Journal of Engineering and Applied Sciences 11 (3): 369-374, 2016
ISSN: 1816-949X

Indexing Android Mobile Malware

[1]Intan Nurfarahin Ahmad, [1,2]Farida Ridzuan, [1,2]Madihah Mohd Saudi
and [1,2]Sakinah Ali Pitchay
[1]Faculty of Science and Technology (FST), [2]Islamic Science
Institute (ISI), Universiti Sains Islam Malaysia (USIM),
Bandar Baru Nilai, Nilai, Malaysia

Abstract: The increasing number of smartphones users have resulted in highly distributed applications that allow users to access information and resources from all over the world. With the advancement of technology, the attacker had created more sophisticated techniques that gave negative impacts to the smartphone users. Moreover, current techniques in mobile malware classification and detection having difficulties to detect the new advanced malware exploitation and threats. Therefore, an efficient mobile malware classification and detection technique are needed. In this research study, a new mobile malware classification and detection technique have been developed and evaluated. Based on the evaluation conducted, the result showed that the current mobile malware available in market is using different technique to avoid from being detected. Therefore in this research study, a way forward to detect such mobile malware is further discussed. Furthermore, the developed mobile malware classification and detection is used as the input for the indexing android mobile malware a framework for indexing android mobile malware also is documented in this research study.

Key words: Exploitation, indexing android mobile malware, detection technique, malware, classification

INTRODUCTION

The mobile phone user has upraised from 12% of the world population in 2000 up to 96% in 2014 which is 6.8 million user (Blondel *et al*., 2015). In the past few years malware has become one of the most serious threats for most of the smartphone user. In contrast to other platforms such as iOS which allow user to install apps that are only available in the iTunes App Store, Android continues to be the most targeted mobile operating system as it allows user to install applications from various sources such as Google Playstore, third-party markets, torrents, or direct download (Wang *et al*., 2015). Obviously, this freedom creates big hole for the attacker to inject the malware into the application while the victims unconsciously execute it.

The typical malware types include virus, worms, spyware, Trojan horse, rootkit and botnet infect and take control of the mobile phone vulnerability and use them to facilitate other criminal activities and gain illegal profit (Hu, 2011). Figure 1 shows the new mobile malware variant motivated by profit (F-Secure, 2014). Hence, defense against mobile malware threat that includes preventing the malware attack from occurring, limiting its activities or recovering from malware after it has occurred is essential.

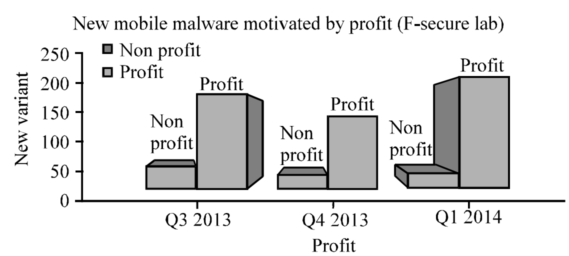

Fig. 1: New mobile malware variant motivated by profit

The ease of malware-mutation process has led to an explosive increase in the number of new malware variant with more advanced features. The characteristics of the malware which can both be structured and unstructured makes it difficult or almost impossible to be processed into knowledgeable structure. In the context of data analysis, the accurate result depends on the veracity of data sources (Wang *et al*., 2015). Therefore, a suitable indexing rule is needed to increase the effectiveness of malware classification and detection (Aamodt *et al*., 1998).

The objectives of this research study are to evaluate the proposed mobile malware classification and detection that will be used as the input for the proposed indexing mobile malware framework.

Corresponding Author: Intan Nurfarahin Ahmad, Faculty of Science and Technology (FST), Universiti Sains Islam Malaysia (USIM), Bandar Baru Nilai, Nilai, Malaysia

Table 1: Comparison of mobile malware detection techniques and features selection

References	Features used	Techniques	Strength	Weakness
Wang *et al.* (2013)	Permission	Static analysis	Generate reliable risk signal for warning the potential malicious activities	Insufficient for detecting more sophisticated application
Wu *et al.* (2012)	Intent filter (manifest file and API calls)	Static analysis	Improvement in malware detection	Two Android malware families failed to be detected which are Droid Kung Fu and BaseBridge
Sanz *et al.* (2014)	Strings	Static analysis	New sample can be detected using previous information	High error rates for anomaly detection system
Saudi *et al.* (2015)	System call	Dynamic analysis	60 patterns of system call combination that exploit call logs	Only focus on system call exploitation
Zhao *et al.* (2011)	System call	Dynamic analysis	High detection rate and low rate of false positive and false negative	Insufficient characteristics to detect more new malicious application
Bläsing *et al.* (2010)	Permissions, java code and system call Android	Static and dynamic analysis	Automatically detect malicious applications	No machine-learning techniques implemented
Wei *et al.* (2012)	Manifest.xml,Java code, user,interaction system call and network traffic	Static and dynamic analysis	Able to discover new behavioral characteristics	

Literature review: Malware is a malicious code that is built by the attacker or criminal to perform any activities in victim's devices such as computer, smartphones and tablet (Saudi *et al*., 2015).

