt et al. (2010) showed how RAT can be used to model the incidence of internet fraud. He showed that the change in consumers’ behavior, especially in the use of online shopping, exposes them to motivated offenders attempting to perpetrate consumer fraud. As such, SCP plays a role in understanding that will be more likely to be targeted in cases of Internet fraud and what they can do to protect themselves. Pratt’s research showed that RAT could be used as a general framework in so much as preventing a very specific case for cybercriminal victimization.

Leukfeldt (2014) did a similar study on phishing attempts which came up with some differing results. Using RAT as a basis, the study attempted to find out if phishing victimization rates were higher in any particular demographic (target) but showed that increasing capable guardianship via target hardening may help. Overall the study showed little effect of changing the circumstances for RAT based approach to crime prevention and suggested that other aspects of SCP should be used.

Although research has been done on the effects of cybercrime on individuals, RAT has also been used to show how highly connected countries have a higher incidence of cybercrime, specifically the incidence of spam and phishing attempts (Kigerl 2012).

This goes to show that more the opportunities for crime are higher in more connected countries.

Navarro (2013) used RAT to map out another aspect of cybercrime, that of cyber bullying or harassment. It showed how RAT fits into explaining the incidences of cyber bullying victimization when a lack of capable guardianship, one of the three conditions for crime to occur, according to the RAT, is present in the online activities of teenagers. Their results at applying the crime opportunity framework were mixed at best because of the way their research was structured around gender lines.

Hinduja and Kooi (2013) utilized general aspects of SCP to address a framework for reducing information security vulnerabilities. They posited that technological solutions are not enough to address vulnerabilities in information systems and that SCP can be used to combat the more opportunistic elements of cybercrime. In their paper, they only considered using Clarke's original 16 opportunity reducing techniques instead of the latter 25 primarily because they determine some of the newer techniques not to be relevant to information security, such as the reduction of provocations due to drugs and alcohol (Hinduja and Kooi 2013).

Similarly, Willison and Backhouse (2006) extended the concepts of SCP to increasing IS security, utilizing the same methods and additional crime theory frameworks. The authors combined the crime script theory with classic SCP techniques and mapped it into potential IS security policies or activities, and specifically on the common cyber security concept of insider threats and how SCP in coordination with IS security policies can be used to mitigate these threats. Willison (2000) also put forward his conceptual Crime Specific Opportunity Structure as a means to understand the circumstances behind information systems risk and to help elaborate the relationships that offenders have with the environment of the crime.

After reviewing the available research on crime opportunity theory and its subsequent SCP applications, it is clear that SCP can be applied to the protection of many aspects of information systems. Although there are SCP techniques for specific threats such as fraud (Samonas 2013) and malicious insiders (Stockman 2014), there does not appear to be an overarching framework that can be applied to protecting cloud computing infrastructure and services. This is a challenge because SCP, according to Clarke (1997), applies to very specific forms of crime and actions against cloud computing can be varied and wide ranging. This suggests different techniques for every threat against cloud computing.

Existing information security frameworks have tended to work on two levels (i.e. technological and process) to mitigate threats. Examples of technological framework solutions include those of Takabi et al. (2010) and Brock and Goscinski (2010), where they break down the cloud infrastructure architecture components and introduce conceptual modules that contain technological solutions to protect each component. In one example, interfaces between cloud users and CSPs can be protected via an access control module that can conceptually contain Role-Based Access (RBAC) control models, such as those proposed by Alam et al. (2017).

The Australian Signals Directorate (2017) has also published specific strategies to combat cybercrime that can be implemented by cloud consumers and CSPs. Strategies such as regular software patching and restricting administrator privileges are practical activities that can be implemented to protect against cyber intrusions. These technology strategies can be made to apply generically across computing infrastructure including cloud computing platforms.

Cyber security can also be driven from a policy perspective as well. NIST has published standards that seek to increase the overall security of information systems through policy controls that map out key information security factors such as security governance, systems development lifecycle, systems acquisitions, systems interconnections, ongoing performance metrics, planning and security incident responses among others (Bowen et al. 2006; NIST 2017; Stoneburner et al. 2001). The International Standards Organization (ISO) has also published information security standards, which is a series of security controls that organizations can implement to increase security. Foremost of these controls is the establishment of security policy. The security policy is meant to contain the organization's security goals and what roles and responsibilities each member of the organization is meant to have. Having an established security policy is the first step to ensuring that information assets are protected, risks are minimized and that organizations are compliant to regulations (AS/NZS 2006).

Given the relationship of cloud computing platforms and the digital information contained therein, the application of information security policies onto cloud computing implementations can be of great benefit and some research has already been undertaken to look at increasing security through this method. For example, Carrol et al. (2011) took the approach of applying specific security policies to a cloud computing scenario where certain elements of the NIST and ISO standards have been used to mitigate risks the authors have identified as applicable to cloud computing services.

After the technological and policy driven solutions for cloud security, we can introduce the third concept of SCP. There has been research that shows SCP to be potentially useful in protecting against specific cybercrime instances. However, there is a possibility that the application for SCP techniques can be used for protection of cloud computing systems.

Choo (2014) has put forward a conceptual framework to mitigate cyber security threats with the use of different crime theories and crime prevention strategies can be "plug and played" into so that a wider ranging plan can be enacted. The framework proposed is to be applicable to a generalized information security model and utilizes various disciplines in order to consolidate the relationships between different objects or actors in an information security context. This means that because there are various objects in play in a cyber security activity (e.g. people, process, and technology), and a recognition that there are different schools of thought as to how and why these incidents occur. It is of use to have a framework where various theories can be used to understand the existing environment and therefore mitigate the ensuing cyber security risks. The author then uses as an example using SCP techniques to understand the cyber security environment and to look at modifying the environment to combat crime.

So far, there has been a number of technological and policy driven approaches to increasing information security. There has also been support for SCP approaches for very specific cybercrime instances. Thus, we posit that using SCP techniques that apply and involve increasing perceived effort and risks of committing the crime, reducing rewards and provocations and removing excuses for the criminals could lead to a reduction in criminal actions against cloud computing services. Specifically, we present a conceptual framework of mitigation of cloud security threats utilizing a combination of these approaches.

The next section will describe this conceptual framework and how this can be applied to cloud computing services.

5 A Conceptual SCP Model for Cloud Security

There are a variety of solutions intended to tackle the problem of cybercrime and cyber security. Because several policy-based solutions are available from different standards bodies, it is important to choose one set of solutions to enact. Controls from the AS/NZS ISO/IEC Standard 27002:2006 (Security Techniques) as the basis for the initial policy solutions, as this provides us with a comprehensive security solution that can be applied to a wide number of information systems including cloud computing platforms.

There are a number of controls available within the ISO Standard; therefore, it is necessary to create designations for each control that could be applied to a cloud computing solution (see Table 1).

Table 1. Policy controls solutions

Security control	Designation
Establishment of a security policy	P1
Organization of information security	P2
Asset management	P3
Human resources security management	P4
Physical and environmental security	P5
Communications and operations management	P6
Establish operational procedures and responsibilities	P7
Third-party service delivery management	P9
Malicious/mobile code	P10
Backup	P11
Network security management	P12
Media handling	P13
Information exchange	P14
Electronic commerce service	P15
Monitoring	P16
Access control	P17
Information systems acquisition, development and maintenance	P18
Information security incident management	P19
Business continuity management	P20
Compliance management	P21

These policy and technology based solutions can be applied to specific cloud threats in the proposed conceptual model. If the conceptual model is implemented by a CSP, then it may be possible to measure the effectiveness of the changes in mitigating these threats to the environment.

According to Clarke (1997), SCP consists of three different measures:

1. Measures directed at highly specific forms of crime.
2. Involves changes to the environment where crime occurs.
3. Strives to make crime less rewarding, more risky and less excusable for offenders.

Because of these measures, it is important for the conceptual model to take into account the risks associated with a cloud computing infrastructure and to modify this particular environment so that criminal activity is discouraged. This model then starts with industry identified threats and vulnerabilities against cloud computing as a source of crime that can be countered or mitigated.

SCP techniques are then applied to each of these threats in order to lower the risk and incidence of these vulnerabilities being exploited, to alter the environment in a way.

With each set of techniques, both technology based and policy based solutions are then invoked to enable the changes in the cloud computing environment. These changes, as directed by SCP, will then be expected to discourage the commission of these crimes. The expected result in turn could lead to increased security for the entire cloud computing environment.

To identify the cloud computing threats that this framework can be applied this paper turns to the major risks identified by the industry via the Cloud Security Alliance, which published their top nine threats to cloud computing (Cloud Security Alliance 2016). Utilizing the top threats in this conceptual model of cloud security and to target each as a specific form of crime, solutions can then be applied so as to change the cloud computing environment and decrease the overall opportunities for crime. Technology and process solutions from standards organizations and governmental authorities such as the ISO 27002 standard and the Australian Signals Directorate (2017) – see also Table 2 – can then be used to change the environment for each threat as a means of mitigation.

Each solution is expected to enact changes in the environment to increase effort, increase risk, decrease reward, remove provocations and remove excuses within the environment. This relates back to the SCP strategies for reducing crime.

The solutions that may be applied could be very specific to the cloud computing components, such as the virtualization infrastructure, or it could be generalized as well, such as to the entire cloud computing organization depending on the set of remediation strategies to use. In Table 3, we can see potential solutions being assigned to cloud threats. The targeted outcomes should still be the same: a change of the environment to lower crime opportunities.

Utilizing this model, we can then start to build the framework by combining various mitigation solutions across the different threats and understanding their effect on modifying the cloud computing environment.

Table 2. ASD mitigation strategies

Mitigation strategy	Designation
Application whitelisting of permitted/trusted programs	A1
Patch applications e.g. Java, PDF viewer, Flash, web browsers and Microsoft Office	A2
Patch operating system vulnerabilities	A3
Restrict administrative privileges to operating systems and applications based on user duties	A4
User application configuration hardening and disabling of vulnerable plugins	A5
Automated dynamic analysis of email and web content run in a sandbox to detect suspicious behavior	A6
Operating system generic exploit mitigation e.g. Data Execution Prevention (DEP), Address Space Layout Randomization (ASLR) and Enhanced Mitigation Experience Toolkit (EMET)	A7
Host-based intrusion detection/prevention system to identify anomalous behavior during program execution	A8
Disable local administrator accounts	A9
Network segmentation and segregation into security zones	A10
Multi-factor authentication especially implemented for remote access, privileged actions or sensitive information access	A11
Software-based application firewall, blocking incoming network traffic	A12
Software-based application firewall, blocking outgoing network traffic	A13
Non-persistent virtualized sandboxed trusted operating environment	A14
Centralized and time-synchronized logging of computer events	A15
Centralized and time-synchronized logging of allowed and blocked network activity	A16
Email content filtering, allowing only whitelisted business related attachment types	A17
Web content filtering of incoming and outgoing traffic	A18
Web domain whitelisting for all domains	A19
Block spoofed emails using Sender ID or Sender Policy Framework (SPF)	A20
Workstation and server configuration management based on a hardened standard operating environment	A21
Antivirus software using heuristics and automated Internet-based reputation ratings	A22
Deny direct Internet access from workstations by using an IPv6-capable firewall to force traffic through a split DNS server, an email server, or web proxy server	A23
Server application configuration hardening	A24
Enforce a strong passphrase policy	A25
Removable and portable media control as part of a data loss prevention strategy	A26
Restrict access to Server Message Block (SMB) and NetBIOS services	A27
User education	A28
Workstation inspection of Microsoft Office files for malicious abnormalities	A29
Signature-based antivirus software that primarily relies on up to date signatures to identify malware	A30
TLS encryption between email servers and perform content scanning after email traffic is decrypted	A31
Block attempts to access websites by their IP address instead of by their domain	A32
Network-based intrusion detection/prevention system using signatures and heuristics	A33
Gateway blacklisting to block access to known malicious domains and IP addresses	A34
Capture network traffic to/from internal critical asset workstations and servers	A35

Table 3. Mapping of cloud threats and mitigation solutions with the SCP-based model

Cloud threat	Increase effort	Increase risk	Decrease reward	Remove provocations	Remove excuses
Data breaches	*P5, A2, A4*	*P4, P16, A33*	*P3, A14, A24*	*P1, P21, A28*	*P1, A1, A28*
Data loss	…	…	…	…	…
Account or service traffic hijacking	…	…	…	…	…
Insecure interfaces	…	…	…	…	…
Denial of service	…	…	…	…	…
Malicious insiders	…	…	…	…	…
Abuse of cloud services	…	…	…	…	…
Insufficient due diligence	…	…	…	…	…
Shared technology vulnerabilities	…	…	…	…	…

There is potential future work and research to develop this framework further by applying it to an existing CSP and observing the changes that this framework brings to the environment in terms of lowering crime opportunities.

References

Ab Rahman, N.H., Choo, K.-K.R.: A survey of information security incident handling in the cloud. Comput. Secur. **49**, 45–69 (2015)

Abdollahifar, A.: Network and Security Challenges in Cloud Computing Infrastructure as a Service Model (2013)

Alam, Q., Malik, S.U.R., Akhunzada, A., Choo, K.-K.R., Tabbasum, S., Alam, M.: A cross tenant access control (CTAC) model for cloud computing: formal specification and verification. IEEE Trans. Inf. Forensics Secur. **12**(6), 1259–1268 (2017)

Antonopoulos, N., Gillam, L.: Cloud Computing: Principles, Systems and Applications. Springer, Heidelberg (2010). https://doi.org/10.1007/978-1-84996-241-4

Archer, J., Boehm, A.: Security guidance for critical areas of focus in cloud computing. Cloud Security Alliance (2009)

AS/NZS: ISO/IEC 27002:2006 - Information Technology - Security Techniques - Code of Practice for Information Security Management (2006)

Attorney-General's Department: National Plan to Combat Cybercrime. Attorney-General's Department, Canberra, ACT, Australia (2013)

Australian Signals Directorate: Strategies to Mitigate Cyber Security Incidents. Australian Department of Defense, Canberra (2017)

Beebe, N.L., Rao, V.S.: Using situational crime prevention theory to explain the effectiveness of information systems security. In: Proceedings of the 2005 Software Conference, Las Vegas (2005)

Bowen, P., Hash, J., Wilson, M.: SP 800-100. Information Security Handbook: A Guide for Managers (2006). https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-100.pdf. Accessed 26 Mar 2018

Brock, M., Goscinski, A.: Toward a framework for cloud security. In: Hsu, C.-H., Yang, Laurence T., Park, J.H., Yeo, S.-S. (eds.) ICA3PP 2010. LNCS, vol. 6082, pp. 254–263. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13136-3_26

Carroll, M., Van Der Merwe, A., Kotze, P.: Secure cloud computing: benefits, risks and controls. In: Information Security South Africa (ISSA), pp. 1–9. IEEE (2011)

Choo, K.-K.R.: Cloud Computing Challenges and Future Directions. Australian Institute of Criminology, Canberra (2010)

Choo, K.-K.R.: A conceptual interdisciplinary plug-and-play cyber security framework. In: Kaur, H., Tao, X. (eds.) ICTs and the Millennium Development Goals, pp. 81–99. Springer, Boston (2014). https://doi.org/10.1007/978-1-4899-7439-6_6

Christie, S.: 2011 CWE/SANS Top 25 Most Dangerous Software Errors (2011). http://cwe.mitre.org/top25/. Accessed 5 Sept 2013

Clarke, R.: Situational Crime Prevention. Criminal Justice Press, Monsey (1997)

Clarke, R.V.: Situational crime prevention: its theoretical basis and practical scope. Crime Justice **4**, 225–256 (1983)

Cloud Security Alliance: ‘The Treacherous Twelve’ Cloud Computing Top Threats in 2016. Cloud Security Alliance (2016)

Cornish, D.B.: The procedural analysis of offending and its relevance for situational prevention. Crime Prev. Stud. **3**, 151–196 (1994)

Cornish, D.B., Clarke, R.V.: Opportunities, precipitators and criminal decisions: a reply to Wortley’s critique of situational crime prevention. Crime Prev. Stud. **16**, 41–96 (2003)

D’Orazio, C.J., Choo, K.-K.R.: A technique to circumvent SSL/TLS validations on iOS devices. Future Gener. Comput. Syst. **74**, 366–374 (2017)

D’Orazio, C.J., Choo, K.-K.R., Yang, L.T.: Data exfiltration from internet of things devices: iOS devices as case studies. IEEE Internet Things J. **4**(2), 524–535 (2017)

Dawoud, W., Takouna, I., Meinel, C.: Infrastructure as a service security: challenges and solutions. In: 2010 The 7th International Conference on Informatics and Systems (INFOS), pp. 1–8. IEEE (2010)

Duncan, A.J., Creese, S., Goldsmith, M.: Insider attacks in cloud computing. In: 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 857–862. IEEE (2012)

Felson, M., Clarke, R.V.G.: Opportunity Makes the Thief: Practical Theory for Crime Prevention. Home Office, Policing and Reducing Crime Unit, Research, Development and Statistics Directorate, London (1998)

Ghorbani, A.A., Lu, W., Tavallaee, M.: Network Attacks, 1st edn. Springer, Boston (2010)

Goodman, S., Straub, D.W., Baskerville, R., Goodman, S.E., Ebrary, I.: Information Security: Policy, Processes and Practices. M. E. Sharpe Incorporated, Armonk (2008)

Greenberg, A., Hamilton, J., Maltz, D.A., Patel, P.: The cost of a cloud: research problems in data center networks. ACM SIGCOMM Comput. Commun. Rev. **39**(1), 68–73 (2008)

Gu, L., Zeng, D., Guo, D.: Vehicular cloud computing: a survey. In: IEEE Globecom Workshops, pp. 403–407 (2013)

Hartel, P., Junger, M., Wieringa, R.: Cyber-crime Science = Crime Science + Information Security (2010). https://research.utwente.nl/en/publications/cyber-crime-science-crime-science-information-security. Accessed 29 Aug 2017

Hiller, J.S., Russell, R.S.: The challenge and imperative of private sector cybersecurity: an international comparison. Comput. Law Secur. Rev. **29**(3), 236–245 (2013)

Hinduja, S., Kooi, B.: Curtailing cyber and information security vulnerabilities through situational crime prevention. Secur. J. **26**(4), 383–402 (2013)

Hollis-Peel, M.E., Welsh, B.C.: What makes a guardian capable? A test of guardianship in action. Secur. J. **27**(3), 320–337 (2014)

Hooper, C., Martini, B., Choo, K.-K.R.: Cloud computing and its implications for cybercrime investigations in Australia. Comput. Law Secur. Rev. **29**(2), 152–163 (2013)

Hunton, P.: The stages of cybercrime investigations: bridging the gap between technology examination and law enforcement investigation. Comput. Law Secur. Rev. **27**(1), 61–67 (2011)

Iqbal, S., Kiah, M.L.M., Dhaghighi, B., Hussain, M., Khan, S., Khan, M.K., Choo, K.-K.R.: On cloud security attacks: a taxonomy and intrusion detection and prevention as a service. J. Netw. Comput. Appl. **74**, 98–120 (2016)

Julidotter, N., Choo, K.-K.R.: CATRA: conceptual cloud attack taxonomy and risk assessment framework. In: Ko, R., Choo, K-K.R. (ed.) Cloud Security Ecosystem. Syngress, an Imprint of Elsevier, Amsterdam (2015)

Khan, M.F., Ullah, M.A., Aziz-Ur-Rehman: An approach towards customized multi-tenancy. Int. J. Mod. Educ. Comput. Sci. **4**(9), 39 (2012)

Kigerl, A.: Routine activity theory and the determinants of high cybercrime countries. Soc. Sci. Comput. Rev. **30**(4), 470–486 (2012)

Leukfeldt, E.R.: Phishing for suitable targets in the Netherlands: routine activity theory and phishing victimization. Cyberpsychology Behav. Soc. Netw. **17**(8), 551–555 (2014)

Litke, P., Stewart, J.: BGP Hijacking for Cryptocurrency Profit (2014). http://www.secureworks.com/cyber-threat-intelligence/threats/bgp-hijacking-for-cryptocurrency-profit/

Lokhande, M.T.S., Shelke, P.R.R.: A review paper on cloud computing security. Int. J. Adv. Res. Comput. Sci. **4**(6), 70 (2013)

Mell, P., Grance, T.: The NIST Definition of Cloud Computing (2011). http://dx.doi.org/10.6028/NIST.SP.800-145. Accessed 29 Aug 2017

Microsoft 2017: Microsoft Security Intelligence Report, vol. 22, January–March 2017. http://download.microsoft.com/download/F/C/4/FC41DE26-E641-4A20-AE5B-E38A28368433/Security_Intelligence_Report_Volume_22.pdf. Accessed 29 Aug 2017

National Institute of Standards and Technology (NIST): Security and Privacy Controls for Information Systems and Organizations (2017). http://csrc.nist.gov/publications/drafts/800-53/sp800-53r5-draft.pdf. Accessed 29 Aug 2017

Navarro, J.N., Jasinski, J.L.: Why girls? Using routine activities theory to predict cyberbullying experiences between girls and boys. Women Crim. Justice **23**(4), 286–303 (2013)

O'Connor, F.: Rackspace DNS Recovers After DDoS Brings System Down. In: PCWorld (2014). http://www.pcworld.com/article/2863592/rackspace-dns-recovers-after-ddos-brings-system-down.html. Accessed 29 Aug 2017

Osanaiye, O., Choo, K.-K.R., Dlodlo, M.: Distributed denial of service (DDoS) resilience in cloud: review and conceptual cloud ddos mitigation framework. J. Netw. Comput. Appl. **67**, 147–165 (2016)

Ouedraogo, M., Mouratidis, H.: Selecting a cloud service provider in the age of cybercrime. Comput. Secur. **38**, 3–13 (2013)

Paganini, P.: 2013 - The Impact of Cybercrime (2013). http://resources.infosecinstitute.com/2013-impact-cybercrime/. Accessed 29 Aug 2017

Pratt, T.C., Holtfreter, K., Reisig, M.D.: Routine online activity and internet fraud targeting: extending the generality of routine activity theory. J. Res. Crime Delinq. **47**(3), 267–296 (2010)

Poh, G.S., Chin, J.J., Yau, W.C., Choo, K.-K.R., Mohamad, M.S.: Searchable symmetric encryption: designs and challenges. ACM Comput. Surv. **50**(3), 1–37 (2017). Article 40

PWC: US Cybercrime: Rising Risks, Reduced Readiness. Key Findings from the 2014 US State of Cybercrime Survey (2014). http://www.pwc.com/us/en/increasing-it-effectiveness/publications/assets/2014-us-state-of-cybercrime.pdf. Accessed 29 Aug 2017

Quick, D., Martini, B., Choo, K.-K.R.: Cloud Storage Forensics. Syngress, an Imprint of Elseiver, Amsterdam (2013)

Rogers, A.: From Peer-to-Peer Networks to cloud Computing: How Technology is Redefining Child Pornography Laws (2012). Available at SSRN 2006664

Samonas, S.: Insider Fraud and Routine Activity Theory (2013). http://eprints.lse.ac.uk/50344/. Accessed 29 Aug 2017

Sawers, P.: Playstation Network and Xbox Live Ddos Arrest: U.K. Authorities Grab an 18-Year-Old Man. Venture Beat (2015)

Smith, R.G.: Responding to organised crime through intervention in recruitment pathways. Trends Issues Crime Crim. Justice **473**, 1–6 (2014)

Stockman, M.: Insider hacking: applying situational crime prevention to a new white-collar crime. In: RIIT Proceedings of the 3rd Annual Conference on Research in Information Technology, pp. 53–56 (2014)

Stoneburner, G., Hayden, C., Feringa, A.: Engineering Principles for Information Technology Security (a Baseline for Achieving Security). DTIC Document (2001)

Takabi, H., Joshi, J.B., Ahn, G.J.: Securecloud: towards a comprehensive security framework for cloud computing environments. In: IEEE 34th Annual Computer Software and Applications Conference Workshops (COMPSACW), pp. 393–398. IEEE (2010)

TrendLabs: Small Business is Big Business in Cybercrime (2015). https://www.trendmicro.de/cloud-content/us/pdfs/internet-safety/tlp_small-business-big-for-cybercrime.pdf. Accessed 29 Aug 2017

Varadharajan, V., Tupakula, U.: Security as a service model for cloud environment. IEEE Trans. Netw. Serv. Manag. **11**(1), 60–75 (2014)

Verizon: Verizon 2015 Data Breach Investigations Report. Verizon Enterprise Solutions (2015). http://www.verizonenterprise.com/resources/reports/rp_data-breach-investigation-report_2015_en_xg.pdf. Accessed 29 Aug 2017

Vidal, C., Choo, K.-K.R.: The current state of an IaaS provider. In: Ko, R., Choo, K.-K.R. (eds.) The Cloud Security Ecosystem: Technical, Legal, Business and Management Issues, pp. 401–426. Syngress, Boston (2015)

Willison, R.: Understanding and addressing criminal opportunity: the application of situational crime prevention to is security. J. Financ. Crime **7**(3), 201–210 (2000)

Willison, R., Backhouse, J.: Opportunities for computer crime: considering systems risk from a criminological perspective. Eur. J. Inf. Syst. **15**(4), 403–414 (2006)

Yar, M.: The novelty of ‘Cybercrime’ an assessment in light of routine activity theory. Eur. J. Criminol. **2**(4), 407–427 (2005)

$SoNeUCON_{ABC}Pro$: An Access Control Model for Social Networks with Translucent User Provenance

Lorena González-Manzano[1](✉), Mark Slaymaker[2], Jose M. de Fuentes[1], and Dimitris Vayenas[3]

[1] Universidad Carlos III de Madrid, Leganés, Spain
{lgmanzan,jfuentes}@inf.uc3m.es
[2] The Open University, Walton Hall, Milton Keynes, UK
mark.slaymaker@open.ac.uk
[3] Oxford University Computing Laboratory, Oxford, UK
dimitris.vayenas@exeter.ox.ac.uk

Abstract. Web-Based Social Networks (WBSNs) are used by millions of people worldwide. While WBSNs provide many benefits, privacy preservation is a concern. The management of access control can help to assure data is accessed by authorized users. However, it is critical to provide sufficient flexibility so that a rich set of conditions may be imposed by users. In this paper we coin the term *user provenance* to refer to tracing users actions to supplement the authorisation decision when users request access. For example restricting access to a particular photograph to those which have "liked" the owners profile. However, such a tracing of actions has the potential to impact the privacy of users requesting access. To mitigate this potential privacy loss the concept of *translucency* is applied. This paper extends $SoNeUCON_{ABC}$ model and presents $SoNeUCON_{ABC}Pro$, an access control model which includes translucent user provenance. Entities and access control policies along with their enforcement procedure are formally defined. The evaluation demonstrates that the system satisfies the imposed goals and supports the feasibility of this model in different scenarios.

Keywords: Social networks · Access control · User provenance Translucency

1 Introduction

The continuing proliferation of Web-Based Social Networks (WBSNs) encourages their study and research. This escalation raises questions about security and privacy due to the amount of managed personal data being shared. For instance, each minute around 2.5 million items are shared on Facebook and 200,000 photos are uploaded to Instagram[1]. Facebook has increased the amount of privacy

[1] http://aci.info/2014/07/12/the-data-explosion-in-2014-minute-by-minute-infographic/.

X. Lin et al. (Eds.): SecureComm 2017 Workshops, LNICST 239, pp. 234–252, 2018.
https://doi.org/10.1007/978-3-319-78816-6_17

controls, enabling users to restrict the content that is viewable by others. Thus, when a user writes a message or adds a friend, privacy controls associated with that content will dictate what is viewable by his friends[2].

Access control has been a challenging matter [5,7]. It is considered such an important thing that [27] considers that the management of who accesses data should be a requirement whatever the cost. An important aspect is to provide an access control mechanism that is both flexible and fine-grained. There has been previous work on *data provenance*, defined as the process of tracing and recording the origin of data and any subsequent change [3,24]. Based on this concept, we coin the term *user provenance* to refer to the process of tracing users' actions, and using that information as a basis for decisions related to granting access. User provenance would make it possible to include additional constrains on users requesting access based on, for instance, where the user comes from or the actions that the user has previously performed.

There are several different user actions in WBSNs, i.e. the addition of comments, the uploading of photos, etc. User provenance could offer interesting access control management alternatives in this respect. The following set of paradigmatic scenarios motivates the development of an access control model that addresses user provenance.

- **Customer acquisition.** Parker's, a well-known restaurant, wants to implement an aggressive marketing campaign to steal clients of competitors. Thus, access to a special promotion is granted only to customers that have *visited* the Facebook profile of competing restaurants at least once in the last week.
- **Loyalty program.** Christian loves keeping up with the latest fashions as well as receiving feedback about his new clothes. He usually uploads photos of his new clothes to Facebook and users who *make comments* on them are allowed to access additional fashion photos he has posted. In this way, Christian limits the number of photos non-interested users can access while allowing interested ones to view a more extensive range of images.
- **Focused access.** Julia went to a Bon Jovi's concert and uploaded photos of the event to Facebook. To prevent Bon Jovi's detractors to post negative comments or create mocking memes based on these photos, she decided to grant access only to actual fans – users who have *liked* Bon Jovi's contents at least five times in the month.

According to these scenarios, the potential for privacy loss cannot be disregarded in the context of user provenance. Tracing user actions means that they are potentially transparent to the other users as these actions become part of the access control process. While tailored access control is desirable, transparency can directly affect privacy. An analogy can be drawn concerning glass-walled houses in which the clear glass walls makes it easy for anybody to look inside them. A potential method of limiting this affect is applying the concept of *translucency*, introduced by Mike Leiter [19], which can be used to balance transparency and privacy [29]. Using a smoked glass-walled home will still allow

[2] http://www.jonloomer.com/2012/05/06/history-of-facebook-changes/.

an onlooker to look inside but reducing the amount of details that can be ascertained. Analogously, integrating a translucency mechanism as part of the user provenance access control management is desirable. In this way, users control actions applied in the access control process and actions that should remain private.

To the best of our knowledge, no single access control model for WBSNs has been proposed enabling the expressiveness permitted by user provenance. $SoNeUCON_{ABC}$ [15] already considers the needs of flexibility, fine-granularity, attribute management and usage control, which are desirable access control properties. In this paper, an extension called $SoNeUCON_{ABC}Pro$ is proposed to address translucent user provenance. The behaviour of users is considered in the access control enforcement process through the management of performed actions but also considering the users right to keep some actions hidden.

The structure of the paper is as follows. Section 2 briefly introduces $SoNeUCON_{ABC}$. The proposed model is presented in Sect. 3. The definition and enforcement of access control policies are described in Sect. 4, with Sect. 5 providing an evaluation. An overview of other related work is described in Sect. 6. Finally, in Sect. 7 conclusions and future work are outlined.

2 Background

In this Section $SoNeUCON_{ABC}$ access control model [15] is introduced. $SoNeUCON_{ABC}$ is an expressive usage control model that manages six WBSN features, namely, common-contacts, clique, distance, multi-path, direction and flexible attributes [4,5,8,13].

$SoNeUCON_{ABC}$ is composed of seven elements: *Subjects* (S) together with *Subject attributes* ($ATT(S)$) refer to WBSN users and their attributes; *Objects* (O) together with *Object attributes* ($ATT(O)$) correspond to WBSN data and their attribute; and *Relationships* (RT) together with *Relationship attributes* ($ATT(RT)$) refer to the set of relations and attributes that exist between a pair of users, with direct relationships denoted as E and $ATT(E)$ their attached attributes; *Rights* (R) correspond to actions that can be performed over objects O; *Authorizations* (A) are rules to be satisfied to grant a subject a right on an object; *Obligations* (B) correspond to requirements to be met before or during the usage process; and *Conditions* (C) are requirements needed regarding the context features, eg. network availability.

In $SoNeUCON_{ABC}$, access control policies (ρ) consist of subjects, objects, relationships predicates (ρ_s, ρ_o and ρ_{rt} respectively), a right (r) is also provided as well as any obligations (∂_b) and conditions (∂_c) to be satisfied. An access control policy ρ is expressed as $\rho(\rho_s; \rho_o; \rho_{rt}; r; \partial_b; \partial_c)$.

An example of an access control policy is presented: *Access is granted to photos entitled "Party" to friends of a friend if they are under 30 years old or if they are under 25 years and have studied computer science.*

$\rho = (((age < 30) \vee ((age < 25) \wedge (studies = c.science))); (title = party); ((((role = friend); (role = friend))), \emptyset, \emptyset); read; \emptyset; \emptyset)$

Note that symbol $\emptyset$ is applied for policy elements which do not need to be involved in the access control management process. The first $\emptyset$ means that multiple paths are not managed, the second one that cliques between users are not considered and the final pair of $\emptyset$ means that conditions and obligations respectively are not included in this policy. See [15] for details.

3 $SoNeUCON_{ABC}Pro$ Proposal

This section outlines the main features of $SoNeUCON_{ABC}Pro$. For the ease of reading Table 1 presents main used notation.

Table 1. Notation table

ρ	Access control policy
ρ_t	Translucency policy
s_i	A subject i
o_i	An object i
rt_i	A relationship i
r	Right to be granted
∂_c	Conditions
χ	Obligations
ξ	User provenance pred.
∂_o	Obligation different from ξ
$P_{ac_{u_i}}$	Path, actions carried out by user u_i
u_i	User i
$ac_{j-u_i:o_k}$	Action j performed by u_i over o_k
e_i	Edge/relationship i

3.1 Goals

$SoNeUCON_{ABC}Pro$ should include the management of **user provenance**, facilitating access control management that is based on the actions performed by WBSN users (called requesters) over other users' data. The system must allow the definition of access control policies that consider previous actions of requesters. The system must also enable **translucency**, allowing requesters to hide some, or all of their, actions when an access control policy is evaluated.

3.2 Supporting Example

This example presents actions carried out by Daniel when he interacts in a WBSN with Alice, Bob and Charly. According to what is described here, access control has to be managed considering interactions performed by Daniel. Moreover, all restrictions, either performed by Daniel or by other user, as it is the case of Bob, have to be excluded in the access control process to respect users' privacy.

In a WBSN Daniel interacts with his direct friends Alice, Bob and Charly. Figure 1 *depicts the interactions that Daniel made between the 1st and 5th of June. On the 1st June Daniel added a like to photo1, photo2 and to the profile of Charly. In addition Daniel posted a comment on Alice's wall. Over the four days of activity covered by* Fig. 1, *Daniel performed a total of 11 actions on various elements of the profiles of his contacts. Moreover, Daniel wants to hide that he has clicked "like" in any of his friends' profiles, Alice's and Charly's profiles in this scenario.*

Additionally, Bob specifies that access to photos entitled "SummerWithAlice" would be only granted to WBSN users who like Alice's profile.

