Mobile Malware Classification for iOS Inspired by Phylogenetics

Muhammad 'Afif Husainiamer, Madihah Mohd Saudi*
Azuan Ahmad, Amirul Syauqi Mohamad Syafiq
Faculty of Science and Technology
Universiti Sains Islam Malaysia
Nilai, Malaysia

***Abstract*—Cyber-attacks such as ransomware, data breaches, and phishing triggered by malware, especially for iOS (iPhone operating system) platforms, are increasing. Yet not much works on malware detection for the iOS platform have been done compared to the Android platform. Hence, this paper presents an iOS malware classification inspired by phylogenetics. It consists of mobile behaviour, exploits, and surveillance features. The new iOS classification helps to identify, detect, and predict any new malware variants. The experiment was conducted by using hybrid analysis, with twelve (12) malwares datasets from the Contagio Mobile website. As a result, twenty-nine (29) new classifications have been developed. One hundred (100) anonymous mobile applications (50 from the Apple Store and 50 from iOS Ninja) have been used for evaluation. Based on the evaluation conducted, 13% of the mobile applications matched with the developed classifications. In the future, this work can be used as guidance for other researchers with the same interest.**

Keywords*—iOS; mobile malware; reverse engineering; exploitation; phylogenetic

I. Introduction

Currently, smartphones based on Android and iOS are commonly and widely used across the world. Yet, they also possess security concerns, especially security exploitation by malware such as ransomware and cryptojacking [1]. Unfortunately, the rapid increase of smartphone users contributed to mobile malware growth in the iOS environment. Malware is referred to as software that can infect devices, software, or networks with malicious attention without the owner's consent. It can harm the victim with malicious activities such as stealing confidential information, identity theft, and spying on the victim. There are different kinds of malware such as viruses, Trojan, spyware, worms, and ransomware. It will cause a lot of chaos when the malware has successfully penetrated the smartphone system.

Whenever new vulnerabilities are released, Apple will update or patch to fix the weaknesses. By keeping the patch up to date, Apple makes sure the devices are secure enough to use. The malware attacks are carried out by attacking the kernel, giving the attacker private APIs (Application Programming Interfaces) and permission, and eventually gaining confidential information about the user. Unfortunately, there is a growing number of malwares attacking iOS devices. For example, it uses private APIs to implement malicious intent and view and steal its data. Fig. 1 shows statistics on the detection of malware for iOS by Welivesecurity [2].

Compared with Android, iOS is considered more secure. For example, in the iOS platform, the hardware, software, and even their booting process are monitored and secured by Apple procedures [3]. This scenario has an impact where many attackers tend to focus on Android malware rather than on iOS. In addition, based on the Mc Afee Labs Threat report on June 2018 shows a drastic increase in malware growth, and there were almost 2.9 million samples recorded [4]. Furthermore, high-risk vulnerabilities were detected in 38 percent of iOS mobile apps in 2019 compared to 43 percent of Android mobile apps [5]. Indeed, 40 percent of iOS malware attacks in 2017 targeted banking services [6]. As in Q1 2020, new mobile malware cases have surged by 71 percent, and new iOS malware grew by over 50 percent [7]. Hence, this paper presents a new mobile malware classification for iOS inspired by phylogenetics to overcome the above challenges. Phylogenetics is a term borrowed from biology and has been mapped into the cybersecurity field. It can be used to detect and predict malicious activity. This approach consists of malware behaviour, vulnerability exploitation, and surveillance features [8].

The proposed malware classification developed in this paper can detect any malware attacks against possible social media and online banking exploitation. This new iOS classification aids in the detection, identification, and prediction of new malware variants.

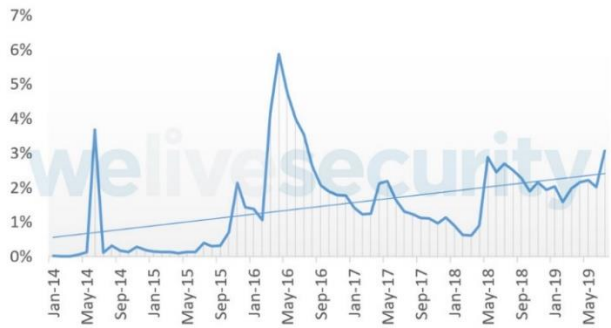

Fig. 1. Detection of Malware for iOS.

