A New Mobile Malware Classification for Camera Exploitation based on System Call and Permission

Madihah Mohd Saudi, Luqman Hakim Zahari, Farida Ridzuan, Nurlida Basir, Sakinah Ali Pitchay, N.F. Nabila

***Abstract*— Currently, there are many attacks and exploitation to Android smartphones by the attackers all over the world. These attacks are based on profit and caused loss of money and productivity to the victim. This exploitation can be done via camera, SMS, call, audio, image or location exploitation by attacking the system call, permission or API inside the Android smartphone. Therefore, this paper presents 32 mobile malware classification based on system call and permission to detect camera exploitation for Android smartphone. The experiment was conducted in a controlled lab environment, by applying reverse engineering with 5560 training dataset from Drebin, where both static and dynamic analyses were used to identify and extract the permission and system call from the mobile applications (apps). These 32 classification have been evaluated with 500 mobile apps from Google Play Store and 19 mobile apps matched with the classification. This new classification can be used as the database and input for the development of new mobile malware detection model for camera exploitation.**

***Index Terms*— Android, camera exploitation, mobile malware, permission, system call.**

I. Introduction

Android is an operating system that acts as platform between user and his smartphone. Due to its technology, it has been widely used by many users all over the world. Unfortunately, many users out there have lack of security awareness about malicious mobile application and mobile malware implication and how to detect it. Mobile malware is defined as malicious software that attacks smartphone systems without user's consent. There are so many techniques have been used by the cyber-criminals to attack Android smartphone. In early year 2017, the cybercriminals have used social chat mobile app known as WhatsApp to access personal data on the phone including banking credentials and pin codes [1]. This can be done by sending embedded file with virus in word, excel or PDF file as the attachment in WhatsApp message. Even worst, the cybercriminals use social engineering technique to convince the victim to open the file attachment by including name of trusted and major organizations in their country in the message. While on April 2017, Eset security researcher, Lukas Stefanko has revealed his finding on an app named as Flashlight LED Widget [2]. This app has Trojan.Android/Charger.B embedded in it. It works as a normal flashlight but the truth is, it has much other hidden functionality once it is executed. It has command and control (C&C) capability where it will able to control victim's smartphone remotely. It is known as Trojan.Android/Charger.B, where it can display fake screen that mimics exactly like a legitimate app. Even worst, it can also lock the infected smartphone and bypass two-factor authentication by intercepting SMS and display fake notification. This app was uploaded to Google Play on March 30, 2017 and up to 5,000 users have downloaded it. It has been taken out from the Google Play on April 10, 2017. This Trojan has evolved from Android/Charger Trojan, which was first discovered on January 2017 [3]. In contrast with Android/Charger, where it has the capability to lock the smartphone and ask as for ransom, the Trojan.Android/Charger.B is more sophisticated type of banking malware.

There are lots of mobile malwares out there ready to attack end user's smartphone. Yet other examples are Trojan-Ransom.AndroidOS.Pletor.d and Trojan-Banker.AndroidOS.Gugi. As for Trojan-Banker.AndroidOS.Gugi, this Trojan is able to bypass Android's permission by integrating social engineering to trick end user. No hard-core coding or vulnerability exploitation is needed to infect the victim. For the past few years, cybercriminals trends have changed. In earlier year, it is more for fun and recognition. Currently, the mobile malwares exploit the Android smartphone to gain super-user right or also known as root exploitation with the aim to steal money and confidential information from the victim. It spreads via Google Play store or third party untrusted source, mimics legitimate apps, turns as mobile banking malware and using advanced threat persistent to defeat security features in Android smartphone [4]. Hence, a good solution is needed to overcome all these issues and threats.

Therefore based on the mobile malwares threats, issues and challenges, this paper objective is to develop a new mobile malware classification for camera exploitation of Android smartphone based on system call and permission. This new classification is useful as a database and input to

Manuscript received July 17, 2017; revised Aug 8, 2017. This work was funded by Ministry of Higher Education (Malaysia), FRGS grant: [USIM/FRGS/FST/32/50114].

Madihah Mohd Saudi is a senior IAENG member and works as an associate professor with the Information Security and Assurance (ISA) programme, Faculty of Science and Technology (FST), Universiti Sains Islam Malaysia (USIM), 71800 Nilai, Malaysia. (E-mail: madihah@usim.edu.my).

Luqman Hakim Zahari graduated from Information Security and Assurance (ISA) programme, Faculty of Science & Technology (FST), Universiti Sains Islam Malaysia (USIM), 71800 Nilai, Malaysia.