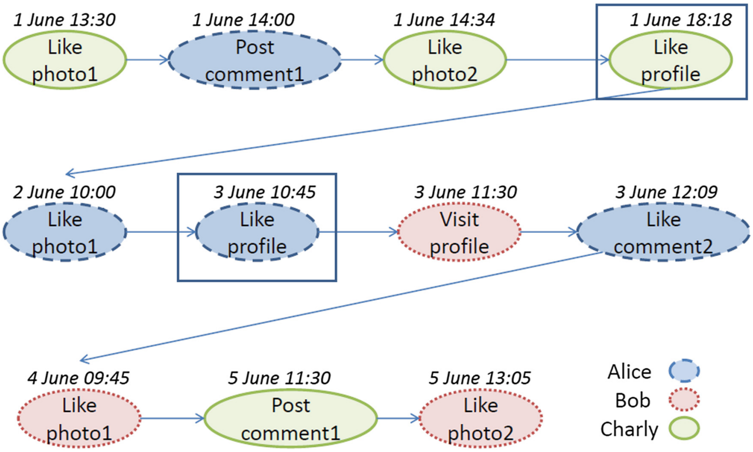


Fig. 1. Actions of Daniel over his contacts' data

3.3 Model Definition

Expressing user provenance in terms of $SoNeUCON_{ABC}$ could be modelled as some Rights r_i that are given after fulfilling some Obligations b_i. Let us consider the supporting example, the Right of accessing photos entitled "SummerWithAlice" is given to users that have liked Alice's profile (Obligation). However, Obligations in $SoNeUCON_{ABC}$ cannot be related to specific objects or subjects. The proposed case needs to express that the obligation is to access the profile (object) of Alice (subject). This lack of expressiveness motivates the extension presented herein.

User provenance management requires including performed actions within the access control process. WBSN actions, defined as *Actions AC*, are modelled as a particular type of *Obligation* b_i. Thus, $SoNeUCON_{ABC}Pro$ extends $SoNeUCON_{ABC}$ including entity *Actions AC* within *Obligations B* together

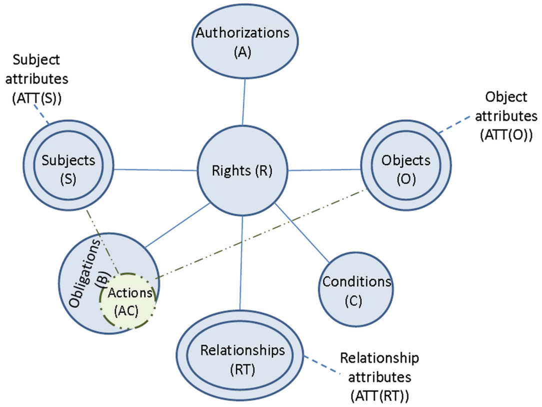


Fig. 2. $SoNeUCON_{ABC}Pro$

with attached links (Fig. 2). AC are performed by subjects S over objects O and then, AC is related to S and O. Consequently, including AC in B comprises new links whose management needs to be specified.

3.4 Translucent User Provenance

Actions management involves the creation of a path per user u_i ($P_{ac_{u_i}}$) where nodes are WBSN actions ($ac_{j-u_i:o_k}$) performed over objects o_k that include the date and time when they are performed; and edges are time relationships (e_{ti}) among nodes. The construction of $P_{ac_{u_i}}$ comprises two steps:

1. Identification of all users U' that have resources over which the requester, a user u_i, has performed an action.
2. Ordering of actions based on date and time. Note that sequential actions over the same object are represented as different nodes connected in temporal order.

Concerning the supporting example a path is constructed based on the date and time of actions performed by Daniel. Figure 3 depicts the formal representation ($P_{ac_{u_4}}$), being Daniel u_4 and Alice, Bob and Charly u_1, u_2 and u_3 respectively. When Daniel, u_4, requests a permission over an object of other user, access control involves verifying some actions of the created path to grant or deny the requested permission accordingly. If Daniel would not mind to disclose any action and if he requests access to "SummerWithAlice" photos, the created path is evaluated and the access granted because he clicked "like" on Alice's, u_1, profile on June 3rd ($ac_{6-u_1:o_6}$).

$SoNeUCON_{ABC}Pro$ also manages translucency. Given the inclusion of AC within B, translucency is based on managing which ac_i performed by the requester u_i over an o_i of a user u_j should remain accessible. In other words, access to chosen nodes $ac_{j-u_i:o_k}$ is denied such that $ac_{j-u_i:o_k}$ and attached e_{ti}

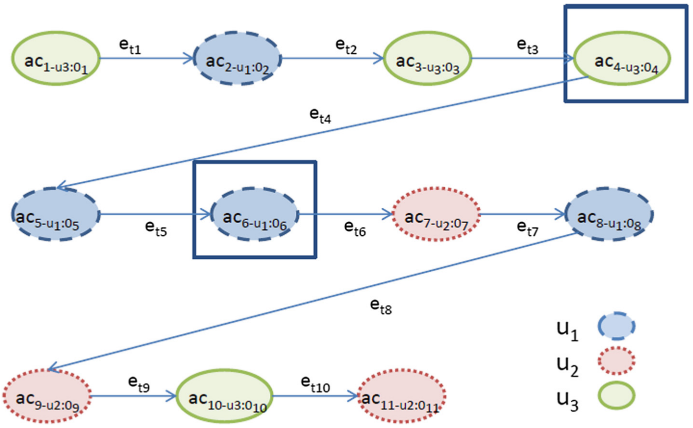


Fig. 3. $P_{ac_{u_4}}$ formal representation of supporting example (Fig. 1)

are deleted from $P_{ac_{u_i}}$ in the access control enforcement process. Recalling the supporting example, as Daniel, u_4, does not want to disclose that he has liked Alice's, u_1, and Charly's, u_3, profiles, actions $ac_{4-u_3:o_4}$ and $ac_{6-u_1:o_6}$ (highlighted in Fig. 3) are not involved in the process. Then, "SummerWithAlice" pictures are denied to Daniel, u_4. More specifically, $ac_{3-u_3:o_3}$ would be linked to $ac_{5-u_1:o_5}$ through e_{t3} and $ac_{5-u_1:o_5}$ would be linked to $ac_{7-u_2:o_7}$ through e_{t5}.

4 Access Control Policies

This Section includes the description and enforcement of access control policies.

4.1 Description

Access control policies are enhanced to address user provenance and translucency. While the former requires the update of $SoNeUCON_{ABC}$ policies, translucency management requires the inclusion of a new set of policies called ***translucency policies*** (ρ_t).

Concerning user provenance, the same operators and attributes as those applied in $SoNeUCON_{ABC}$ [15] are considered. Nonetheless, access control policies are defined in terms of $ATT(S)$, $ATT(O)$, $ATT(RT)$, R, C, B and AC. In particular, an access control policy is formally defined as $\rho(\rho_s; \rho_o; \rho_{rt}; r; \chi; \partial_c)$.

Recalling Sect. 2, the only difference is that χ replaces ∂_b. In fact, χ is a superset containing $SoNeUCON_{ABC}$ obligations as well as the user provenance actions (ξ) introduced in $SoNeUCON_{ABC}Pro$. χ is described as follows using BNF notation [28]:

- $\chi ::= (\emptyset|\xi^*|\partial_b) ::= (\emptyset|(act_i; dt; \rho'_w)^*|\partial_b) ::= (\emptyset|(act_i; dt; \rho'_s; \rho'_o; \rho'_{rt})^*|\partial_b)$
  - ξ comprises predicates applied for user provenance management.

- ∂_b refers to any type of obligation different from those related to user provenance, e.g. the need to have 10 contacts at least. This type of obligation is analogous to the ones presented in $SoNeUCON_{ABC}$.
- act_i refers to all possible actions that can be applied in the WBSN context. For instance, Liked, Visited, Commented, etc. Note that xAPI can be used to represent user actions [1].
- dt refers to the date (d) in the form YYYY/MM/DD and the time (t) in the form HH:MM:SS when act_i is performed. Data and time follow ISO 8601 [30]. Any element, e.g. DD, can take symbol $*$ meaning that no restrictions are established. dt can take symbol $\emptyset$ meaning that no element has restrictions.
- ρ'_w refer to subjects, objects and relationship predicates, where w can take three possible values – s, o or e –, to indicate its relation to subjects, objects or relationships. Specifically, ρ'_s refers to subjects with a relationship type ρ'_{rt} with the owner of the requested object who perform r_i over objects linked to ρ'_o.

Example. Recalling the supporting example, the following user provenance policy expresses that access to "SummerWithAlice" photos is only granted to users who like Alice's, u_1, profile, χ $(Liked; */*/*-*:*:*; (name = Alice);\ (title = profile); ((((\emptyset))),\ \emptyset,\ \emptyset))$.

On the other hand, ρ_t are proposed to limit which actions are applied in the access control process and which ones remain hidden. All WBSN actions are accessible to all users by default – the whole $p_{ac_{u_i}}$ is applied in managing access control. However, if ρ_t exist, they are firstly evaluated against $P_{ac_{u_i}}$. Actions $ac_{j-u_i:o_k}$ which satisfy established ρ_t are removed from the graph. ρ_t are formally described as follows again applying BNF notation [28].

- $\rho_t ::= (act_i; dt; \rho''_w) ::= (act_i; dt; \rho''_s; \rho''_o; \rho''_{rt})$
  - act_i refers to WBSN actions performed over objects linked to ρ''_o.
  - dt refers to the date and time when act_i is performed. Its structure is analogous to the one presented in χ.
  - ρ''_w again involves subject, object and relationship predicates (ρ''_s, ρ''_o and ρ''_{rt} respectively) but meaning that removed nodes are those where objects are linked to ρ''_o whose owner satisfies ρ''_s and has a relationship ρ''_{rt} with the data requester.

Example. Considering Fig. 3 and the supporting example, Daniel, u_4, does not want to disclose that he has liked his direct friends' profiles. Thus, he establishes $\rho_{t_1}(Like; \emptyset;\ (title = profile); ((((role = friend))),\ \emptyset,\ 1))$, such that the resulting $P_{ac_{u_4}}$ would be the one depicted in Fig. 3 removing nodes within rectangles. In this way, Daniel, u_4 limits which actions are accessible becoming translucent.

4.2 Policy Enforcement

$P_{ac_{u_i}}$ is defined as an ordered list ($lpath_{u_i}$) where each position is an action $ac_{j-u_i:o_k}$ together with attached object o_k. $lpath_{u_i}$ is formally represented as $lpath_{u_i}\{o_t, ac_{k-u_i:o_t}; o_{t+1}, ac_{(k+1)-u_i:o_{k+1}}; \ldots\}$. For instance, based on Fig. 3, $lpath_{u_4}$ $\{o_1, ac_{1-u_3:o_1}; o_2, ac_{2-u_1:o_2}; o_3, ac_{3-u_3:o_3}; o_4, ac_{4-u_3:o_4}; \ldots; o_{11}, ac_{11-u_2:o_{11}}\}$.

After the construction of $P_{ac_{u_i}}$ per access request (s, o, r) (where s is the requester, o the requested object and r the requested right over o), all access control policies ρ and translucency policies ρ_t are evaluated. ρ_t are firstly evaluated against *lpath*. Function *evaluateTransPolicies* is executed with the inputs of ρ_t, *lpath* and *req*, where *req* is a reference to data pertaining to the requester, such as his objects and relationships. If the result of the evaluation is 'true' the appropriate elements of *lpath* are removed and thus, *newlpath* is created and applied in the evaluation of ρ. Pseudo-code of *evaluateTransPolicies* is depicted in Algorithm 1 where functions *Match*, *MatchRT*, *GetSubAtt*, *GetObjAtt*, *CreateRT* and *GetAdmin* are developed in $SoNeUCON_{ABC}$ (see [15] for details). These functions are used to verify that objects O, subjects S and relationships RT predicates ρ''_w involved in ρ_t match O, S and RT involved in *lpaths*. Note that symbol "." is used to access the content of an element and the expression $list[pos]$ refers to accessing the element of *list* located at position *pos*.

Algorithm 1. evaluateTransPolicies

```
procedure EVALUATETRANSPOLICIES($\rho_t, lpath, req$ )
    for $lpath_{u_i} \leftarrow (i = 1)$ to $sizeOf(lpath)$ do
        if $\rho_t.act_i = lpath_{u_i}[j]$ then
            if $verifyDateTime(\rho_t.dt, lpath_{u_i}[j])$ then
                $adminLpath = GetAdmin(lpath_{u_i}[j].o)$
                $attSubj = GetSubAtt(adminLpath, lpath_{u_i}[j].\rho_s)$
            end if
            if $Match(attSubj, \rho_t.\rho''_s)$ then
                $attObj = GetObjAtt(lpath_{u_i}[j].o, lpath_{u_i}[j].\rho_o)$
            end if
            if $Match(attObj, \rho_t.\rho''_o)$ then
                $rt = CreateRT(adminLpath, req, 1)$
            end if
            if $MatchRT(\rho_t.\rho''_r t, rt, u_i, 1)$ then
                return $lpath_{u_i}$ node marked as not usable. newlpath
            end if
        end if
    end for
end procedure
```

Subsequently, access control policies ρ are evaluated. The evaluation of predicates ρ_o, ρ_s and ρ_{rt}, conditions ∂_c and the subset of obligations ∂_b are analogous to the corresponding elements of $SoNeUCON_{ABC}$ [15]. Then, the evaluation of χ is what needs to be described herein (Function *evaluate*χ, Algorithm 2). It is similar to *evaluateTransPolicies*, the only difference is when all $\xi \in \chi$ are evaluated over *newlpath*. If the result is 'true' for all ξ, the requested r_i over o_i is granted whether results of evaluating the remaining elements in ρ are also 'true'.

Algorithm 2. evaluateχ

```
procedure EVALUATE$\chi$($\rho_r$, $newlpath$, $req$ )
    for $\chi.\xi[h] \leftarrow (h = 1)$ to $sizeOf(\chi)$ do
        for $newlpath_{u_i} \leftarrow (i = 1)$ to $sizeOf(newlpath)$ do
            if $\chi.\xi[h].act_i = newlpath_{u_i}[j]$ then
                if $verifyDateTime(\chi.\xi[h].dt, newlpath_{u_i}[j])$ then
                    $adminLpath = GetAdmin(newlpath_{u_i}[j].o)$
                    $AttSubj = GetSubAtt(adminLpath, newlpath_{u_i}[j].\rho_s)$
                end if
                if $Match(attSubj, \chi.\xi[h].\rho''_s)$ then
                    $AttObj = GetObjAtt(newlpath_{u_i}[j].o, newlpath_{u_i}[j].\rho_o)$
                end if
                if $Match(AttObj, \chi.\xi[h].\rho''_o)$ then
                    $rt = CreateRT(adminLpath, req, 1)$
                end if
                if $MatchRT(\chi.\xi[h].\rho''_r t, rt, u_i, 1))$ then
                    return $\chi$ verified. Result true
                end if
            end if
        end for
    end for
end procedure
```

5 Evaluation

The evaluation of $SoNeUCON_{ABC}Pro$ comprises a goals analysis and a temporal workload assessment.

5.1 Goals Analysis

$SoNeUCON_{ABC}Pro$ addresses user provenance together with translucency. The former feature is achieved by the inclusion of actions within obligations together with management issues, namely the update of access control policies and the enforcement procedure. Translucency is achieved by creating and managing policies by which users only disclose chosen actions.

Note that to apply $SoNeUCON_{ABC}Pro$ in a real WBSN the following three guidelines should be considered: (1) WBSNs should allow the establishment of ρ and ρ_t; (2) attributes within ρ such as age, role, etc. which are already used and stored by WBSNs, should be involved in the access control process; and (3) user actions have to be recorded by WBSNs and those actions included in ρ_t removed from the access control enforcement process.

5.2 Temporal Workload Assessment

In $SoNeUCON_{ABC}Pro$ access control management is based on the evaluation of policies ρ and translucency policies ρ_t per user request. A critical aspect is to keep the temporal workload under usability limits. In this regard, ρ_t will be managed off-line whereas ρ are managed on-line. Each part will be analyzed separately.

Experimental Settings. $SoNeUCON_{ABC}Pro$ is devoted to the very same goal as its ancestor, $SoNeUCON_{ABC}$ – access control. Therefore, experiments are focused on measuring how much time it takes to assess the policies at stake in different social network scenarios. The settings mainly relate to three aspects – the social networks, the policies and the computational resources. Regarding the first aspect, the same four WBSNs created in the evaluation of $SoNeUCON_{ABC}$ have been considered herein. For illustration purposes, Table 2 depicts the number of nodes ($\#v_i$), relationships ($\#e_i$) and relationships per node ($\overline{e_i/v_i}$) of proposed WBSNs.

Table 2. WBSNs structure

WBSNs id	$\#e_i$	$\#v_i$	$\overline{e_i/v_i}$
1	2,980,388	50,000	60
2	5,965,777	50,000	120
3	8,949,375	50,000	185
4	10,929,713	50,000	219

As $SoNeUCON_{ABC}$ policies did not consider user provenance or translucency, they could not be directly applied to assess $SoNeUCON_{ABC}Pro$. In this case we consider that common actions that users perform in a WBSN such as Facebook are *Liked*, *Photos uploaded*, *Sent messages*, *Shared items* and *Comments*, where the percentage of actions usage is the one presented in Table 3. However, apart from actions, the elements involved in policies of $SoNeUCON_{ABC}$ are similar to those involved in user provenance and translucency – they affect subjects, objects and relationships. Thus, we assume that these policies have similar computational requirements than user provenance or translucency ones. For simplicity we keep the same policies than $SoNeUCON_{ABC}$ (see [15]).

Table 3. Percentage of actions usage in Facebook[a,b]

Likes	Photos uploaded	Sent messages	Shared items	Comments
43.75	0.30	9.72	46.18	0.00071

[a] http://blog.wishpond.com/post/115675435109/40-up-to-date-facebook-facts-and-stats, last access June 2017.

[b] https://zephoria.com/top-15-valuable-facebook-statistics, last access June 2017.

The experiments were carried out on a Intel Core Due E8400 3.2 GHz processor with 4 GB of RAM and Ubuntu 12.04. This experiment is designed to act as a crude proof of principle as it is running on a modest system.

Off-line Part: Translucency. The evaluation of ρ_t consists of creating $P_{ac_{u_i}}$ per user u_i and enforcing the verification of ρ_t over such path. Most WBSNs store a timeline of our activities[3]. Then, $P_{ac_{u_i}}$ can be created at runtime avoiding the cost of its creation when policy enforcement is carried out. Likewise, the separation of actions could benefit performance to a great extent, for instance, creating a path $P_{ac_{j,u_i}}$ for each type of action j. In the same way, the evaluation of ρ_t over each $P_{ac_{j,u_i}}$ can be carried out off-line also benefiting performance, that is after the execution of a set of actions instead of per user's request. This simplifies the implementation of translucent user provenance in a real environment.

This workload is measured as follows. For each action in $P_{ac_{u_i}}$, it is necessary to evaluate if conditions are met (thus hiding the action from access control evaluation) or not. As aforementioned, this evaluation involves the same elements as those existing in $SoNeUCON_{ABC}$ policies and then, in this previous model the evaluation of proposed policies takes 13 ms at minimum and 184.5 ms on average (see Appendix for details). Therefore, we take these values as the expected time to assess each action.

Regarding the amount of actions (i.e. the length of $P_{ac_{j,u_i}}$), we propose different scenarios based on the amount of contacts and the number of actions over each contact's data. Particularly, we consider 25, 50, 100, 300 contacts and 10, 25, 50, 75, 300, 450, 750, 1000, 10000 actions per contact. Thus, $P_{ac_{j,u_i}}$ ranges from 250 to 3000000 actions, though for performance reasons different paths per type of action could be distinguished. Note that the amount of contacts is in line with current figures, as 338 users is the average amount of Facebook friends[4].

Considering established parameters, temporal workload of evaluating translucency policies ρ_t is presented in Table 4. Depending on the type of action within ρ_t, the temporal workload is highly affected because the higher the usage of actions (recall Table 3), the higher the nodes in $P_{ac_{u_i}}$ to evaluate. Though results are better when actions *Photos uploaded* or *Comments* are involved in ρ_t, as this process is performed off-line, the impact of temporal workload is not a big issue. For instance, when the action type is *Comments*, the evaluation takes 3 ms for 10 actions and around 0.39 ms for 1000 actions and 300 contacts in the average case. However, when other actions are at stake, i.e. *Shared items*, the evaluation takes 1.4 min for 10 actions and 42 min for 300 actions and 100 contacts in the average case.

On-line Part: Access Control with User Provenance. The evaluation of ρ that include user provenance is carried out on-line. As opposed to

[3] https://es-la.facebook.com/notes/radio-949/timeline/309814275719798/, last access June 2017.

[4] http://www.pewresearch.org/fact-tank/2014/02/03/6-new-facts-about-facebook/, last access June 2017.

Table 4. Off-line part: translucency assessment. Temporal workload (sc)

		# of actions per user's data								
		10	25	50	75	300	450	750	1000	100000
Like										
Best time										
# contacts per user	25	1.42	4.98	12.09	14.93	42.66	63.99	106.66	142.21	14220.92
	50	2.84	9.95	24.17	29.86	85.32	127.98	213.31	284.41	28440.94
	100	5.69	19.91	48.35	59.72	170.64	255.96	426.61	568.81	56881.08
	300	17.06	59.72	145.04	179.17	511.92	767.89	1279.81	1706.41	170641.24
Average time										
# contacts per user	25	20.13	60.38	140.88	181.14	603.79	905.69	1509.48	2012.64	201263.71
	50	40.25	120.76	281.77	362.27	1207.58	1811.37	3018.95	4025.27	402526.75
	100	80.51	241.52	563.54	724.55	2415.16	3622.74	6037.90	8050.53	805052.83
	300	241.55	724.55	1690.61	2173.64	7246.47	10869.71	18116.18	24154.90	2415490.00
Photos uploaded										
Best time										
# contacts per user	25	0.01	0.04	0.09	0.12	0.33	0.50	0.83	1.11	110.61
	50	0.02	0.08	0.19	0.23	0.66	1.00	1.66	2.21	221.21
	100	0.04	0.15	0.38	0.46	1.33	1.99	3.32	4.42	442.41
	300	0.13	0.46	1.13	1.39	3.98	5.97	9.95	13.27	1327.21
Average time										
# contacts per user	25	0.16	0.47	1.10	1.41	4.70	7.04	11.74	15.65	1565.38
	50	0.31	0.94	2.19	2.82	9.39	14.09	23.48	31.31	3130.76
	100	0.63	1.88	4.38	5.64	18.78	28.18	46.96	62.62	6261.51
	300	1.88	5.64	13.15	16.91	56.36	84.54	140.90	187.87	18787.10
Sent messages										
Best time										
# contacts per user	25	0.32	1.11	2.69	3.32	9.48	14.22	23.70	31.60	3160.20
	50	0.63	2.21	5.37	6.64	18.96	28.44	47.40	63.20	6320.21
	100	1.26	4.42	10.74	13.27	37.92	56.88	94.80	126.40	12640.24
	300	3.79	13.27	32.23	39.82	113.76	170.64	284.40	379.20	37920.27
Average time										
# contacts per user	25	4.47	13.42	31.31	40.25	134.18	201.26	335.44	447.25	44725.27
	50	8.95	26.84	62.62	80.51	268.35	402.53	670.88	894.50	89450.39
	100	17.89	53.67	125.23	161.01	536.70	805.05	1341.75	1789.01	178900.63
	300	53.68	161.01	375.69	483.03	1610.33	2415.49	4025.82	5367.76	536775.56
Shared items										
Best time										
# contacts per user	25	1.50	5.25	12.76	15.76	45.03	67.55	112.58	150.11	15010.97
	50	3.00	10.51	25.52	31.52	90.06	135.09	225.16	300.21	30020.99
	100	6.00	21.01	51.03	63.04	180.12	270.19	450.31	600.41	60041.14
	300	18.01	63.04	153.10	189.12	540.36	810.55	1350.91	1801.21	180121.31
Average time										
# contacts per user	25	21.24	63.73	148.71	191.20	637.34	956.00	1593.34	2124.45	212445.03
	50	42.49	127.47	297.42	382.40	1274.67	1912.00	3186.67	4248.89	424889.35
	100	84.98	254.93	594.84	764.80	2549.33	3824.00	6373.33	8497.78	849777.98
	300	254.97	764.80	1784.53	2294.40	7649.05	11473.58	19122.63	25496.84	2549683.89
Comments										
Best time										
# contacts per user	25	$2\cdot10^{-5}$	$8\cdot10^{-5}$	$2\cdot10^{-4}$	$2\cdot10^{-4}$	$7\cdot10^{-4}$	$1\cdot10^{-3}$	$1\cdot10^{-3}$	$2\cdot10^{-3}$	0.23
	50	$5\cdot10^{-5}$	$1\cdot10^{-4}$	$3\cdot10^{-4}$	$4\cdot10^{-4}$	$1\cdot10^{-3}$	$2\cdot10^{-3}$	$3\cdot10^{-3}$	$4\cdot10^{-3}$	0.46
	100	$9\cdot10^{-5}$	$3\cdot10^{-4}$	$7\cdot10^{-4}$	$9\cdot10^{-4}$	$2\cdot10^{-3}$	$4\cdot10^{-3}$	$6\cdot10^{-3}$	$9\cdot10^{-3}$	9.29
	300	$2\cdot10^{-4}$	$9\cdot10^{-4}$	$2\cdot10^{-3}$	$2\cdot10^{-3}$	$8\cdot10^{-3}$	0.01	0.02	0.02	2.78
Average time										
# contacts per user	25	$3\cdot10^{-4}$	$9\cdot10^{-4}$	$2\cdot10^{-3}$	$2\cdot10^{-3}$	$9\cdot10^{-4}$	0.01	0.02	0.03	3.28
	50	$6\cdot10^{-4}$	$1\cdot10^{-3}$	$4\cdot10^{-3}$	$5\cdot10^{-3}$	0.01	0.02	0.04	0.06	6.56
	100	$1\cdot10^{-3}$	$3\cdot10^{-3}$	$9\cdot10^{-3}$	0.01	0.03	0.05	0.09	0.13	13.13
	300	$3\cdot10^{-3}$	0.01	0.02	0.03	0.11	0.17	0.29	0.39	39.42

$SoNeUCON_{ABC}$ assessment, in this proposal obligations B are critical – recall that user provenance can be seen as obligations involving actions, $\xi \in \chi$.

Policy ρ enforcement can be divided into two main parts. First, the evaluation of predicates regarding the object (ρ_o), the subject (ρ_s) and its relationships (ρ_{rt}). For this part the temporal workload is exactly the time of $SoNeUCON_{ABC}$ policies. Second, the evaluation of the obligations B that the requester needs to fulfill. In this second part, we consider policies with a single user provenance obligation ξ. Given that this obligation involves the same elements that the first part (i.e. subjects, objects and relationships), we assume that it takes the same time – 13 ms (best case) and 184.5 ms (on average) – per element in the path.

Table 5. On-line part: user provenance assessment. Temporal workload (sc)

		# of actions per user's data								
		10	25	50	75	300	450	750	1000	100000
		Like								
		Best time								
# contacts per user	25	**1.43**	**4.98**	**12.09**	14.95	42.83	64.25	107.08	142.78	14277.80
	50	**2.85**	**9.96**	24.18	29.88	85.49	128.24	213.73	284.98	28497.82
	100	**5.69**	19.91	48.35	59.74	170.81	256.22	427.03	569.38	56937.96
	300	17.07	59.73	145.05	179.18	512.09	768.14	1280.24	1706.98	170698.12
		Average time								
# contacts per user	25	20.21	60.46	140.96	181.38	606.21	909.32	1515.53	2020.71	202070.95
	50	40.33	120.84	281.85	362.52	1210.00	1815.00	3025.00	4033.34	403333.99
	100	80.59	241.60	563.62	724.79	2417.58	3626.37	6043.95	8058.60	805860.06
	300	241.63	724.63	1690.69	2173.88	7248.89	10873.34	18122.23	24162.97	2416297.23
		Photos uploaded								
		Best time								
# contacts per user	25	**0.01**	**0.04**	**0.09**	**0.12**	**0.33**	**0.50**	**0.83**	**1.11**	111.05
	50	**0.02**	**0.08**	**0.19**	**0.23**	**0.66**	**1.00**	**1.66**	**2.22**	221.65
	100	**0.04**	**0.15**	**0.38**	**0.46**	**1.33**	**1.99**	**3.32**	**4.43**	442.85
	300	**0.13**	**0.46**	**1.13**	**1.39**	**3.98**	**5.97**	**9.96**	**13.28**	1327.65
		Average time								
# contacts per user	25	**0.16**	**0.47**	**1.10**	**1.41**	**4.71**	**7.07**	**11.79**	15.72	1571.66
	50	**0.31**	**0.94**	**2.19**	**2.82**	**9.41**	**14.12**	23.53	31.37	3137.03
	100	**0.63**	**1.88**	**4.38**	**5.64**	18.80	28.21	47.01	62.68	6267.79
	300	**1.88**	**5.64**	**13.15**	16.91	56.38	84.57	140.95	187.93	18793.38
		Sent messages								
		Best time								
# contacts per user	25	**0.32**	**1.11**	**2.69**	**3.32**	**9.52**	**14.28**	23.80	31.73	3172.84
	50	**0.63**	**2.21**	**5.37**	**6.64**	19.00	28.50	47.50	63.33	6332.85
	100	**1.27**	**4.43**	**10.75**	**13.28**	37.96	56.94	94.90	126.53	12652.88
	300	**3.79**	**13.27**	32.23	39.82	113.80	170.70	284.50	379.33	37932.91
		Average time								
# contacts per user	25	**4.49**	**13.44**	31.33	40.31	134.71	202.07	336.78	449.05	44904.66
	50	**8.96**	26.85	62.63	80.56	268.89	403.33	672.22	896.30	89629.77
	100	17.91	53.69	125.25	161.06	537.24	805.86	1343.10	1790.80	179080.01
	300	53.70	161.03	375.71	483.08	1610.86	2416.30	4027.16	5369.55	536954.94
		Shared items								
		Best time								
# contacts per user	25	**1.51**	**5.26**	**12.76**	15.78	45.21	67.82	113.03	150.71	15071.01
	50	**3.01**	**10.51**	25.52	31.54	90.24	135.36	225.61	300.81	30081.03
	100	**6.01**	21.02	51.04	63.06	180.30	270.46	450.76	601.01	60101.18
	300	18.02	63.05	153.10	189.14	540.54	810.82	1351.36	1801.81	180181.34
		Average time								
# contacts per user	25	21.33	63.82	148.80	191.46	639.89	959.84	1599.73	2132.97	213297.11
	50	42.57	127.55	297.51	382.66	1277.22	1915.84	3193.06	4257.41	425741.43
	100	85.06	255.02	594.93	765.05	2551.89	3827.84	6379.73	8506.30	850630.07
	300	255.05	764.88	1784.61	2294.65	7651.61	11477.41	19129.02	25505.36	2550535.97
		Comments								
		Best time								
# contacts per user	25	$\mathbf{2}\cdot10^{-5}$	$\mathbf{8}\cdot10^{-5}$	$\mathbf{2}\cdot10^{-4}$	$\mathbf{2}\cdot10^{-4}$	$\mathbf{7}\cdot10^{-4}$	$\mathbf{1}\cdot10^{-3}$	$\mathbf{1}\cdot10^{-3}$	$\mathbf{2}\cdot10^{-3}$	**0.23**
	50	$\mathbf{5}\cdot10^{-5}$	$\mathbf{1}\cdot10^{-4}$	$\mathbf{3}\cdot10^{-4}$	$\mathbf{4}\cdot10^{-4}$	$\mathbf{1}\cdot10^{-3}$	$\mathbf{2}\cdot10^{-3}$	$\mathbf{3}\cdot10^{-3}$	$\mathbf{4}\cdot10^{-3}$	**0.46**
	100	$\mathbf{9}\cdot10^{-5}$	$\mathbf{3}\cdot10^{-4}$	$\mathbf{7}\cdot10^{-4}$	$\mathbf{9}\cdot10^{-4}$	$\mathbf{2}\cdot10^{-3}$	$\mathbf{4}\cdot10^{-3}$	$\mathbf{6}\cdot10^{-3}$	$\mathbf{9}\cdot10^{-3}$	**9.29**
	300	$\mathbf{2}\cdot10^{-4}$	$\mathbf{9}\cdot10^{-4}$	$\mathbf{2}\cdot10^{-3}$	$\mathbf{2}\cdot10^{-3}$	$\mathbf{8}\cdot10^{-3}$	**0.01**	**0.02**	**0.02**	**2.78**
		Average time								
# contacts per user	25	$\mathbf{3}\cdot10^{-4}$	$\mathbf{9}\cdot10^{-4}$	$\mathbf{2}\cdot10^{-3}$	$\mathbf{2}\cdot10^{-3}$	$\mathbf{9}\cdot10^{-4}$	**0.01**	**0.02**	**0.03**	**3.29**
	50	$\mathbf{6}\cdot10^{-4}$	$\mathbf{1}\cdot10^{-3}$	$\mathbf{4}\cdot10^{-3}$	$\mathbf{5}\cdot10^{-3}$	**0.01**	**0.02**	**0.04**	**0.06**	**6.58**
	100	$\mathbf{1}\cdot10^{-3}$	$\mathbf{3}\cdot10^{-3}$	$\mathbf{9}\cdot10^{-3}$	**0.01**	**0.03**	**0.05**	**0.09**	**0.13**	**13.15**
	300	$\mathbf{3}\cdot10^{-3}$	**0.01**	**0.02**	**0.03**	**0.11**	**0.17**	**0.29**	**0.39**	39.43

Table 5 shows the time taken for the evaluation of ρ. The time needed is practically the same as the one required for translucency. The rationale behind this is that in user provenance we need to add the time for assessing the related predicates ρ_o, ρ_s and ρ_{rt} which turns out to be small as compared to the time to evaluate obligations ξ.

Despite of the similarity, the acceptance criterion for these times involves usability aspects because this is an on-line evaluation. Results are suitable if they do not negatively affect to the user experience. Establishing 15 sc as the maximum threshold for keeping users attention[5] [22], values in bold on Table 5

[5] Although 2 sc would be a desirable threshold [22], we believe that the proposed limit is illustrative enough as it is the maximum acceptable upper limit.

are suitable. Only when types of actions *Photos uploaded* and *Comments* are involved within B the temporal workload remains within the established limit, as well as a significant amount of actions can be considered, i.e. 10000 actions for *Comments* in the average case. Nevertheless, for the remaining types of actions these results are subject to improvement, as discussed below.

Discussion. Results achieved in the experiments lead to different considerations:

- Regarding translucency, it can be performed off-line (thus not affecting usability) and with immensely greater computational resources.
- The proposed study presents the worst-case analysis. The evaluation of predicates ρ'_o, ρ'_s and ρ'_{rt} within each obligation ξ are evaluated for every $ac_{j-u_i:o_k} \in P_{ac_{j,u_i}}$. Conversely, in a real scenario not all policies ρ include subjects, objects and relationships predicates, thus reducing the measured temporal workload. Additionally, the algorithm applied in the evaluation can be enhanced, e.g. a divide and conquer algorithm to search in ordered lists may be used.
- Computational resources currently applied by WBSNs are much more powerful than those applied herein, e.g. parallelism could alleviate the problem.

In sum, this worst-case analysis has shown that even with constrained computational resources and with heavyweight policies, the proposed approach is feasible.

