Mobile Malware Classification based on Phylogenetics

Madihah Mohd Saudi, Sazali Sukardi, Amirul Syauqi Mohamad Syafiq, Azuan Ahmad, Muhammad 'Afif Husainiamer

Abstract*: Security researchers and practitioners face many challenges in mitigating mobile malware attacks against smartphones. Ranges of techniques have been developed by different developers to ensure that smartphones remain free from such attacks. However, we still lack efficient techniques to mitigate mobile malware attacks, especially for the iOS platform. Hence, this paper presents mobile malware classifications based on phylogenetics that can be used for mobile malware detection with regard to the iOS platform. Phylogenetics have been used as the basis concept associated with forming a mobile malware classification based on similar malware behavior, vulnerability exploitation and mobile phone surveillance features that originate from the same family of specific malware practices. A mobile malware classification based on the phylogenetic concept and on mathematical formulations has been developed for this purpose, and proof of the concept has been sought to support this new classification. This research was conducted in a controlled lab environment using open source tools and by applying dynamic analysis. Consequently, this paper can be used as reference for other researchers with the same interest in future.

Keywords: iOS, Malware Classification, Mathematical Formulation, Mobile Malware, Phylogenetic, Surveillance Feature, Vulnerability Exploitation.

I. INTRODUCTION

A report by McAffee in the first quarter of 2019, showed the increase in mobile malwares attacks against smartphones using hidden mobile applications (app) was the number one threat to users of these phones [1]. Malware such as Timp Door, Fortnite, Dress Code, and Milky Door are examples of Trojan malware with backdoor and camouflage capabilities. These malwares exploit and expose smartphone users to such security risks for examples identity theft, data loss, privacy invasion, fraud and ransomware. As for Timp Door, it exploits a victim's smartphone via SMS. It will ask the user to download an app from outside the genuine Google Play store. If the user does so, he can be exploited using SMS or also known as smishing. This has wreaked havoc across the globe by infecting more than 5,000 Android devices.

Revised Manuscript Received on September 22, 2019.

Madihah Mohd Saudi, Cybersecurity and System Research Unit, Islamic Science Institute, Universiti Sains Islam Malaysia, Nilai, Malaysia
Sazali Sukardi, CyberSecurity Malaysia, Cyberjaya, Malaysia
Amirul Syauqi Mohamad Syafiq, Faculty of Science and Technology, Universiti Sains Islam Malaysia, Nilai, Malaysia
Azuan Ahmad, Cybersecurity and System Research Unit, Islamic Science Institute, Universiti Sains Islam Malaysia, Nilai, Malaysia
Muhammad 'Afif Husainiamer, Faculty of Science and Technology, Universiti Sains Islam Malaysia, Nilai, Malaysia

Surprisingly, although malwares targeting Android peaked from 2013 to 2017, the incidence of malware exploitation is currently showing sign of slowing. This is in contrast to MacOS malware, which increased by 165% in 2018, with 73,024 phones being infected worldwide [2]. Consequently, Apple users should prioritize security and more aware of the threat posed by MacOS malware, since the Project Zero Team from Google has identified 14 forms of malware related to iOS [3]. In addition, with the rise of internet-enabled crime across the world, we need a solution that could effectively mitigate these cybercrimes triggered by malwares. Though there are a number of different techniques that are currently used to detect malwares such as those described by [4-6], they have tended to focus on Android, with fewer mentions of the iOS platform. Such works have tended to focus on app-level profiling, multi-modal deep learning feature extraction, permission, behavior features in network traffic, and deep learning of codes. Works by [7] and [8] have discussed analysing iOS malwares but only briefly. This indicates that more work is needed to support analysis and to suggest solutions for iOS platform protection. To fill this gap, we propose a mobile malware classification for the iOS environment.

Consequently, in this paper we present a mobile malware classification based on phylogenetic by using a mathematical formulation. Subsequently, this classification could be used as a basis for developing a mobile malware detection model.