Farida Ridzuan, Nurlida Basir, Sakinah Ali Pitchay, N.F.Nabila are senior lecturer with the Information Security and Assurance (ISA) programme, Faculty of Science and Technology (FST), Universiti Sains Islam Malaysia (USIM), 71800 Nilai, Malaysia.

ISBN: 978-988-14047-5-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

detect mobile malware that exploits camera for cybercriminal activities. There are 32 classification have been developed and these have been tested by using 500 mobile apps that have been chosen randomly from Google play store. Surprisingly there are 3.8% of the mobile apps matched with the proposed classification.

This paper is organized as follows. Section 2 presents related work on mobile malware detection techniques for camera exploitation. Section 3 describes the methodology used in this paper. Section 4 presents the experimental result and Section 5 concludes this paper and discusses the future work.

II. RELATED WORK

There are few works done related with mobile malware and camera exploitation. Based on work done by Zhou and Jiang, camera permission is ranked as the 12th most requested in mobile apps and ranked as top 20 in malware [5]. While Wu et al. discussed about the attacks that are based on the use of phone camera [6]. They proposed a detection scheme by integrating computer vision technique where it is based on activity pattern. A very promising solution but it would be a better detection mechanism if other features such as permission and system call are taken into consideration. While Pore and Bartere, presented a review on camera-based attacks for android smartphones [7]. Xu et al. presented a Trojan that is able to exploit video camera in Android smartphones by integrating computer vision technique and remote-controlled and real-time monitoring attacks [8]. Their work mechanism is similar as done by [6]. As for Maggi et al., they implemented shoulder surfing technique to steal whatever written by the victim in his smartphone by using camera [9].

While Raguram et al. [10], applied the same concept as work done by [8]. Both of these works have challenges on how to place the camera near the victim without noticing it. As for Kundu et al., they proposed energy attacks to Android phone and camera is one of the elements that can be used [11]. It is based on hardware feature and a good reference and input to build a camera exploitation detection mechanism based on the model proposed. As for Kynigos et al., they developed a malware that exploits camera in Android smartphone without being detected by digital forensics software [12]. The images were captured and then being transferred to secondary cloud location without user's consent and knowledge. It is a good empirical analysis and same like work done by [11], it is very useful as a reference and input to build a camera exploitation detection mechanism. Furthermore, a comprehensive review by Wolfe on the vulnerabilities and countermeasures for smartphones has been discussed [13]. Same like [13], Deshpande and Dharmadhikari, made a comprehensive review on camera attacks and defenses mechanism for Android smartphone from year 2008 to 2015[14]. As for Chouhan et al., they activated the camera and captured the cybercriminal image once the Android smartphone was stolen [15]. This technique is very useful as the detection and response mechanism for a stolen phone.

All of above discussed existing works have their own strengths and challenges. Therefore, to overcome these challenges this paper has developed mobile malware classification based on the combination of system call and permission, which is yet not being discussed thoroughly in any of previous works.

III. METHODOLOGY

The overall processes involved in this research are summarized and displayed in Fig. 1. These processes are very time-consuming and need great analytical skill to extract the system call and the permission. The system call and permission were extracted by applying reverse engineering technique and the controlled lab architecture for this experiment as displayed in Fig. 2. It is called as controlled lab architecture, as no outgoing network connection is allowed from this lab. More than 80% software installed and used in this lab are based on open source and free. In this research, hybrid analysis which consists of both static and dynamic analyses was applied. Static analysis is a process where the source code of the mobile app being decompiled and reverse engineered without executing it. In contrast with dynamic analysis, the mobile app is executed and the behavior and payload are being monitored. There are few existing works that used static analysis such as by [16, 17] and dynamic analysis such by [18, 19, 20] as their method for malware analysis. Hybrid analysis is seen as comprehensive way of doing the analysis for our research. The permissions from the mobile apps were extracted and reverse engineered by using the static analysis (refer Fig. 3), while the system calls extraction by using dynamic analysis (refer Fig. 4). Total of 5560 Drebin dataset were used as the training dataset and 500 random anonymous mobile apps from different categories were selected for testing.