6 Related Work

Lots of WBSN models have been developed. Many of them are based on assorted features, i.e. roles [18], trust [5], relationships [13], attributes [21] and ontology [20]. Besides, dealing with attributes management but looking for expressiveness $SoNeUCON_{ABC}$ was proposed [15].

Other proposals manage data provenance in WBSNs. The origin and traces of data is involved in the access control management process. A data provenance based access control model is proposed by Park et al. [23,24]. It provides dynamic separation of duties, origin-based control and objects versioning in environments like WBSNs. Pei and Ye [25] define a framework to capture data provenance and create access control policies from collected data being possible its application in WBSNs. Cheng et al. [9] look for the administrative management of a relationship-based access control model including provenance management.

A step forward can be taken by managing user provenance access control. Considering this feature as a trustworthiness analysis focused on tracing WBSN users' actions, some works can be pointed out. A monitoring system to capture and analyse WBSN users behaviour is proposed in [17]. Sybildefender [32], SybilInfer [11] and Sybilguard [33] focus on identifying sybil WBSN nodes. In addition, [26] works with policies that involve users' actions but they are applied for dynamic access control instead of provenance management.

The negative side of provenance management is the privacy problems it involves [12] as the identification of data or user traces may reveal private data. Multiple proposals work on the establishment of anonymous interactions [31,35]. Others focus on protecting users' data [2,10,16] or users' relationships [6,34] by applying cryptography. Several works in the context of social translucency have been proposed [14]. The idea is the management of which relationships are established and to whom by being aware of the situation and accountable at the same time.

Despite existing WBSN access control models, user provenance has not already been addressed by any of them. In the same way, those which have worked with user provenance in the form of capturing users' behaviours, do not consider privacy at all, in contrast to the concept of translucency proposed in this model.

7 Conclusion

The massive expansion of WBSNs together with the amount of security issues they involve, foster their research and innovation. The origin and trace of actions performed by a WBSN user, called user provenance, together with translucency to avoid privacy problems are requirements to include within WBSN access control models. $SoNeUCON_{ABC}Pro$ extends a previous version, $SoNeUCON_{ABC}$ an expressive access control model for WBSNs, including the management of user provenance together with translucency. From the authors knowledge this is the first time both concepts are applied for access control management purposes. Its implementation has been empirically studied and it is feasible in different scenarios. While translucency management could be performed without restrictions, some settings are acceptable for user provenance management.

Future work will focus on facilitating selective translucency. Users have to be able to choose to whom translucency policies are applied instead of hiding performed actions for everyone. Also the improvement of performance is an issue to consider, as well as usability issues regarding the specification and management of policies by WBSN users.

Acknowledgments. This work was supported by the MINECO grants TIN2013-46469-R (SPINY: Security and Privacy in the Internet of You) and TIN2016-79095-C2-2-R (SMOG-DEV); by the CAM grant S2013/ICE-3095 (CIBERDINE: Cybersecurity, Data, and Risks); and by the Programa de Ayudas para la Movilidad of Carlos III University of Madrid, Spain (J. M. de Fuentes and L. Gonzalez-Manzano grants). Finally, authors would like to thank Security and Privacy Research Group of University of Birmingham for their comments in an early version of this paper.

Appendix: Temporal Workload Enforcement in $SoNeUCON_{ABC}$

Coloured in gray in Table 6, the evaluation of proposed policies in $SoNeUCON_{ABC}$ takes 13 ms at minimum and 184.5 ms on average.

Table 6. Policy enforcement temporal workload in $SoNeUCON_{ABC}$

WBSN id = 1						
rt id	P1-TW (ms)	P2-TW (ms)	P3-TW (ms)	P4-TW (ms)	P5-TW (ms)	P6-TW (ms)
1	4143	4144	4143	4143	4142	4142
2	435	435	435	435	435	435
3	28	28	28	28	28	28
4	54	55	54	54	54	54
5	38	38	38	38	38	38
6	51	51	51	51	51	51
7	13	13	13	13	13	13
WBSN id = 2						
9	21291	21304	21291	2289	21289	21287
9	712	712	712	713	712	712
10	58	57	57	57	58	57
11	88	88	89	89	88	88
12	61	60	60	60	61	60
13	62	62	63	62	62	62
14	31	30	30	30	30	30
WBSN id = 3						
15	56825	56816	56815	56813	56820	56811
16	274	273	274	273	273	273
17	80	80	80	81	80	80
18	110	111	110	110	110	110
19	89	88	89	88	88	88
20	86	86	86	86	87	86
21	36	37	36	37	36	36
WBSN id = 4						
22	105549	105545	105554	105558	105496	105563
23	1721	1722	1721	1721	1721	1722
24	44	45	44	44	44	45
25	135	134	134	134	134	135
26	96	97	96	96	96	97
27	83	83	83	83	83	84
28	46	46	46	46	46	47
Average	184.6	184.5	184.5	184.5	184.5	184.6
Total average	184.5					

*Results of evaluating proposed policies (see [15]) in created WBSNs over 28 pairs of random users are presented in this Table. Considering 2000 ms a usability limit for being approximately the tolerable waiting time of WBSN users for information retrieval [22], removing cases that exceed this threshold (details in [15]).

References

1. The Advanced Distributed Learning (ADL) Initiative. Experience API, version 1.0.1 (2013). http://www.adlnet.org/wp-content/uploads/2013/10/xAPI_v1.0.1-2013-10-01.pdf. Accessed July 2016
2. Beato, F., Kohlweiss, M., Wouters, K.: Scramble! Your social network data. In: Fischer-Hübner, S., Hopper, N. (eds.) PETS 2011. LNCS, vol. 6794, pp. 211–225. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22263-4_12
3. Buneman, P., Khanna, S., Tan, W.-C.: Data provenance: some basic issues. In: Kapoor, S., Prasad, S. (eds.) FSTTCS 2000. LNCS, vol. 1974, pp. 87–93. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44450-5_6
4. Carminati, B., Ferrari, E.: Access control and privacy in web-based social networks. Int. J. Web Inf. Syst. **4**, 395–415 (2008)
5. Carminati, B., Ferrari, E., Perego, A.: Rule-based access control for social networks. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006, Part II. LNCS, vol. 4278, pp. 1734–1744. Springer, Heidelberg (2006). https://doi.org/10.1007/11915072_80
6. Carminati, B., Ferrari, E., Perego, A.: Private relationships in social networks. In: ICDE, pp. 163–171. IEEE (2007)
7. Carminati, B., Ferrari, E., Perego, A.: Enforcing access control in web-based social networks. TISSEC **13**(1), 6 (2009)
8. Cheng, Y., Park, J., Sandhu, R.: Relationship-based access control for online social networks: beyond user-to-user relationships. In: SocialCom, pp. 646–655 (2012)
9. Cheng, Y., Bijon, K., Sandhu, R.: Extended ReBAC administrative models with cascading revocation and provenance support. In: SACMAT, pp. 161–170. ACM (2016)
10. Cutillo, L.A., Molva, R., Strufe, T.: Safebook: a privacy-preserving online social network leveraging on real-life trust. IEEE Commun. Mag. **47**(12), 94–101 (2009)
11. Danezis, G., Mittal, P.: Sybilinfer: detecting sybil nodes using social networks. In: NDSS (2009)
12. Davidson, S.B., et al.: On provenance and privacy. In: EDBT/ICDT, pp. 3–10. ACM (2011)
13. Fong, P.W.L., Siahaan, I.: Relationship-based access control policies and their policy languages. In: SACMAT, pp. 51–60. ACM (2011)
14. Gilbert, E.: Designing social translucence over social networks. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 2731–2740. ACM (2012)
15. González-Manzano, L., González-Tablas, A.I., de Fuentes, J.M., Ribagorda, A.: $SoNeUCON_{ABC}$, an expressive usage control model for web-based social networks. Comput. Secur. **43**, 159–187 (2014)
16. Jahid, S., et al.: DECENT: a decentralized architecture for enforcing privacy in online social networks. In: PERCOM Workshops, pp. 326–332. IEEE (2012)
17. Lalas, E., Papathanasiou, A., Lambrinoudakis, C.: Privacy and traceability in social networking sites. In: PCI, pp. 127–132. IEEE (2012)
18. Li, J., et al.: Role based access control for social network sites. In: JCPC, pp. 389–394. IEEE (2009)
19. Lynch, S.: The Agency "Cannot Survive Without Being More Transparent". https://www.gsb.stanford.edu/insights/former-nsa-head-michael-hayden-agency-cannot-survive-without-being-more-transparent. Accessed July 2016 (2014)
20. Masoumzadeh, A., Joshi, J.: OSNAC: an ontology-based access control model for social networking systems. In: SOCIALCOM, pp. 751–759. IEEE Computer Society (2010)

21. Munckhof, C.V.D.: Content based access control in social network sites. Master's thesis. Eindhoven University of Technology (2011)
22. Nah, F.F.H.: A study on tolerable waiting time: how long are web users willing to wait? Behav. Inf. Technol. **23**(3), 153–163 (2004)
23. Park, J., Nguyen, D., Sandhu, R.: On data provenance in group-centric secure collaboration. In: CollaborateCom, pp. 221–230. IEEE (2011)
24. Park, J., Nguyen, D., Sandhu, R.: A provenance-based access control model. In: PST, pp. 137–144. IEEE (2012)
25. Pei, J., Ye, X.: Towards policy retrieval for provenance based access control model. In: TrustCom, pp. 769–776. IEEE (2014)
26. Power, D., Slaymaker, M., Simpson, A.: Conformance checking of dynamic access control policies. In: Qin, S., Qiu, Z. (eds.) ICFEM 2011. LNCS, vol. 6991, pp. 227–242. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-24559-6_17
27. Sandhu, R.S., Samarati, P.: Access control: principle and practice. IEEE Commun. Mag. **32**(9), 40–48 (1994)
28. Scowen, R.S.: Extended BNF-a generic base standard. Technical report, ISO/IEC 14977 (1998). http://www.cl.cam.ac.uk/mgk25/iso-14977.pdf
29. Simcox, R.: Surveillance After Snowden: Effective Espionage in an Age of Transparency. The Henry Jackson Society, London (2015)
30. ISO Standards. Date and time format - ISO 8601 (1988)
31. Sweeney, L.: k-anonymity: a model for protecting privacy. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. **10**(05), 557–570 (2002)
32. Wei, W., et al.: Sybildefender: defend against sybil attacks in large social networks. In: INFOCOM, pp. 1951–1959. IEEE (2012)
33. Yu, H., et al.: Sybilguard: defending against sybil attacks via social networks. ACM SIGCOMM Comput. Commun. Rev. **36**, 267–278 (2006)
34. Zheng, Y., Wang, B., Lou, W., Hou, Y.T.: Privacy-preserving link prediction in decentralized online social networks. In: Pernul, G., Ryan, P.Y.A., Weippl, E. (eds.) ESORICS 2015, Part II. LNCS, vol. 9327, pp. 61–80. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24177-7_4
35. Zhou, B., Pei, J.: Preserving privacy in social networks against neighborhood attacks. In: ICDE, pp. 506–515. IEEE (2008)

1st Workshop on Security and Privacy in the Internet of Things (SePrIoT)

SAFEDroid: Using Structural Features for Detecting Android Malwares

Sevil Sen[1(✉)], Ahmet I. Aysan[1], and John A. Clark[2]

[1] WISE Laboratory, Hacettepe University, Ankara, Turkey
ssen@cs.hacettepe.edu.tr, aysan@hacettepe.edu.tr
[2] University of Sheffield, Sheffield, UK
john.clark@sheffield.ac.uk

Abstract. Mobile devices have become a popular target for attackers, whose aims are to harm the devices, illegally obtain personal information and ultimately to reap financial benefit. In order to detect such malicious attempts, security solutions based on static analysis are mainly preferred due to resource-constraints of these devices. However, in general, static analysis-based solutions are not very effective against new mobile malwares and new variants of existing mobile malwares appear on a daily basis. In this study, new features for static analysis are investigated in order to detect mobile malwares. While studies found in the literature mostly employ API calls and permissions, this current study explores some novel structural features. Results show the relative effectiveness of these features on malware detection. Furthermore, it is shown that these features detect new malwares better than solely applying API-based features.

Keywords: Android security · Malware detection · Static analysis
Structural features · Machine learning

1 Introduction

Android malware is one of the biggest threats today. With mobile devices having become an integral part of modern lifes, attackers focus more and more on harming mobile devices and stealing private information from them. New mobile malwares are emerging every day. Kaspersky [1] reported that 884,774 new malicious applications appeared in 2015 alone, three times more than the number of new mobile malwares seen in 2014. According to the recent McAfee Labs Threats Report [2], the number of mobile malwares is still on the rise.

There are two main approaches to the detection of mobile malwares: static analysis and dynamic analysis. Static analysis is generally preferred over dynamic analysis due to its lower overhead on mobile devices. Therefore, a significant amount of work has been proposed on static-based malware detection for Android devices over the last five years, and different features have been investigated in different approaches.

X. Lin et al. (Eds.): SecureComm 2017 Workshops, LNICST 239, pp. 255–270, 2018.
https://doi.org/10.1007/978-3-319-78816-6_18

In the literature, permissions and API calls are among the most used features for static analysis [3]. However, the structural features of an application have not been explored as part of the proposed approaches, hence it forms the main aim of this current study. It can be seen from the literature that the structural information of portable executables has a positive impact on malware detection [4]. Therefore, this current study investigates the use of structural features of applications on mobile malware detection. To the best of our knowledge, some of these proposed features are novel and have not been the subject of previous studies. The proposed system is also applied to new malwares in order to see its capability of detecting new malwares and new variants of malwares. In recent years, there has been a significant increase in Android malware variants, much more than new malware families [5]. Therefore, detecting variants of existing malwares is an important characteristic of an anti-malware system.

To summarize, the contributions of this paper are as follows:

- SAFEDroid, an Android malware detection system based on static analysis, is introduced.
- Rigorous analysis of the original features of SAFEDroid is conducted in order to increase the accuracy of the system.
- SAFEDroid is evaluated on new variants of existing malwares as well as new malwares in order to demonstrate the impact of the original features employed by the proposed system.

The remainder of this paper is organized as follows. Section 2 discusses the related works in the literature, and outlines existing malware detection systems based on static analysis together with the features they employ. Section 3 introduces the features and classification algorithms employed in this study. Section 4 evaluates the results and also discusses the limitations of the proposed approach. Finally, the study is concluded in Sect. 5.

2 Related Work

Static analysis is the most employed technique on mobile devices due to its efficiency. One of the early studies relying on static analysis is by Kirin [6] which detects certain types of malware by evaluating the configuration of an application against a collection of security rules. These rules are defined by using permissions and action strings in Intent filters extracted from the manifest file. Even though it is not a complete solution for the problem, it was shown that it could mitigate certain types of malware. Stowaway [7] detects overprivileged apps by analyzing API Calls. ComDroid [8] detects application' vulnerabilities by analyzing inter-application communications; with their analysis showing that Android applications are vulnerable to attack by other applications.

RiskRanker [9] proposed a two-level analysis. High-risk and medium-risk applications are determined in the first-order analysis, and applications employing obfuscating, encryption or dynamic class loading techniques are extracted among these risky applications in the second-order analysis. Only static analysis is employed to identify applications employing suspicious behaviors. DroidAnalyzer [10] employs static analysis in order to detect the presence of root exploits, and identifies the potential risks of related apps. Droid Analytics [11] generate signatures at the app/class/method level, hence it can detect 0-day repackaged malwares.

In another approach for malware detection, Drebin [12] uses hardware and app components, permissions, API calls and network address as features in order to apply machine learning techniques. Drebin not only classifies the applications, but also produces explanations for the detection model. Similarly, DroidAPIMiner [13] achieves a high detection rate and low false positive rate by using kNN classifier on API calls. They firstly remove API calls invoked by third-party packages, and then consider only top APIs with the highest difference between malware and benign apps in training. Therefore, it is not susceptible to evasive attacks that add more benign APIs to the code in order to evade detection. SafeDroid [14] is also based on the top distinguished API calls extracted from dex files. Although our study has used the same acronym as SafeDroid [14], it is an independent work which evaluates not only the effect of API calls but also the effect of structural features on malware detection. This replication occurred purely coincidentally without realization on the part of the authors. Another ML-based approach firstly groups applications with similar functionalities in clusters, then employs kNN classifier in order to detect mobile malwares [15]. The performance of the approach is shown to be better than AndroGuard [16]; however, it does not perform well on the detecting of updated attacks such as BaseBridge and DroidKungFu.

The usage of ML-based techniques for mobile malware detection is increasing. One of the recent studies employs Bayesian classifiers to distinguish malicious from benign applications [17]. They extract features based on API calls, system and Android commands, permissions in the manifest file, and other information such as the usage of encryption, and the presence of a second .apk file. Mutual Information calculation is employed to rank features, and as a result the top 25 features are outlined and used in training. They also show that as the training sample set increases, the performance of the classifier on the same feature set also increases. A recent study employed different machine learning techniques, including deep learning to the problem [18]. Another recent work evaluates ML-based mobile malware detectors [19].

One of the studies in the literature shows similarity to this current study, employing data mining algorithms on static features obtained from apk, dex, and XML files of an application [20]. However, due to the unavailability in 2010 of malware datasets, the proposed model was not applied for malware classification. Its evaluation is carried out to differentiate between game and tool applications. Moreover, only one feature from [20] is used common to this current study.

To summarize, the studies found in the literature mainly use permissions and API calls for static analysis-based malware detection. Application metadata has also been proposed as complementary to static and dynamic analysis [25]. Recently, studies that use metadata for malware detection [22,23] have been

Table 1. Existing Android static analysis tools

Static tools	Purpose	Features
Kirin [6]	Mitigating certain types of malwares	Permissions Action strings
Stowaway [7]	Determining overprivilege	API calls Action strings
ComDroid [8]	Detecting apps communication vulnerabilities	Permissions Intents Components
RiskRanker [9]	Risk analysis for detecting 0-day malwares	Data flow analysis Control flow analysis Suspicious activities
DroidMat [15]	Detecting malwares	Permissions Intents Components API calls
Droid analytics [11]	Generating signatures Detecting 0-day repackaged malwares	API calls
DroidAPIMiner [13]	Detecting malwares	API calls
Bayesian [17]	Detecting malwares	Permissions API calls Commands
Drebin [12]	Detecting malwares Producing explanations for the model	Permissions Intents Components API calls Network addresses
DroidAnalyzer [10]	Detecting the presence of root exploits	API calls Keywords
Anastasia [18]	Detecting malwares	Permissions Intents API calls System commands Malicious activities
ML-based detectors [19]	Detecting malwares Evaluating ml-based detectors	Basic blocks from CFG
Resource usage-based [21]	Detecting malwares	Resource usage
Metadata-based [22]	Detecting malwares	Permissions API calls Metadata
ADRoit [23]	Detecting malwares	Permissions Metadata
AndroDialysis [24]	Detecting malwares	Permissions Intents

introduced. A brief survey in [3] provides further information on the features used for mobile malware detection. The static analysis tools employed in the literature are outlined and sorted according to their publication date in Table 1.

3 Methodology

In this study, a static analysis tool is developed that is based on API calls and the structural features of applications. Figure 1 illustrates a simplified system architecture of SAFEDroid.

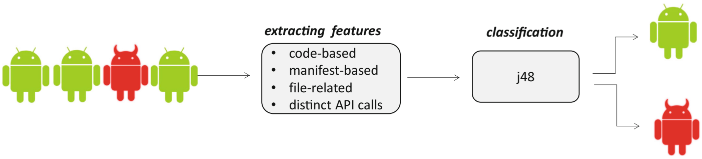


Fig. 1. Simplified schema of the proposed approach

First of all, a dataset consisting of malicious and benign malwares is constructed. Then, a rigorous study is conducted on the dataset in order to select the right features to increase the detection module's accuracy. The choice of which characteristics of an application can be used for machine learning is very important, and must contain sufficient information to allow the fundamentals to be developed. However, irrelevant or an excessive number of features could degrade the performance of the learning algorithms. Based on feature analysis, the feature subset, which is believed to discriminate malicious applications from goodwares, is selected and extracted. Machine learning techniques are applied for the classification of applications by using this feature subset. Finally, the model produced is evaluated by using a new dataset obtained from Drebin [12]. The results support the researchers' hypotheses that the structural features of an application are indicative of malicious behavior, and could detect new malware variants. Each step of the proposed approach and the experimental results are presented in the subsequent sections.

3.1 Feature Selection

In the literature, API calls and permissions of applications are features that are extensively employed for static analysis in mobile malware detection [3]. It is shown that the systems relying on API calls-based features achieve better detection performance than systems using a permission-based feature set [13]. Moreover, the usage of API and permission-based features together does not engender improvements over the usage of only API-based features [26]. Therefore, this study has not included permission features; the top distinguished API calls have been added to the features used. In addition to API calls, the proposed system in this study also extracts some structural features of an application based on analysis of malwares and goodwares. The features employed in this study are classified into three groups: code-based, manifest-based, and file-related features. All of these features are summarized in Table 2.

Table 2. Category of features

Category	Features	Descriptions	Previous works
Code-based	DexClassLoader	DexClassLoader API usage	Yes
	Crypto API	Cryptographic API usage	Yes
	Goto	Number of goto statements	No
	Annotation	Number of annotations	No
	Methods	Number of methods	No
	Classes	Number of classes	No
	Used permissions	Number of used permissions	No
	Used dangerous permissions	Number of used dangerous permissions	No
	API calls	API calls usage	Yes
Manifest-based	Permissions	Number of requested permissions	Yes [34]
	Dangerous permissions	Number of requested dangerous permissions	No
	Lines	Number of lines of the manifest file	No
File-related	File size	Size of the application (bytes)	Yes [20]
	Files	Number of files	No
	Directories	Number of directories	No
	Resource files	Number of files stored in resource directory	No

Code-Based Features. The features in this group are extracted from the application code. The following three features in this group have previously been used for the detection of malwares in other studies: DexClassLoader, Crypto API usage and, API calls. As far as the researchers of this study are aware, the other features are being employed for the first time in this study.

- **DexClassLoader.** Android gives developers an opportunity to load and use classes in runtime. These files can be dynamically loaded from a remote server or from any path of the device. Surprisingly, according to analysis, the percentage usage of dynamic code loading in the malware dataset is much less than its usage in the benign dataset. It was noted that almost half of the benign applications use dynamic code loading. This result is also consistent with the analysis of one million apps submitted to Andrubis [27]. There is a substantial increase in the number of applications using the updating techniques, especially dynamic class loading [28]. It could be concluded that dynamic code loading alone may no longer be an indicator for malicious behavior due to its rising popularity among goodwares as well.
- **Crypto API.** Attackers mainly use cryptography in order to evade static analysis. On the other hand, its usage statically among goodwares has also

increased over the last few years [27]. Analysis conducted in this study also supports the increasing usage of crypto API in benign applications.

- **Goto.** The *goto* statement could be inserted into code in order to evade signature-based detection tools. By using goto statements, attackers can change the order of the code, but preserve the code execution sequence at runtime.
- **Annotation.** Annotations are types of metadata that can be added to the code, but have no effect on the runtime. Our analysis shows that benign applications tend to use many more annotations than malicious applications.
- **Methods.** This feature represents the number of methods in an application package. Benign applications appear to use many more methods than malwares.
- **Classes.** This feature represents the number of classes in an application package. The number of classes seems to be much higher in benign applications than malwares.
- **Used Permissions.** The application authors generally request more permissions than they use. Since used permissions give more information about an application, the number of used permissions are also taken into account in this study. PScout's API-permission list [29] are used in order to extract the real permission list of applications. Recently, a few studies in the literature analyze and employ the combination of used permissions for malware detection [30–32]. However, to the best of our knowledge, the number of used permissions employed in this current study is a first.
- **Used Dangerous Permissions.** This feature represents the used dangerous permissions by an application and to the best of knowledge of the researchers of this study it is also a first to be employed.
- **API Call.** The top distinguished API calls, whose usage in the malware dataset is higher than the benign dataset, as in DroidAPIMiner [13], are extracted.

Manifest-Based Features. Every Android application must have a manifest file (*AndroidManifest.xml*) that contains essential information about the application such as information about the package and application components such as activities, services, broadcast receivers, and content providers. In this study, only the number of permissions listed in the file, and the size of the manifest file is used.

- **Permissions.** This feature represents the number of permissions requested in the manifest file. According to analysis in this study, benign applications use 8 permissions on average, whereas malwares use 14 permissions. Attackers generally use more permissions in order to obtain the control of the device. They can also obtain permissions in advance for use in the dynamically loaded code at runtime [33]. Permissions are the most used features in static analysis [3]. While studies mainly extract the permissions listed in the manifest file [18,20], to the best of knowledge of the researchers the number of permissions is not explicitly used as a feature. Only in [34] the count of xml elements in the manifest file besides permissions is also considered.

- **Dangerous Permissions.** This feature represents the number of dangerous permissions requested in the manifest file. It is shown that malwares generally request more dangerous permissions than benign apps [35].
- **Lines.** Another important feature about the manifest file is the size of the file which is evaluated by the number of lines here. The researchers observed that benign applications mostly generate well-written manifest files with more information. Research experimentation for this study revealed that the average line count of the manifest files of goodwares is 81, whereas it is 49 for malwares.

File-Related Features. Android application is similar to *jar* file and it contains class files and hierarchical directories. Besides, resource files used by applications are stored in the *resource* directory. These features might have a positive effect in order to classify samples. To the best of our knowledge, the file-related features have not been included in previous studies.

- **File Size.** This feature represents the file size of an application measured in bytes. The researchers found that benign applications generally have larger file sizes than malicious applications. In the literature, this feature has been used to differentiate game applications from tools and, has been suggested for use in malware detection [36].
- **Files.** This feature represents the number of files in an application package. Benign applications appear to have many more files than malwares.
- **Directories.** This feature represents the number of directories in an application package. It is observed that benign applications have many more directories in their packages.
- **Resource Files.** This feature indicates the number of resource files used by an application. The resources in Android could be used for various reasons such as language support and, providing images for UI. Applications in the benign dataset are observed to use more resources than those in the malware dataset.

When the correlation between the features and the class labels (i.e. benign or malicious) is analyzed, all non-zeros values are observed. The smallest absolute correlation value (0.14) belongs to *Lines* feature. However when principle component analysis (PCA) is carried out, it is seen that even this feature has considerable weight for the eigen vector corresponding to the 5th highest eigen value. When the correlation among the features are analyzed, it is observed that some features are highly correlated as expected, such as *Methods* and *Classes*. However in general, the cross-correlation matrix shows that the feature set exhibits low inter-dependency, even with feature couples having zero correlation, such as *DexClassLoader* and *Dangerous Permissions*.

3.2 Classification

All applications are firstly disassembled into .smali files using Apktool [37] and then features are extracted. In this stage well-known machine learning algorithms are applied in order to classify applications as either benign or malicious based on these features. A decision tree algorithm is used that is named J48 [38], which is an implementation of the C4.5 algorithm. Weka tool [39] is used in the application of the C4.5 algorithm for this study. Other well-known machine learning algorithms such as kNN and SVM are also evaluated in this study. However since J48 produces the highest level of accuracy, it was elected to be employed across all other relevant experiments of this study. J48 is one of the top ten data mining algorithms [40], which also allows us for interpretation of the tree in order to see the features that are best separating malicious and benign applications.

In training, MalGenome dataset is used for representing malicious applications. This dataset contains 1260 malwares from 49 malware families. A benign dataset consisting of 1260 applications downloaded from Google Play was also created. A tool based on Android Market API [41] was utilized in obtaining the benign samples from Google Play. Particular attention was paid to select applications downloaded more than 5 million times to ensure that they were not malicious. Furthermore, they were checked with VirusTotal [42] in order to ensure these samples are not malware. Applications are only included in the dataset if they are not detected by any of the antivirus solutions. Hence the benign dataset is reduced to 978 applications. However the benign dataset is aimed to be extended in the future. In order to see whether or not a generated model could detect new attacks, an evaluation of the models using the Drebin dataset [12] was also conducted. This public dataset contains 5,560 applications from 179 different malware families. Hence, it introduces new malware families that do not exist in MalGenome; however it also contains some samples from MalGenome too. In the experiments conducted for this study, the malware samples also common to MalGenome were extracted to specifically evaluate the detection performance of the generated models on unknown malwares.

4 Evaluation

In order to evaluate the performance of each model, the following metrics were employed: true positive ratio; false positive ratio; and accuracy. True positive ratio shows the ratio of correctly classified malicious applications (TP: true positives) to all malicious applications in the dataset. False positive ratio represents the misclassified applications as malicious (FP: false positives) to all benign applications in the dataset. TN represents true negatives, FN represents false negatives in the equations below.

$$True\ Positive\ Ratio = \frac{(TP)}{(TP + FN)} \tag{1}$$

$$False\ Positive\ Ratio = \frac{(FP)}{(TN + FP)} \tag{2}$$

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{3}$$

Firstly the classifiers were trained by using only API calls which are shown to be very effective against detecting mobile malwares [13]. The effect of the number of API calls can be seen in Fig. 2. The result seen is consistent with DroidAPIMiner's outcome in that 169 API calls produce the highest accuracy. Therefore, 169 API calls were employed for the remainder of the experiments. The top 20 API calls having the highest difference between malware and benign datasets are also given in Fig. 3. The top 20 APIs shows similarity with DroidAPIMiner's results [13], for which analysis was performed on a larger dataset.

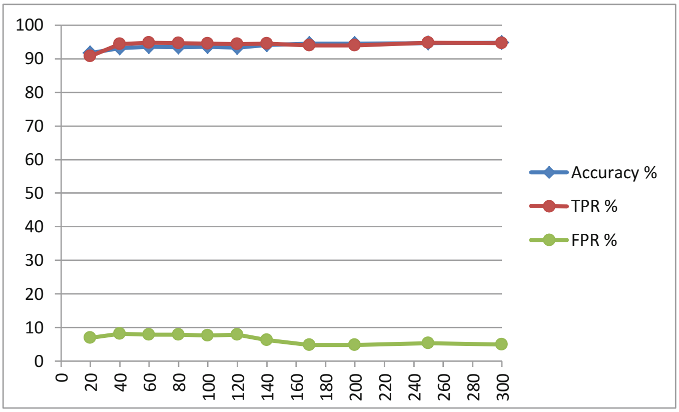


Fig. 2. Effect of the number of API calls

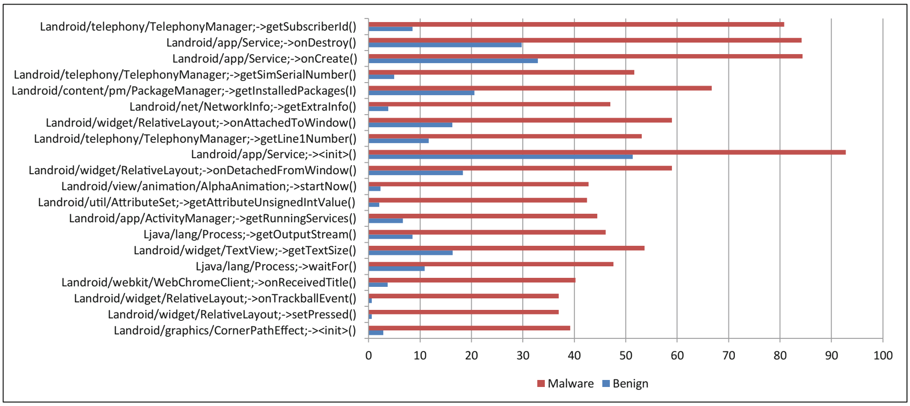


Fig. 3. The top 20 distinguishable API calls

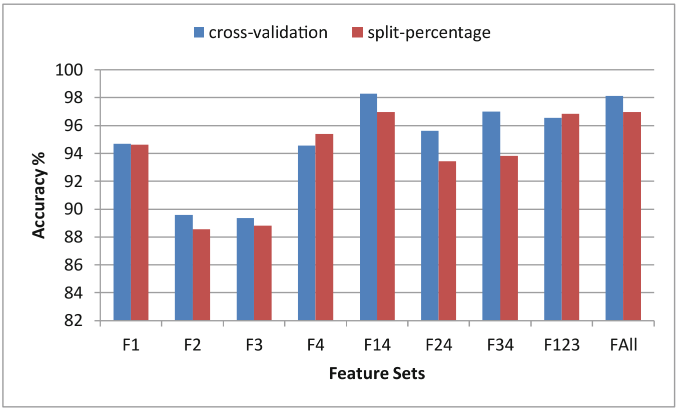


Fig. 4. Percentage of correctly classified instances

Table 3. Performance of classifiers

Feature sets	DR	FPR
F14: Code-based and API calls	98.4%	1.8%
FAll: All	98.3%	2.0%

Classifiers were also trained using different combinations of features and the results compared. F1 represents the code-based features except for API calls, F2 represents manifest-based features, F3 is used for file-related features, and F4 indicates the most distinguishable API calls. Based on the outcome of DroidAPIMiner [13], 169 API calls were employed. Two training schemes are employed: cross-validation and 66% split.

Figure 4 shows the percentage of correctly classified applications. As can be seen, manifest-based and file-related features do not perform as well as code-based features in distinguishing malicious from benign applications. The combination of code-based features and API calls produces the lowest error rate. The performance of this combination is quite close to the combination of all features. Hence, both combinations are evaluated on the new malwares. The detection and false positive ratios of both combinations are exhibited in Table 3.

Finally, in order to see the performance of this approach against new variants of existing malwares and new malwares, evaluation was also conducted against the Drebin dataset [12], which is a larger dataset than MalGenome [43]. It should be noted that, all applications that exist in MalGenome were removed from the test dataset before the evaluation took place. The results are presented in Table 4. In particular, the malware families considered in training are shown. The results show that the proposed approach effectively detects new variants of existing malwares. While API-based approach could not detect some families at all, the new features introduced here make them distinguishable from benign applications. On the other hand, the proposed approach is ineffective against

Table 4. Detection ratio on new malwares

Family	Family size	API	Code-based and API	All
Adrd	24	91.67%	91.67%	91.67%
BaseBridge	22	63.64%	63.64%	72.73%
Bgserv	1	0.00%	0.00%	100.00%
DroidDream	34	79.41%	94.12%	94.12%
DroidKungFu	193	90.76%	94.81%	95.34%
FakePlayer	11	0.00%	90.91%	90.91%
Geinimi	25	80.00%	91.30%	91.30%
GGTrack	2	0.00%	100.00%	100.00%
GinMaster	338	91.41%	96.15%	96.15%
GPSpy	2	0.00%	100.00%	100.00%
JiFake	28	0.00%	89.29%	89.29%
KMin	96	100.00%	100.00%	100.00%
Nickspy	9	77.78%	88.89%	88.89%
Plankton	614	29.48%	2.12%	14.50%
Spitmo	10	100.00%	100.00%	100.00%
TapSnake	2	50.00%	100.00%	100.00%
Yzhc	15	100.00%	100.00%	100.00%
Zitmo	13	69.23%	76.92%	76.92%
Zsone	2	50%	100%	100%
DREBIN	4432	**42.95**%	**70.21**%	**61.86**%

Plankton family, which is a type of update attack. This family could not be detected effectively by applying API-based features as well. Other than Plankton family, we could say that the proposed approach considerably increases the accuracy against new variants of existing malwares and new malware families.