*Corresponding Author

This paper is organized as follows. Section II discusses the related works, while Section III presents the methods used, and Section IV explains the findings. Finally, Section V discusses the conclusions reached by this paper.

II. Related Work

A. iOS Malware Attacks

iOS malware attacks have been increased rapidly from years to years, and many researchers try to invade the issues and solve them to reduce the impact. Attackers evolve with the latest technology to ensure their intention to exploit user data can be conducted smoothly without interruption. Work by [9] found Trident worm and exploited three types of vulnerabilities once the link is clicked. Once it has been executed, the attackers will have the privilege to read, write, and any software in the infected device. Next, work by [10] found iKee Worm, which gathered logs on many jailbroken devices by scanning the OpenSSH port and used the root account and default password, and once infected, it will scan the surrounding IP address to spread the worm. They also found the YiSpecter worm, where the malware used ISP (Internet Service Provider) traffic, Window SNS (Self/Non-self) worm and offline applications installation, and other routes for transmission. It installed malware applications intending to collect private user information. Then, a previous study by [11] used Xcode Ghost worm where malware was sitting in the background of legitimate apps, then it did the data mining and injected malware in the apps when compiled. It possesses a new capability to prompt a fake alert dialogue to phish user credentials, hijack opening specifics URLs, and read and write data in the user clipboard. Work by [12] found AceDeciever that infects any Apple device connected to infected PC (personal computer) were capable of obtaining Apple ID (identification) and password. Finally, work by [13] found Keyraider worm, where it intercepted iTunes traffic and stole user login credentials, GUID (Globally Unique Identifier) devices Apple requests push service certificates and private keys, and iTunes receipts for purchase. It then sends this data to a remote server. Based on these previous studies, it can be concluded that there is a growing number of malwares attacking iOS users, and a solution to overcome these challenges is urgently needed. So, this paper proposes a new mobile malware classification as one of the mitigation solutions for the above challenges.

B. Phylogenetics

The phylogenetics aims to discover the origin of malware genes evolving [8]. It deals with evolutionary history and uses a tree diagram for different organisms and taxonomic groups. Malware phylogenetics emphasizes the similarities and relationships between a set of malwares. For example, a few types of phylogenetics tree models are the minimum spanning tree (MST), the persistent phylogeny tree, and the dendrogram [14]. Works by [15] used process mining which detects temporal logic properties designed to detect Android malware families and track the phylogenetic tree. [16] also used process mining where the program calls trace from a mobile application to classify associations and repeat execution patterns. Work by [17] used fuzzy clustering algorithm, where a malware program's syscalls can be modelled to produce a malware fingerprint with a number of associations and recurrent execution patterns. The author in [18] used discrete time Markov chain (DTMC) due to the paired KLD (Kullback-Leibler Divergence) and JSD (Jensen-Shannon Divergence) track calculation, it is computationally intensive. Bayesian network algorithm used by [19], learn a Directed Acyclic Graph (DAG) from observational data using statistical inference of conditional dependence and an informative antecedent to partial variable ordering. Work by [20] used extension of graphical lasso to discover a precision sparse matrix based on the kernel's combined matrix. An example of the phylogenetics diagram is depicted in Fig. 2. In this paper, there are three features mapped into phylogenetics to develop the classification. The identified features are malware behaviour, iOS version, and surveillance features.

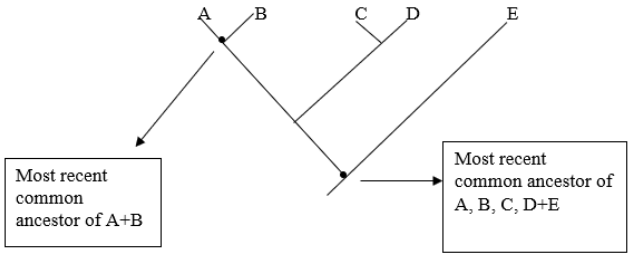

Fig. 2. Phylogenetic Diagram.