This paper is organized as follows. Section II explains the methodology we used in this research. It consists of the formulation of a mobile malware classification based on phylogenetic and mathematical formulation. Section III presents our methodology, while Section IV consists of the findings and proof of the concept underpinning the proposed classification. Section V presents the findings and suggests future work based on this paper.

II. RELATED WORKS

A number of works such as [9-12] have used bio-inspired concepts to provide cybersecurity solutions. They show the significance of such concepts as the basis for developing new self-resilient algorithm. We summarize the comparison with other bio-inspired algorithms in Table 1. Based on our experiment and study, we chose phylogenetic as a basic concept in forming a mobile malware classification. There are a number of existing works related to malware phylogenetic, namely [13-20].

These works can be further improved by integrating them with mobile phone surveillance technology which includes SMS, call logs, cameras, audio and geolocation facilities (GPS) for better detection. We mapped the phylogenetic concept with the mobile malware classification as displayed in Table 2.

Table. 1 Comparison with Other Bio-Inspired Algorithms

Algorithm	Concept Description	Limitations
Genetic Algorithm (GA)	GA is a heuristic search method used in artificial intelligence and computing.	- GA finds it difficult to ensure constant optimization in response times. - GA applications in real time are limited because of random solutions and convergence
Fuzzy Logic (FL)	FL is a generalization of standard logic, in which a concept can possess a degree of truth.	- When the solution is not known, the problem cannot be solved using fuzzy logic - Fuzzy logic systems are expensive, and require extensive testing
Negative Selection Algorithm (NSA)	NSA is a non-self pattern which is used to build an anomaly detection system.	-The main barriers to the success of this algorithm as an effective detection system are scalability and coverage.
Danger Theory (DT)	This theory states that the immune system discriminates between dangerous and safe pathogens by recognition such pathogens or detecting alarm signals from injured or stressed cells and tissues.	- This is only works for self- & non-self systems, but has some complexity - Difficult to identify danger signals
Phylogenetic	This deals with evolutionary history and is related to different organisms and taxonomic groups by using a tree diagram.	Not applicable

Table 2. Mapping Phylogenetics to Mobile Malware Classifications

Phylogenetics	Mobile Malware
Based on evolutionary history and the relatedness of different organisms and taxonomic groups by using a tree diagram.	Evolution ancestry of malware genes based on the integration of malware behavior, vulnerability exploitation and mobile phone surveillance features. Behavior is referred to as infection, payload, operating algorithm, activation, propagation. Mobile phone surveillance features are SMS, call logs, cameras, audio and GPS. Vulnerability exploitation is based on iOS version and chain of exploitation. The three elements are mapped into a tree diagram.

Phylogenetic is a study aiming to discover the evolution ancestry of malware genes. An example of the phylogenetic diagram is depicted in Figure 1. It helps us to understand malware evolutionary trends, extracting the crucial element inside malware codes and strategizing the best detection mechanism.

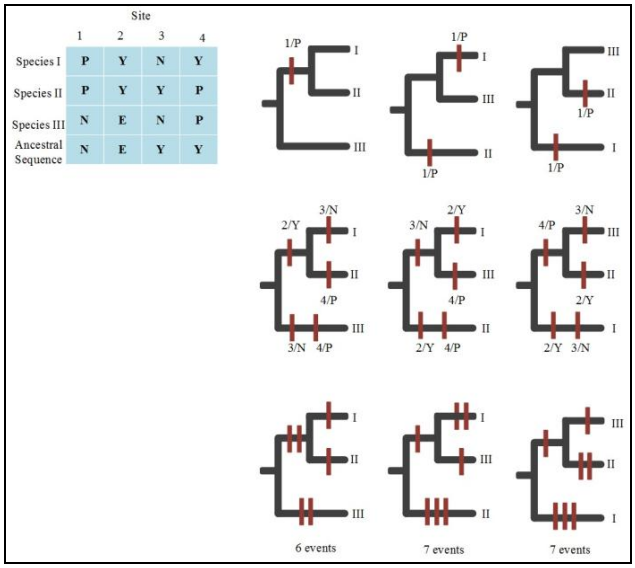

Fig. 1 Phylogenetic Diagram

III. METHODOLOGY

The overall processes involved in this experiment are summarized in Figure 2.