4.1 Limitations

An attacker who knows that the device is protected with SAFEDroid could change its code while preserving its malicious behavior in order to evade the system. For example, it could increase the size of the manifest file and the file size in order to look less suspicious. Actually, it is the common vulnerability for any detection system based on static analysis. Attackers always employ evasive techniques to be successfully installed and run on the device. It is the nature of an arms race mechanism between virus and antivirus systems. It should however be emphasized that the attacker has to change critical features employed in this study in order to evade SAFEDroid, which could cause both a decrease in the impact and an increase in the cost of the attack. For example, an attacker could change the number of permissions requested in the manifest file, and seem non-overprivileged, but then the attacker might not be able to run dynamically loaded code at runtime. An attacker could also uses less *goto* statements by not using evasion techniques such as code reordering, but then might need to use other evasion techniques in order to hide its malicious code. Furthermore, an attacker could not evade API-based features by increasing the size of benign features, since API calls considered here are the most distinguishable API calls from benign applications in malware applications [13]. To summarize, an attacker, who pursues to achieve his goals no matter what happens, could avoid detection by SAFEDroid by changing the features, but that would come with a cost for the attacker.

5 Conclusions

There have been many studies proposed for malware detection on mobile devices. Many of these approaches are based on static analysis due to resource constraints of these devices. Those approaches mainly employe API calls and permissions as features. In this current study, structural features such as the number of methods/classes, the size of the application, and the number of *goto* statements are explored for the purpose of malware detection. In particular, three groups of features are analyzed: code-based, manifest-based, and file-related. The analysis shows that code-based features together with API calls achieve a high detection rate with a low false positive ratio. Furthermore, these novel features are shown to be effective against new malwares and new variants of malwares. The results obtained were an improvement on solely applying API-based features.

Acknowledgements. This study is supported by the Scientific and Technological Research Council of Turkey (TUBITAK-115E150). We would like to thank TUBITAK for its support.

References

1. Kaspersky Lab: The volume of new mobile malware tripled in 2015, March 2016. http://www.kaspersky.com/about/news/virus/2016/The_Volume_of_New_Mobile_Malware_Tripled_in_2015
2. McAfee Lab: McAfee lab threats report, June 2016. https://www.mcafee.com/us/resources/reports/rp-quarterly-threats-jun-2017.pdf, https://www.mcafee.com/us/resources/reports/rp-quarterly-threats-jun-2017.pdf
3. Feizollah, A., Anuar, N.B., Salleh, R., Wahab, A.W.A.: A review on feature selection in mobile malware detection. Digit. Investig. **13**, 22–37 (2015)
4. Shafiq, M.Z., Tabish, S.M., Mirza, F., Farooq, M.: PE-Miner: mining structural information to detect malicious executables in realtime. In: Kirda, E., Jha, S., Balzarotti, D. (eds.) RAID 2009. LNCS, vol. 5758, pp. 121–141. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04342-0_7
5. Symantec: Internet security threat report, April 2016. https://www.symantec.com/content/dam/symantec/docs/reports/istr-21-2016-en.pdf

6. Enck, W., Ongtang, M., McDaniel, P.: On lightweight mobile phone application certification. In: Proceedings of the 16th ACM Conference on Computer and Communications Security, pp. 235–245. ACM (2009)
7. Felt, A.P., Chin, E., Hanna, S., Song, D., Wagner, D.: Android permissions demystified. In: Proceedings of the 18th ACM Conference on Computer and Communications Security, pp. 627–638. ACM (2011)
8. Chin, E., Felt, A.P., Greenwood, K., Wagner, D.: Analyzing inter-application communication in Android. In: Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services, pp. 239–252. ACM (2011)
9. Grace, M., Zhou, Y., Zhang, Q., Zou, S., Jiang, X.: RiskRanker: scalable and accurate zero-day Android malware detection. In: Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services, pp. 281–294. ACM (2012)
10. Seo, S.-H., Gupta, A., Sallam, A.M., Bertino, E., Yim, K.: Detecting mobile malware threats to homeland security through static analysis. J. Netw. Comput. Appl. **38**, 43–53 (2014)
11. Zheng, M., Sun, M., Lui, J.C.S.: Droid analytics: a signature based analytic system to collect, extract, analyze and associate Android malware. In: 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom, pp. 163–171. IEEE (2013)
12. Arp, D., Spreitzenbarth, M., Hübner, M., Gascon, H., Rieck, K., CERT Siemens: DREBIN: effective and explainable detection of Android malware in your pocket. In: Proceedings of the ISOC Network and Distributed System Security Symposium, NDSS (2014)
13. Aafer, Y., Du, W., Yin, H.: DroidAPIMiner: mining API-level features for robust malware detection in Android. In: Zia, T., Zomaya, A., Varadharajan, V., Mao, M. (eds.) SecureComm 2013. LNICST, vol. 127, pp. 86–103. Springer, Cham (2013). https://doi.org/10.1007/978-3-319-04283-1_6
14. Goyal, R., Spognardi, A., Dragoni, N., Argyriou, M.: SafeDroid: a distributed malware detection service for Android. In: 2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA), pp. 59–66. IEEE (2016)
15. Wu, D.-J., Mao, C.-H., Wei, T.-E., Lee, H.-M., Wu, K.-P.: DroidMat: Android malware detection through manifest and API calls tracing. In: 2012 Seventh Asia Joint Conference on Information Security, Asia JCIS, pp. 62–69. IEEE (2012)
16. Androguard, March 2017. https://github.com/androguard/androguard/
17. Yerima, S.Y., Sezer, S., McWilliams, G., Muttik, I.: A new Android malware detection approach using Bayesian classification. In: 2013 IEEE 27th International Conference on Advanced Information Networking and Applications, AINA, pp. 121–128. IEEE (2013)
18. Fereidooni, H., Conti, M., Yao, D., Sperduti, A.: ANASTASIA: Android mAlware detection using STatic analySIs of Applications. In: 2016 8th IFIP International Conference on New Technologies, Mobility and Security, NTMS, pp. 1–5. IEEE (2016)
19. Allix, K., Bissyandé, T.F., Jérome, Q., Klein, J., Le Traon, Y., et al.: Empirical assessment of machine learning-based malware detectors for Android. Empir. Softw. Eng. **21**(1), 183–211 (2016)
20. Shabtai, A., Fledel, Y., Elovici, Y.: Automated static code analysis for classifying Android applications using machine learning. In: 2010 International Conference on Computational Intelligence and Security, pp. 329–333, December 2010

21. Canfora, G., Medvet, E., Mercaldo, F., Visaggio, C.A.: Acquiring and analyzing app metrics for effective mobile malware detection. In: Proceedings of the 2016 ACM on International Workshop on Security And Privacy Analytics, pp. 50–57. ACM (2016)
22. Ban, T., Takahashi, T., Guo, S., Inoue, D., Nakao, K.: Integration of multi-modal features for Android malware detection using linear SVM. In: 2016 11th Asia Joint Conference on Information Security, AsiaJCIS, pp. 141–146. IEEE (2016)
23. Martín, A., Calleja, A., Menéndez, H.D., Tapiador, J., Camacho, D.: ADROIT: Android malware detection using meta-information. In: 2016 IEEE Symposium Series on Computational Intelligence, SSCI, pp. 1–8. IEEE (2016)
24. Feizollah, A., Anuar, N.B., Salleh, R., Suarez-Tangil, G., Furnell, S.: AndroDialysis: analysis of Android intent effectiveness in malware detection. Comput. Secur. **65**, 121–134 (2017)
25. Teufl, P., Ferk, M., Fitzek, A., Hein, D., Kraxberger, S., Orthacker, C.: Malware detection by applying knowledge discovery processes to application metadata on the Android Market (Google Play). Secur. Commun. Netw. **9**(5), 389–419 (2013)
26. Aysan, A.I., Sen, S.: API call and permission based mobile malware detection. In: 2015 23rd Signal Processing and Communications Applications Conference, SIU, pp. 2400–2403. IEEE (2015)
27. Lindorfer, M., Neugschwandtner, M., Weichselbaum, L., Fratantonio, Y., Van Der Veen, V., Platzer, C.: ANDRUBIS-1,000,000 apps later: a view on current Android malware behaviors. In: Proceedings of the the 3rd International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security, BADGERS (2014)
28. Aysan, A.I., Sen, S.: "Do you want to install an update of this application?" A rigorous analysis of updated Android applications. In: 2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing, CSCloud, pp. 181–186. IEEE (2015)
29. Au, K.W.Y., Zhou, Y.F., Huang, Z., Lie, D.: PScout: analyzing the Android permission specification. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security, pp. 217–228. ACM (2012)
30. Moonsamy, V., Rong, J., Liu, S.: Mining permission patterns for contrasting clean and malicious Android applications. Future Gener. Comput. Syst. **36**, 122–132 (2014)
31. Liu, X., Liu, J.: A two-layered permission-based Android malware detection scheme. In: 2014 2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, MobileCloud, pp. 142–148. IEEE (2014)
32. Sheen, S., Ramalingam, A.: Malware detection in Android files based on multiple levels of learning and diverse data sources. In: Proceedings of the Third International Symposium on Women in Computing and Informatics, pp. 553–559. ACM (2015)
33. Zhauniarovich, Y., Ahmad, M., Gadyatskaya, O., Crispo, B., Massacci, F.: StaDynA: addressing the problem of dynamic code updates in the security analysis of Android applications. In: Proceedings of the 5th ACM Conference on Data and Application Security and Privacy, pp. 37–48. ACM (2015)
34. Samra, A.A.A., Yim, K., Ghanem, O.A.: Analysis of clustering technique in Android malware detection. In: 2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IMIS, pp. 729–733. IEEE (2013)

35. Wang, Y., Zheng, J., Sun, C., Mukkamala, S.: Quantitative security risk assessment of Android permissions and applications. In: Wang, L., Shafiq, B. (eds.) DBSec 2013. LNCS, vol. 7964, pp. 226–241. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39256-6_15
36. Shabtai, A., Fledel, Y., Elovici, Y.: Automated static code analysis for classifying Android applications using machine learning. In: 2010 International Conference on Computational Intelligence and Security, CIS, pp. 329–333. IEEE (2010)
37. Apktool. https://ibotpeaches.github.io/Apktool/. Accessed April 2017
38. Quinlan, J.R.: Induction of decision trees. Mach. Learn. **1**(1), 81–106 (1986)
39. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. ACM SIGKDD Explor. Newsl. **11**(1), 10–18 (2009)
40. Wu, X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Philip, S.Y., et al.: Top 10 algorithms in data mining. Knowl. Inf. Syst. **14**(1), 1–37 (2008)
41. Android Market API. http://code.google.com/p/android-market-api. Accessed April 2017
42. Virus Total. https://www.virustotal.com/. Accessed April 2017
43. Zhou, Y., Jiang, X.: Dissecting Android malware: characterization and evolution. In: 2012 IEEE Symposium on Security and Privacy, no. 4, pp. 95–109, May 2012

Manipulating the Five V's in the Next Generation Air Transportation System

Dustin Mink[1(✉)], William Bradley Glisson[1], Ryan Benton[1], and Kim-Kwang Raymond Choo[2,3]

[1] School of Computing, University of South Alabama, 150 Jaguar Drive, Suite 2101, Mobile, Al 36688, USA
dmm1521@jagmail.southalabama.edu, {bglisson,rbenton}@southalabama.edu

[2] Department of Information Systems and Cyber Security, The University of Texas at San Antonio, San Antonio, TX 78249-0631, USA

[3] School of Information Technology and Mathematical Sciences, University of South Australia, Adelaide, SA 5095, Australia
Raymond.Choo@fulbrightmail.org

Abstract. The U.S. Next Generation Air Transportation System (NextGen) is designed to increase the capacity, safety and efficiency of the air traffic control via the integration of past experiences and advances in technology. However, the system is expected to greatly increase the amount and types of data generated as well as the knowledge to be managed. Additionally, as with all new technology, U.S. NextGen opens the specter of the potential impacts created by cyberattacks. Given this, it appears logical to view the U.S. NextGen system from the lens of Big Data. This study evaluates the U.S. NextGen system using the five differentiated qualitative characteristics of big data: Volume, Velocity, Variety, Veracity and Value. The results indicate that U.S. NextGen system has several big data challenges that must be addressed in order to obtain its maximal potential.

Keywords: NextGen · ADS-B · IoT · Big data · Cybersecurity

1 Introduction

The impact of the aviation industry in today's globally integrated societies is evident from both economic and governmental perspectives. A 2016 report by the U.S. FAA indicates U.S. Gross Domestic Product (GDP) will increase from $16.3 trillion U.S. dollars in 2015 to $26.2 trillion in 2036 [1]. Furthermore, the world GDP is forecasted to increase from 74.4 trillion U.S. dollars in 2015 to $136.3 trillion in 2036.

While the issue of funding security is always challenging particularly in a tight fiscal climate [2], the escalation of cyber-security concerns in the aviation environment, from the government perspective, is very visible through legislative activities like the Senate subcommittee approving a bill to investigate aviation security and cybersecurity [3]. An article on the World Economic Forum highlights the fact that the proliferation and equalization of technology accessibility increases the potential number of attackers [4].

X. Lin et al. (Eds.): SecureComm 2017 Workshops, LNICST 239, pp. 271–282, 2018.
https://doi.org/10.1007/978-3-319-78816-6_19

It also goes on to note that the integration of cyber and physical environments not only create new vulnerabilities but, potentially, has extensive impacts in the aviation industry. The importance of cybersecurity is reinforced in incidents such as those involving Brussels' airport [5], MH17 in the Ukraine [6] and the missing Malaysia Flight ML370 [7].

In an attempt to mitigate security concerns, the U.S. Government Accountability Office [8] states that the aviation industry is in the process of rolling out the U.S. Next Generation (NextGen) Air Traffic System. While all of the U.S. NextGen component programs are at various stages of development, they are targeted to be operational no later than the 2020 [8]. U.S. Government Accountability Office (US GAO) indicates that the U.S. NextGen system is, currently, comprised of six parts, namely: Automatic Dependent Surveillance Broadcast (ADS-B), Collaborative Air Traffic Management Technologies (CATMT), Data Communication, National Airspace System Voice System, U.S. NextGen Air Transportation System Weather, and System Wide Information Management. According to the US GAO, a major element of this system is the ADS-B capability, which is directed to be the future of air traffic control through advancements in aircraft tracking and flow management. They also state that the U.S. NextGen ADS-B messages are sent continually every five seconds. Furthermore, there are three different ADS-B message types, namely: position messages, velocity messages, and identification messages. CATMT is the program that is responsible for enhancing the existing traffic flow management system and subsequently will have to handle the volume of data the ADS-B will be producing [8]. Complicating matters, there are documented exploitations of ADS-B system [9–11]. Hence, spoofing aircraft with fake ADS-B messages is a viable concern. Fingerprinting aircraft transponders transmitting ADS-B and cross referencing with aircraft equipment transponders allows for the inference of airline communications. This environment prompted the idea that the ADS-B message system should be examined from the perspective of the five differentiated qualitative characteristics of big data, namely: Volume, Variety, Velocity, Variability, and Value [12]. In this environment, each aircraft can be thought of as a very complex device or node that communicates with other aircraft and Air Traffic Control Facilities (ATCF). The goals are two-fold. First, identify the big data issues within the U.S. NextGen Air Transportation System architecture. Second, understand which of the five differentiated qualitative characteristics apply to the unique U.S. NextGen Air Transportation System to categorize big data issues.

The next section summarizes the relevant works within a big data and the U.S. NextGen Air Transportation System context. In Sect. 3, we discuss the hypothesis: *Does the U.S. NextGen Air Transport System have unaddressed big data issues?* Section 4 examines each of the five-differentiated qualitative big data characteristics within the context of the U.S. NextGen architecture. Finally, the last section presents conclusions and identifies future U.S. NextGen system research from a big data perspective.

2 Relevant Literature

The increasing amalgamation of technology into the aviation industry is stimulating research interest into the possible risk associated with the U.S. NextGen Air Traffic

System. Interest in this area is being encouraged through the continued escalation of residual data in legal environments [13, 14] along with an absence of clarity on conducting aircraft forensics investigations [15]. Coupling this with the increasing capabilities of technology that allow a single entity/node to generate vast volumes of data quickly, U.S. NextGen Air Traffic System starts to resemble a big data problem, especially when multiple entities/nodes are considered from a real-world perspective. This is supported further when one considers the variety of research interests pertaining to NextGen, which range from *communication* data flow [16] and *encryption* [17], to *cyber-physical systems*, to the *Internet of Things (IoT)* [18], to *big data* applications [19], to *defense-in-depth* [20], and so on.

From a *communication* perspective, many researchers agree that the ADS-B system is the most important program out of the ten programs that make up the configuration of the U.S. NextGen Air Transport System [17, 21–23]. Aircraft will be required to be equipped with ADS-B systems to transmit messages to other aircraft and Air Traffic Control Centers. The unencrypted structure of the ADS-B system means the National Airspace System is susceptible to breath of cyber-physical attacks. As He, et al. [17] noted, an important objective of the ADS-B system is the security of the National Airspace System by 2020. To address both authentication and integrity issues they proposed a "three-level hierarchical identity-based signature" solutions. However, a key limitation in the scheme of He et al. [17] is the sending of identities in plaintext, which could be exploited by attackers.

The *unencrypted structure* of the ADS-B system means the national airspace system is susceptible to variety of cyber-physical attacks [11]. OpenSky is a sensor network in Central Europe, which can capture 30% of the European air traffic communications on ADS-B. The ADS-B system can augment traditional means of surveillance: radar and transponders. Radars can indicate there is something in the sky the same size as an aircraft, while a transponder will broadcast or squawk the identity of the aircraft when activated. An ADS-B message field can contain information on traffic, weather, and flight information. ADS-B vulnerabilities transgress confidentiality, integrity, and availability. First, anyone with an ADS-B radio can transmit and receive messages showing no signs of confidentiality. Data integrity is affected by attacks such as Ghost Aircraft Injection, Aircraft Disappearance, Virtual Trajectory Modification, and Aircraft Spoofing. Ghost Aircraft Injection occurs when an ADS-B radio transmits a fake message and other aircraft now believe there is an aircraft that does not really exist. Aircraft Disappearance happens when skillfully timed malformed ADS-B messages are sent with a real aircraft's identification, resulting in ADS-B messages with the particular aircraft to be disregarded. In other words, the remaining aircrafts do not believe this particular aircraft exists. Virtual Trajectory Modification is the act of jamming an aircraft or ground station to create false alarms. Aircraft Spoofing is simply using another aircraft's identification to send ADB-S message with false information. Finally, availability is loss associated with Ground Station and Ghost Aircraft Flooding. Ground Station Flooding occurs when ground-based radios are jammed. Ghost Aircraft Flooding happens when a large number of fake ADB-S messages are sent that there are too many real and fake aircrafts that nothing is distinguishable. No solutions were presented on how to address the unencrypted structure of the ADS-B system.

From an *IoT* perspective, Varga et al. [18] presented a solution for a real-time air traffic monitoring and tracking system that is based upon the ADS-B system. The solution is implemented via a software defined radio, integrating hardware and software into a high-performance wireless communication system. The software defined radio solution, however, does not allow for the use of multiple radios or the correlation of data between systems.

From a *big data* perspective, researchers are beginning to investigate architectural solutions for analyzing ADS-B records. Boci and Thistlethwaite [19] developed a Hadoop-based solution that can be used to analyze billions of ADS-B radio messages in approximately 35 min. The results of their research are visualized using density maps. However, the maps produced are very busy. It would be beneficial, from a security (or forensic) perspective, to be able to filter the messages on key words or phrases to reduce noise [24]. As the authors noted, a reduction in computational times would assist with enormous data asset as well as assisting with real time processing aspirations [19, 24].

Other researchers are beginning to look at U.S. NextGen Air Transportation systems from the perspective of *defense-in-depth* [20]. The research recommends the Flight Information Exchange Model based on experience with the Mini Global II for the advancement of the U.S. Federal Aviation Administration NextGen Air Transportation System Wide Information Management. The research is to extend the Flight Information Exchange Model beyond the 3.0 version for the benefit of the public and private organizations. The Mini Global II is part of the Federal Aviation Administration and international aviation community to unite sharing of flight, weather, and aeronautical data. The research demonstrates how the International Civil Aviation Organization Flight and Flow Information for a Collaborative environment could be leveraged to share information on a global scale to the Air Navigation Service Providers and Air Transportation Operators. The research version of the Mini Global II (e.g. Flight Information Exchange Model) includes the Weather Exchange Information Model and Aeronautical Information Exchange Model. The expanded Global Enterprise Messaging Services Support Air Navigation Service Providers' Flight Operations Centers. The simulated global environment allows for the testing of the Flight Information Exchange Model for data collection and exchange. The results indict development needs to use of the exchange model for flight objects.

While existing literature on U.S. NextGen security focuses to a large degree on communications, cyber-physical vulnerabilities, and IoT perspectives, there is minimal research investigating U.S. NextGen air transportation systems from a big data perspective.

3 Methodology

In order to investigate the U.S. NextGen system from a big data perspective, we use a case study research strategy. Specifically, this involves a documentation data generation method along with quantitative data analysis, as defined by Oates [25]. Key concepts in big data defined by Katal et al. [12] are the five characteristics, also known as the 5 v's of big data, namely: volume, variety, velocity, veracity, and value.

- Data volume measures the scale of the data within the system;
- Data variety refers to the different structures and sources of data;
- Data velocity is the analyzation of the data as the data is generated;
- Data veracity illustrates the uncertainty of the data; and
- Data value is the evaluation of the impact the data has on research.

We posit that the U.S. Next Generation Air Transport System has unaddressed big data issues; thus, we seek to obtain a better understanding of the following research challenges.

Q1: Can the combined ADS-B messages within the U.S. National Airspace system be stored with current storage technologies?

Q2: Can the combined ADS-B messages within the U.S. National Airspace system be processed with current processing technologies?

Q3: Are there too many ADS-B message formats, which creates undue complexity of the processing unit?

Q4: Are there cybersecurity issues with the ADS-B that create uncertainty about the data being transmitted?

Q5: Is the U.S. NextGen system capable of providing timely analysis in order to meet its maximum potential in enhancing public safety of air transportation?

4 Analysis and Results

The research results are described using the five considerations of big data, namely: volume, velocity, variety, veracity, and value.

4.1 Volume

The volume is calculated for the ADS-B system using the size of the message, the rate messages are sent, the amount of aviation flight hours, and a conversion factor from hours to seconds. The ADS-B systems uses fixed length 112 Bytes messages [26], and averages 6.2 messages every second from an individual aircraft. In 2015, U.S. recorded 18,103,000 general aviation flight hours [27]. Finally, there are 3,600 s in one hour. This results in 41TiB per a one year time frame, as seen in the following calculation:

$$\begin{aligned}(112\,\text{Bytes/Message}) \ast (6.2\,\text{Messages/Second}) \ast (3{,}600\,\text{Second/Hour})\\ \ast (18{,}103{,}000\,\text{Flight Hours/Year}) = 41\,\text{TiB/Year}\end{aligned} \tag{1}$$

The combined ADS-B messages within the U.S. National Airspace system can be stored with current storage technologies. It should be noted, however, that another study [19], processed CAT033 messages that were generated from ADB-S signals received by 71 radio stations in March 2014. Compressed, this dataset size was approximately 4 TB. Given that the stations only cover a small part of the country, there does seem to be a mismatch in data generated and data stored. This could be due to the adding of additional meta-data, overlap of stations and so forth. While still in bounds with conventional storage, it does point to potential issues of assuming that the source transmittions are indicative of archival size. It should also be noted that the data collection, to our knowledge, assumes that the data is trustworthy and accurate.

Additionally, it should be noted that simply storing data does not facilitate data analysis. Hence, while the storage of the raw information can be achieved with current technologies, it is important to ensure the data is stored in a means to facilitate analysis (the rationale behind Big Data). Marsh and Ogaard [28] noted much of the information stored in the ADS-B data they received was not relevant to their analysis. Moreover, the data they received were organized in files based upon the receiving stations; hence, to track a flight, it would be often necessary to search through multiple files. To extract the relevant data, and preprocess it to be amendable to analysis, took approximately three hours; the raw data was approximately 22 gigabytes in size. Thus, in order to facilitate timely access and retrieval of the ADS-DB data for analysis, the data will need to be stored in databases, with various fields (and combination of fields) being indexed to support anticipated types of analysis. Other precomputed operations may include the ability to search and retrieve based upon aggregation of certain data elements. This, of course, adds to the storage and other costs.

4.2 Velocity

The velocity is calculated for the ABS-D system by using the size of the message, the rate message is sent, the average amount of flights in the National Air Space at any given time. An average of 7,000 flights in the U.S. National Air Space at any given time [27] results in 404,058,960,000 messages per year, as shown in the next two equations.

$$\begin{aligned} &(6.2\ \text{Messages/Second}) * (60\ \text{Seconds/Minutes}) * (60\ \text{Minutes/Hour}) \\ &* (18{,}103{,}000\ \text{Flights Hours/Year}) = (404{,}058{,}960{,}000\ \text{Messages/Year}) \end{aligned} \tag{2}$$

$$\begin{aligned} &(404{,}058{,}960{,}000\ \text{Messages/Year})/\left(\begin{aligned} (365\ \text{Days/Year}) * (24\ \text{Hours/Day}) \\ * (60\ \text{Minutes/Hour}) * (60\ \text{Seconds/Minute}) \end{aligned}\right) \\ &\qquad = (\sim 13\ \text{Messages/Millisecond}) \end{aligned} \tag{3}$$

The combined ADS-B messages within the U.S. National Airspace system cannot be processed efficiently in real-time with existing standard processing technologies. A proposed ADS-B Data Lake Architecture used to process one month of messages covering the en route air traffic for Boston, New York, and Washington DC [19] took over 35 min. This dealt with approximately 17 million ADS-B messages sent at the 1090 channel; or approximately only 0.001% of the total expected volume of ADS-B messages. Assuming there is any real-time need to collect, process, compare and transmit results to other locations, this can become a true bottleneck in the process.

4.3 Variety

One means in which variety is shown within the U.S. NextGen Air Transportation is through the multitude of message type [17, 18, 29]. The message types of U.S. NextGen Air Transportation are Mode A, Mode C, Mode S, and ADS-B In and Out. Mode S, in turn, has three message types, which are (a) Data Block Surveillance Interrogation and Reply Message Format, (b) Data Block Surveillance and Communication Interrogation and Reply-Communication-A and Communication-B Message Format, and (c) Data Block Surveillance Communication Interrogation and Reply-Extended Length Message Format. The ADS-B system inherits its message types from Mode S; hence, ADS-B has three different message types.

The Mode S Data Block Surveillance Interrogation and Reply Message Format comprises of three parts, which is displayed in Table 1. The three parts are Format Number, Surveillance and Communication Control, and Address and Parity; the format is also displayed in Table 1. The Format Number is a 5-bit message representing the sequence number of the message. The Surveillance and Communication Control is a 27-bit message, which includes commands and flight information. The Address and Parity is a 24-bit message intended to represent a unique aircraft identifier.

Table 1. Mode S data block surveillance interrogation and reply message format.

Format number	Surveillance and communication control	Address and parity
5-bits	27-bits	24-bits

The Mode S Data Block Surveillance and Communication Interrogation and Reply-Communication-A and Communication-B Message Format comprises four parts, which are shown in Table 2. The four parts of the Mode S Data Block Surveillance and Communication Interrogation and Reply-Communication-A and Communication-B Message Format are Format Number, Surveillance and Communication Control, Message Field, and Address and Parity.

Table 2. Mode S data block surveillance and communication interrogation and reply – communication–A and communication-B message format.

Format number	Surveillance and communication control	Message field	Address and parity
5-bits	27-bits	56-bits	24-bits

The Format Number is a 5-bit message representing the sequence number of the message. The Surveillance and Communication Control is a 27-bit message, which includes commands and flight information. The Message Field is a 56-bit that contains additional flight information. The Address and Parity is a 24-bit message intended to represent a unique aircraft identifier.

The Mode S Data Block Surveillance Communication Interrogation and Reply-Extended Length Message Format comprise four parts: Format Number, Communication Control, Message Field, and Address and Parity (see Table 3). The Format Number is a 2-bit message representing the sequence number of the message. The Communication Control is a 6-bit message, which includes commands. The Message Field is an 80-bit contains additional flight information. The Address and Parity is a 24-bit message intended to represent a unique aircraft identifier.

Table 3. Mode S data block surveillance communication interrogation and reply – extended length message format.

Format number	Communication on control	Message field	Address and parity
2-bits	6-bits	80-bits	24-bits

While the varying length message format is an asset where data transmission and storage is concerned, the varying length message formats creates additional complexity for processing, similar to that of the Complex Instruction Set Architecture (CISC). CISC uses varying length instruction, while Reduced Instruction Set Architecture (RISC) uses fixed length instructions. CISC saves on the storage of the instructions, but additional complexity resides within the processor to decode the varying length instructions. The fixed length instructions of the RISC processor suffer from internal fragmentation because of the unused space within the instruction format. However, the processor only processes a one size instruction, reducing the complexity on the processor. In this case, the ADS-B protocol favored optimizing storage over reducing complexity.

Aside from the variability in the messages themselves, it has been noted that the formats used to store ADS-B formats vary. As noted earlier, the study conducted by [19] used CAT033 messages that contained ADS-B data. Marsh and Ogaard [28] received ADS-B data from around the world. However, they noted the three storage formats received were “Comma-Separated Value (CVS), Extensible Markup Language (XML) and the binary format used by Garmin GDL 90 ADS-B transceiver”. Hence, the Automatic Dependent Surveillance system can be viewed as having multiple tiers of variety.

4.4 Veracity

The veracity is depicted by the known and peer-reviewed security vulnerabilities within the ADS-B protocol. The vulnerabilities to the ADS-B system include ground station flooding, ghost aircraft injection or flooding, aircraft disappearance, virtual trajectory modification or false alarm attack, and aircraft spoofing [11, 30]. Ground station flooding is the jamming of the 1090 MHz frequency. The exploitation of the ground station flooding vulnerability has a low level of difficulty. The attacker is required to have a signal power greater than the legitimate communications to the Area Control Center. The exploitation would require the Area Control Center to use a legacy system incapable of handling high density airspaces. Ghost aircraft injection or flooding is the insertion of an ADS-B message spoofing an existing aircraft. The scaling of ghost aircraft injection into many ghost aircraft injections is called ghost aircraft flooding. The injected messages are indistinguishable from the legitimate communications. The ghost aircraft flooding causes a denial of service. Aircraft disappearance is caused by the deletion of all ADS-B messages sent from a legitimate aircraft. Virtual trajectory modification or false alarm attack is achieved by combining an illegitimate message with illegitimate modified trajectory date with the legitimate and valid 24-bit International Civil Aviation Organization identifier. Aircraft spoofing is accomplished by combining the illegitimate message with the valid 24-bit International Civil Aviation Organization identifier of the legitimate aircraft being spoofed.

Hence, it is safe to conclude that there are cybersecurity issues with the ADS-B system such as the lack of integrity demonstrated by the vulnerabilities to the ADS-B. This creates uncertainty about the data being transmitted, which in turn, indicates that veracity is an issue. It should also be noted, in this case, that the volume of the data and velocity of data, as well as the distributed nature of the collection and storage, exasperate the problem of verifying the data veracity. Attempts to mitigate the veracity concerns include two mitigation solutions: intrusion detection [9, 31] and cryptographic solution implementation [32–34]. However, the problem is still not definitively solved.

4.5 Value

The value is shown through the lens of public safety. U.S. NextGen Air Transportation aims to improve safety, increase efficiency and capacity. Aviation Safety Information Analysis and Sharing creates an aggregate of data from industry and government. One use of the aggregate data is to detect safety tendencies. Aviation Safety Information Analysis and Sharing is used by incident responders to replay the events leading to an incident. The data points are derived from surface monitoring systems. System Safety Management and Transformation allows for visualization of safety trends and further analysis is used for forecasting. The System Wide Information Management system creates an interconnection between otherwise unshared information, which could enhance public safety of air transportation.

A key issue in value is the timeliness of the analysis. Hence, for the air traffic controller, determining that a contact is a matter of Ghost Aircraft Injection requires a system that can analyze, within seconds, the array of historical and current, to determine the likelihood of the contact actually being true. In the case of determining if a Ghost Aircraft Injection occurred as a postmortem of a security alert, the value of the analysis does not decrease if it take a few minutes. Thus, the question if the Next Generation Air Transportation has a value problem, in terms of big data, becomes a rather complex determination of what questions need to be answered, when they need to be answered and by whom needs the data. Given this complexity, at present, the question is not resolvable at this time.

Not all Big Data considerations were addressed by the U.S. NextGen Air Transportation System, as shown in Table 4.

Table 4. Results from the characteristics of big data.

Characteristics of big data	Results
Volume	(41 TiB/Year)
Velocity	(~13 Message/ms)
Variety	Mode A, Mode C, and Mode S
Veracity	No encryption
Value	Public safety

While the 41 TiB per year volume is manageable, the results only address existing ADS-B systems. One would expect more volume from a voice system, which will be provided by the U.S. NextGen Air Transportation System Data Communication. Unlike Twitter, the Federal Aviation Agency would have to address and processes voice communication. There is variety within variety for the U.S. NextGen Air Transportation System.

5 Conclusions and Future Work

In this paper, we explained the unaddressed big data issues in the U.S. NextGen Air Transport System. For example, We pointed out that the System Wide Information Management does not address the veracity of the data received via the ADS-B protocol, which is untrustworthy due to the lack of encryption for both confidentiality and integrity. Potential mitigation solutions include intrusion detection and Public Key Infrastructure implementation. The goal of the research, to identify Big Data issues with the U.S. NextGen Air Transport System, was achieved by identifying issues with the velocity, variety, and veracity of the U.S. NextGen Air Transport System.

Future work will investigate the creation of a U.S. NextGen Air Transportation System command and control model to address the outlined big data issues, namely: velocity, variety, and veracity. In order to add in command and control models, each of the remaining five parts of the U.S. NextGen system, plus the overall system, will be analyzed from the Big Data perspective. The research will need to identify combinations that pose unique challenges and problems from the 5 V perspective. In addition, the applicability of these newly created command and control models will need to be examined for automobile and drone environments. As the U.S. Department of Transportation progresses in its effort to automate automobiles, many of the lessons learned within the U.S. NextGen Air Transportation System may be applicable to ground transportation infrastructures. Future research will also examine the viability of adopting these command and control models to Unmanned Arial Vehicles (UAV) environments.

Another potential research agenda is to integrate forensic requirements and techniques into the design of the U.S. NextGen Air Transportation System. Such an approach, coined forensic-by-design [35], can facilitate the identification, collection and analysis of data during forensic investigations on a cybersecurity incident [36, 37].