C. Features Mapped to Phylogenetic

Malware behaviour can be classified into five parts: infection, activation, payload, operating algorithm, and mitigation [8]. In dynamic analysis, these five components are significant to classify malware based on their behaviour. The malware behaviour used EDOWA (Efficient Detection of Worm Attack) worm classification as the underlying concept [21]. Apple keeps satisfying their customer by serving them with the best version of iOS. The version must be updated to make sure the user is secured enough from any current security issues. The update also has some new features that can help user's life easier [22]. Surveillance features used in this research come from 5 basic functions in the smartphone, consisting of call, SMS (Short Message Service), photos, audio, and GPS (Global Positioning System). All these features are dangerous whenever been exploited. The attacker can profit by exploiting either one of its features [23].

As one of the mitigating options for the stated challenges above, this research presents a new mobile malware classification based on phylogenetics. Three features are mapped into phylogenetics, which are malware behaviour, iOS version, and surveillance features.

III. Methodology

The overall process involved in this experiment is summarized in Fig. 3.

The analysis took place once the malware had been executed. The findings were mapped during the research regarding malware behaviour, vulnerability exploitation, and mobile phone surveillance features to allow malware classification. Malware behavior is referred to as infection,

payload, operating algorithm, activation, and propagation. Vulnerability exploitation refers to the iOS platform version, either iOS 10.x, 11.x, or 12.x, and the type of exploitation used. At the same time, mobile phone surveillance features are the features that attackers could use to exploit a mobile phone in the form of SMS,call log, camera, audio, and GPS. The mathematical formula for the proposed mobile malware classification is as follows:

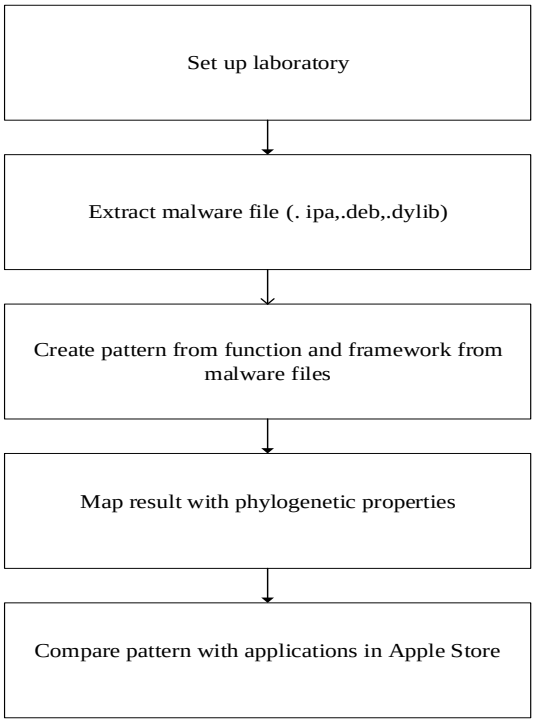

Fig. 3. Experimental Process.

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 asset 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:

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

$$\text{where } M(\alpha, \beta, \delta) = \alpha + \beta + \delta \quad (2)$$

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

Where M represents the malware classification, T represents 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

$\beta 1 - \beta 5$: iOS 10.x, iOS 11.x, iOS 12.x, iOS 13.x, iOS 1x.x

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

IV. Results

After the exploitation has been discovered, all the exploitation script's functions will be traced to their main frameworks to see what framework they are attacking during the malicious act. Then, the exploit is mapped into phylogenetics. The mapping result showed either the malware might lead to possible social media or online banking exploitation. If SMS or call is being exploited, it can be concluded as online banking exploitation. If any of those five features are being exploited, it is social media exploitation. Table I shows malware analysis results based on the mapping with the phylogenetics.