They can perform bad tasks such as destruction of data, steal personal information or gain access to system resources in order to control the devices.

Mobile malware detection and classification techniques: Mobile malware comes in with different type of structure, characteristics and behaviors. Generally, they have similar propagation and exploitation methods. A few common characteristics exists in Android malware are listed as follow (Wang *et al*., 2013; Saudi *et al*., 2015).

Activation: As the application install, the process of the application activated in the mobile phone.

Malicious payloads: For Android malware, the payload functionalities can be divided into four different categories which are privilege escalation, remote control, financial charges and personal information.

Malware installation: For malware installation characteristics, generally Android malware can be categorized into three main groups based on social engineering techniques which are repackaging, update attack and drive-by download.

Permission uses: For Android applications that do not exploits the root, their abilities are limited and constrained by the user permissions granted to them.

To cope with the growing amount and complex malware's characteristics, a large number of concepts and techniques have been proposed and they are mainly categorized into static analysis and dynamic analysis (Wang *et al*., 2015).

Static analysis: Research includes (Zhao *et al*., 2011; Wu *et al*., 2012) was carried out using static analysis for mobile malware detection. The mechanism of static analysis is by looking at the files by disassembly and de-compilation without actually running the program.

Dynamic analysis: Research such as (Saudi *et al*., 2015; Zhao *et al*., 2011; Blasing *et al*., 2010) use dynamic analysis for malware classification and detection. Dynamic analysis includes executing the mobile malware dataset in the controlled lab and carefully watch their behavior and actions.

Comparison between existing researches based on the features used, techniques, strengths and weaknesses are summarized in Table 1.

Existing indexing mobile malware research: Graph similarity had widely implemented to help malware analyst to detect new malicious application and has commonly be used for information indexing (Hu *et al*., 2009). Other than that graph similarity was used as the new method to classify new mobile malware (Park *et al*., 2010). However, graph similarity has several disadvantages. It is unsuitable to be implemented for large malware database and cannot effectively capture similarity among malware as all the indexing features need to be exactly matched. In Case-Based Reasoning (CBR), indexing rule helps to retrieve the information needed to represent the knowledge (Esmaili *et al*., 1996). CBR technique gives a promising result for a system on how to handle a specific security incident for mobile malware attack (Zakaria, 2015; Micarelli and Sansonetti, 2007). The main principle behind CBR is based on the concept that similar problem has similar solution (Fanoiki *et al*., 2010). CBR is used to make the best matching case in the case base and approximate malware classification is retrieved.

Table 2 summarizes the research that implemented indexing algorithm in order to improve malware classification and detection. Research related to

Table 2: Comparison table of indexing approach used in rsearch

Refences	Domain	Algorithm	Strength	Weakness
Micarelli and Sansonetti (2007)	Malware system call	Earth Mover's Distance (EMD)	Enables evaluation of dissimilarity between two multi-dimensional distribution	Every new case was discarded
Park *et al.* (2010)	Malware system call	Graph matching	Automated malware classification	Graphs distortions
Hu *et al.* (2009)	Malware function-call	Graph matching	Detection with large graph similarity	Fail to identify all the functions in a malware binary
Fanoiki *et al.* (2010)	UCI machine learning repository	Fuzzy similarity relationships	AUTOGUARD converts the low level audit trail into high level class representation	Undesirable side effect if deriving the whole set of hyper edges from hyper graph of events
Berkat (2011)	Virus dataset	Calculate similarities between new and stored cases	Detection of new viruses is stored automatically in the database	Limited of uses of data set

mobile malware analysis usually uses CBR to increase the efficiency of malware classification and detection. However, the huge range of features used drew the limit to the proposed methodology.

MATERIALS AND METHODS

The overall processes involved in this research is illustrated in Fig. 2. The dataset used consist of 1260 training dataset from android Malware Genome Project (Zhou and Jiang, 2012) and 100 testing dataset gathered from Google Apps Store website.