References

1. U.S. Department of Transportation: Fact Sheet - FAA Forecast Fact Sheet-Fiscal Years 2016–2036, 2017 (2016)
2. Gillen, D., Morrison, W.G.: Aviation security: costing, pricing, finance and performance. J. Air Transp. Manage. **48**, 1–12 (2015)
3. Committee on Appropriations: FY2017 Homeland Security Appropriations Bill Cleared for Committee Debate, 2017 (2016)
4. Kaspersen, A.: Four threats to aviation security – and four responses, 2017 (2016)
5. BBC News: Brussels explosions: what we know about airport and metro attacks, 2017 (2016)
6. BBC News: MH17 Ukraine plane crash: what we know, 2017 (2016)
7. AirlineReporter: Updated: Malaysia Airlines Flight 370 Has Likely Crashed But Where? 2017 (2014)

8. United States Government Accountability Office: FAA Needs a More Comprehensive Approach to Address Cybersecurity As Agency Transitions to NextGen, 2017 (2015)
9. Strohmeier, M., Martinovic, I.: On passive data link layer fingerprinting of aircraft transponders. In: Proceedings of the First ACM Workshop on Cyber-Physical Systems-Security and/or PrivaCy, pp. 1–9. ACM, Denver (2015)
10. Costin, A.: Ghost is in the air (traffic): on insecurity of ADS-B protocol and practical attacks on ADS-B devices. Black Hat USA (2012)
11. Strohmeier, M., Schafer, M., Lenders, V., Martinovic, I.: Realities and challenges of nextgen air traffic management: the case of ADS-B. IEEE Commun. Mag. **52**, 111–118 (2014)
12. Katal, A., Wazid, M., Goudar, R.H.: Big data: issues, challenges, tools and good practices. In: 2013 Sixth International Conference on Contemporary Computing (IC3), pp. 404–409 (2013)
13. Berman, K., Glisson, W.B., Glisson, L.M.: Investigating the impact of global positioning system (GPS) evidence in court cases. In: Hawaii International Conference on System Sciences (HICSS-48). IEEE, Kauai (2015)
14. McMillan, J., Glisson, W.B., Bromby, M.: Investigating the increase in mobile phone evidence in criminal activities. In: Hawaii International Conference on System Sciences (HICSS-46). IEEE, Wailea (2013)
15. Mink, D., Yasinsac, A., Choo, K.-K.R., Glisson, W.B.: Next generation aircraft architecture and digital forensic. In: Americas Conference on Information Systems (AMCIS). Americas Conference on Information Systems, San Diego (2016)
16. Moallemi, M., Castro-Peña, C.A., Towhidnejad, M., Abraham, B.: Information security in the aircraft access to system wide information management infrastructure. In: 2016 Integrated Communications Navigation and Surveillance (ICNS), pp. 1A3-1–1A3-7 (2016)
17. He, D., Kumar, N., Choo, K.K.R., Wu, W.: Efficient hierarchical identity-based signature with batch verification for automatic dependent surveillance-broadcast system. IEEE Trans. Inf. Forens. Secur. **12**, 454–464 (2017)
18. Varga, M., Polgár, Z.A., Hedeşiu, H.: ADS-B based real-time air traffic monitoring system. In: 2015 38th International Conference on Telecommunications and Signal Processing (TSP), pp. 215–219 (2015)
19. Boci, E., Thistlethwaite, S.: A novel big data architecture in support of ADS-B data analytic. In: 2015 Integrated Communication, Navigation and Surveillance Conference (ICNS), pp. C1-1–C1-8 (2015)
20. Li, W., Kamal, P.: Integrated aviation security for defense-in-depth of next generation air transportation system. In: 2011 IEEE International Conference on Technologies for Homeland Security (HST), pp. 136–142 (2011)
21. Samuelson, K., Valovage, E., Hall, D.: Enhanced ADS-B research. In: IEEE Aerospace Conference, pp. 1–7 (2006)
22. Robinson, R.V., Sampigethaya, K., Li, M., Lintelman, S., Poovendran, R., Oheimb, D.V.: Secure network-enabled commercial airplane operations: it support infrastructure challenges. In: First CEAS European Air Space Conference, pp. 1–10 (2007)
23. Kacem, T., Wijesekera, D., Costa, P.: Integrity and authenticity of ADS-B broadcasts. In: IEEE Aerospace Conference, pp. 1–8 (2015)
24. Tassone, C.F.R., Martini, B., Choo, K.-K.R.: Visualizing digital forensic datasets: a proof of concept. J. Forensic Sci. **62**, 1197–1204 (2017)
25. Oates, B.J.: Researching Information Systems and Computing (2006)
26. Dong, X.L., Srivastava, D.: Big data integration. In: IEEE 29th International Conference on Data Engineering (ICDE), pp. 1245–1248 (2013)

27. U.S. Department of Transportation's Bureau of Transportation Statistics: Transportation Statistics Annual Report 2016 (2016)
28. Marsh, R., Ogaard, K.: Mining heterogeneous ADS-B data sets for probabilistic models of pilot behavior. In: IEEE International Conference on Data Mining Workshops, pp. 606–612 (2010)
29. Finke, C., Butts, J., Mills, R.: ADS-B encryption: confidentiality in the friendly skies. In: Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop, pp. 1–4. ACM, Oak Ridge (2013)
30. Chen, T.-C.: An authenticated encrption scheme for automatic dependent surveillance-broadcast. IEEE Commun. Mag. (2012)
31. Baek, J., Young-jj, B., Hableel, E., Al-Qutavri, M.: Making air traffic surveillance more reliable: a new authentication framework for automatic dependent sureillance-broadcast (ADS-B) based on online/offline identity-based signature. Secur. Commun. Netw. **8**, 740–750 (2015)
32. Lauf, A.P., Peters, R.A., Robinson, W.H.: A distributed intrusion detection system for resource-constrained devices in ad-hoc networks. Ad Hoc Netw. **8**, 253–266 (2010)
33. Mitchell, R., Chen, I.-R.: A survey of intrusion detection techniques for cyber-physical systems. ACM Comput. Surv. (CSUR) **46**, 55 (2014)
34. Wesson, K.D., Humphreys, T.E., Evans, B.L.: Can cryptography secure next generation air traffic surveillance. IEEE Secur. Priv. Mag. (2014)

A Framework for Acquiring and Analyzing Traces from Cryptographic Devices

Alfonso Blanco Blanco[1], Jose María de Fuentes[2], Lorena González-Manzano[2],
Luis Hernández Encinas[1](✉), Agustín Martín Muñoz[1],
José Luis Rodrigo Oliva[1], and J. Ignacio Sánchez García[1]

[1] Departamento de Tecnologías de la Información y las Comunicaciones,
Instituto de Tecnologías Físicas y de la Información,
Consejo Superior de Investigaciones Científicas, Madrid, Spain
{alfonso,luis,agustin,joseluis.rodrigo,nacho.sanchez}@iec.csic.es

[2] Departamento de Ciencias de la Computación,
Universidad Carlos III de Madrid, Leganés, Madrid, Spain
{jfuentes,lgmanzan}@inf.uc3m.es

Abstract. We present a Side-Channel Analysis Platform (SCAP) Framework developed to acquire and study the traces derived from a cryptographic device when cryptographic computations are done. The main goal of this work is to develop a tool for performing side-channel attacks against these cryptographic devices. The characteristics of the SCAP Framework are described and a case study with a smartphone is presented.

Keywords: Android · Power consumption · RSA traces
Side-channel attack · Smartphone

1 Introduction

The uses of electronic devices with cryptographic features in order to perform personal and business operations has largely increased worldwide in the last years. In general, most of these uses (identification, payment with credit cards, access to the cloud, browsing, etc.) are relevant enough to ensure important security measures.

The cryptographic community considered that the security of a cryptosystem lied in the strength of the mathematical problem in which it was based on. For instance, the security of the RSA algorithm is based on the difficulty of solving the integer factorization problem [1], or the strength of the elliptic curve cryptography (ECC) is based on the intractability of the discrete logarithm problem over elliptic curves defined on finite fields [2].

Nevertheless, this paradigm has been questioned since the publication in 1996 of a paper by Kocher [3]. Kocher demonstrated that it was possible to break the

X. Lin et al. (Eds.): SecureComm 2017 Workshops, LNICST 239, pp. 283–300, 2018.
https://doi.org/10.1007/978-3-319-78816-6_20

security of embedded cryptographic protocols by means of an attack which, instead of trying to solve the underlying mathematical problem, made use of the information obtained from the cryptographic device, i.e., from the fact that the cryptosystem was physically implemented in such device. Kocher showed that measuring the time required to perform private key operations could be useful to find Diffie-Hellman exponents, to factor RSA keys, etc.

Nowadays, the continuous development of device-implemented cryptography [4] is accompanied by an increasing number of physical attacks [5–7].

The information obtained from the implementation of a cryptographic algorithm in a device is related to the execution of the code and can be obtained by measuring the computation time that the algorithm takes to execute [3], the consumption of electric power to execute the process that is running [8], the generation of electromagnetic fields during their computations [9,10], the temperature reached by the chip [11], the noise produced while doing calculations [12], etc. All channels which permit to obtain extra information about the cryptographic processes are called *side channels* and the attacks derived from the information obtained are denoted as *side-channel attacks*. Note that these attacks are passive in the sense that they do not modify the device where computations are done; hence, they are difficult to be detected.

Side-channel attacks consider that the mentioned measurable quantities depend on the instructions, mathematical operations, and the data used by the processor to perform its cryptographic computations. In this way, if the implementation is not sufficiently protected it is possible to obtain information related to the keys through these side channels.

On the contrary, when the attacker provokes a fault or malfunctioning in the device due to the alteration of its normal execution such as modifying the temperatures accepted by the device, triggering a laser that alters the memory contents or the execution flow of an algorithm, etc. [13], then we are considering *fault-injection attacks*.

In this work we present SCAP (Side-Channel Analysis Platform) framework, a framework to develop a tool for acquiring and studying the obtained traces when a cryptographic device is carrying out cryptographic computations. To illustrate the benefits of our framework we have applied it to the acquisition of the power consumption traces of an implementation of the RSA cryptosystem over a smartphone. We have selected the RSA cryptosystem because it is the most extended system nowadays. Nevertheless, the framework also includes an implementation of the ElGamal (EG) cryptosystem. In a general way, the framework could be easily extendable to any other cryptosystem as ECC, for example. Our framework has been developed in LabView$^{\mathrm{TM}}$.

The rest of this paper is organized as follows. In Sect. 2, a short review about side-channel attacks is presented. Section 3 presents our Framework, including a description of the smartphone interface and the used hardware and software. In Sect. 4, the experimental results obtained with the mentioned hardware and software are shown. Finally, Sect. 5 includes the conclusions of the work and possible future work.

2 Side-Channel Attacks

As we have mentioned, problems related to the security of implementations arise due to the existence of side channels on the device from which it is possible to obtain sensitive information, analyzing either the behavior of the software or the hardware, and inducing faults in the behavior of the circuit to deduce information about the keys.

These attacks on physical devices are more specific than the classic ones since they are carried out depending on the algorithm implementation, the chip architecture, etc. They are classified as invasive, semi-invasive or non-invasive [14,15], depending on whether the device is manipulated or only the available information is used. Another classification of the attacks is: active or passive, depending on whether they manipulate the device or only observe its behavior, respectively.

Finally, it is noteworthy that, as in traditional attacks, it is assumed that the Kerckhoffs principle [16] is verified, i.e., the starting point is that the attacker has access to the device, he knows the cryptographic algorithm implemented in the chip, the characteristics of that implementation, and he can execute the protocol with the parameters he estimates appropriate as many times as desired.

The main types of attacks against physical devices are summarized below.

2.1 Timing Analysis

The attacks for *timing analysis* try to obtain information about a cryptographic protocol by measuring the time that the attacked physical device takes in performing the operations of the algorithm being attacked [3].

For example, in the case of the RSA cryptosystem, the attacker wants to obtain the private key while executing the operation of the modular exponentiation. This computation is carried out by the squaring and multiplying algorithm. The initial hypothesis is that the time required for a multiplication is constant, but if a modular reduction must be performed due to the fact that the multiplication is greater than the module, then this new operation implies an increase in the execution time. This way, it is possible to obtain some information about the sizes of the numbers considered in each step of the algorithm.

2.2 Power Analysis

In some cases, certain information can be extracted by measuring the power consumption of the microprocessors. This consumption can be closely related, for example, to the number of bits that change in memory or register. Thus, an attacker can take advantage of this feature to try to guess a secret value used in a cryptographic operation by observing the power consumption trace, for example.

There are several attack methods related to this side channel. The most simple methods are the *Simple Power Analysis* (SPA) attacks. These attacks use the power consumption traces measured when the cryptographic device is working. Traces are obtained by a digital oscilloscope that measures the voltage drop in a resistor that is connected to the power supply of the device. From the measurements of a trace (or a few traces), the attacker will try to obtain some information about the secret key used in the implemented cryptosystem.

When the obtained signal is weak or if the relationship between the secret key and the consumed power is not clear, SPA attacks do not give enough information to break the algorithm. In these cases, a new type of attack, which uses statistical techniques, is considered: *Differential Power Analysis* (DPA). DPA uses a lot of traces of power consumption and requires a synchronization and alignment of the measured traces. Then, a statistical analysis can be made between the values of the power consumed along the execution of the algorithm with the values of the hypothetical model [17].

In the *Correlation of the Power Attacks* (CPA), the correlation between the measurement of the instantaneous consumed power and the data processed is analyzed [18]. As this correlation is, in general, very small, it is necessary to obtain a large set of measurements in order to have a lot of traces which are compared, by means of correlation coefficients, with the traces from the outputs of a theoretical model of the device.

Finally, the *High-Order Differential Power Analysis* (HODPA) are a generalization of the DPA attacks. In this case, several points of the power trace are used [19].

2.3 Electromagnetic Analysis

Electromagnetic fields emitted by a circuit due to the displacement of charges along the tracks of the metal layers of the circuit can be measured when the transistors switch state [20], which gives rise to the *ElectroMagnetic Attacks* (EMA).

Once these emanations are measured, they are analyzed in a similar way as the power traces, so that they give rise to *Simple ElectroMagnetic Analysis* (SEMA) or *Differential ElectroMagnetic Analysis* (DEMA).

Traditionally, EMAs have been used to attack smart cards, FGPAs and other small devices; nevertheless, attacks against laptops have also been carried out. In [10] a laptop has been attacked by using an antenna of 0.5 m which is connected by a coaxial cable to a low-pass filter and two amplifiers.

2.4 Other Type of Attacks

The sound produced by a device can be used as a side channel as well. They are called *acoustic attacks*. For example, in [12] two attacks are described. In the first attack, the microphone of the smartphone Samsung NOTE II, located at 30 cm, points to the ventilator vents of the notebook and an attack against the secret key used in computations of the notebook is performed. The second one considers a parabolic microphone connected to a laptop in a padded case which attacks to another laptop, located at a distance of 4 m from the first one.

Attacks denoted as *non-invasive physical attack* can be mounted by measuring the fluctuations in the electrical potential of the chassis of a laptop by means of the grounding or with a conductor cable connected to an input/output port, or even by touching the equipment by hand and measuring the potential of the body [21].

In some situations, attacks can be mounted by using several methods among those discussed above. In this way, it is possible to increase the power of the attack being carried out. For example, by simultaneously combining power consumption and electromagnetic emanations [22].

Next, we include other type of attacks. These attacks are specifically proposed against different implementations of cryptographic primitives on smartphones. In [23], authors have proposed a pre-processing composition to mount a Simple Side-Channel Analysis (SSCA) on RSA and ECC, i.e., an attack that uses one single waveform to uncover a secret key. In particular, they explain how a composition of time-frequency pre-processing manages to extract the relevant information obtained from one signal of an asymmetric cryptographic operation (RSA and ECC) running on an Android system.

On the other hand, electromagnetic emanations of smartphones have been used in [24] to obtain secret keys of public key cryptosystems by means of standard radio equipment in combination with far-field antennas. Moreover, in [25,26], side-channel resistance of the implementation of the ECDSA signature scheme in Android's standard cryptographic library is studied. Authors show that, for elliptic curves over prime fields, it is possible to recover the secret key very efficiently on smartphones using EMA side channel and lattice reduction techniques.

In [27], it is shown that elliptic-curve cryptography implementations on mobile devices are vulnerable to electromagnetic and power side-channel attacks. Authors prove that full extraction of ECDSA secret signing keys from OpenSSL and CoreBitcoin running on iOS devices, and partial key leakage from OpenSSL running on Android and from iOS's CommonCrypto are possible. The mounted non-intrusive attacks use a magnetic probe placed in the proximity of the device, or a power probe on the phone's USB cable.

3 The SCAP Framework

We have designed a framework, called SCAP (Side-Channel Analysis Platform) Framework, for capturing and studying the traces obtained when a cryptographic device is carrying out cryptographic computations. In order to show the behavior of this framework, we have considered the capture of traces derived from the power consumption when the RSA cryptosystem is running under an Android smartphone. In fact, we have implemented both the RSA and the EG cryptosystems in the smartphone so they are accesible by means of an Android application used as the interface to communicate with the smartphone. We will try to determine if it is possible to obtain side-channel information that allows breaking that implementation when the smartphone is deciphering.

The RSA and EG implementations have been made by using two different libraries: the Spongy Castle, SC, (https://rtyley.github.io/spongycastle/) and the BouncyCastle, BC, libraries (https://bouncycastle.org). The specific library can be chosen by the user when he launches the application.

3.1 Overview of the Framework

Side-channel attacks against cryptographic devices or, in general, against a Device Under Test (DUT), involve two types of activities: (1) The creation of a database of traces obtained from the measurement of different types of parameters when the device is computing (time, power consumption, emanations, etc.), and (2) the processing of such data. This processing is done by taking into account the knowledge of the algorithm, the protocol performed in the DUT, and a mixture of observation and intuition together with techniques of signal processing, statistics, etc.

The database is elaborated by collecting informative fragments from the repetition of observations, with similar or different parameters. The files with the obtained data are stored with some extra information about the involved parameters and variables, due to the number of files considered (sometimes more than 100 000 files). So, it is very convenient to automate, as much as possible, the acquisition and storage of data.

Moreover, given the large amount of data considered in each file, it is necessary to use an efficient data model in the sense that it is necessary to get a balance between data volume and the ease of its processing, i.e., the amount of data stored and its format must be adequate in order to avoid the slowing down due to the use of limited memory resources, virtual memory, type of processor, etc. Finally, it is important to consider that side-channel attacks involve the interaction with a set of hardware devices like oscilloscopes, data acquisition cards, etc., and software applications.

Indeed, in our SCAP Framework, we have considered the challenge of developing a database through a double strategy: on the one hand, the creation of a big data pool, with sorted, nominated and indexed files; and on the other hand, the management of these processed files, using database tools and data mining, making use of the parameters and results stored in such files.

To do this, we name the files with prefixes and suffixes generated automatically or manually, so that their names contain the test performed, the date and time of creation, and an extension. The files are stored in a directory tree with several levels. In the first level there are two branches: one branch stores the files with source data, and the other one, with different directories depending on the type of analysis, contains the files with the processed data. If the file size is large and a high efficiency is essential, strictly binary files are used. However, the “TDMS” (Technical Data Management Streaming) format is preferred as it handles variables in a standardized way together with metadata with group names, experiment parameters, etc., and moreover, it includes an implicit indexing system that facilitates subsequent data mining.

Data management is done using data mining tools (DIADEM$^{\text{TM}}$, National Instruments), proprietary applications developed by our group, and small programmed tools (SQLitle); all of them in order to extract the parameters of the data files, create files summary, etc.

3.2 Components of SCAP Framework

SCAP framework is composed of hardware devices and a set of software applications (see Fig. 1). In the following elements of SCAP framework are described.

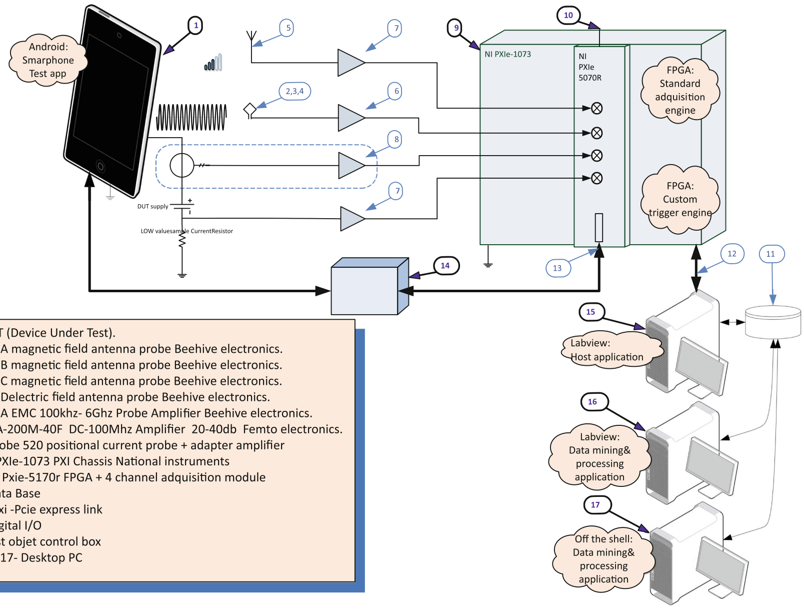

Android:
Smarphone
Test app
1
5
7
9
10
NI PXIe-1073
NI
PXIe
5070R
FPGA:
Standard
adquisition
engine
2,3,4
6
8
7
FPGA:
Custom
trigger engine
DUT supply
+
LOW valuesamble CurrentResistor
14
13
12
11
15
Labview:
Host application
16
Labview:
Data mining&
processing
application
17
Off the shell:
Data mining&
processing
application
1- DUT (Device Under Test).
2- 100A magnetic field antenna probe Beehive electronics.
3- 100B magnetic field antenna probe Beehive electronics.
4- 100C magnetic field antenna probe Beehive electronics.
5- 100Delectric field antenna probe Beehive electronics.
6- 105A EMC 100khz- 6Ghz Probe Amplifier Beehive electronics.
7- HVA-200M-40F DC-100Mhz Amplifier 20-40db Femto electronics.
8- I-probe 520 positional current probe + adapter amplifier
9- NI PXIe-1073 PXI Chassis National instruments
10- NI Pxie-5170r FPGA + 4 channel adquisition module
11- Data Base
12- Mxi -Pcie express link
13- Digital I/O
14-Test objet control box
15,16,17- Desktop PC

Fig. 1. General scheme of the SCAP Framework

- **DUT** (1)**:** It is the entity to be observed, where possible leakages are produced. In general, DUT contains some type of application or software (Test app), with known characteristics and algorithms for the observation.
- **Test Object Control Box** (14)**:** It produces an excitation in the DUT, adapted to the type of test. The excitation can be generated manually or through some automatically generated vector.
- **Data Acquisition System** (9)–(10)**:** The DUT produces some type of emanations or leakages that are captured by probes. These probes, individually or collectively (2)–(8), provide to the Data Acquisition System analog signals that correspond to emanations of different types: magnetic, electrical or electromagnetic radiation, variations in power consumption, temperature, etc.

  These signals are introduced into Data Acquisition System, where an application, developed in Labview$^{\mathrm{TM}}$, is installed as a firmware (we will talk about it later) on a hardware equipped with Analog Digital Converters.

The received data sequence is controlled, preprocessed and filtered at low level by a FPGA (Field Programmable Gate Array) with local memory. The FPGA creates a specially selected data stream, which is transferred at high speed, via PCiexpress (12), to the host PC. The PC host processes the data through a host application designed for this purpose, and stores the processed data in a database (11) for sharing and subsequent post-processing.
On the other hand, the FPGA (10) controls via the digital lines, supplied by the Digital IO port (13), different communication protocols for the DUT excitation (vectors or triggers) through the Test Object Control Box (14).

- **Processing Set** (15)–(17)**:** Several applications access to the database to process the attack; they are developed ad hoc in Labview^{TM}, C, and standard packages off the shell (Matlab, Diadem, etc.). These applications can be used cooperatively, from other workstations (desktop, laptop, etc.), for curve view, data browsing, data minning, and manipulating the information of the files.

The described architecture is flexible, so we are going to explain in more detail, for a particular implementation, some of the main parts.

3.3 Test Object Control Box

There are several ways to produce some kind of excitation to the DUT (in our case a smartphone). In general, the DUT needs to receive, in some way, the command to generate a set of events that produce the possible leakages. A simple way to do this is to develop an app where one can induce this type of situations, for example, by touching the smartphone screen or by an external mechanical push button switch and fixing the smartphone, if many trials are needed.

Another way to execute the command is through the digital IO port interface (see (13) in Fig. 1), which improves the determinism of the test. In our case we execute the push button switch action with an off-the-shelf pulse generator: an Agilent 33220A (see Fig. 2).

The connection of the pulse generator with the DUT can be made by using cheap hardware (Arduino boards, for example) or, as in our case, with the earphone connector and a small interface like the selfie stick to activate the camera button.

To detect the trigger that marks the start of the operation to analyze, we must add a suitable routine in the app running on the DUT, and an output line that provides a trigger signal to be used by another hardware (Data acquisition system, Oscilloscope, etc.) that helps to the synchronization of the traces.

3.4 Data Acquisition System

This device performs the acquisition of leakage signals. In general, this functionality is well known and it is typically carried out by using data acquisitions boards inside the PC, and especially by digital oscilloscopes. However, we prefer the use of hardware with high-speed, high-resolution A/D converters, FPGA, and PCIexpress link to PC.

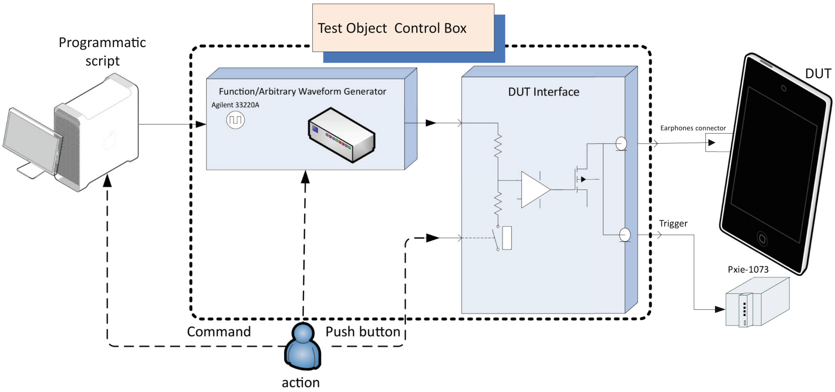


Fig. 2. Test Object Control Box

To study a DUT, fragments of information varying from several thousands of a second to several tens of seconds have to be observed and recorded. However, useful leakage information is usually contained between tenths and tens of milliseconds. In order to have sufficient temporal resolution and to be able to discern the useful part, it is necessary to sample between 40 ms and 250 ms. On the other hand, once the useful area has been determined, and after being filtered, it is possible to repeat the sampling reaching values between 10 ms and 20 ms, i.e., the information can be summarized as 0.001% and 0.010% of the total.

Another question to consider is that different side-channel attacking techniques require generating tracks of data with respect to a well-known temporal source, as accurate as possible, with little jitter with respect to the initial trigger. This is an important problem because the synchronization of traces is necessary. Thus, it is very important to make the collection data as accurately as possible with reference to the same source of temporal coordinates, and to have a flexible, combinable and masked trigger subsystem.

In the designed framework, our Data Acquisition System is constituted by a NI PXIe-1073 Chassis and an NI PXIe-5170R module, manufactured by National Instruments$^{\text{TM}}$ (see Fig. 3). The NI PXIe-1073 Chassis allows you to feed the modules that are introduced, to synchronize them and to establish a communication link with a PC through an MX interface. Moreover, the NI PXIe-5170R consists of a general purpose reconfigurable multichannel digital oscilloscope whose internal controller consists of a reprogrammable FPGA. It consists of 4 channels of simultaneous acquisition of up to 250 ms/s, 14 bits of vertical resolution, amplifiers and anti-aliasing filters, with inputs programmable, digital outputs, individually or with serial protocols: I2C, SPI, etc.

There are multiple possibilities for synchronizing and exchanging data with other modules communicated with a PC using PXIexpress Gen2 x8 (up to 3.2 Gb/s), which are inserted in a NI PXIe-1073 (up to 200 Mb/s) chassis.

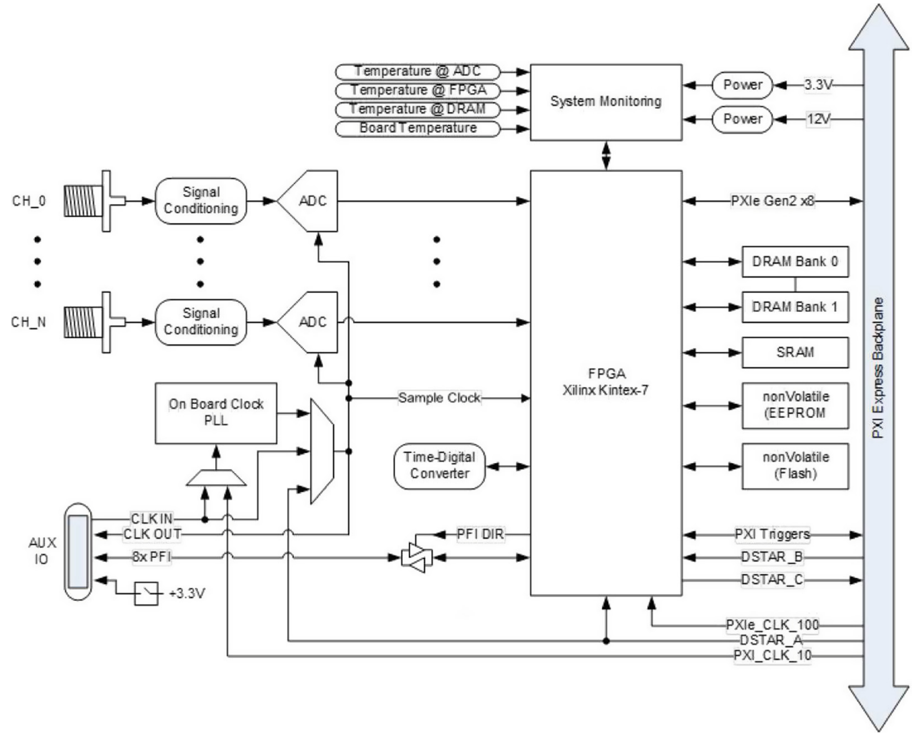


Fig. 3. NI PXIe-5170R block diagram. (Courtesy of National Instruments$^{\text{TM}}$)

Regarding the implementation of the Data Acquisition System, the main goal is to obtain a functionality similar to that of a digital oscilloscope, with extended and customized triggering capabilities (for masks and combined multi-shot), a pre-processed (typically filtered and decimated) that, together with the customized trigger, provides a specially selected data flow. This flow will be transferred at high speed, via PCIexpress, to the computer, where it is processed by an ad-hoc application. The flow is stored in a database that allows sharing and post-processing by means of two types of applications: firmware in PXIe-5170R FPGA Oscilloscope Emulation Module, and Host Application for Data Transactions.

Next, we will describe with more detail both applications: firmware and Host, which were made in Labview$^{\text{TM}}$ environment using the Xilinx$^{\text{TM}}$ ISE and Vivando tools.

Firmware in PXIe-5170R FPGA Oscilloscope Emulation Module. The implementation of the firmware (to be run in the FPGA) is performed to handle all hardware acquisition such as A/D converters, analogue digital filters, sequencers, DMA, etc.

The strategy of mixing two types of libraries is used: a standard realization with Labview^{TM} Instrument Device Libraries (IDL), and a specific customization of it. This way the emulation of a multichannel oscilloscope with its most characteristic primitives is performed: vertical and horizontal range change, trigger mode (simple, continuous), data storage size, transparent mode of transport of data to the computer memory, and FIFO communications.

The triggers customization allows the detection of events generated by various trigger forms from any channel, with storage, multi-trigger, multichannel combinational logic conditions trigger (And, Or, If then, etc.) and arbitrary mask trigger with pattern and (stored in a memory pattern) trigger tolerance.

These deeply coupled to acquisition process triggering methods allow the detection of an event or anomaly in a particular pulse, with a certain value, with a given waveform, without any need of storing previous data. This fact, together with the possibility of "selective decimation", constitutes a very important basis for saving data transactions and storage space over long periods of data observation.

Host Application for Data Transactions. This application, developed in Labview^{TM}, allows sending or receiving commands and data to the acquisition system, the transfer of files to the database and their processing.

The architecture is based on a typical multitasking cue loops producer-consumer, with isolated tasking loops: to the human interface, graphics curves, data acquisition transactions. The graphical interface has an aesthetic of folders with tabs containing functions (see Fig. 4).

Some options available in the host app are: communication port, data acquisition parameters, vertical-horizontal range, conversion rate, decimation, filter (media, average, median, envelope extraction, etc.), data fetch visualization, pre- and post- trigger (size and count), trigger mode (edge, level, channel, condition, combination, mask), Hilbert transform, etc. Moreover, it is possible to change of graphic mode view, colors, interpolation mode, channel in view, etc.

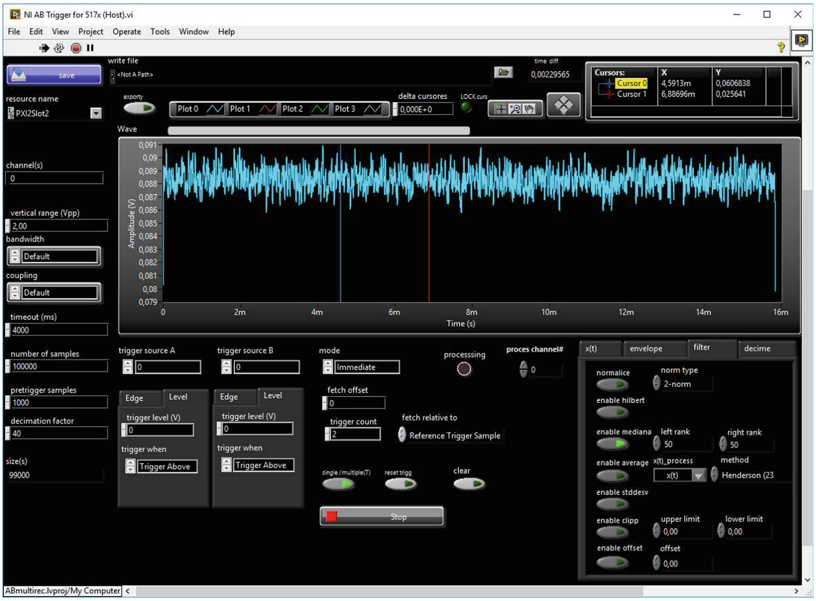
Moreover, graphs can be exported to elaborate reports or documents and save data in different formats for data export purposes.

3.5 Smartphone Interface

In this section we briefly describe the environment where SCAP Framework has been used.

The smartphone (DUT) used has the following characteristics: it is a Samsung, model Galaxy S3 GT-i9300 16 GB. The operating system is an Android 4.4.4 CyanogenMod, version 11-20141115-SNAPSHOT-M12-i9300, the kernel version is 3.0.64-CM-g4ca83ff, Build02@cyanogenmod #1, Fri Nov 14 21:44:13 PST 2014, and the CPU is ARMv7 Processor rev 0 (v7I).