TABLE I. Malwares Mapped To Phylogenetic

Classification	**Description**	**Social Media Exploitation**	**Online Banking Exploitation**
E1: Unflod Malware			
α_1. α_5 (Malware Behaviour)	**Payload:** Phishing. Stole the device's apple id, password and sent them in plaintext to the server. **Infection:** Host. From Chinese Cydia repositories **Operating Algorithm:** Stealth. **Activation:** Self-activation. Initiated their execution by exploring vulnerabilities in services that are available. **Propagation:** Passive monitoring. The malware infected those jailbroken devices that have download piracy Chinese repositories.	Possible exploitation	Possible exploitation
β_1 - β_5 (iOS Chain)	iOS version: Chain 1 (iOS 10.x and below)		
δ_1 - δ_5 (Surveillance Features)	GPS, SMS, call, audio, and photo		

E2: Spad Malware			
α_1- α_5 (Malware Behaviour)	**Payload:** Destructive. Change the actual owner id for the ads into their id to obtain the revenue. **Infection:** Host. Result of some action taken by a user. The criminal hooks the legitimate functions and adds their tweaks. **Operating Algorithm:** Stealth. **Activation:** Human trigger. The malware will be functioning whenever the ads operate. **Propagation:** Passive monitoring. The malware only can infect jailbroken devices.	No possible exploitation	No possible exploitation
β_1 - β_5 (iOS Chain)	iOS version: Chain 1(iOS 10.x and below)		
δ_1 - δ_5 (Surveillance Features)	No surveillance features are involved as the attacker only exploits affect ads section		
E3+E4+E5+E6: Inception Malware			
α_1- α_5 (Malware Behaviour)	**Payload:** Phishing. Collect device information and recording audio and send it to the c2(Command and Control) server. **Infection**: Host, Result of some action taken by a user. The malware will inject into the host whenever the user clicks a malicious link or download any doc files. **Operating Algorithm:** Stealth. **Activation:** Human activation. The malware will be executed when the file has been opened. **Propagation:** Passive monitoring. The malware only can infect those jailbroken devices that click the link or open the trojanized file.	Possible Exploitation	Possible Exploitation
β_1 - β_5 (iOS Chain)	iOS version: Chain 1(iOS 10.x and below)		
δ_1 - δ_5 (Surveillance Features)	GPS, SMS, Call Log, Audio, and Photos		
E7+E8: Xcodeghost Malware			
α_1- α_5 (Malware Behaviour)	**Payload:** Destructive. Modifies Xcode, infects and steals some device information, and then sends it to the c2 server. **Infection:** Host. Infected the apps that Xcodeghost produced. **Operating Algorithm:** Stealth. **Activation:** Human trigger. The malware will be functioning whenever the apps been open. **Propagation:** Passive monitoring. The malware only can infect the user of the apps that have been created using Xcodeghost	No possible Exploitation	No possible Exploitation
β_1 - β_5 (iOS Chain)	iOS version: Chain 1(iOS 10.x and below)		
δ_1 - δ_5 (Surveillance Features)	No surveillance features are involved.		
E9+E10+E11+E12: Wirelurker Malware			
α_1- α_5 (Malware Behaviour)	**Payload:** Phishing. Exfiltration of user data and exfiltration of application usage and device serial number information. **Infection:** Host. Infected through USB (Universal Serial Bus) when the device connects with the infected Mac. **Operating algorithm**: Stealth. **Activation:** Self-activation. It will start the malicious act once the malware has been injected into the device. **Propagation:** Passive monitoring. The malware only can infect users that connect their device with the infected Mac.	Possible exploitation	Possible exploitation
β_1 - β_5 (iOS Chain)	iOS version: Chain 1(iOS 10.x and below)		
δ_1 - δ_5 (Surveillance Features)	GPS, SMS, Photos, Audio, and Call All the surveillance features can be exploited by the attacker if they can brute force the victim's Apple ID as they already get the Apple ID.		
E13: Zerghelper malware			
α_1- α_5 (Malware Behaviour)	**Payload:** Installing backdoor. Installed via a backdoor, requests the 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.	Possible exploitation	Possible exploitation