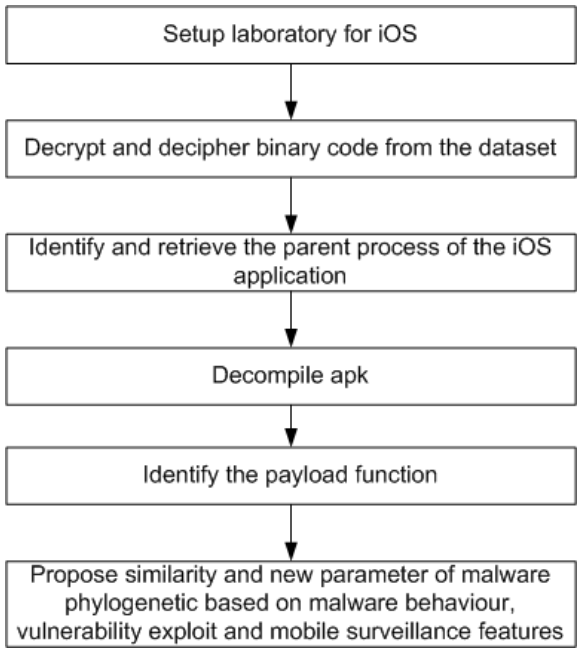

Fig. 2 Experimental processes

The experiment was conducted in a controlled lab environment as displayed in Figure 3. The software used for this experiment is displayed in Table 3.

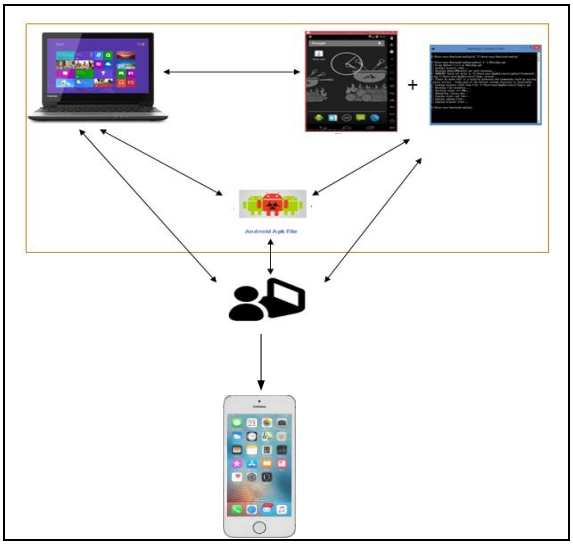

Fig. 3 Lab Setup

Table. 3 Tools for Experiment

Software name	Function
Notebook (i7,8GB RAM,1TB Hard disk)	Notebook to run the emulator
iPhone (Model 5s,version iOS 10.3.3)	To run the malware for analysis and testing purposes
VMware	As emulator to run MacOS operating system
Advanced Apk tool	To decompile/ recompile Android apk
Filezilla	To analyse the mobile apk

We used VMware as the emulator for MacOS. 90% of the software used are open source and we ran a dynamic analysis for this experiment. In contrast with static analysis, dynamic analysis is a procedure whereby the coding is executed in order to see its payload or implications. We chose dynamic analysis to ensure that the payload could be captured for further analysis. For this experiment, we ran a mobile malware called as ZergHelper in the emulator and in the iPhone to support our proof of concept for the proposed malware classification. Further analysis based on the phylogenetic concept was carried out once the ZergHelper had been executed. In the emulator, the ZergHelper was uploaded and recognized as EnglishStudy.app (see Figure 4). It camouflaged itself inside the software. The details of the findings are explained in Section IV.