A controlled laboratory environment is developed as illustrated in Fig. 3. Almost 80% of the tools used for this research were open sources.

Dynamic analysis is used to extract the system call from both dataset. The behavior of an application was monitored through system calls that can be generated based on the user interaction with the application. The processes of analysis involve:

- Start the Android Virtual Device from the Software Development Kit (SDK)
- Installation of the binaries using (adb install xxx.apk)
- Emulate the device using Android Debug Bridge (ADB)
- List up the parent process of the Android application (ps)
- Monitor the running application's system call using Strace tools

There were two methods implemented to classify the bad system call, which are percentage of occurrence and covering algorithm. In this research, the occurrence of the system calls is noted as 1 to indicate the presence of system call and 0 to indicate the absence of system call in an application. Covering algorithm was used to generate system call pattern for each application. The system call classification was developed by identifying the rule that cover some instances of an application.

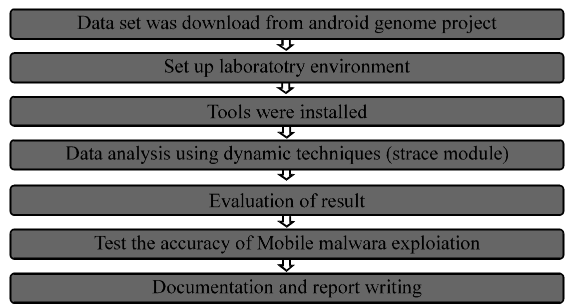

Fig. 2: Overall Research Process Involved

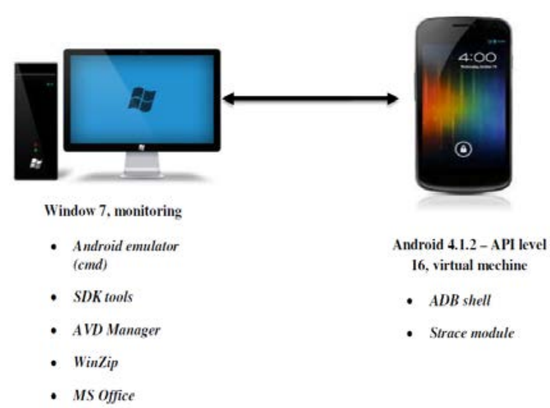

Fig. 3: Laboratory eanvironment

RESULTS AND DISCUSSION

In this experiment, thousands of system calls have been retrieved. Based on the 1260 samples extracted, there are 60 patterns of system calls lead to financial charges such as automatically cause financial charges such as automatically making phone call and reroute outgoing calls (Saudi *et al.*, 2015). Figure 4 shows example of system call patterns (Saudi *et al.*, 2015). The details of the mobile malware is not discussed in their research study. Further details can be referred in study (Saudi *et al.*, 2015).

No	Patterns
P1	a3+a4+a5+a6+a7+a8+a9+a10+a11+a12+a13+a14+a16+a17+a18+a20+a21+a28+a29+a31+a32+a39+a40+a42+a43+a47+a56+a57+a58+a59+a60+a62+a66
P2	a3+a4+a5+a6+a7+a8+0+a10+a11+a12+a13+a14+a16+a17+a18+a20+a21+a28+a29+a39+a40+a42+a56+a57+a58+a59+a62+a66
P3	a3+a4+a5+a6+a7+a8+a9+a10+a11+a12+a13+a14+a16+0+a18+0+a20+a21+a28+a29+a40+a42+a43+a47+a56+a57+a58+a59+a62+a66
P4	a3+a4+a5+a6+a7+a8+a9+a10+a11+a12+a13+a14+a16+a17+a18+0+a20+a21+a28+a29+a31+a32+a39+a40+a43+a47+a56+a57+a58+a59+a60+a62+a66
P5	a3+a4+a5+a6+a7+a8+a9+a10+a11+a12+a13+a14+a16+a18+a20+a21+a28+a29+a40+a43+a47+a56+a57+a58+a59+a62+a66
P6	a3+a4+a5+a6+a7+a8+a10+a11+a12+a13+a14+a16+a17+a18+a20+a21+a28+a29+a56+a58+a59+a62+a66
P7	a3+a4+a5+a6+a7+a8+a9+a10+a11+a12+a13+a14+a16+a17+a20+a21+a26+a29+a41+a56+a58+a59+a62