	Infection: Host. Spread via host file (mobile app). It camouflaged itself in a genuine mobile app. **Operating algorithm:** Stealth. **Activation:** Scheduled process. It is based on the location of the user and activates its payload only in China. **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.		
β_1 - β_5 (iOS Chain)	iOS version: Chain 1(iOS 10.x and below)		
δ_1 - δ_5 (Surveillance Features)	SMS		
E14: Oneclickfraud malware			
α_1- α_5 (Malware Behaviour)	**Payload**: Destructive. Installing another app through OTA. **Infection:** Host. Distributed through an adult site. **Operating algorithm:** Stealth. **Activation:** Human trigger. The malware will be functioning whenever the play button been clicked. **Propagation**: Passive monitoring. The malware only can infect the visitor of the adult site	No possible exploitation	No possible exploitation
β_1 - β_5 (iOS Chain)	iOS version: Chain 1(iOS 10.x and below		
δ_1 - δ_5 (Surveillance Features)	No surveillance features involved		
E15: Xsser Malware			
α_1- α_5 (Malware Behaviour)	**Payload:** Phishing. Act as spyware and harvesting information from user device thus send it to c2 server. **Infection:** Host. Installed via a rogue repository on Cydia, the most popular third-party application store for jailbroken iPhones. **Operating algorithm**: Terminate and resident. When triggered, xRAT will clean out its installation directory before issuing a package manager command to uninstall itself. Additionally, the developers behind xRAT created an alert system, flagging the malware operator if any of the following antivirus applications are present on a compromised device. **Activation:** Self-activation. It will start the malicious act once the malware has been injected into the device. **Propagation:** Passive monitoring. The malware only can infect users that use the repositories.	Possible exploitation	Possible exploitation
β_1 - β_5 (iOS Chain)	iOS version: Chain 1(iOS 10.x and below)		
δ_1 - δ_5 (Surveillance Features)	GPS, SMS, and Photo		
E16+E17: Muda malware			
α_1- α_5 (Malware Behaviour)	**Payload:** Display advertisements over other apps or in the notification bar and ask users to download iOS apps it promoted. **Infection:** Host. Spreads via third-party Cydia sources in China and only affects jailbroken iOS devices. **Operating algorithm**: Stealth. **Activation:** Human trigger. The malware will be functioning whenever the apps are open. **Propagation:** Passive monitoring. The malware only can infect jailbroken devices that using third-party Cydia repositories.	No possible exploitation	No possible exploitation
β_1 - β_5 (iOS Chain)	iOS version: Chain 1(iOS 10.x and below)		
δ_1 - δ_5 (Surveillance Features)	No surveillance features are involved.		
E18+E19+E20: Tinyv malware			
α_1- α_5 (Malware Behaviour)	**Payload:** Destructive. It connects with its C2 server to get remote commands and install specified IPA file or DEB file(s) in the background, uninstalling specified IPA app or DEB package(s) in the background and changing the /etc/hosts file. **Infection:** Hosts. Repackaged into some pirated iOS apps for jailbroken devices.	No possible exploitation	No possible exploitation

	Operating algorithm: Stealth. **Activation:** Human trigger. The malware will be executed whenever the apps are being used. **Propagation:** Passive monitoring. The malware only can infect the user that installed the pirated iOS apps.		
β_1 - β_5 (iOS Chain)	iOS version: Chain 1(iOS 10.x and below		
δ_1 - δ_5 (Surveillance Features)	No surveillance features involved		
E21+E22+E23+E24+E25+E26+E27: Yispecter Malware			
α_1- α_5 (Malware Behaviour)	**Payload:** Destructive. Abusing enterprise certificates, installing malicious apps, uninstall apps, and self-monitoring and updating, collecting, and uploading device information, changing safari configurations, hijacking other apps execution, and pretending to be system apps. **Infection:** Host. Through hijacking of traffic from nationwide ISPs, an SNS worm on Windows, and an offline app installation and community promotion. **Operating algorithm:** Stealth. **Activation:** Self-activation. It will start the malicious act once the malware has infected the device. **Propagation:** Passive monitoring. The malware only can infect users that use the repositories.	No possible exploitation	No possible exploitation
β_1 - β_5 (iOS Chain)	iOS version: Chain 1(iOS 10.x and below)		
δ_1 - δ_5 (Surveillance Features)	No surveillance features are involved.		
E28+E29: Keyraider malware			
α_1- α_5 (Malware Behaviour)	**Payload:** Destructive. It is stealing Apple account data, certificates, and private keys. **Infection:** Host. Distributed through third-party Cydia repositories in China. **Operating algorithm:** Stealth. **Activation:** Human trigger. The malware will be functioning whenever the apps are open. **Propagation:** Passive monitoring. The malware only can infect jailbroken devices that using third-party Cydia repositories.	Possible exploitation	Possible exploitation
β_1 - β_5 (iOS Chain)	iOS version: Chain 1(iOS 10.x and below)		
δ_1 - δ_5 (Surveillance Features)	GPS, SMS, Call Log, Audio, and Photos		