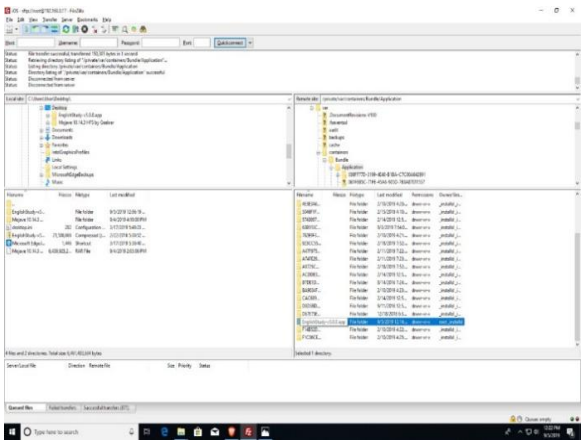

Fig. 4 Screenshot for uploaded dataset into the emulator

Once the malware had been executed, analysis took place. During the analysis, the findings were mapped in terms of the malware behavior, vulnerability exploitation and mobile phone surveillance features in order to allow malware classification formation. Malware behavior is referred to as infection, payload, operating algorithm, activation, and propagation. Vulnerability exploitation is based on the version of the iOS platform, either iOS 10.x, 11.x or 12.x and the type of exploitation used. At the same time, mobile phone surveillance features are those features that could be used by the attacker to exploit a mobile phone in the form of the call log, camera, audio and GPS.

The mathematical formula for mobile malware classification is as follows.

Let α_1 be a malware architecture I, and $\alpha = \bigcap_{i=1}^{p} \alpha_i, \beta_j$ be a mode attack j, and $\beta = \bigcup_{i=1}^{m} \beta_i$, γ_k be a connected assets in network k, and $\gamma = \bigcap_{i=1}^{p} \gamma_i$.

Let M be the malware detection and T be a target asset. S is the detection model which can be defined in terms of the following function:

$$(M,T)=S \quad (1)$$

where $M(\alpha, \beta, \delta) = \alpha + \beta + \delta$ (2)

$$f(M_i, T_j) = S_{ij} \quad (3)$$

where M represents the malware classification, T represent the target asset and S is the detection model.

$$M(\alpha, \beta, \delta) = \alpha + \beta + \delta$$
$$\alpha = \alpha 1 \cap \alpha 2 \cap \alpha 3 \cap \alpha 4 \cap \alpha 5$$
$$\beta = \beta 1 \cup \beta 2 \cup \beta 3 \cup \beta 4 \cup \beta 5$$
$$\delta = \delta 1 \cup \delta 2 \cup \delta 3 \cup \delta 4 \cup \delta 5$$
$$\begin{matrix} M_i & \alpha & \beta & \gamma \\ \vdots & & \ddots & \vdots \\ M_n & & \cdots & \delta_n \end{matrix}$$

where:

$\alpha 1 - \alpha 5$: payload, infection, operating algorithm, activation and propagation

$\beta 1 - \beta 5$: chain1, chain2, chain3, chain4 and chain5

$\delta 1 - \delta 5$: SMS, call log, gps, audio and camera.

The findings from the experiment are explained in the next section.

IV. FINDING

The experimental findings are as follows. In Apple iOS architecture, there are generally 4 different layers - the core OS, core services, media and cocoa touch. The lower layers in the iOS (core OS and core services) provide the basic services. The higher layers (media and cocoa touch), provide graphics and the user interface. The ZergHelper's payload as depicted in Figure 5, was executed and further analysed.