When the user launches the developed app, he must select some possible options (see Fig. 5). The first action is to select the cryptosystem, in this case, RSA. Then, he chooses the library used for the implementation: SC or BC, and



write file: <Not A Path>

time diff: 0,00229565

resource name: PXI2Slot2

delta cursores: 0,000E+0

Cursors:	X	Y
Cursor 0	4,5913m	0,0606838
Cursor 1	6,88696m	0,025641

channel(s): 0

vertical range (Vpp): 2,00

bandwidth: Default

coupling: Default

timeout (ms): 4000

number of samples: 100000

pretrigger samples: 1000

decimation factor: 40

size(s): 99000

trigger source A: 0

trigger level (V): 0

trigger when: Trigger Above

trigger source B: 0

trigger level (V): 0

trigger when: Trigger Above

mode: Immediate

fetch offset: 0

trigger count: 2

fetch relative to: Reference Trigger Sample

proces channel#: 0

norm type: 2-norm

left rank: 50

right rank: 50

x(t)_process: x(t)

method: Henderson (23

upper limit: 0,00

lower limit: 0,00

offset: 0,00

Fig. 4. Screenshot of the host app

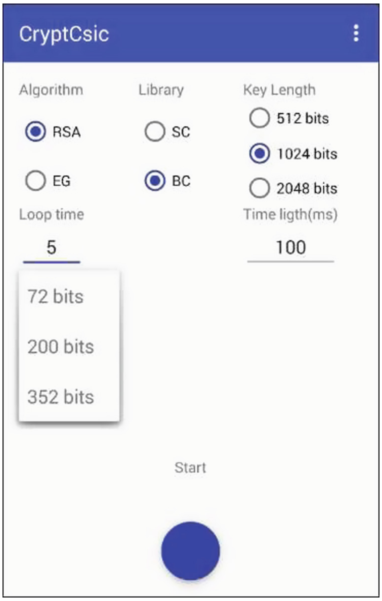
next, the user must choose the bitlength of the RSA key, where three options are available: 512, 1024, and 2048 bits.

Once the characteristics of the cryptosystem have been selected, the user selects how long the light of the flash (Time light) will be activated (in ms), the number of times (Loop time) that he wants to run the decryption process, that is, the number of times that the ciphertext will be decrypted with the same key, and the length of the plaintext to be used, which depends on the bitlength of the key. The plaintext is formed by repeating the chain ‘0123456789’ the number of times needed until texts of length 72, 200, 352, 472, or 792 bits, are obtained.

The app is launched by pressing the Start button, which can be activated in two ways: by pressing it with a finger or by means of an external trigger connected to the audio jack (like the system used to make a selfie).

The first time the application is launched, it generates the couple of keys (public and private) and the plaintext with the selected length. Then, it encrypts the plaintext obtaining the ciphertext and stores both the keys and the ciphertext in the smartphone. When the application is launched again, it will verify if such data is stored. If the data exists, it will use it; if not, it will repeat the previous process to generate the keys and the ciphertext for the current options. After these verifications, the decryption process is executed as many times as selected. Note that the flash is turned on before and after this process is carried out. The light of the flash is used as a signal or trigger.

The developed app turns on the flash of the camera in each encryption a unit of time, two units of time in the decryption and finishing with four units of time at the end of the cycle.



CryptCsic

Algorithm: ● RSA ○ EG

Library: ○ SC ● BC

Key Length: ○ 512 bits ● 1024 bits ○ 2048 bits

Loop time: 5

Time ligth(ms): 100

Fig. 5. Smartphone interface with some options selected

4 Experimental Results

This section presents some experimental results using SCAP Framework in the developed app.

For example, when a capture of 6 decryption iterations is released, the graph shown in Fig. 6 is obtained. In this first capture the used trigger has been a level detector (see Fig. 7). When the flash is triggered, an increase in the current supplied by the battery occurs. If the deciphering is repeated several times it is possible to see that the trace has repetitions.

In successive captures, we try to limit the moment in which the encryption/decryption process takes place, by modifying parameters of time of ignition of the flash and the length of the key. For the options of RSA, SC, 512 bits, Loop time 2, 25 ms flash time, with the same trigger, we have obtained several similar graphs (see Figs. 8 and 9). Both figures are similar, but they are not the same and it is not possible to appreciate the decryption start flashes, only the cipher start flash is noticed.

Two traces for the parameters RSA, SC, 2048 bits, Loop time 2, 50 ms flash time are shown in Figs. 10 and 11. Initially, the encryption/decryption time should be the same in each sample since the same text is always encrypted and decrypted with the same key, but there are differences between successive executions.

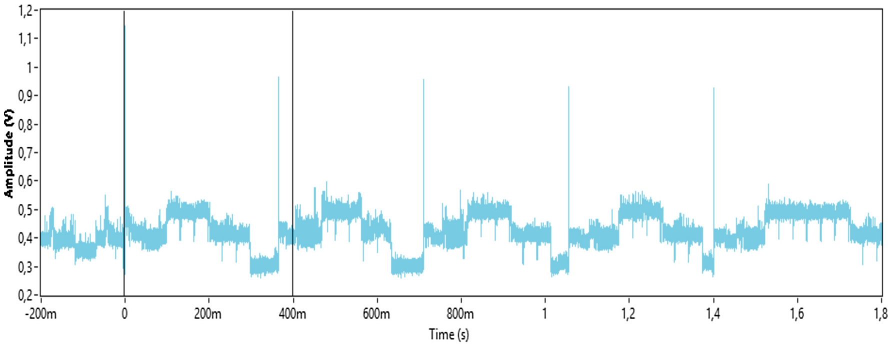
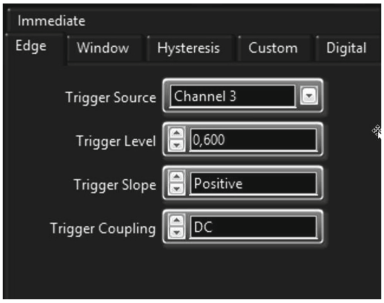


Fig. 6. Trace with 6 iterations



Trigger Source Channel 3

Trigger Level 0,600

Trigger Slope Positive

Trigger Coupling DC

Fig. 7. Example of trigger selected

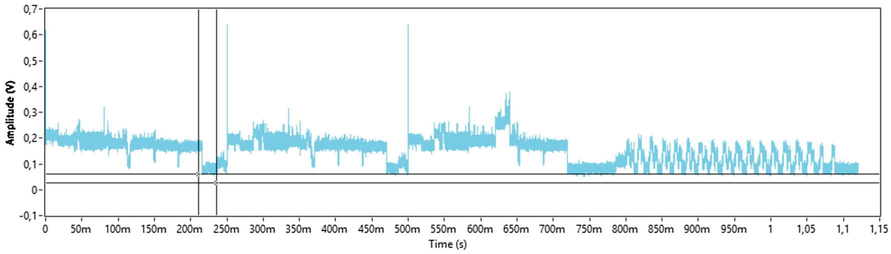


Fig. 8. Example of a trace: RSA, SC, 512 bits, loop time 2, 25 ms flash time

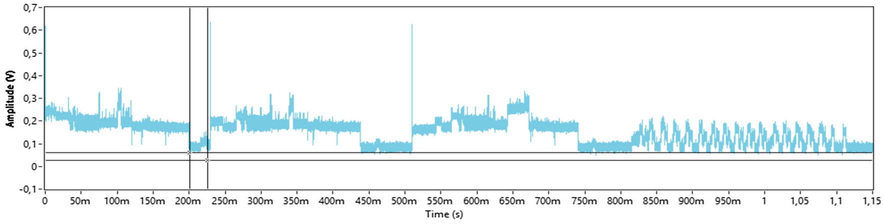


Fig. 9. A different example of trace: RSA, SC, 512 bits, loop time 2, 25 ms flash time

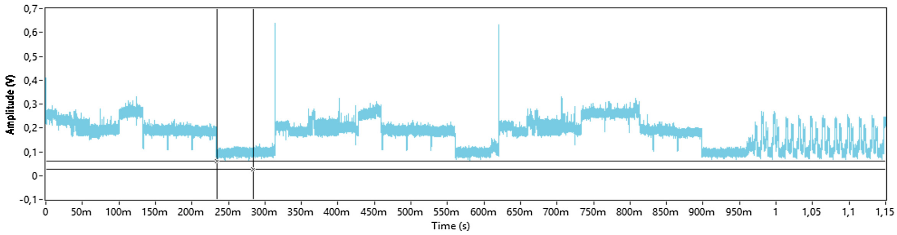


Fig. 10. Example of a trace: RSA, SC, 2048 bits, loop time 2, 50 ms flash time

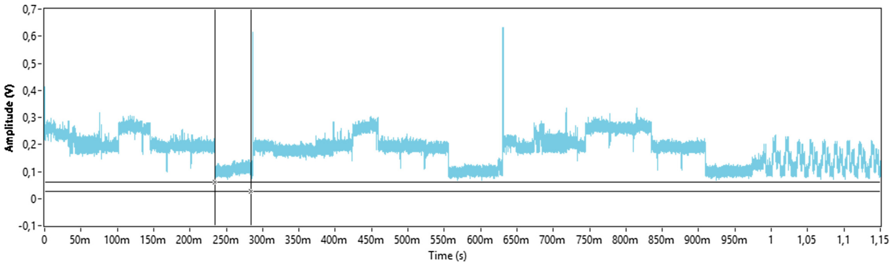


Fig. 11. A different example of trace: RSA, SC, 2048 bits, loop time 2, 50 ms flash time

5 Conclusions and Future Work

In this work we have developed a framework to capture and study the traces derived from a cryptographic device when cryptographic computations are performed. The main goal is to develop a tool for performing side-channel attacks against this type of device. The characteristics of the SCAP Framework have been described and a case study with a smartphone has been presented.

Some of the conclusions obtained from performed experiments and some of the future work to develop can be summarized as follows:

1. It is necessary to modify the software that interacts with the oscilloscope for improving the capture conditions. To do this new trigger systems, filters, etc. have to be implemented. The goal is to synchronize with much greater precision the moment when the decryption begins.
2. We have detected that the greater consumption produced in the smartphone is due to the ignition and refreshment of the screen, which masks the obtained results. Therefore, it is necessary to attenuate the backlight of the screen to obtain a less noisy trace.
3. In order to obtain a large number of traces and avoid causing vibrations in the smartphone screen that alter the position of the current probe, causing unnecessary electrical noise, it is important to automate the application with an external trigger.

4. The flash duration does not seem to be significant, so the smartphone software should be modified in order to try to limit the periods in which the flash is on. This could allow a better interpretation of the data.
5. Finally, it is necessary to modify the application of the smartphone so that it does not execute unnecessary code every time the encryption/decryption operation is launched. That is, the reading of the keys and the encryption of the text, among other things, should be optimized. This way it will be more feasible to determine the region where traces must be analyzed.

Acknowledgments. This work has been partly supported by Ministerio de Economía y Competitividad (Spain) under the projects TIN2014-55325-C2-1-R (ProCriCiS), TIN2013-46469-R (SPINY), TIN2016-79095-C2-2-R (SMOG-DEV), and by Comunidad de Madrid (Spain) under the project S2013/ICE-3095-CM (CIBERDINE), cofinanced with the European Union FEDER funds. We thank National Instruments for its support.

References

1. Rivest, R., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21**(2), 120–126 (1978)
2. Hankerson, D., Menezes, A., Vanstone, S.: Guide to Elliptic Curve Cryptography. Springer, New York (2004). https://doi.org/10.1007/b97644
3. Kocher, P.C.: Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 104–113. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-68697-5_9
4. Wold, K., Petrovic, S.: Behavioral model of TRNG based on oscillator rings implemented in FPGA. In: Proceedings of the 14^{th} IEEE International Symposium on Design and Diagnostics of Electronic Circuits Systems (DDECS), pp. 163–166 (2011)
5. Moradi, A., Kasper, M., Paar, C.: Black-box side-channel attacks highlight the importance of countermeasures. In: Dunkelman, O. (ed.) CT-RSA 2012. LNCS, vol. 7178, pp. 1–18. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-27954-6_1
6. De Mulder, E., Örs, S.B., Preneel, B., Verbauwhede, I.: Differential power and electromagnetic attacks on a FPGA implementation of elliptic curve cryptosystems. Comput. Electr. Eng. **33**(5–6), 367–382 (2007)
7. Sun, S., Yan, Z., Zambreno, J.: Experiments in attacking FPGA-based embedded systems using differential power analysis. In: Proceedings of the IEEE International Conference on Electro/Information Technology (EIT), pp. 7–12 (2008)
8. Kocher, P., Jaffe, J., Jun, B., Rohatgi, P.: Introduction to differential power analysis. J. Cryptogr. Eng. **1**, 5–27 (2011)
9. Mangard, S.: Exploiting radiated emissions-EM attacks on cryptographic ICs. In: 2003 Proceedings of Austrochip, pp. 13–16 (2003)
10. Genkin, D., Pachmanov, L., Pipman, I., Tromer, E.: Stealing keys from PCs using a radio: cheap electromagnetic attacks on windowed exponentiation. In: Güneysu, T., Handschuh, H. (eds.) CHES 2015. LNCS, vol. 9293, pp. 207–228. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48324-4_11. https://eprint.iacr.org/2015/170.pdf

11. Hutter, M., Schmidt, J.-M.: The temperature side channel and heating fault attacks. In: Francillon, A., Rohatgi, P. (eds.) CARDIS 2013. LNCS, vol. 8419, pp. 219–235. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08302-5_15. https://eprint.iacr.org/2014/190.pdf
12. Genkin, D., Shamir, A., Tromer, E.: RSA key extraction via low-bandwidth acoustic cryptanalysis. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8616, pp. 444–461. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44371-2_25. https://www.cs.tau.ac.il/ tromer/papers/acoustic-20131218.pdf
13. Joye, M., Tunstall, M. (eds.): Fault Analysis in Cryptography. Springer publishing, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29656-7
14. Anderson, R., Bond, M., Clulow, J., Skorobogatov, S.: Cryptographic processors-a survey. Proc. IEEE **94**(2), 357–369 (2006)
15. Skorobogatov, S.: Semi-invasive attacks-a new approach to hardware security analysis. Ph.D. thesis, University of Cambridge, Darwin College, UK (2005). http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-630.pdf
16. Kerckhoffs, A.: La cryptographie militaire. J. des Sci. Militaires **IX**, 1–2, 5–38, 161–191 (1883)
17. Kocher, P., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48405-1_25
18. Brier, E., Clavier, C., Olivier, F.: Correlation power analysis with a leakage model. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 16–29. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28632-5_2
19. Mangard, S., Oswald, E., Popp, T.: Power Analysis Attacks: Revealing the Secrets of Smart Cards. Advances in Information Security. Springer Science+Business Media, Heidelberg (2007). https://doi.org/10.1007/978-0-387-38162-6
20. Quisquater, J.-J., Samyde, D.: Electro magnetic analysis (EMA): measures and counter-measures for smart cards. In: Attali, I., Jensen, T. (eds.) E-smart 2001. LNCS, vol. 2140, pp. 200–210. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45418-7_17
21. Genkin, D., Pipman, I., Tromer, E.: Get your hands off my laptop: physical side-channel key-extraction attacks on PCs. J. Cryptogr. Eng. **5**(2), 95–112 (2015). http://link.springer.com/content/pdf/10.1007%2Fs13389-015-0100-7.pdf
22. Agrawal, D., Rao, J.R., Rohatgi, P.: Multi-channel attacks. In: Walter, C.D., Koç, Ç.K., Paar, C. (eds.) CHES 2003. LNCS, vol. 2779, pp. 2–16. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45238-6_2
23. Nakano, Y., Souissi, Y., Nguyen, R., Sauvage, L., Danger, J.-L., Guilley, S., Kiyomoto, S., Miyake, Y.: A pre-processing composition for secret key recovery on android smartphone. In: Naccache, D., Sauveron, D. (eds.) WISTP 2014. LNCS, vol. 8501, pp. 76–91. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-43826-8_6. https://hal.inria.fr/hal-01400921
24. Goller, G., Sigl, G.: Side channel attacks on smartphones and embedded devices using standard radio equipment. In: Mangard, S., Poschmann, A.Y. (eds.) COSADE 2014. LNCS, vol. 9064, pp. 255–270. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21476-4_17
25. Belgarric, P., Fouque, P.A., Macario-Rat, G., Tibouchi, M.: Side-channel analysis of Weierstrass and Koblitz curve ECDSA on android smartphones. Cryptology ePrint Archive, Report 2016/231, pp. 1–26 (2016). https://eprint.iacr.org/2016/231.pdf

26. Belgarric, P., Fouque, P.-A., Macario-Rat, G., Tibouchi, M.: Side-channel analysis of Weierstrass and Koblitz curve ECDSA on android smartphones. In: Sako, K. (ed.) CT-RSA 2016. LNCS, vol. 9610, pp. 236–252. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29485-8_14
27. Genkin, D., Pachmanov, L., Pipman, I., Tromer, E., Yarom, Y.: ECDSA key extraction from mobile devices via nonintrusive physical side channels. Cryptology ePrint Archive, Report 2016/230, pp. 1–23 (2016). https://eprint.iacr.org/2016/230.pdf

Sensitive Data in Smartphone Applications: Where Does It Go? Can It Be Intercepted?

Eirini Anthi(✉) and George Theodorakopoulos

School of Computer Science and Informatics, Cardiff University, Cardiff, UK
{anthies,theodorakopoulosg}@cardiff.ac.uk

Abstract. We explore the ecosystem of smartphone applications with respect to their privacy practices towards sensitive user data. In particular, we examine 96 free mobile applications across 10 categories, in both the *Apple App Store* and *Google Play Store*, to investigate how securely they transmit and handle user data. For each application, we perform wireless packet sniffing and a series of man-in-the-middle (MITM) attacks to capture personal identifying information, such as usernames, passwords, etc. During the wireless packet sniffing, we monitor the traffic from the device when a specific application is in use to examine if any sensitive data is transmitted unencrypted. At the same time, we reveal and assess the list of ciphers that each application uses to establish a secure connection. During the MITM attacks, we use a variety of methods to try to decrypt the transmitted information.

The results show that although all tested applications establish a secure TLS connection with the server, 85% of them support weak ciphers. Additionally, 60% of iOS and 25% of Android applications transmit unencrypted user data over the Wi-Fi network. By performing a MITM attack we capture the username, password, and email in various apps, e.g. Instagram, Blackboard, Ebay, and Spotify. We manage to bypass certificate pinning in 75% of the iOS applications, including Facebook. Finally, we observe that data is being forwarded to third party domains (mostly to domains that belong to Google and Apple).

Keywords: Mobile security · Man-in-the-middle attacks
Wireless network security · Network sniffing · SSL/TLS

1 Introduction

In the last decade, the number of smartphone users has increased dramatically [36]. Smartphones are Internet-enabled devices with an operating system (e.g. iOS, Android, Windows), capable of executing a variety of applications. Most of these devices are also equipped with voice control functionality, a camera, a Wi-Fi antenna, Bluetooth, and GPS. Due to their capabilities, smartphone owners not only use their devices to communicate but also to perform important

X. Lin et al. (Eds.): SecureComm 2017 Workshops, LNICST 239, pp. 301–319, 2018.
https://doi.org/10.1007/978-3-319-78816-6_21

everyday life activities. Such activities include researching a health condition, accessing education resources, navigating, and managing their money [34].

Most of the time users are required to share personal information with the mobile applications they use. However, it is often not clear to them how exactly these applications handle their personal data. A study by Boyels et al. [9] showed that 54% of smartphone users decided not to install an application once they discovered how much personal information they would need to share. Additionally, 30% of the users uninstalled an application that was already on their mobile phone when they learned it was collecting personal information they did not wish to share. The same study also showed that users are particularly sensitive about location data, with 19% of the users turning off the location tracking feature on their phone due to concerns about who could possibly access this information.

The rapid growth of the number of smartphone users has led to the increase of security threats related to smartphones. According to ENISA (European Union Agency for Network and Information Security), the number one threat is the leakage of data [13], which can happen in various ways: When a smartphone gets lost or stolen, its memory or removable media are unprotected, allowing an attacker to access the user's data [13]. Moreover, most of the applications used on a smartphone device will require the user to change their privacy settings in order to allow the application to access sensitive information such as contacts, photographs, etc. Many of these applications have been reported for sharing users' personal information with third parties without their consent. A recent study by Zang et al. [20] showed that 73% of Android and 47% of iOS applications shared personal information with third parties, including email addresses and location data. Finally, there is data loss that can occur when a smartphone is connected to Wi-Fi [22].

Although many smartphone users are aware that the mobile applications they use may share their personal data with third parties, many do not realise how often this happens [10]. Specifically, a recent survey [35], showed that many users are completely unaware of the risks that come when they share their personal data over a Wi-Fi connection, and particularly over public Wi-Fi networks. The most severe threat is the unauthorized access to their device which can lead to identity theft and compromised bank accounts [35].

This paper examines in depth the data leakage that occurs when users share personal information with various mobile applications over a Wi-Fi connection. Such information includes usernames, passwords, search terms, and location/geo-coordinates data. Additionally, we examine how these applications handle a user's personal information by observing the type of data they might share with third parties. Finally, we investigate methods to avoid data leakage. We perform tests on both *Android* and *iOS* devices; as they have different operating systems, we expect their behavior as to how they transmit and handle user data to differ.

The rest of the paper is organized as follows: Sect. 2 presents related work. Section 3 describes the experimental set up. Sections 4, 5, and 6 describe the main experiments and their results. Section 7 discusses the findings and evaluates the research. Finally, Sect. 8 covers the conclusion and future work.

2 Related Work

Previous studies have mainly focused on investigating the types of sensitive data that various mobile applications share with third parties, using dynamic analysis to capture mobile network traffic [6]. The major disadvantage of this approach is that requires human intervention, which can limit the scaling of the experiment. Various methodologies fall under this approach and have been used successfully in the past.

For instance, Rao et al. [32] used a Virtual Private Network (VPN) to monitor mobile traffic, involving tools such as *Meddle*. They showed that a significant number of Apple *iOS* and Google *Android* applications shared sensitive information such as emails, locations, names, and passwords as plaintext. A different way to observe network traffic is directly on the device. The *TaintDroid* application [4] for the *Android* platform allows users to track how private information is obtained and released by mobile applications in real time. A study by Enck et al. showed that 15 applications sent user location data to third parties and 30 sent the unique phone identifier, phone number, and SIM card serial number. Zang et al. [20] used a third method to monitor network traffic, during which they performed a man-in-the-middle attack over the Wi-Fi network that the device was connected. They showed that a very large percentage of mobile applications shared personal data with third parties and connected to unknown domains.

Another study which used the same method as [20] was that of Thurm and Kane [38]. This study revealed that a music *iOS* application shared personal information with eight different domains. Furthermore, the Federal Trade Commission [16] applied the same method to research the behavior of 15 fitness applications. The results of this study showed that 12 of the applications transmitted identifying information to 76 third party domains.

These studies focus on investigating the types of sensitive data that various mobile applications share with third parties. However, how securely these applications transmit this data over Wi-Fi networks has not yet been examined.

In this paper, we build on previous work by testing 96 free applications that require personal information. We investigate how user sensitive data is transmitted and handled, using wireless packet sniffing and dynamic analysis with man-in-the-middle attacks over a Wi-Fi network.

3 Experimental Setup

3.1 Selecting Mobile Applications

The Google *Play Store* for *Android* and the Apple *App Store* for *iOS* are the two largest distribution channels for mobile applications [41], which is why we focus on these two platforms. From a total of 96 applications that we test, 51 are *iOS* and 45 are *Android*. These are the most popular applications as of January/February 2016 that handle sensitive user data, across 10 different categories: Business, Finance, Food and Drink, Games, Health and Fitness, Music, Productivity, Shopping, Social Networking and Travel. We test the *iOS* applications on an *iPhone 6/ iOS v9.0.1* and the *Android* applications on a *Motorola Moto e/ Kit Kat v4.1*. Table 1 in the Supplemental Material[1] contains all the applications that we examine in this research.

3.2 Testing the Mobile Applications

In order to test each application we manually simulate a typical use for 10 to 15 min. The time spent on each application varies and exclusively depends on its type. During the simulation we explore the basic functions of the application. These include: create a user account, search using various keywords, perform actions that require personal identifying data, and complete a level of a game. We record specific keywords and personal user data that are used during each simulation. We then search for these keywords and personal data in the captured communications. To ensure the integrity of the captured data and to avoid possible interference from other applications, we take the following measures: during testing only the tested application is open and no other. We achieve this by terminating all other applications and by observing whether any data is transmitted, while no applications are open. For each application, we allow all requested permissions, such as for sharing location data, except for push notifications. The reason we disable push notifications is because they keep sending data in the background even after the application is closed [15]. This would result in capturing data not only from the application being tested at any single time, but also from any previously tested applications that enabled push notifications.

4 Experiment 1: Examining Network Data Following SSL Employment

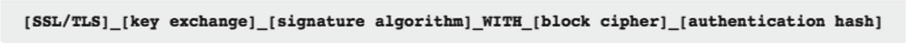
To identify if any of the applications transmit unencrypted data over the Wi-Fi network, we perform wireless packet sniffing using *Wireshark* [26]. During this process we passively monitor the mobile traffic from the smartphone. After configuring *Wireshark* to monitor mobile traffic from the smartphone, we start using an application. For each application, we test all the captured packets for user sensitive data using *Wireshark*'s built-in filter functionality.

If the mobile applications do not employ the Secure Sockets Layer (SSL)/ Transport Layer Security (TLS) protocol [33], the data that gets transmitted is not encrypted, hence it can simply be intercepted by performing passive network sniffing on the operating channel. If the SSL/TLS is employed, the transmitted data is encrypted and no third party is able to eavesdrop on or interfere with any of the transmitted messages [29]. As a result, for any application that employs SSL, we are unable to read or modify any of the transmitted messages. However,

[1] The supplemental material has been placed in our institutional repository due to space constraints. It can be accessed at this link: http://orca.cf.ac.uk/id/eprint/101448.

the SSL connection can be weakened in a number of ways and hence it is possible to decrypt the transmitted data.

In order for an SSL connection to be established, the client and the server make use of cipher suites. A cipher suite consists of a key exchange algorithm, a signature algorithm, a block cipher algorithm, and a hashing algorithm which computes the authentication key [29] (see Fig. 1). There is a variety of cipher suites available that provide different levels of security. The choice of cipher suites is crucial as they can compromise the security of the communication. Even if one of the listed cipher suites is cryptographically insecure, it is enough to break the secure connection between the client and the server and hence intercept the communication. This is possible via the *TLS Protocol Downgrade* attack [25] and it is one of the ways in which the SSL/TLS connection can be weakened.



```
[SSL/TLS]_[key exchange]_[signature algorithm]_WITH_[block cipher]_[authentication hash]
```

Fig. 1. Format of a cipher suit

Via *Wireshark* we are able to view the list of the cipher suites that each application supports to establish a secure connection with the server and as a result we can assess how secure they are. To achieve this we use data from the *O-Saft* [28] tool, which is used to inspect information about SSL/TLS certificates and tests the SSL/TLS connection, according to a given list of cipher suites. The code within *O-Saft* contains an evaluation of the strength of different cipher suites. To rate a cipher suite as weak or strong, the script examines the level of security of the individual algorithms (including the length of the key they use - if applicable) that compose the cipher suit. The script contains all possible combinations of cipher suites followed by a description of the level of their security, described as weak, medium, and high. Immediately afterwards, it displays a break down of each cipher, which explains the algorithms they contain and their key lengths in further detail.

Results: All the tested mobile applications for both *iOS* and *Android* platforms employ the latest version of SSL to establish a secure channel for communication. As a result, although we are able to capture the transmitted data, it is not possible for us to read it because it is encrypted. The only case in which we have the opportunity to capture transmitted data in plaintext is when we test the mobile browsers, *Safari* on the *iPhone* and *Google Chrome* on the *Motorola*, and perform requests that do not require a secure connection.

We examine and assess the cipher suites in 51 *iOS* applications, and we find that 45 use the same set of 26 cipher suites. From these 26 suites, 4 are considered to be weak and should not be used. Only 6 of the tested applications use less than 26 suites and do not support any weak suites (see Fig. 2). From the 45 *Android* applications, 27 use the same set of 35 cipher suites, of which 4 are considered insecure. Moreover, 11 of the applications use less than 35 cipher suites and from these only 6 do not support any insecure suites. Just 3 applications use more than 35 suites and only 1 does not support weak cipher suites. Finally, it was not possible to capture the *ClientHello* message for 4 applications and as a result their cipher suites could not be assessed (see Fig. 3).

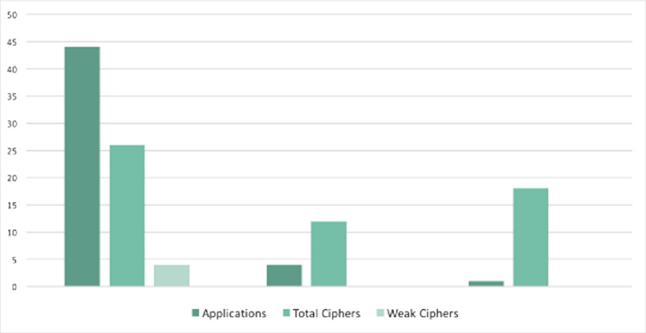


Fig. 2. Number of cipher suites that *iOS* applications support and how many of these are considered to be weak.

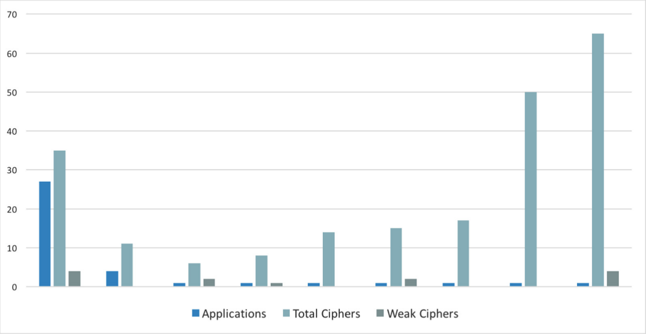


Fig. 3. Number of cipher suites that *Android* applications support and how many of these are considered to be weak.

Table 3 in the Supplemental Material shows in detail the number of cipher suites each application uses and how many of these are considered to be weak. For both systems we find that the applications support the same 4 insecure cipher suites, which are:

1. TLS_ECDHE_ECDSA_WITH_RC4_128_SHA
2. TLS_ECDHE_RSA_WITH_RC4_128_SHA
3. TLS_RSA_WITH_RC4_128_SHA
4. TLS_RSA_WITH_RC4_128_MD5

The order in which the suites appear in the *ClientHello* message denotes the client's preferred suites (with the client's highest preference first). In the *ClientHello* message, for all *iOS* applications, we observe that these 4 suites are at the bottom of the list, as opposed to the *Android* applications where the suites are found to be at the top of the list, which shows that these are the client's most preferred suites. Therefore, in the first case, the four weak cipher suites are the least preferred suites by the client and are unlikely to be used to establish a secure connection [1]. In the second case, the weak suites seem to be the client's most preferred suites. If the server accepts the client's preferences (the server is free to ignore the client's order and can pick the cipher suite that thinks it is best [1]) a connection will be established using one of these insecure suites, making the application vulnerable to MITM attacks. Regardless of the order in which these weak cipher suites appear in the application's *ClientHello* messages, they should not be used, as a *TLS Downgrade Attack* [25] could be used against them.

5 Experiment 2: Examining Network Data After Bypassing SSL

To examine how various applications transmit and handle user data other than sniffing the packets on the wireless network, we also use dynamic analysis with MITM attacks. The MITM attack is a technique used to intercept the communication between two systems, in this case between the client (application) and the server [27].

There are many tools that can be used to perform such an attack. Specifically, in this paper we use *Burp Suite* [37] and *mitmproxy* [8]. These also help us identify only HTTP-based traffic. We note that a recent study by Raoa et al. [32] showed that TCP flows (HTTP/HTTPS) are responsible for over 90% of the total traffic volume. Finally, in order to perform the attacks described above, we need to setup a Wi-Fi hot-spot on a computer that runs these tools and connect the smartphone device to the Internet via this hot-spot (Fig. 4).

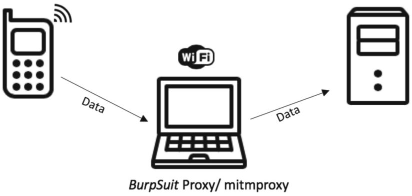


Fig. 4. Man-in-the-middle attack using *Burp Suite* and *mitmproxy*.

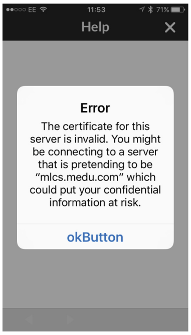
5.1 Man-in-the-Middle Attack Using *Burp Suite*

To examine if an application is accepting self-signed certificates, it is necessary to configure the smartphone to use a proxy. In this case we use *Burp Suite*, which generates a self-signed certificate and presents it to the client. We then monitor the behavior of the application in use and observe if it functions as expected. Additionally, we check if we are able to capture any HTTPS traffic on the proxy software. The steps of the procedure are described below [39]:

1. We ensure that the smartphone does not have any existing custom proxy certificates in its trust store.
2. On the computer, we disable the firewall and start the *Burp Suite* proxy. It is necessary to configure it to listen to all external network interfaces by specifying the port and address.
3. Then we configure the smartphone device to use the proxy. (Settings, Wi-Fi, we choose the desired Wi-Fi network, select HTTP Proxy Manual). The IP address and port of the proxy are the same to the computer in use.
4. Finally, we launch the application we want to test and simulate a typical use, while we monitor the proxy to detect if any HTTPS data is being intercepted.

If *Burp Suite* is able to intercept HTTPS traffic from the device without us having to install the proxy's certificate on the device's trust store, we know that the application does indeed accept self-signed certificates and is vulnerable to eavesdropping and modification via MITM attacks [39].

Results: We find that none of the applications for both platforms accept the unverified certificate that *Burp Suite* generates, and they prompt us with a message as shown in Fig. 5. As a result, we are not able to capture any of the HTTPS traffic that occurs during the simulation of a typical use for each application.



Error

The certificate for this server is invalid. You might be connecting to a server that is pretending to be "mlcs.medu.com" which could put your confidential information at risk.

Fig. 5. *Blackboard* application rejecting *Burp Suite*'s self-signed certificate

5.2 Man-in-the-Middle Attack Using *mitmproxy*

On applications that do not accept self-signed certificates, we are not able to capture the encrypted traffic that occurs from the device using the previous method. In order to overcome this, we perform a MITM attack using *mitmproxy*.

Once again, we configure the smartphone to use the proxy. However, this time we install the proxy's certificate in the device's trust store. *mitmproxy* contains a Certificate Authority (CA) implementation and is able to generate digital certificates [24]. Furthermore, to make the client (device) trust certificates issued by *mitmproxy*, we register it manually on the device as a trusted CA. It is necessary to emphasize that this method will only work if the application does not employ certificate pinning [12]. More details about this mechanism and how to bypass it are in Sect. 6.

To intercept traffic with the *mitmproxy* we follow the steps below [23]:

1. We start *mitmproxy* and configure the device to use it by setting the correct proxy details (port and IP address).
2. We then open the browser on the smartphone and visit www.mitm.it.
3. We select the relevant icon and follow the instructions, as to how to install the proxy's certificate in the device's trust store. When the installation is completed, we open an application and start observing the *mitmproxy*'s screen for HTTPS traffic.