Table I shows 13 of the 29 malware classifications mapped to phylogenetics, which can be used against possible exploitation for social media and online banking. The identified classification are E1 (Unflod malware), E3+E4+E5+E6 (Inception malware), E9+E10+E11+E12 (Wirelurker malware), E13 (Zerghelper malware), E15 (Xsser malware and E28+E29 (Keyraider malware).

In summary, the malwares have been reported based on the classification proposed. The analysis fits with the elements required for the classification based on this classification. Furthermore, every malware examined contains components that can be used for further exploitation.

Next, for the evaluation process, 50 anonymous apps from Apple Store and another 50 from the third-party store were selected. This is to test the practicality of the proposed classification in detecting any possible exploitation in the tested apps. As a result, 13% of the tested apps were identified with possible security exploitation. Two apps were from Apple Store, and 11 apps were from iOS Ninja. This 13% represents the possible exploitation either against social media or online banking.

V. Conclusion

Based on these experimental results, the proposed malware classification developed in this paper can be used to detect any malware attacks against possible exploitation for social media and online banking. These new iOS classification helps to identify, detect, and predict any new malware variants. Another important consideration for future improvement would be to revise the frameworks and functions involved in the iOS architecture from time to time and integrate with artificial intelligence-based alarms. Malicious apps will use the combinations of frameworks and functions in the iOS architecture to exploit the targeted feature successfully. Furthermore, as a newer iOS version will be introduced, a new framework and functions may also be offered. Hence, there is a need to add more malware classifications based on the mobile malware classification formulation developed in this paper.

Acknowledgment

The authors would like to express their gratitude to the Universiti Sains Islam Malaysia (USIM) for the support and facilities provided. This research paper project is under the grant: [P1-17-16120-UNI-CVD-FST].

REFERENCES

[1] S. Garg and N. Baliyan, "Comparative analysis of Android and iOS from security viewpoint," Comput. Sci. Rev., vol. 40, p. 100372, May 2021, doi: 10.1016/j.cosrev.2021.100372.

[2] D. G. Bilić, "Semi-annual balance of mobile security 2019 | WeLiveSecurity," welivesecurity, 2019. https://www.welivesecurity.com/2019/09/05/balance-mobile-security-2019/ (accessed Jun. 13, 2021).

[3] M. Reddy Gangula, "Overcoming Forensic Implications with Enhancing Security in iOS," 2019. Accessed: Jun. 13, 2021. [Online]. Available: https://repository.stcloudstate.edu/msia_etds/77.

[4] McAfee, "Gold Dragon Expands the Reach of Olympics Attacks Lazarus Rises Again, Targeting Cryptocurrency Users Advanced Data-Stealing Implants GhostSecret and Bankshot Have Global Reach and Implications REPORT 2 McAfee Labs Threats Report," 2018.

[5] "Vulnerabilities and threats in mobile applications," 2019.

[6] "Mobile Cyberattacks Impact Every Businesss," Apr. 2017.

[7] "McAfee Labs COVID-19 Threats Report," 2020.