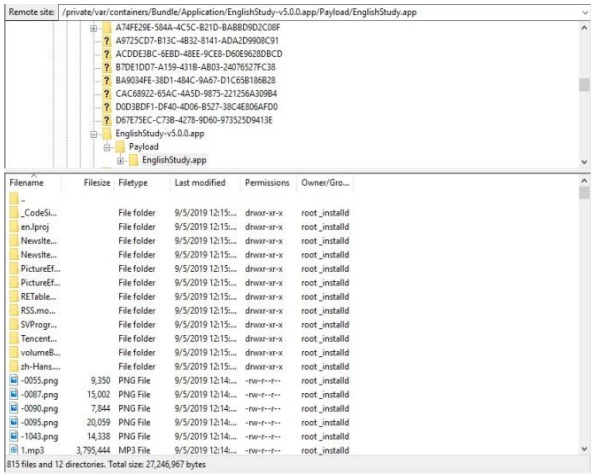

Fig. 5. Screenshot for Zerg Helper payload

The three elements proposed for the new malware classification - the integration of malware behavior, vulnerability exploitation and mobile phone surveillance features - are summarized in Table 4.

Table. 4 Findings for the Malware Classification

Mobile Malware Classification	**Description**
Mobile behavior ($\alpha 1 \cap \alpha 2 \alpha 3 \cap \alpha 4 \cap \alpha 5$)	*1) Infection*: Spread via host file (mobile app). It camouflaged itself in a genuine mobile app. *2) Activation*: Scheduled process. It is based on the location of the user and activates its payload only in China. *3) Payload*: Installed via a backdoor, requests user to give Apple ID, shared Apple ID to other users, abuses the Apple ID by running different operations in the background and abuses enterprise and personal certificates. *4) Operating algorithm*: Stealth. Runs in the background. *5) Propagation*: Passive monitoring. The malware camouflaged itself by claiming to resolve stability issues. It then guides the installation for two configurations. Only users in China will see the payload.
Vulnerability exploitation ($\beta 1$)	iOS version 10.x (Chain 1)
Surveillance features ($\delta 1$)	SMS

It could be concluded that Zerg Helper consists of:

$$M(\alpha, \beta, \delta) = \alpha + \beta + \delta$$

$$= (\alpha 1 \cap \alpha 2 \alpha 3 \cap \alpha 4 \cap \alpha 5) + (\beta 1) + (\delta 1)$$

Based on this mathematical formulation, Zerg Helper possesses all the elements necessary for malware classification. In addition, Zerg Helper, has elements for further possible exploitation. We can also conclude that in the case of iOS, especially for iPhone users, there are two elements that could lead to exploitation in the form of unpatched iOS with the latest patch, and the removal of security restrictions by users of the iPhone. Furthermore, based on the findings of this experiment, the proposed malware classification can identify all the elements for infecting an iPhone. Once all of these elements have been identified, it will be easier to implement a detection and removal plan.

V. CONCLUSION

Based on the experiment described in this paper, it is proven that the proposed malware classification works well with real malware on vulnerable iOS platform. Once the malware classification has been developed, it will be easier for solution providers to decide a more effective way forward to mitigate the dangers of malware. This work is part of a bigger project for mobile malware detection modeling for the iOS platform. In future, the research will be using more huge dataset for evaluation purposes.

ACKNOWLEDGMENT

The authors would like to express their gratitude to Ministry of Education (MOE),Malaysia, Universiti Sains Islam Malaysia (USIM) and Cyber Security Malaysia (CSM) for the support and facilities provided. This research paper project is under grant: [FRGS/1/2019/ICT04/USIM/02/3].