In the *mitmproxy*'s main screen, we are able to view the mobile traffic that occurs when an application is in use. *mitmproxy* displays the full flow summary, including the methods used and the full *Uniform Resource Identifiers (URIs)* of the HTTP/HTTPS requests. By selecting one of the requests, the software allows us to inspect and manipulate the data it contains [24]. If the application is not using any encryption on the transmitted data, we are able to view it as plaintext. Therefore, this method helps us identify if the applications transmit unencrypted information over the network and examine if they send any of it to unknown third parties. To analyze further the captured communications, we export all captured data to a text file and use a *Python* script to search in it for any user sensitive data that might have been transmitted in plaintext. Specifically, the data we look for includes: Personal Identifying Information (PII) such as names and passwords, search terms, and geo-coordinate data, including longitude and latitude values. In Table 1, we present all the types of user data that the script looks for in the text files. The complete list of the keywords that are used throughout the simulations and therefore we look to find in the captured data, can be found in Table 2 in the Supplemental Material. Moreover, in our *Python* script we identify all the URIs of the requests that the application performed POST requests for. This way we are able to discover if any of the applications transmit personal user data to unknown domains.

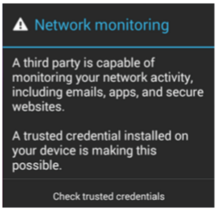
In order to ensure that our results are reliable, every time that the script identifies an occurrence of a keyword within a text file, we manually inspect the findings to confirm that they are correct and identify any further information. For instance, if the script finds a match for the string "`1990`", we manually examine the result to ensure that the finding is indeed the user's year of birth

Table 1. Types of user data.

Categories of data	Data types
Behavior	Employment (job searches)
	Medical
	Private messaging (chats, texts, etc.)
	Searching
Location	Latitude
	Longitude
PII	Address
	Age
	Date of birth
	Device information (e.g. Device ID)
	Email address
	Gender
	Name
	Password
	Post code
	Telephone number
	Username

and not a part of some other information such as long integer [20]. This process is also necessary in order to discover the destination domain, of the data that is transmitted and identified as plaintext.

Results: In order to perform this MITM attack it is necessary to install the certificate that *mitmproxy* generated in the device's trust store. After we complete this procedure, we observe that the *Android* device displays a warning message (see Fig. 6) to inform us that an unauthenticated certificate is currently being used. In contrast, on the *iOS* device we do not get any warnings about the fake certificate. Nevertheless, at this point we are able to capture HTTPS traffic from both devices, hence we start testing the applications, the results of which are presented in the following sections.



Network monitoring

A third party is capable of monitoring your network activity, including emails, apps, and secure websites.

A trusted credential installed on your device is making this possible.

Fig. 6. Warning message on the *Android* device, regarding the *mitmproxy*'s fake certificate.

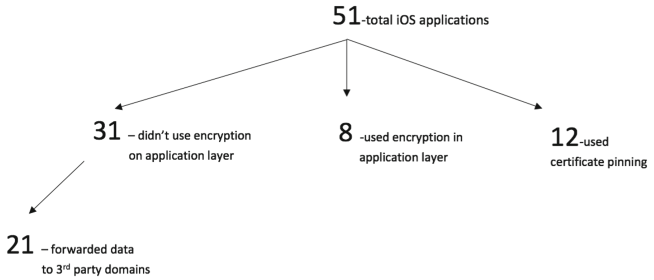

51-total iOS applications
31 – didn't use encryption on application layer
8 -used encryption in application layer
12-used certificate pinning
21 – forwarded data to 3rd party domains

Fig. 7. The number of *iOS* applications that use encryption in the application layer, employ certificate pinning, and transmit sensitive data to 3rd party domains.

Results for *iOS* Applications: From the 51 applications, we find that 30 transmitted the data unencrypted over the network, of which 20 forward it to third party domains. Just 8 of the applications encrypt user data in the application layer (i.e. before passing it to SSL), therefore although we can capture the transmitted data, we are unable to read it. Finally, 12 applications employ certificate pinning and do not function at all (see Fig. 7), claiming that there is a problem with the network.

Table 5 in the Supplemental Material shows the sensitive data that we capture for each application and the domains that each one forwards data to. In the same table we mark applications that employ certificate pinning with an xmark and use "n/a" for data that is not being forwarded to any third party domains.

The Burger King, Indeed Jobs, Lose it!, and Ebay applications transmit the most unencrypted user data, which includes: usernames, passwords, emails, location, gender, and search terms. Additionally, we manage to capture usernames and passwords for Spotify, Blackboard, Instagram, and EasyJet. The applications that forward the most data to third party domains are Indeed Jobs and Burger King. Gaming applications mainly transmit and share information about the device such as: phone model, screen size, etc. Moreover, the third party domains that receive the most sensitive user data are googleanalytics.com, googleservices.com, and apple.com. Figure 8 shows the types of data that the 20 *iOS* applications share with third parties.

Being able to capture the username, password, and email for Instagram, EasyJet, Blackboard, Ebay, and Spotify is a vulnerability. If an unauthorised person logs into these applications using these credentials, they could have access to much more sensitive information such as PayPal, bank accounts, home address, passport details, etc. Therefore, we decided to report our observations to each of the application's development teams as per the *Responsible Disclosure*[2] procedure. Facebook (for Instagram), Spotify, and Blackboard replied to us thanking us for reporting this issue, confirming that it is indeed a security flaw.

[2] This procedure involves privately notifying affected software vendors of vulnerabilities. The vendors then typically address the vulnerability at some later date, and the researcher reveals full details publicly at or after this time [18].

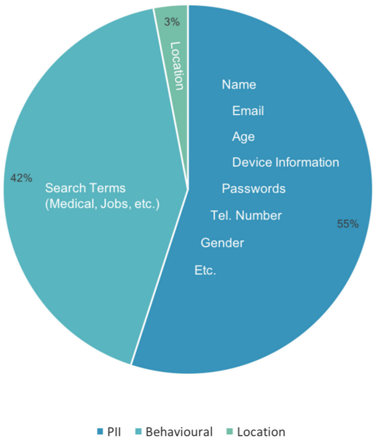


Fig. 8. The number of *iOS* applications that use encryption in the application layer, employ certificate pinning, and transmit sensitive data to 3rd party domains.

Results for *Android* Applications: From the 45 applications that we examine, 11 do not use any encryption in the application layer, hence the data gets transmitted unencrypted over the Wi-Fi network. Only 9 applications use encryption on the actual user data, so although we are able to capture the network traffic we are not able to read it. Furthermore, 25 applications employ certificate pinning and do not function during this process (see Fig. 9). Table 6 in the Supplemental Material shows the transmitted sensitive data that we capture for each *Android* application and also the third party domains to which it is being sent.

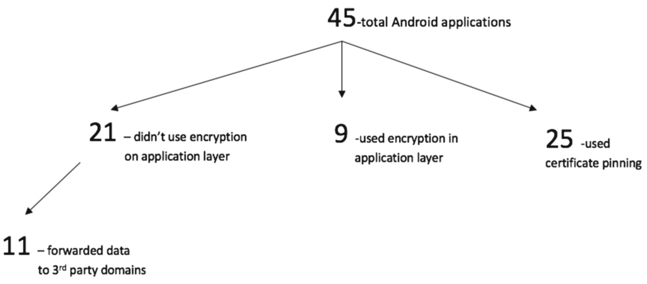

45-total Android applications
21 – didn't use encryption on application layer
9 -used encryption in application layer
25 -used certificate pinning
11 – forwarded data to 3rd party domains

Fig. 9. The number of *Android* applications that use encryption in the application layer, employ certificate pinning, and transmit sensitive data to 3rd party domains.

Ebay, Gumtree, and Booking.com, are the only applications that transmit unencrypted usernames and passwords. Domino's Pizza, Gumtree, and Booking.com share with third parties all the terms that were searched for in the

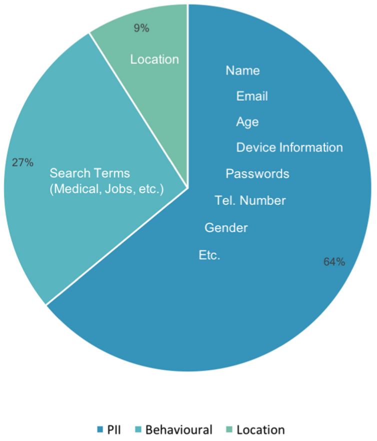


Fig. 10. The number of *Android* applications that use encryption in the application layer, employ certificate pinning, and transmit sensitive data to 3rd party domains.

application. Finally, location data is only shared by Just Eat and gaming applications mainly transmit and share device information. The third party domains that receive the most user sensitive data are googleads.com and apple.com. Figure 10, shows the types of data that the 11 *Android* applications share with third parties.

6 Experiment 3: Bypassing Certificate Pinning

Certificate pinning is a technique used widely in mobile applications to prevent the possibility of the device's trust store being compromised, by manually installing unverified certificates [12]. Specifically, this technique pins the certificate that the server uses in the application's source code, forcing it to ignore the device's trust store. As a result, it will only establish a connection to hosts signed with certificates that are pinned in the application's source code. To applications that employ this mechanism, we use *iOS SSL Kill Switch* to attempt to bypass it.

We perform this procedure only on *iOS* applications, and we are required to *Jailbreak/Rooting* [11] the tested device. This allows us to remove all the software restrictions of *Apple*'s operating system and grants us access to the *iOS* file system and manager. As a result, we are able to download extra items that are unavailable on the official *Apple App Store* [11].

After *jailbreaking* the *iPhone 6* following the instructions on [30], we gain access to *Cydia*, the unofficial *iOS App Store*. From there we can download and install in the device *iOS SSL Kill Switch* [2]. This tool disables the certificate validation process on the client side (the device), leaving it exposed to MITM attacks. Having installed and enabled *iOS SSL Kill Switch*, we use *mitmproxy* following the method described in the previous Sect. 5 to check if we can capture any HTTPS traffic.

Results: We find that this tool is effective on 75% of the applications, allowing us to capture the traffic that is transmitted while we are testing them. The remaining 25% of the applications are able to detect that the device is *Jailbroken* and do not operate (e.g. banking & social media applications).

7 Discussion

We perform wireless packet sniffing to investigate if any of the mobile applications transmit data unencrypted over the Wi-Fi network. Our results show that all the applications for both *iOS* and *Android* platforms use SSL to establish a secure channel for communication with the server. This protocol is fairly widely employed by developers, as it provides protection against passive eavesdropping [8]. Anyone performing wireless packet sniffing over the network will be able to capture the traffic, but they won't be able to read it as it is encrypted. SSL may provide privacy and data integrity between a client and a server, however it can be weakened and the cipher suites that applications use to establish this connection have an important role in this. We examine all the cipher suites that applications support in order to establish a secure connection and we find that the majority of them in both platforms (90% of the *iOS* and 80% of the *Android* applications) support four insecure cipher suites. These suites were the same for both operating systems:

1. TLS_ECDHE_ECDSA_WITH_RC4_128_SHA
2. TLS_ECDHE_RSA_WITH_RC4_128_SHA
3. TLS_RSA_WITH_RC4_128_SHA
4. TLS_RSA_WITH_RC4_128_MD5

These cipher suites are considered to be weak mainly because they use the RC4 stream cipher. Even though RC4 is widely supported and preferred by most servers, it has been known to have a variety of cryptographic weaknesses, making it unable to provide a sufficient level of security [3,19]. For this reason, according to the Internet Engineering Task Force (IETF), the RC4 algorithm is prohibited and clients must not include RC4 ciphers in their *ClientHello* message. Additionally, the MD5 hash algorithm is also known to have cryptographic weaknesses and cipher suites that employ it should not be used [14,29]. A few of the reasons that applications support these suites although they are considered to be insecure and have been prohibited include: compatibility with most servers, simple design, and speed due to the reduced number of operations they need to perform [31]. Nevertheless, 85% of all the tested *iOS* and *Android* applications that support these suites, even though they use SSL, are considered to potentially be vulnerable to MITM attacks.

We also test the applications in order to investigate if they accept self-signed certificates. We find that none of the applications, for both *iOS* and *Android*, accept the self-signed certificate that *Burp Suite* proxy generates. This is an indication that accepting self-signed certificates is indeed a severe security issue that developers are aware of, making the certificate validation processes as robust as possible [39].

Using *mitmproxy* we establish that approximately 60% of the *iOS* and 25% of the *Android* applications transmit and forward sensitive unencrypted data to third party domains. The most common data forwarded by applications to third party domains is Personal Identifying Information (PII) and Behavioral including: device information, email, name and search terms. For both platforms, gaming applications mainly transmitted and forwarded information about the device. A reason why PII and behavioural types of data are shared with third parties could be that this information is used by these organisations to develop targeted advertising [40]. The percentage of *Android* applications that share user data with third party domains seems to be significantly less than the percentage of the *iOS* applications. This is due to the fact that 20% of *Android* applications encrypt the actual user data and 56% employ certificate pinning. On the other hand, only 15% of the *iOS* applications encrypt the user data and only 23% employ certificate pinning. Therefore, for the applications that encrypt the data and use certificate pinning we are unable to investigate if they share sensitive information with third parties.

Comparing our results with a recent study by Zang et al. [20], which also investigated data sharing by applications, we can observe some differences. In the previous study, more applications shared location and other sensitive user data and very few employed certificate pinning. On the contrary, our results show that fewer applications share location and other sensitive user data with third parties. Additionally, the number of applications that use certificate pinning, specifically when it comes to *Android* applications, has increased dramatically. The overall increase in applications employing certificate pinning may be because, without it, data can be intercepted by installing fake certificates in the device's trust store [12]. Additionally, penetration testing recently performed on various mobile applications [20,21] could also explain why more of them started using certificate pinning. The fact that significantly more *Android* applications employ certificate pinning compared to *iOS* is because certificate pinning is one of the many security enhancements introduced in the new firmware version, Android 4.2 [12].

The domains to which applications from both platforms send the most user sensitive data are: googleanalytics.com, googleservices.com, googleads.com, and apple.com. Previous studies [20,32] have also found these domains to be dominant. This may be due to Google and Apple owning a variety of mobile advertisement networks and services such as AdMob, Google Analytics, Double CLick and iAds [5,17].

Finally, we use *SSL Kill Switch* on a *Jailbroken* iPhone, in order to attempt to bypass certificate pinning on applications that employ it, and we successfully manage to do so in 75% of the applications. Finance applications (Barclays, PayPal, Pingit) detected that the device was *jailbroken* and did not operate. To conclude, *Jailbreaking* or *Rooting* the smartphone introduces security issues and unless the applications are designed to not operate in such a device, the user's data is in danger of being stolen.

Overall, the methods we choose to evaluate how securely mobile applications transmitted and handled user data over a Wi-Fi network are effective but have limitations. To begin with, all the methods we use require human intervention which limits significantly the number of applications that we are able to test. The MITM attacks we perform to both platforms, although they were able to provide us with valuable information about the applications certificate validation process and data sharing behaviour, require physical access to the device in order to install fake certificates. Therefore, even though we are able to intercept any transmitted sensitive data, these methods would be very difficult to apply in real life. Additionally, the tools we use to perform these attacks focus only on HTTP/HTTPS traffic, limiting the scope of the research. The *SSL Kill Switch* allows us to successfully bypass the certificate pinning mechanism; however, we need to *jailbreak* the iPhone. This is a very time consuming and insecure process. To analyse the captured data, we write a Python script to search for sensitive data in the captured communications text files. The script is very effective in analysing our data, however if these files were larger in size, Python would run very slowly and would not be the most appropriate language to use to implement it.

8 Conclusion and Future Work

Our study aims to explore and analyse how user data is transmitted and handled by various mobile applications. We select 51 *iOS* and 45 *Android* mobile applications and carry out 4 different experiments, while we simulate a typical use for each application. The results show that all applications use SSL protocol to establish a secure channel for communication with the server, which protects data from passive eavesdropping, specifically when transmitted over public networks. However, this does not mean that user data is secure, as our findings show that only a very small percentage of these applications encrypt the actual user data and approximately 85% of these applications support 4 weak cipher suites which make them vulnerable to MITM attacks. Moreover, our results show that 60% of the *iOS* and 15% of *Android* applications forward sensitive user data, mostly PII and Behavioral, to third party domains mainly owned by Google and Apple.

Although our research methodology has its limitations, we still manage to arrive at significant conclusions as to how securely user data gets transmitted and handled by various applications, over a Wi-Fi network. Additionally, two of the methods we use are designed to break or bypass the basic security mechanisms that developers employ, such as SSL and certificate pinning. This is proof that these security measures are not invulnerable. As a result, users need to become fully aware that their personal information can never be 100% secure and the only way to protect their privacy is to understand these security risks.

To expand on the results of this research, future study could focus on testing more applications from each category, for both operating systems. Non-TCP traffic could also be investigated for sensitive data leakage using *tcpdump*, which monitors traffic that is not on TCP. To the applications that support weak cipher suites *TLS Downgrade Attack* could be performed, to explore if SSL can indeed be compromised this way. In this paper, we manage to apply tools to bypass certificate pinning only to *iOS* devices. Future studies could also *root* an *Android* device and then use *Android-SSL-TrustKiller* [7] to try to bypass certificate pinning in this operating system as well. Furthermore, tools that track the data-sharing behavior of applications directly from the smartphone device such as *TaintDroid* could be used to monitor both the operating system and the application. As a result, it would be possible to clearly distinguish any leakage that happens due to the application's activity and the background system processes [4, 20].

Additionally, paid applications could also be tested for data leakage. The results could then be compared to free applications in order to review any difference in the data sharing behavior. Finally, tools that limit data sharing, such as *Limit ad Tracking* and *Opt out of interest based ads*, can be employed to examine any differences in the activity of the applications.

References

1. RFC 5246 - The Transport Layer Security (TLS) Protocol Version 1.2. https://tools.ietf.org/html/rfc5246. Accessed 05 Jan 2017
2. Diquet, A.: iOS SSL Kill Switch (2016). https://github.com/iSECPartners/ios-ssl-kill-switch. Accessed 20 Apr 2017
3. AlFardan, N.: On the Security of RC4 in TLS. http://www.isg.rhul.ac.uk/tls/. Accessed 25 Apr 2017
4. Appanalysis. Realtime Privacy Monitoring on Smartphones (2016). http://www.appanalysis.org/index.html/. Accessed 9 Apr 2017
5. Apple. Ad for Developers. Apple Developer. https://developer.apple.com/iad/. Accessed 03 May 2017
6. Ball, T.: The concept of dynamic analysis. In: Nierstrasz, O., Lemoine, M. (eds.) ESEC/SIGSOFT FSE -1999. LNCS, vol. 1687, pp. 216–234. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48166-4_14
7. Blanchou, M.: iSECPartners/Android-SSL-TrustKiller. Bypass SSL certificate pinning for most applications. https://github.com/iSECPartners/Android-SSL-TrustKiller. Accessed 03 May 2017
8. Boneh, D., Inguva, S., Baker, I.: SSL, MITM Proxy (2007). http://crypto.stanford.edu/ssl-mitm
9. Boyles, J.L., Smith, A., Madden, M.: Privacy and data management on mobile devices. Pew Internet Am. Life Project **4** (2012)
10. Carnegie Mellon University. Knowledge of Location Sharing by Apps Prompts Privacy Action (2015). https://www.sciencedaily.com/releases/2015/03/150323132846.html. Accessed 4 Apr 2017
11. Cohen, A.: The iPhone Jailbreak: A Win Against Copyright Creep. Time.com (2010)
12. Elenkov, N.: Certificate Pinning in Android 4.2 (2012)
13. ENISA. Top Ten Smartphone Risks (2016). https://www.enisa.europa.eu/activities/Resilience-and-CIIP/critical-applications/smartphone-security-1/top-ten-risks. Accessed 4 Apr 2017

14. MIT Laboratory for Computer Science and RSA Data Security. RFC 1321 - The MD5 Message-Digest Algorithm. https://tools.ietf.org/html/rfc1321. Accessed 25 Apr 2017
15. Fox, M.A., King, P.F., Ramasubramani, S.: Method and apparatus for maintaining security in a push server. US Patent 6,421,781, 16 July 2002
16. FTC. Federal Trade Commission (2016). https://www.ftc.gov/search/site/fitness~app. Accessed 9 Apr 2017
17. Google. Monetize and Promote with Google Ads.Google Developers. https://developers.google.com/ads/?hl=en. Accessed 03 May 2017
18. Google. Rebooting Responsible Disclosure: A Focus on Protecting End Users. https://security.googleblog.com/2010/07/rebooting-responsible-disclosure-focus.html. Accessed 30 Apr 2017
19. Internet Engineering Task Force (IETF). RFC 7465 - Prohibiting RC4 Cipher Suites. https://tools.ietf.org/html/rfc7465#section-1. Accessed 25 Apr 2017
20. Zang, J., Dummit, K., Graves, J., Lisker, P., Sweeney, L.: Who knows what about me? A survey of behind the scenes personal data sharing to third parties by mobile apps (2015). http://techscience.org/a/2015103001/. Accessed 14 Feb 2017
21. Mense, A., Steger, S., Sulek, M., Jukic-Sunaric, D., Mészáros, A.: Analyzing privacy risks of mhealth applications. Stud. Health Technol. Inform. **221**, 41 (2016)
22. Cooney, M.: 10 Common Mobile Security Problems to Attack (2012). http://www.pcworld.com/article/2010278/10-common-mobile-security-problems-to-attack.html. Accessed 4 Apr 2017
23. mitmproxy. About certificates (2016). Accessed 20 Apr 2017
24. mitmproxy. How mitmproxy works (2016). Accessed 20 Apr 2017
25. Moeller, B., Langley, A.: RFC 7507: TLS fallback signaling cipher suite value (SCSV) for preventing protocol downgrade attacks (2015)
26. Orebaugh, A., Ramirez, G., Beale, J.: Wireshark & Ethereal Network Protocol Analyzer Toolkit. Syngress, Rockland (2006)
27. OWASP. Man-in-the-Middle Attack (2016). https://www.owasp.org/index.php/Man-in-the-middle_attack/. Accessed 18 Apr 2017
28. OWASP. O-Saft (2016). https://www.owasp.org/index.php/O-Saft/. Accessed 20 Apr 2017
29. OWASP. Transport Layer Protection Cheat Sheet (2016). https://www.owasp.org/index.php/Transport_Layer_Protection_Cheat_Sheet. Accessed 18 Apr 2017
30. Pangu. Pangu Jailbreak (2016). http://en.pangu.io. Accessed 20 Apr 2017
31. Paul, S., Preneel, B.: On the (in)security of stream ciphers based on arrays and modular addition. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 69–83. Springer, Heidelberg (2006). https://doi.org/10.1007/11935230_5
32. Raoa, A., Kakhkib, A.M., Razaghpanahe, A., Tangc, A., Wangd, S., Sherryc, J., Gille, P., Krishnamurthyd, A., Legouta, A., Misloveb, A., et al.: Using the middle to meddle with mobile. Technical report, Northeastern University (2013)
33. Rescorla, E.: SSL and TLS: Designing and Building Secure Systems, vol. 1. Addison-Wesley Reading, Boston (2001)
34. Smith, A.: Us Smartphone Use in 2015. Pew Research Center, pp. 18–29 (2015). Accessed 1 Apr 2017
35. Statista. The Hidden Dangers of Public WiFi (2016). http://www.privatewifi.com/wp-content/uploads/2015/01/PWF_whitepaper_v6.pdf/. Accessed 5 Apr 2017
36. Statista. Number of Smartphone Users Worldwide from 2014 to 2019 (2016). http://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/. Accessed 1 Apr 2017

37. Stuttard, D.: Burp Suite (2007)
38. Thurm, S., Kane, Y.I.: Your apps are watching you. Wall Str. J. **17**, 1 (2010)
39. Chell, D., Erasmus, T., Colley, S., Whitehouse, O.: The Mobile Application Hacker's Handbook, 1st edn. Wiley, Hoboken (2015)
40. Varshney, U., Vetter, R.: Mobile commerce: framework, applications and networking support. Mob. Netw. Appl. **7**(3), 185–198 (2002)
41. Victor, H.: Android's Google play beats app store with over 1 million apps, now officially largest (2013). Accessed 16 Jan 2014

Fault-Tolerant and Scalable Key Management Protocol for IoT-Based Collaborative Groups

Mohammed Riyadh Abdmeziem(✉) and François Charoy

Université de Lorraine Inria-CNRS-LORIA, Nancy, France
{mohammed-riyadh.abdmeziem,francois.charoy}@loria.fr

Abstract. Securing collaborative applications relies heavily on the underlying group key management protocols. Designing these protocols is challenging, especially in the context of the Internet of Things (IoT). Indeed, the presence of heterogeneous and dynamic members within the collaborative groups usually involves resource constrained entities, which require energy-aware protocols to manage frequent arrivals and departures of members. Moreover, both fault tolerance and scalability are sought for sensitive and large collaborative groups. To address these challenges, we propose to enhance our previously proposed protocol (i.e. DBGK) with polynomial computations. In fact, our contribution in this paper, allows additional controllers to be included with no impact on storage cost regarding constrained members. To assess our protocol called DsBGK, we conducted extensive simulations. Results confirmed that DsBGK achieves a better scalability and fault tolerance compared to DBGK. In addition, energy consumption induced by group key rekeying has been reduced.

Keywords: Collaborative applications · Internet of Things (IoT) Security · Group key management · Polynomial computation · Contiki

1 Introduction

With the rise of the Internet of Things (IoT) and its integration in information systems, collaborative applications have taken a new dimension. Pervasive devices and objects are able to perceive our direct environment and act autonomously upon it to help users to reach their goals. Applications flourished in healthcare, transportation and military environments [4] that combine input from users and objects to reach goals in a collaborative way. In these domains, stakeholders would only accept these systems in their environment if they have strong guarantees on the security, privacy and integrity of the data they produce and share. The distributed nature of such systems and the requirement for encryption of data shared among participants lead to one of the most important challenges in such evolving environments: the management of cryptographic group keys [2,6,32].

X. Lin et al. (Eds.): SecureComm 2017 Workshops, LNICST 239, pp. 320–338, 2018.
https://doi.org/10.1007/978-3-319-78816-6_22

Group key management is challenging in this context. In fact, collaborative groups involve heterogeneous members with different requirements and resources capabilities [17]. This gap can hinder end-to-end communications. Indeed, constrained members with limited processing power and storage space can not run heavy cryptographic primitives [5]. Moreover, collaborative applications may present a high rate of leaving and joining members within tight time lapses, which makes the issue more difficult to handle. The scalability of these systems needs to be addressed bearing in mind the increasing number of entities taking part in the collaborative groups. Last, fault tolerance is at utmost importance especially for critical and sensitive applications (e.g. health related and military applications) [31].

We address this problematic of designing a secure and efficient protocol to establish shared group credentials for Peer-to Peer collaborative groups. These credentials will be used to ensure the required security properties such as data confidentiality, data integrity, and data authentication. The proposed protocol has to be energy aware allowing an implementation on constrained devices, which take part in the collaborative process. In addition, the protocol must be scalable, as well as tolerant to possible failures of the entity in charge of managing the group key.

To achieve this goal, we rely on our previously proposed group key management protocol called DBGK (Decentralized Batch-based Group Key) [3]. This protocol considers a network topology composed of several sub groups. Each sub group is managed by an area key management server, while the whole group is managed by a general group key management server. The established group key is composed of a long term key and short terms keys (called tickets), which are different for each time interval. Constrained members in terms of resources (e.g. connected objects) are only involved in the re-keying process if these latter have recently been active. In addition, keying materials are distributed to joining members based on their resources capabilities. Experiments showed that DBGK [3] is energy efficient and outperforms similar existing protocols in the literature.

Although efficient and secure, DBGK relies on key management servers to maintain the group key. Including additional servers to improve fault tolerance would impose a high storage overhead on constrained members. This makes DBGK inappropriate to be directly implemented in sensitive collaborative applications. In this paper, we propose a distributed extension for DBGK called DsBGK (Distributed Batch-based Group Key). In this extension, we keep the core functioning of DBGK, while significantly distributing the operations which were based on a central entity. We achieve this by integrating a polynomial based scheme inspired from [24,25]. In addition, we improve the efficiency of the original scheme to suit the constrained IoT environment. We conducted extensive experiments to assess the performances of DsBGK and compared the results with DBGK performances. The results showed that DsBGK provides an enhanced scalability and fault tolerance, as additional key management servers (controllers) can be included without impacting the storage overhead on constrained members. Furthermore, energy cost due to rekeying operations is reduced compared to DBGK, which extends the life cycle of battery powered entities.

The remaining of the paper is organized as follows. In Sect. 2, we present a use case scenario to motivate our contribution. In Sect. 3, we discuss, in detail, existing solutions in the literature. For the sake of clarity, we summarize in Sect. 4, the required background. In Sect. 5, we present our network model, along with our assumptions and the used notations. In Sect. 6, we thoroughly present our approach before introducing and analyzing the experimental results in Sect. 7. Section 8 concludes the paper and sets our future direction.

2 Use Case Scenario: Personal Health Record (PHR)

A personal heath record [33] (Fig. 1) is a typical example of a document that can be accessed and edited by multiple participants, including medical sensors attached to patients. This is also an example of a document that contains highly private and sensitive information. To edit a medical record, some participants (e.g. medical staff) collaborate using unconstrained devices, such as Personal Computers (PC) and smartphones. However, sensors planted in or around the human body are considered as constrained since they have limited computing power and may operate on battery. These sensors can either communicate their sensed data to medical staff through the unconstrained entities (e.g. PC, smartphones) or directly edit patient's medical record. Medical staff can also control the sensors (trigger or stop the sensing of a particular physiological data), and add more sensors to the collaboration. New members can join or leave the collaboration around the medical record as the situation of the patient evolves. The different entities collaborate in a distributed way to maintain the medical

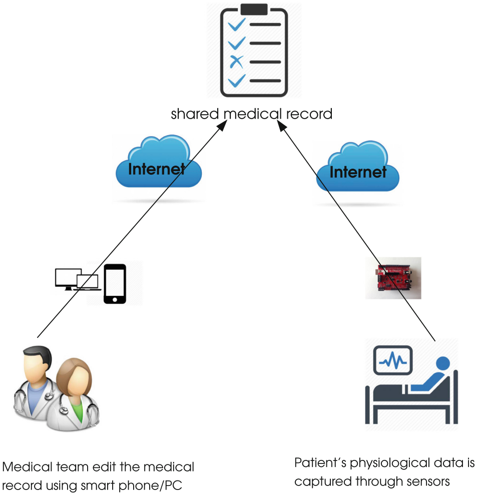

shared medical record
Internet
Internet
Medical team edit the medical record using smart phone/PC
Patient's physiological data is captured through sensors

Fig. 1. Use case scenario

record. This latter can be replicated among different entities and the modifications can be executed on the different replicas, which need to be synchronized. This is important in order to avoid a single point of failure on the record management architecture. It is also important to control the entities that have access and can modify the record over time. This clearly highlights the importance of securing communications in such a hybrid and heterogeneous group of entities by efficiently managing the security credentials used to provide data authentication and data confidentiality. Personal Health Record (PHR) is a typical case of collaboration among health-care personal, insurers, caregivers, patients and sensors to maintain a document that reflects the patient status, health history and treatment. There is an obvious need to provide a decentralized, secure, safe, privacy preserving and scalable solution to share these documents among people and sensors (objects).

3 Related Work

In this section, we review the main categories under which group key management protocols are usually categorized [11,28], namely, the centralized, the decentralized, and the distributed categories.

Centralized protocols are based on an unconstrained central entity (i.e. Key Management Server (KMS)), which is responsible for generating, distributing, and updating the group key for the whole group. Authors in [15] introduced the Group Key Management Protocol (GKMP), which is based on a Group Key Packet (GKP). This latter encompasses a Group Traffic Encryption Key (GTEK) to secure data traffic, and a Group Key Encryption Key (GKEK) to secure transmissions related to rekeying operations. Following a leave event, the central entity broadcasts the new GKP to all remaining members creating a complexity of $O(n)$. This complexity makes GKMP not scalable with regards to dynamic and large groups. To reduce the impact of leave events, authors in [34] proposed an interval-based protocol, which generates the keying materials corresponding to the predicted period of time during which the members are expected to remain in the group. Doing so, following a leave event, no rekeying is required. However, this solution is not suited to dynamic groups with unexpected join and leave events, as predicting the leaving moment of members is neither realistic nor practical. In addition, constrained members which are part of the group for a long period of time might suffer from storage issues, as a large number of keying materials needs to be stored.

To further improve efficiency, several hierarchical based protocols have been proposed. Among them, the Logical Key Hierarchy (LKH) protocol [37], later improved by the One-way Function Tree protocol [7] are typical examples. The basic idea of these protocols is that the KMS shares pre-established credentials with subsets of the group. Following an event, the KMS relies on these credentials to target specific subgroups during the rekeying, thus, reducing the number of required rekeying messages (i.e. $OLog(n)$).

Thanks to their efficiency, polynomial based approaches are used to manage group keys in collaborative applications. In fact, polynomial based schemes allow overcoming the storage cost related to multicast inter-group communications. Moreover, polynomial evaluation can be, under certain conditions, more efficient than encryption/decryption primitives. Polynomials have originally been included in threshold secret sharing schemes [30]. More recently, authors in [35,36] used polynomials to enable group members decrypting received messages. Doing so, the members are no longer required to store a secret key shared with each sender. Nevertheless, polynomials are usually generated and broadcasted by the KMS. To reduce this overhead on the KMS, authors in [25] propose a self-generation technique to generate the polynomials by the members of the group. In a nutshell, centralized protocols are characterized by their efficiency due to the use of symmetric primitives. Furthermore, these protocols do not require peer-to-peer communications during rekeying operations. However, the single point of failure and scalability issues constitute their main weaknesses.

Decentralized protocols consider the group divided into various areas, with an Area Key Management Server (AKMS) in charge of managing local events. This class of protocols is generally categorized into two sub categories [11]: *common Traffic Encryption Key (TEK) per area* [9,27], and *independent TEK per area* [22,25]. In the former category, a unique TEK is implemented for the various areas of the group. As a result, if an event happens, the whole group is affected by the rekeying. In the latter category, a different TEK is implemented for each area. As a result, the *1-affects-n* issue is attenuated, as rekeyings only affect specific areas. However, data transmitted across areas has to be translated at the border of each area. This classification of decentralized protocols can further be refined [10] by including *time-driven* rekeying subcategory [9,29] and *membership-driven* rekeying subcategory [8,27]. In membership-driven protocols, the group key is updated following each membership event, whereas, in time-driven protocols, the update of the group key is carried out at the end of a defined period of time without taking into consideration membership events. Consequently, the impact of frequent and consecutive events is limited. Nevertheless, ejected members are still able to access exchanged data up to the end of the interval. Likewise, a new member would have to temporize until the start of a new interval prior of being able to access exchanged data in the group.