[8] M. M. Saudi, S. Sukardi, A. S. M. Syafiq, A. Ahmad, and M. 'Afif Husainiamer, "Mobile Malware Classification based on Phylogenetics," Int. J. Eng. Adv. Technol., vol. 9, no. 1, pp. 3661–3665, 2019, doi: 10.35940/ijeat.A2710.109119.

[9] Bill Marczak and John Scott-Railton, "The Million Dollar Dissident: NSO Group's iPhone Zero-Days used against a UAE Human Rights Defender - The Citizen Lab," 2016. Accessed: Jun. 13, 2021. [Online]. Available: https://citizenlab.ca/2016/08/million-dollar-dissident-iphone-zero-day-nso-group-uae/.

[10] [Z. Yixiang and Z. Kang, "Review of iOS Malware Analysis," in Proceedings - 2017 IEEE 2nd International Conference on Data Science in Cyberspace, DSC 2017, Aug. 2017, pp. 511–515, doi: 10.1109/DSC.2017.104.

[11] A. Vakil, "Smartphone and Security Issues," Int. J. Business, Humanit. Technol., vol. 9, no. 3, pp. 11–17, Sep. 2019, doi: 10.30845/ijbht.v9n3p2.

[12] P. Lewandowski, A. Felkner, and M. Janiszewski, "Security analysis for authentication and authorisation in mobile phone," PRZEGLĄD ELEKTROTECHNICZNY, Aug. 2019, doi: 10.15199/48.2019.08.29.

[13] B. Amro, "Malware Detection Techniques for Mobile Devices," Int. J. Mob. Netw. Commun. Telemat., vol. 7, no. 4/5/6, pp. 01–10, 2017, doi: 10.5121/ijmnct.2017.7601.

[14] J. Liu, P. D. Xie, M. Z. Liu, and Y. J. Wang, "Having an insight into malware phylogeny: Building persistent phylogeny tree of families," IEICE Trans. Inf. Syst., vol. E101D, no. 4, pp. 1199–1202, 2018, doi: 10.1587/transinf.2017EDL8172.

[15] M. G. C. A. Cimino, N. De Francesco, F. Mercaldo, A. Santone, and G. Vaglini, "Model checking for malicious family detection and phylogenetic analysis in mobile environment," Comput. Secur., vol. 90, p. 101691, 2020.

[16] M. L. Bernardi, M. Cimitile, D. Distante, F. Martinelli, and F. Mercaldo, "Dynamic malware detection and phylogeny analysis using process mining," Int. J. Inf. Secur., vol. 18, no. 3, pp. 257–284, 2019.

[17] G. Acampora, M. L. Bernardi, M. Cimitile, G. Tortora, and A. Vitiello, "A fuzzy clustering-based approach to study malware phylogeny," IEEE Int. Conf. Fuzzy Syst., vol. 2018-July, pp. 1–8, 2018.

[18] K. Ghosh, J. Mills, and J. Dorr, "Phylogenetic-inspired probabilistic model abstraction in detection of malware families," AAAI Fall Symp. - Tech. Rep., vol. FS-17-01-, pp. 200–205, 2017.

[19] D. Oyen, B. Anderson, and C. Anderson-Cook, "Bayesian networks with prior knowledge for malware phylogenetics," AAAI Work. - Tech. Rep., vol. WS-16-01-, pp. 185–192, 2016.

[20] B. Anderson, T. Lane, and C. Hash, "Malware phylogenetics based on the multiview graphical lasso," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 8819, pp. 1–12, 2014.

[21] M. M. Saudi, M. Tamil, S. Aishah, M. Nor, and K. Seman, "EDOWA Worm Classification," Jul. 2008.

[22] "Update your iPhone, iPad, or iPod touch - Apple Support," 2021. https://support.apple.com/en-us/HT204204 (accessed Jun. 13, 2021).

[23] M. M. Saudi, M. Z. A. Rahman, A. A. Mahmud, N. Basir, and Y. S. Yusoff, "A New System Call Classification for Android Mobile Malware Surveillance Exploitation via SMS Message," Adv. Comput. Commun. Eng. Technol., vol. 362, pp. 103–112, 2016, doi: 10.1007/978-3-319-24584-3