Fig. 4: Example of combination of system calls patterns

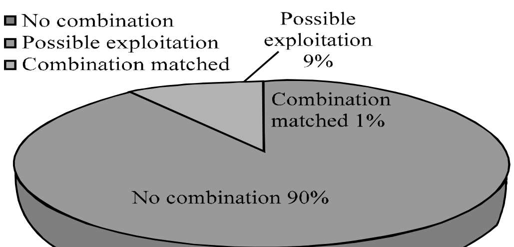

Fig. 5: Mobile malware detection result for system Call exploitation

Table 3: Mobile malware detection category

Category	Description
No combination	Not matched with any of the proposed patterns
Possible exploitation	Not matched with the proposed patterns but allowed permission shows money exploitation
Combination matched	Match with the proposed patterns and allowed permission shows money exploitation

For evaluation purpose, it was carried out by using 100 anonymous dataset gathered from Google Playstore. The accuracy of data classification were evaluated based on three category as illustrated in Table 3. Figure 5 presents the overall result of mobile malware detection accuracy.

The results in Fig. 5 shows that 90% out of 100 samples used to test the accuracy of the proposed patterns did not match with any pattern. No pattern matched means that the application do not execute any suspicious activity regarding to financial exploitation through call logs. The result also shows that 9% of the sample has the possibility in exploiting user call logs. This category describes that the system call of an application do not match with any of the proposed patterns but there is possibility that they also execute bad activities in user devices. Based on the 100 samples gathered from Google Playstore, only one that matched combination of patterns. The study shows that although the system call of an application does not shows the exploitation of call logs, but the bad activities can possibly occur in user devices. Therefore, an efficient mobile malware classification and detection technique are needed. Furthermore, the pattern will be used as the input for indexing mobile malware as displayed in Fig. 6. While for indexing mobile malware CBR is seen as one of the primary result for the implementation. Figure 6 shows the indexing mobile malware framework that uses CBR approach.

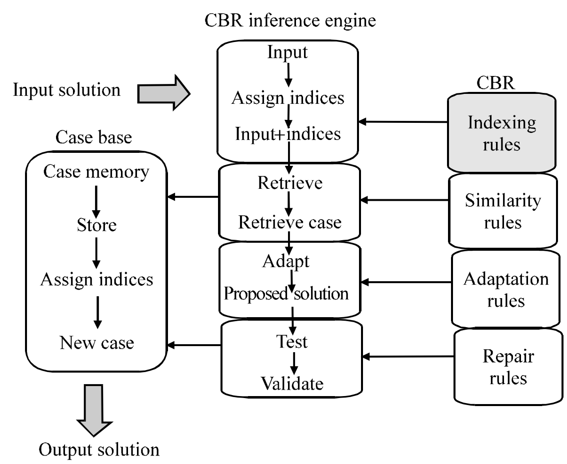

Fig. 6: Case-Based Reasoning (CBR) architecture for indexing mobile malware

CONCLUSION

As a conclusion, this study presents the evaluation of the developed mobile malware classification and a framework of an indexing rule for mobile malware information retrieval. It shows the significant of identifying and using mobile malware patterns for mobile malware classification and detection. Case-Base Reasoning (CBR) approach is implemented to enhance the effectiveness of the indexing mobile malware. This is part of larger research project to design automated indexing mobile malware retrieval system.

ACKNOWLEDGEMENTS

The researchers would like to express their gratitude to Ministry of Education Malaysia, Universiti Sains Islam Malaysia and Islamic Science Institute (ISI), USIM for the support and facilities provided. The research study is supported by MOE [FRGS/1/ICT04/USIM/02/1], [PPP/UCG-0114/FQS/30/11714], [USIM/NRGS-P4/FST/8404/52113],RAGSgrants[RAGS/FST/36/51013], [RAGS/FST/36/50914] and [RAGS/FST/36/50814].

REFERENCES

Aamodt, A., H.A. Sandtorv and O.M. Winnem, 1998. Combining case based reasoning and data mining-A way of revealing and reusing RAMS experience. Proceedings of the European Conference on Safety and Reliability, June 16-19, 1998, Trondheim, Norway, pp: 1345-1351.

Berkat, A., 2011. Using Case-Based Reasoning (CBR) for detecting computer virus. Int. J. Comput. Sci. Issues, 8: 606-610.

Blasing, T., L. Batyuk, A.D. Schmidt, S.A. Camtepe and S. Albayrak, 2010. An android application sandbox system for suspicious software detection. Proceedings of the 5th International Conference on Malicious and Unwanted Software, October 19-20, 2010, Nancy, Lorraine, pp: 55-62.