Distributed protocols do not rely on any central entity. Instead, all members contribute in the management of the group key in a peer-to-peer way. Distributed protocols are usually based on the n-party version of the well known Diffie-Hellman protocol [18,19]. Hence, these protocols are highly reliable, as the group is free from any single point of failure. Nevertheless, distributed protocols involve a high number of exchanged messages during rekeying operations, in addition to an important computation cost due to the use of heavy asymmetric primitives.

To alleviate this cost, authors in [13] propose a probabilistic based protocol. Members of the group establish communication channels composed of sequences of adjacent members between which a key is shared. Indeed, members propagate the key, which is shared between the first adjacent members to the remaining members. This propagation is achieved using local keys. However, if no local key is found between two specific members, these members proceed with a pairing attempt by exchanging a set of global keys generated from a pool of keys. In spite of its improved performances compared to deterministic protocols, this protocol suffers from a lack of connectivity. In fact, members could be disconnected from the group if several pairing attempts fail. To further mitigate the complexity of distributed protocols, authors in [12] introduce a protocol which proceeds within two phases. In the first phase, members of the group autonomously generate the group key using pre-defined seeds and hash functions. In the second phase, members synchronize their generated keys taking into account delays due to the loose synchronization of members clocks. Compared to other solutions based on DH primitives, one of the drawbacks of this protocol lies in the pre-sharing assumption of the seeds, which affects both its scalability and feasibility.

In this context, we address the issue of group key management for dynamic and heterogeneous collaborative groups. The originality and features of our approach are detailed through the remaining sections. But first, to ease the understanding of our contribution, we provide the reader with a broad overview of the protocols upon which our approach is built.

4 Background

4.1 DBGK [3]

DBGK considers the group divided into sub groups. Each sub-group is managed by an Area Key Management Server ($AKMS$). The time axis is split into several time slots. For each time slot, a different ticket (piece of data) is issued. The group Traffic Encryption Key (TEK) for slot i is computed using a one way function F as follows:

$$TEK_i = F(SK, T_i)$$

where SK is a long term key, and T_i is the ticket issued for slot i.

Once an object (or member, both terms are used indistinguishably) O_i wants to join the group, it initiates DBGK which goes through successive phases. The object sends a join request through an anycast message. Based on the object location, the nearest $AKMS$ handles the join. Let us assume that the $AKMS$ of area j is the nearest one. In case of a successful authentication, the object is initialized (through a secure channel) with a long term key (i.e. SK), and a shared key with its $AKMS$. Despite being a valid member of the group, the new member O_i is not yet able to derive the current TEK. Backward secrecy is therefore inherently ensured while no rekeying operation is required for the group. If O_i is involved in a message exchange (sending/receiving), it has to be able to encrypt and decrypt the messages. To do so, O_i has to compute the current TEK. Thus, O_i sends a request to $AKMS_j$ asking for a ticket corresponding to the current time slot. In order to reduce the amount of exchanges in case O_i is highly active, the object can request several tickets corresponding to multiple future intervals. The request contains information about the objects specifications, in particular, data regarding its storage capabilities and resources. Based on this data, and on the trust level of O_i (if the object has previously been a member of the group), AKMS decides on the number of tickets to be granted to O_i.

When O_i leaves the network, forward secrecy has to be guaranteed to prevent the object from accessing future communications in the area. Two possible scenarios arise. In the first case, O_i leaves the network or is ejected with one or several valid tickets stored in its internal memory. In this case, $AKMS$ checks its AOL (Active Object List, which keeps track of the issued tickets) and sends a multicast notification to all the objects that have received the same tickets owned by the leaving member. The semantics of the notification is as follows. The tickets ranging from T_t to T_{t+k} (k corresponds to the number of tickets that O_i has received) are no longer valid. The recipients of the notification that are not active anymore (i.e. not in the process of exchanging messages) just ignore the notification. However, the active objects send a request to $AKMS$ in order to receive new tickets. Based on experimental results (see section IV.B in [3]), DBGK outperforms its peers within a proportion of around 50% of the members in possession of the same tickets as the leaving (ejected) member. If the proportion exceeds 50%, a state of the art approach (i.e. LKH [37]) is considered to rekey the whole group. In the second case, the leaving O_i does not own any valid ticket. In this situation, forward secrecy is ensured without any rekeying operation.

4.2 Piao et al. [25] and Patsakis and Solanas [24] Schemes

Piao et al. proposed a scalable and efficient polynomial based centralized group key management protocol to secure both inter-group and intra-group communications. Nevertheless, this scheme contains security breaches. In [16], authors show that Piao et al. scheme does not ensure neither backward nor forward secrecy. In [21] authors show that Piao et al. is based on a mathematical problem computable within a reasonable amount of resources (time and computation power). An attacker can easily factorize the polynomial over a finite field and retrieve the private keys of the members, as well as the exchanged secrets.

To address these issues, Patsakis and Solanas [24] proposed a modified version of Piao et al. [25] scheme to take advantage of its efficiency while strengthening its security properties. They base their scheme on a NP-hard mathematical problem which is finding the roots of univariate polynomials modulo large composite numbers for which the factorization is not known [26]. This is in contrast with the weak mathematical problem upon which Piao et al. [25] scheme is based. Moreover, they introduce an additional virtual term in the generation of the polynomial (called salting parameter) upon every rekeying to prevent backward and forward secrecy breaches.

In DsBGK, we build upon Patsakis and Solanas [24] scheme to secure the transmission of secrets using polynomial computation instead of using encryption. Hence, efficiency and scalability are both increased. Furthermore, we enhance Patsakis and Solanas scheme to ensure forward and backward secrecy more efficiently and to increase the collusion freeness of the protocol.

5 Network Model

Our network architecture models a group of entities collaborating to achieve a defined and common goal. This group is heterogeneous, and composed of both unconstrained and constrained entities. The unconstrained entities are powerful enough to perform asymmetric primitives (e.g. desktop computers, servers, smart phones, etc.). The constrained entities are limited in terms of energy, computational, communication and storage capabilities (e.g. sensors, RFID, NFC, etc.), hence, unable to perform asymmetric primitives. Unlike in DBGK, no General Key Management Server (GKMS) is considered. Furthermore, the group is not partitioned into subgroups with Area Key Management Servers (AKMS) controlling each sub group. In fact, we consider a single logical group where the unconstrained entities play the role of controllers. These controllers maintain a consistent, distributed and open AOL (Active Object List). This list can be maintained using one of the existing solutions in the literature, such as [23]. Figure 2 illustrates our network architecture.

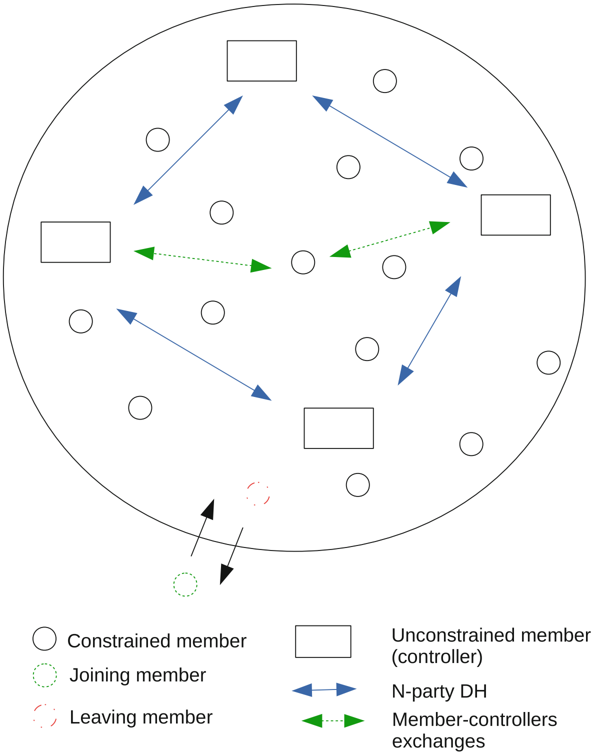

Constrained member
Joining member
Leaving member
Unconstrained member (controller)
N-party DH
Member-controllers exchanges

Fig. 2. Network architecture

5.1 Assumptions and Definitions

- we consider a heterogeneous group. More precisely, we assume the existence of both unconstrained members, powerful enough to perform periodic n-party Diffie-Hellman (DH) rekeyings [10], and constrained members unable to run the resource consuming n-party DH.
- the powerful entities are considered as controllers. Controllers are in charge of initiating a key update following specific events (e.g. join and leave).
- during the initialization phase, each new member is set (offline) with a private binding ID.
- during the initialization phase, at least one controller is pre-loaded (offline) with the binding ID of each new member (the ID can then be securely propagated to all controllers).
- a distributed AOL (i.e D-AOL) is maintained consistent between all controllers through the different updates.
- members are IP-enabled (6Lowpan for constrained members, and IPV6 for unconstrained members).
- we consider at a particular moment, only one active controller.

The different notations used throughout the remaining of this paper are summarized in Table 1.

6 Protocol Functioning

6.1 DsBGK General Overview

The goal of DsBGK is to establish and maintain a group key to secure communications in collaborative environments. This has to be achieved while remaining efficient and secure, ensuring both forward and backward secrecy. DsBGK is based on DBGK, we recommend the reader to refer to [3] for a comprehensive presentation of the protocol.

DsBGK proceeds within several phases. The first phase is related to the initialization of the entities. In fact, a set of unconstrained entities are designated off-line to play the role of controllers based on their capabilities. n-party DH is run within this sub-group of controllers to establish shared credentials. These latter are used to secure the communications required to update the distributed AOL (D-AOL). In addition, at least one controller is set with the secret binding ID of each new member. To become active, the new member sends a request to the active controller. The member requests one or more tickets according to its level of trust and resources capabilities. Upon successfully passing the authentication and authorization phase, the member receives the tickets along with SK (SK is only sent during the first exchange). The member will then be able to derive the group key using both the current ticket and the long term key SK. To secure the transmission of these tickets to the requesting members, the active controller builds a univariate polynomial of degree m. Upon its reception, the member computes the polynomial using its private binding ID to retrieve the

Table 1. Terminology table

Notation	Description
Group	A set of entities (members and controllers) collaborating by exchanging data in a Peer to Peer way to reach a common goal
Member (node)	An object of the group with limited resources capabilities (e.g. RFID, IP-enabled sensors, etc.)
Controller	An object of the group without hard resource constraints (e.g. personal computers, smartphones, servers, etc.)
TEK (Traffic Encryption Key)	The group key used to secure communications within the group. $TEK = F(SK, T_i)$
F	A one way function (easy to compute but hard to reverse)
SK	A long term key transmitted to each new member during its first exchange
Ticket (T_i)	Piece of data used in the generation of the TEK. T_i refers to the ticket issued for time slot i
Time slot	A defined period of time (e.g. seconds, minutes, days, etc.)
ID	Binding private identity of members. ID is used in the computation of polynomials
PublicID	Identity of the member
P(x)	Univariate polynomial modulo a composed large number n (product of two large primes $p * q$)
D-AOL	Distributed Active Object List: records all active members including the tickets they have received
SpecData	Data related to storage, processing capabilities, and trust level of members
Nslot	Number of requested time slots (tickets)

transmitted secret (i.e. tickets). The security of this scheme relies on the strength of the underlying mathematical problem. In this case, the problem comes down to finding the roots of univariate polynomials modulo large composite numbers. Upon a leave event, two situations arise. If the leaving member has not recently been active, then, no rekeying is required. However, if the leaving member is active, its tickets are no longer valid. As a result, the information stating that these tickets are no longer valid has to be propagated to the concerned members by the active controller. In the following, we present the details of DsBGK phases.

6.2 Initialization (Joining)

During this phase, the private binding ID of the member is communicated to at least one controller (typically the active controller). Upon successful authentication and authorization, the controller propagates the ID to the rest of controllers. We assume that the ID of a new members is set offline. This ID will be used to compute the received polynomials from controllers to retrieve exchanged secrets. Once the ID is set, the member is valid and can become active at any moment.

6.3 Activation

Algorithm 1 depicts the behaviour of DsBGK following a join event. After successfully joining the group, a member becomes active by requesting one (or several) tickets from the active controller. Indeed, any controller is able to deliver tickets to members, as D-AOL is distributed and maintained between all controllers. This provides a better fault tolerance compared to DBGK where only the controller, in charge of a specific area, can deliver the tickets. Upon receiving a request, a controller grants or deny the request based on several parameters related to the requesting member such as, resources capabilities and the level of trust. To secure the transmission of tickets, the active controller generates a univariate polynomial $P(x)$ modulo the product of two large prime numbers (see Algorithm 2).

$$P(x) = (x - r_1)(x - ID)(x - r_2) \dots (x - r_m) + T_i \ mod \ n$$

This polynomial represents the product of m terms plus the transmitted secret (i.e. T_i). One of the terms (i.e. $x - ID$) allows the receiving member to compute $P(ID) = 0$ to retrieve the secret. The remaining terms are set randomly.

In both Patsakis and Solanas [24] and Piao et al. [25] schemes, the terms are composed of the private credentials of the members (i.e. ID). As a result, to mitigate collusion attacks and to provide backward and forward secrecy, Patsakis and Solanas in [24] introduce the use of additional terms upon each rekeying (called salting parameters). In DsBGK, we propose to avoid using additional parameters, which can quickly increase the ratio between the polynomial degree and the actual number of users (members) within the group.

In the original Piao et al. scheme, if a new member l joins the group, this latter could breach backward secrecy (i.e. accessing data exchanged prior to the joining).

Indeed, let us consider $P_{old}(x)$ the polynomial generated before the joining, $P_{new}(x)$ the polynomial generated after the joining, n the number of users, and s the transmitted secret.

$$P_{old}(x) = (x - ID_1) \dots (x - ID_n) + s \ mod \ n$$

$$P_{new}(x) = (x - ID_1) \dots (x - ID_l) \dots (x - ID_{n+1}) + s' \ mod \ n$$

The new member m would derive the old secret s by computing:

$$s = P_{old}(x) - \frac{P_{new}(x) - s'}{x - ID_l}$$

In DsBGK, this attack would not possible, as computing $\frac{P_{new}(x)-s'}{x-ID_l}$ would give no extra knowledge considering that the terms are defined randomly (except the term that contains the ID of the recipient member) and thus vary across the different polynomials.

Furthermore, DsBGK ensures collusion freeness as the disclosure of the private ID of colluding users brings no additional knowledge to retrieve private IDs of non-colluding members. Indeed, in each polynomial, apart from the term containing the recipient ID, the remaining terms are random and different across the polynomials. Besides, we set the degree m of the polynomial in a way to keep the factorization not easily feasible while maintaining efficiency. In [20], experimentations on MICA2 sensor showed that the computation of a polynomial of a degree up to 40 is more efficient than symmetric encryption (i.e. RC5). In DsBGK, we set m accordingly and regardless of the number of users in the group. Thus, the size of the polynomial does not grow with the growth of the number of users (members), which has a positive impact on scalability.

6.4 Leaving

To ensure forward secrecy upon a leaving event, the TEK is changed. In DsBGK, two scenarios are considered. If the leaving (ejected) member at time slot i is not in possession of valid tickets T_{i+k} (with $k \geq 0$), no rekeying is required. In fact, the leaving member will not be able to derive future TEK given the fact that group keys are partly composed of dynamic tickets. As a result, the leaving member will not have access to future communications. However, if the leaving member is in possession of tickets, the members in possession of the same tickets need to be notified. In case they are still active, they will ask for new tickets. The exchange of these secret credentials is secured using univariate polynomials generated by the active controller (see Algorithm 3).

Algorithm 1. Activation algorithm

```
procedure Activation (member, controller)
    $request \leftarrow Ticket\_request\{PublicID, SpecData, Nslot\}$
    $Member.send(request, controller)$
    if $member\ is\ authenticated$ then
        if $member\ is\ authorized$ then
            while $i < number\ of\ granted\ tickets$ do
                $P_1 \leftarrow GeneratePoly(T_i)$
                $i \leftarrow i + 1$
            if $first\ activation$ then
                $P2 \leftarrow GeneratePoly(SK)$
                $Controller.Send(P1, member)$
                $Controller.Send(P2, member)$
            else
                $Controller.Send(P1, member)$
            $Update\ D\_AOL(controller,\ PublicID)$
```

7 Analysis

7.1 Security Properties

Backward secrecy violation occurs when a legitimate member tries to access communications, which took place before its joining. In DsBGK, backward secrecy is

Algorithm 2. Polynomial generation algorithm

```
procedure GeneratePoly (secret)
    $p \leftarrow randomly\ generated\ large\ prime\ number$
    $q \leftarrow randomly\ generated\ large\ prime\ number$
    $n \leftarrow p \times q$
    $m \leftarrow fixed\ threshold$
    $P \leftarrow (x - ID)$
    while $i < m - 1$ do
        $r \leftarrow random\_value()$
        $P \leftarrow P \times (x - r)\ mod\ n$
    $P \leftarrow P + secret$
    $return(P)$
```

Algorithm 3. Leaving algorithm

```
procedure Leaving (member, controller)
                                        ▷ retrieving tickets of the leaving member
    $tickets \leftarrow controller.lookup(D\_AOL, member)$
    if $tickets \neq null$ then
                                    ▷ retrieving members holding the same tickets
        $list \leftarrow controller.lookup(D\_AOL, tickets);$
        $threshold \leftarrow 50\%\ of\ total\ number\ of\ members$
        if $list.length < threshold$ then
            while $list \neq null$ do
                                                  ▷ concerns only active members
                $controller.notify(member)$
                $activation(member, controller)$
        else                                   ▷ rekey the whole group using LKH
            $LKH(SK)$
```

ensured inherently, as joining members are not able to derive group keys which have been established prior to their joining. In fact, the group key is composed of a fixed long term key and varying tickets following each time slot. As a result, new members are unable to derive previous keys.

Forward secrecy violation occurs when a former member of the group tries to access communications, which take place after its departure from the group. In DsBGK, this property is ensured based on whether the leaving member is in possession of tickets or not. If the member is not in possession of tickets, no rekeying is required. In fact, the leaving member will not be able to derive any future group keys. However, if the member is in possession of valid tickets, using $D\text{-}AOL$, the active controller notifies only the active members which are in possession of the same tickets about their non-validity. In case the number of active members reaches a certain threshold (set experimentally to 40–50% of the total number of members in the group), the active controller relies on the state of the art LKH protocol to rekey the long term key SK. As a result, the leaving member will not be able to use its tickets to derive future group keys, either because they are not valid anymore (and thus not used in the generation of the group key) or because the long term key has been modified.

Collusion attacks occur when two or more legitimate members collude to retrieve the security credentials of other members. In DsBGK, secret credentials are securely exchanged using univariate polynomials modulo a composite number of large primes. We ensure collusion freeness by considering variable terms, which are not based on the credentials of the users (members). Indeed, the collusion of a subset of members will not help in any form to compose polynomials with the goal of retrieving the security credentials of the remaining members. Nevertheless, this solution requires from the controller to compose a different polynomial for each member. It is worth noting, however, that the controllers are not considered as constrained members, and DsBGK main goal is to reduce the overhead with respect to the constrained members of the group.

7.2 Performance Evaluation

To analyze the performances of DsBGK and compare the results with DBGK [3], we relied on Cooja, which is the built-in network simulator of Contiki 2.7 [1]. Contiki is an open source Operating System (OS) for IP-enabled constrained devices (objects). This OS is used by the research community in several domains, such as, networked electrical systems, industrial monitoring, e-health sensors, and in Internet of Things (IoT) related applications in general. With the purpose of assessing our protocol's performances compared to DBGK's performances, we considered the same experimental setups as those used in the evaluation of DBGK. In fact, we use Tmote Sky nodes, which are equipped with the CC2420 radio chip and the MSP430 microcontroller (10k RAM, 48k Flash). Furthermore, energy consumption is computed using Powertrace tool [14]. This tool measures the time (number of ticks) during which each element (e.g. CPU, transmission, reception, etc.) of the sensor is active. This duration is combined with other data (specific to the sensor, such as the current draw, and voltage) to evaluate the energy consumption. We evaluated DsBGK performances with respect to the following metrics: storage overhead, polynomial degree, and members leave cost.

Storage overhead: In this experiment, we considered an event where a new constrained member (denoted merely by 'member' in the remaining of this analysis) joins a group. We varied the number of controllers (KMS) in order to assess the impact of additional controllers on the overhead resulting from the storage of security materials by members. The results, depicted in Fig. 3, show that for DBGK, storage overhead increases linearly with the inclusion of additional controllers. However, for DsBGK, storage overhead is steady and not related to the number of controllers. In fact, in DBGK, a pre-shared key is established between each member and each controller. This leads to a proportional dependency between the number of controllers and the number of stored keys. Indeed, in DsBGK, thanks to the use of polynomials, a pre-shared material (i.e. ID) is only set in the controller side for each additional member. Nevertheless, no material is stored in the member side. Consequently, unlike DBGK, DsBGK allows adding controllers with no impact on storage overhead.

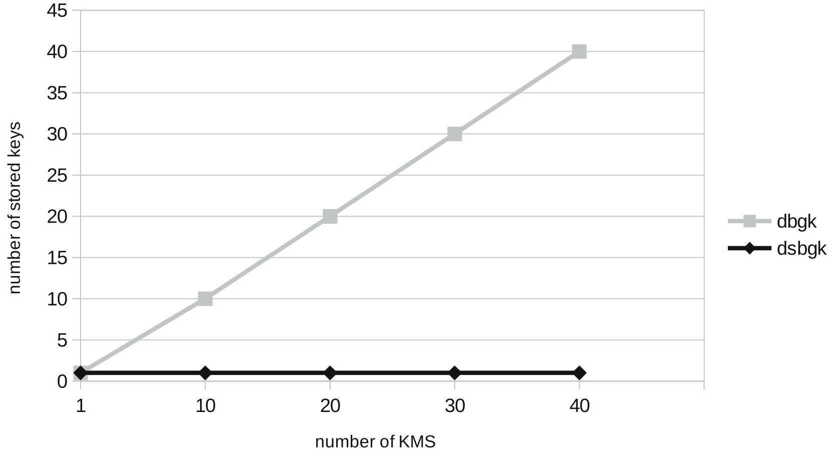


Fig. 3. Storage overhead

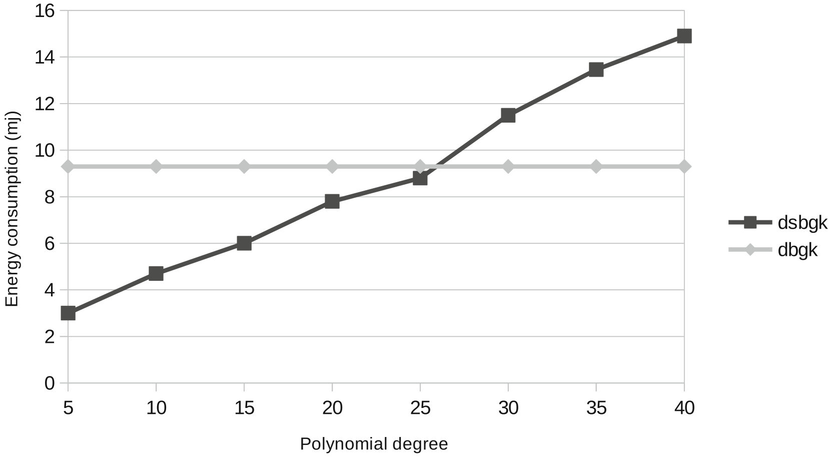


Fig. 4. Polynomial degree

The next step in our evaluation was to evaluate the impact of this gain in storage on the energy consumption induced by rekeying operations. In particular, when members leave or are ejected from the group. But first, we ran extensive simulations to set the optimal degree of the polynomial to achieve the best trade-off between security and efficiency.

Polynomial degree: We considered a group of 1000 members. We simulated a member leaving the group (or being ejected) with a proportion of 40% of remaining members holding the same tickets as the leaving member. Based on DBGK evaluation (see section IV.B in [3]), around 40–50% represents the maximum proportion above which DBGK efficiency drops and a state of the art protocol

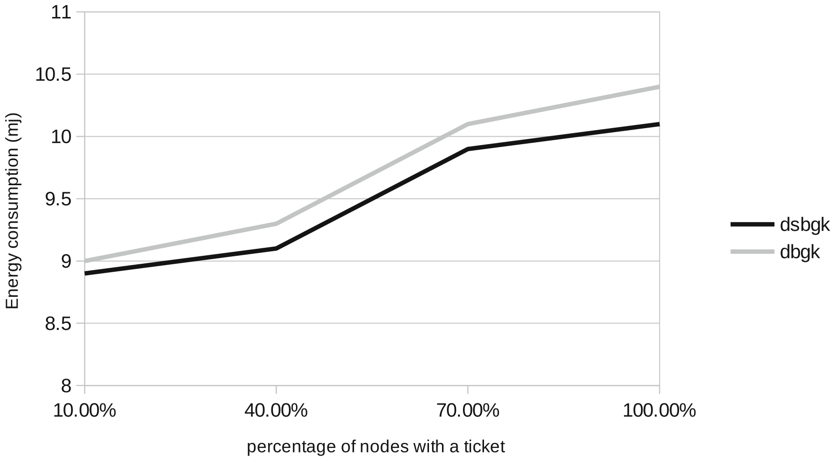


Fig. 5. Member leaving cost

(i.e. LKH [37]) is preferred to update the group key. Furthermore, $NSlot$ has been set to 20, which we consider being a realistic value. We varied the degree of the polynomial and compared energy cost with DBGK. The results presented through Fig. 4 highlight a steady raise in energy consumption with the increase of the polynomial degree. It is worth mentioning that DBGK energy cost is not impacted by polynomial degree variation, hence the constant energy consumption. Eventually, DsBGK energy cost exceeds DBGK energy cost when the degree reaches a value around 25.

Our results were slightly different compared to the experimental results presented in [20] (previously mentioned in Sect. 6.3), where performances using polynomial computation were better, up to a degree of 40. We explain this difference by the fact that we used a different sensor in our experiment (Sky mote) in addition to a different encryption primitive for DBGK (i.e. AES). Nonetheless, this variation does not alter the security foundations of DsBGK, as the NP-hard mathematical problem upon which DsBGK is based is not altered [26]. Following this experiment, we compared the energy consumptions of DBGK and DsBGK in case of a leave event to make sure that the gain in storage cost has not been achieved at the expense of other metrics.

Member leave cost: We estimated the energy cost related to the departure (or ejection) of a member in possession of a valid ticket. Similarly to DBGK's evaluation, we consider a group of users composed of 1000 members. We record several measures, while varying the proportion of members with tickets similar to those in possession of the leaving member. Moreover, we define the number of tickets requested by notified members as equal to 20 time slots (i.e. $NSlot = 20$). We depict the results in Fig. 5. It is clear that DsBGK energy consumption increases with the increase of the percentage of members in possession of the same tickets as leaving members. However, this raise in energy cost is slightly lower compared to the raise noticed in DBGK energy consumption. This is mainly due to the superior efficiency of polynomial computation compared to cryptographic symmetric primitives.

Based on the obtained results, we can affirm that compared to DBGK, DsBGK provides a considerable improvement in fault tolerance and scalability. Not only this result does not incur additional overhead with respect to rekeying operations, but an improvement in energy consumption is also achieved. Back to our e-health use case scenario, presented in Sect. 2, DsBGK can be applied to efficiently secure data exchanges in such sensitive environment where the unconstrained entities (e.g. PC, smartphones, etc.) can play the role of controllers. These controllers will be in charge of efficiently managing the group key for the constrained members of the group (i.e. health related sensors). Additional controllers can be included without incurring any additional storage cost on constrained members. Thus, the failure of one or several controllers does not hinder the protocol functioning, as other controllers can take over. Furthermore, the improved efficiency is highly sought for battery powered e-health sensors. Indeed, these sensors can be planted inside human bodies. Increasing the life time of their battery would reduce the cycle of surgical interventions required for their replacement.

8 Conclusions and Perspectives

Securing distributed collaborative applications in the era of the Internet of Things relies heavily on strong and efficient group key management protocols. In this paper, we combined a polynomial based approach with our previously proposed protocol (DBGK) to propose a new protocol called DsBGK. Experimental analysis showed that DsBGK improves both fault tolerance and scalability which are highly sought in sensitive applications, such as e-health systems. Energy gains are also achieved, which makes DsBGK suitable for heterogeneous, and dynamic collaborative groups. We plan to further investigate DsBGK security strength by thoroughly assessing properties such as data integrity, data authentication, and data confidentiality through an implementation using automated formal validation tools (e.g. Avispa, Scyther). In addition, we are currently investigating a lightweight blockchain based scheme to allow sensors authenticating genuine controllers.

References

1. The Contiki Operating System. http://www.contiki-os.org
2. Abdmeziem, M.R., Tandjaoui, D.: An end-to-end secure key management protocol for e-health applications. Comput. Electr. Eng. **44**, 184–197 (2015)
3. Abdmeziem, M.R., Tandjaoui, D., Romdhani, I.: A decentralized batch-based group key management protocol for mobile internet of things (DBGK). In: 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), pp. 1109–1117. IEEE (2015)

4. Abdmeziem, M.R., Tandjaoui, D., Romdhani, I.: Architecting the internet of things: state of the art. In: Koubaa, A., Shakshuki, E. (eds.) Robots and Sensor Clouds. SSDC, vol. 36, pp. 55–75. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-22168-7_3
5. Abdmeziem, M.R., Tandjaoui, D., Romdhani, I.: A new distributed MIKEY mode to secure e-health applications. In: Proceedings of the International Conference on Internet of Things and Big Data, IoTBD, vol. 1, pp. 88–95. SciTePress (2016)
6. Abdmeziem, M.R., Tandjaoui, D., Romdhani, I.: Lightweighted and energy-aware MIKEY-ticket for e-health applications in the context of internet of things. Int. J. Sens. Netw. (2017, in press)
7. Balenson, D., McGrew, D., Sherman, A.: Key management for large dynamic groups: one-way function trees and amortized initialization. Internet-Draft, February 1999
8. Ballardie, A.: Scalable multicast key distribution. RFC 1949, May 1996
9. Briscoe, B.: MARKS: zero side effect multicast key management using arbitrarily revealed key sequences. In: Rizzo, L., Fdida, S. (eds.) NGC 1999. LNCS, vol. 1736, pp. 301–320. Springer, Heidelberg (1999). https://doi.org/10.1007/978-3-540-46703-8_19
10. Challal, Y., Seba, H.: Group key management protocols: a novel taxonomy. Int. J. Inf. Technol. **2**(1), 105–118 (2005)
11. Daghighi, B., Kiah, M., Shamshirband, S., Rehman, M.: Toward secure group communication in wireless mobile environments: issues, solutions, and challenges. J. Netw. Comput. Appl. **50**, 1–14 (2015)
12. Di Pietro, R., Mancini, L.V., Jajodia, S.: Providing secrecy in key management protocols for large wireless sensors networks. Ad Hoc Netw. **1**(4), 455–468 (2003)
13. Dini, G., Lopriore, L.: Key propagation in wireless sensor networks. Comput. Electr. Eng. **41**, 426–433 (2015)
14. Dunkels, A., Eriksson, J., Finne, N., Tsiftes, N.: Powertrace: network-level power profiling for low-power wireless networks (2011)
15. Harney, H., Muckenhirn, C.: Group key management protocol (GKMP) architecture. RFC 2093, July 1997
16. Kamal, A.A.: Cryptanalysis of a polynomial-based key management scheme for secure group communication. IJ Netw. Secur. **15**(1), 68–70 (2013)
17. Keoh, S.L., Kumar, S.S., Tschofenig, H.: Securing the internet of things: a standardization perspective. IEEE Internet Things J. **1**(3), 265–275 (2014)
18. Kim, Y., Perrig, A., Tsudik, G.: Tree-based group key agreement. ACM Trans. Inf. Syst. Secur. (TISSEC) **7**(1), 60–96 (2004)
19. Lee, P., Lui, J., Yau, D.: Distributed collaborative key agreement and authentication protocols for dynamic peer groups. IEEE/ACM Trans. Netw. **14**(2), 263–276 (2006)
20. Liu, D., Ning, P.: Security for Wireless Sensor Networks, vol. 28. Springer Science & Business Media, Heidelberg (2007). https://doi.org/10.1007/978-0-387-46781-8
21. Liu, N., Tang, S., Xu, L.: Attacks and comments on several recently proposed key management schemes. IACR Cryptology ePrint Archive 2013:100 (2013)
22. Mittra, S.: Iolus: a framework for scalable secure multicasting. ACM SIGCOMM Comput. Commun. Rev. **27**(4), 277–288 (1997)
23. Oster, G., Urso, P., Molli, P., Imine, A.: Data consistency for P2P collaborative editing. In: Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work, pp. 259–268. ACM (2006)
24. Patsakis, C., Solanas, A.: An efficient scheme for centralized group key management in collaborative environments. IACR Cryptology ePrint Archive 2013:489 (2013)

25. Piao, Y., Kim, J., Tariq, U., Hong, M.: Polynomial-based key management for secure intra-group and inter-group communication. Comput. Math. Appl. **65**(9), 1300–1309 (2013)
26. Plaisted, D.A.: New NP-hard and NP-complete polynomial and integer divisibility problems. Theor. Comput. Sci. **31**(1–2), 125–138 (1984)
27. Rafaeli, S., Hutchison, D.: Hydra: a decentralized group key management. In: 11th IEEE International WETICE: Enterprise Security Workshop, June 2002
28. Rafaeli, S., Hutchison, D.: A survey of key management for secure group communication. ACM Comput. Surv. (CSUR) **35**(3), 309–329 (2003)
29. Setia, S., Koussih, S., Jajodia, S., Harder, E.: Kronos: a scalable group re-keying approach for secure multicast. In: Proceedings IEEE Symposium on Security and Privacy, pp. 215–228 (2000)
30. Shamir, A.: How to share a secret. Commun. ACM **22**(11), 612–613 (1979)
31. Sicari, S., Rizzardi, A., Grieco, L.A., Coen-Porisini, A.: Security, privacy and trust in internet of things: the road ahead. Comput. Netw. **76**, 146–164 (2015)
32. Sicari, S., Rizzardi, A., Miorandi, D., Coen-Porisini, A.: Internet of things: security in the keys. In: Proceedings of the 12th ACM Symposium on QoS and Security for Wireless and Mobile Networks, pp. 129–133. ACM (2016)
33. Tang, P.C., Ash, J.S., Bates, D.W., Overhage, J.M., Sands, D.Z.: Personal health records: definitions, benefits, and strategies for overcoming barriers to adoption. J. Am. Med. Inform. Assoc. **13**(2), 121–126 (2006)
34. Veltri, L., Cirani, S., Busanelli, S., Ferrari, G.: A novel batch-based group key management protocol applied to the internet of things. Ad Hoc Netw. **11**(8), 2724–2737 (2013)
35. Wang, W., Bhargava, B.: Key distribution and update for secure inter-group multicast communication. In: Proceedings of the 3rd ACM Workshop on Security of ad Hoc and Sensor Networks, pp. 43–52. ACM (2005)
36. Wang, W., Wang, Y.: Secure group-based information sharing in mobile ad hoc networks. In: IEEE International Conference on Communications, ICC 2008, pp. 1695–1699. IEEE (2008)
37. Wong, C., Gouda, M., Lam, S.: Secure group communications using key graphs. IEEE/ACM Trans. Netw. **8**(1), 16–30 (2000)

Author Index