REFERENCES

1. Raj Samani, Gary Davis (2019). McAfee Mbile Threat Report Q1,2019 URL:https://www.mcafee.com/enterprise/en-us/assets/reports/rp-mobile-threat-report-2019.pdf
2. Adebayo, O.S. and Aziz, N.A., 2019. The trend of mobile malwares and effective detection techniques. In Multigenerational Online Behavior and Media Use: Concepts, Methodologies, Tools, and Applications (pp. 668-682). IGI Global. [last accessed: 9 Sept 2019].
3. Arslan, R.S., Doğru, İ.A. and Barişçi, N., 2019. Permission-Based Malware Detection System for Android Using Machine Learning Techniques. International Journal of Software Engineering and Knowledge Engineering, 29(01), pp.43-61.
4. Wang, S., Chen, Z., Yan, Q., Yang, B., Peng, L. and Jia, Z., 2019. A mobile malware detection method using behavior features in network traffic. Journal of Network and Computer Applications.
5. Milosevic, J., Malek, M. and Ferrante, A., 2019. Time, Accuracy and Power Consumption Tradeoff in Mobile Malware Detection Systems. Computers & Security..
6. Cai, H., Meng, N., Ryder, B. and Yao, D., 2019. Droidcat: Effective android malware detection and categorization via app-level profiling. IEEE Transactions on Information Forensics and Security, 14(6), pp.1455-1470.
7. Cimitile, A., Martinelli, F. and Mercaldo, F., 2017. Machine Learning Meets iOS Malware: Identifying Malicious Applications on Apple Environment. In ICISSP (pp. 487-492).

8. Pajouh, H.H., Dehghantanha, A., Khayami, R. et al. J Comput Virol Hack Tech (2018) 14: 213.
9. Demertzis, K. and Iliadis, L., 2016. Bio-inspired hybrid intelligent method for detecting android malware. In Knowledge, Information and Creativity Support Systems (pp. 289-304). Springer, Cham.
10. Firdaus, A., Anuar, N.B., Ab Razak, M.F. and Sangaiah, A.K., 2018. Bio-inspired computational paradigm for feature investigation and malware detection: interactive analytics. Multimedia Tools and Applications, 77(14), pp.17519-17555.
11. Demertzis, K. and Iliadis, L., 2016. Ladon: a cyber-threat bio-inspired intelligence management system. Journal of Applied Mathematics & Bioinformatics, 6(3), pp.45-64.
12. Gracanin, D., D'Amico, A., Manuel, M., Carson, W., Eltoweissy, M. and Cheng, L., 2018, May. Biologically Inspired Safety and Security for Smart Built Environments: Position Paper. In 2018 IEEE Security and Privacy Workshops (SPW)(pp. 293-298). IEEE.
13. Liu, J., Dai Xie, P., Liu, M.Z. and Wang, Y.J., 2018. Having an Insight into Malware Phylogeny: Building Persistent Phylogeny Tree of Families. IEICE TRANSACTIONS on Information and Systems, 101(4), pp.1199-1202.
14. Acampora, G., Bernardi, M.L., Cimitile, M., Tortora, G. and Vitiello, A., 2018, July. A Fuzzy Clustering-based Approach to study Malware Phylogeny. In 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE) (pp. 1-8). IEEE.
15. Bernardi, M.L., Cimitile, M., Distante, D., Martinelli, F. and Mercaldo, F., 2018. Dynamic malware detection and phylogeny analysis using process mining. International Journal of Information Security, pp.1-28.
16. Karim, M.E., Walenstein, A., Lakhotia, A. and Parida, L., 2005. Malware phylogeny generation using permutations of code. Journal in Computer Virology, 1(1-2), pp.13-23.
17. Hayes, M., Walenstein, A. and Lakhotia, A., 2009. Evaluation of malware phylogeny modelling systems using automated variant generation. Journal in Computer Virology, 5(4), p.335.
18. Anderson, B., Lane, T. and Hash, C., 2014, October. Malware phylogenetics based on the multiview graphical lasso. In International Symposium on Intelligent Data Analysis (pp. 1-12). Springer, Cham.
19. Oyen, D., Anderson, B. and Anderson-Cook, C., 2016, March. Bayesian networks with prior knowledge for malware phylogenetics. In Workshops at the Thirtieth AAAI Conference on Artificial Intelligence.
20. Catalin Cimpanu., 2019.Google finds malicious sites pushing iOS exploits for years, URL: https://www.zdnet.com/article/google-finds-malicious-sites-pushing-ios-exploits-for-years/ [last accessed: 9 Sept 2019].