Blondel, V.D., A. Decuyper and G. Krings, 2015. A survey of results on mobile phone datasets analysis. EPJ Data Sci., Vol. 4. 10.1140/epjds/s13688-015-0046-0

Esmaili, M., B. Balachandran, R. Safavi-Naini and J. Pieprzyk, 1996. Case-based reasoning for intrusion detection. Proceedings of the 12th Annual Computer Security Applications Conference, December 9-13, 1996, San Diego, CA., USA., pp: 214-223.

F-Secure, 2014. Mobile threat report. https://www.f-secure.com/documents/996508/1030743/Mobile_Threat_Report_Q1_2014.pdf.

Fanoiki, T.O., I. Drummond and S. Sandri, 2010. Case-based reasoning retrieval and reuse using case resemblance hypergraphs. Proceedings of the IEEE International Conference on Fuzzy Systems, July 18-23, 2010, Barcelona, Spain, pp: 1-7.

Hu, X., 2011. Large-scale malware analysis, detection and signature generation. Ph.D. Thesis, The University of Michigan.

Hu, X., T. Chiueh and K.G. Shin, 2009. Large-scale malware indexing using function-call graphs. Proceedings of the 16th ACM conference on Computer and Communications Security, November 9-13, 2009, Chicago, IL., USA., pp: 611-620.

Micarelli, A. and G. Sansonetti, 2007. A case-based approach to anomaly intrusion detection. Proceedings of the 5th International Conference on Machine Learning and Data Mining in Pattern Recognition, July 18-20, 2007, Leipzig, Germany, pp: 434-448.

Park, Y., D. Reeves, V. Mulukutla and B. Sundaravel, 2010. Fast malware classification by automated behavioral graph matching. Proceedings of the 6th Annual Workshop on Cyber Security and Information Intelligence Research, April 21-23, 2010, Oak Ridge, TN., USA., pp: 1-4.

Sanz, B., I. Santos, X. Ugarte-Pedrero, C. Laorden, J. Nieves and P.G. Bringas, 2014. Anomaly detection using string analysis for android malware detection. Proceedings of the International Joint Conference SOCO'13-CISIS'13-ICEUTE'13, September 11-13, 2013, Salamanca, Spain, pp: 469-478.

Saudi, M.M., F. Ridzuan, N. Basir, N.F. Nabila, S.A. Pitchay and I.N. Ahmad, 2015. Android mobile malware surveillance exploitation via call logs: Proof of concept. Proceedings of the 17th UKSim-AMSS International Conference on Modelling and Simulation, March 25-27, 2015, Cambridge, UK., pp: 176-181.

Wang, Y., J. Zheng, C. Sun and S. Mukkamala, 2013. Quantitative security risk assessment of android permissions and applications. Proceedings of the IFIP 27th Annual Conference on Data and Applications Security and Privacy, July 15-17, 2013, Newark, NJ., USA., pp: 226-241.

Wang, X., Y. Yang and Y. Zeng, 2015. Accurate mobile malware detection and classification in the cloud. SpringerPlus, Vol. 4. 10.1186/s40064-015-1356-1.

Wei, X., L. Gomez, I. Neamtiu and M. Faloutsos, 2012. ProfileDroid: multi-layer profiling of android applications. Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, August 22-26, 2012, Istanbul, Turkey, pp: 137-148.

Wu, D.J., C.H. Mao, T.E. Wei, H.M. Lee and K.P. Wu, 2012. DroidMat: Android malware detection through manifest and API calls tracing. Proceedings of the 7th Asia Joint Conference on Information Security, August 9-10, 2012, Tokyo, pp: 62-69.

Zakaria, W.Z.A., 2015. Application of case based reasoning in IT security incident response. Proceedings of the 3rd International Conference Recent treads in Engineering and Technology, September 2-3, 2015, Istanbul, Turkey -.

Zhao, M., F. Ge, T. Zhang and Z. Yuan, 2011. AntiMalDroid: An efficient SVM-based malware detection framework for Android. Proceedings of the 2nd International Conference on Information Computing and Applications, October 28-31, 2011, Qinhuangdao, China, pp: 158-166.

Zhou, Y. and X. Jiang, 2012. Dissecting android malware: Characterization and evolution. Proceedings of the IEEE Symposium on Security and Privacy, May 20-23, 2012, San Francisco, California, pp: 95-109.