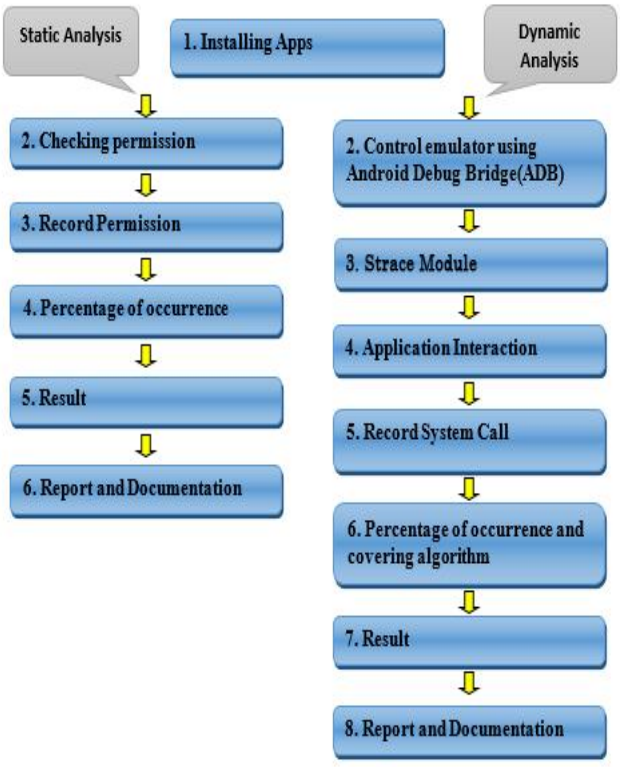

Fig.1. Overall Research Processes.

ISBN: 978-988-14047-5-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

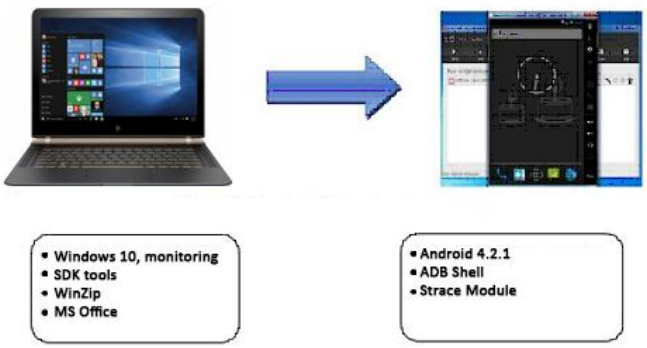

Fig. 2. Mobile Malware Lab Architecture.

Once the system calls and permissions were extracted, the frequency or also known as the percentage of occurrence being applied (refer Fig. 1). Total number for each permission and system call was being calculated.

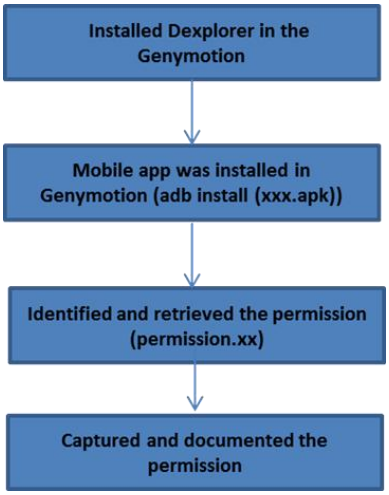

Fig.3. Static Analysis Work Flow.

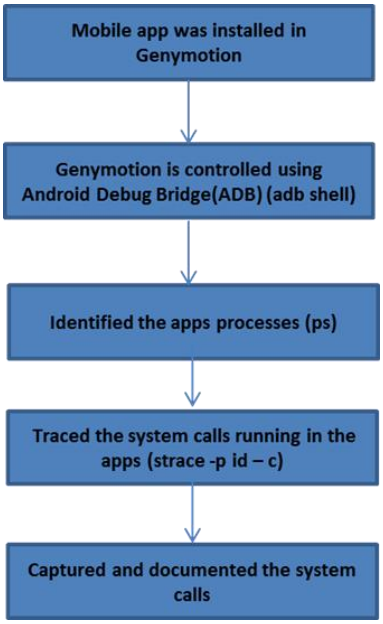

Fig.4. Dynamic Analysis Work Flow.

Then covering algorithm is used to generate system call and permission classification for each app. The covering algorithm generates rule by concentrating on a specific class and maximizing the probability of the desired classification. It is based on if-then rule. This is summarized as the following:

- The classification rule,

$$R = <P, C> \quad (1)$$

consists of:
P: precondition a series of tests that be valuated as true or false.
C: conclusion the class or the classes that apply to instances covered by rule R.

- Step1: Generate R on training dataset.
- Step 2: Remove the training data covered by R
- Step 3: Repeat the process step1 and step 2.

In each stage, covering approach identifies rule that cover some of the training dataset. The classification of the system call and permission were developed based on this method. As for the evaluation, it was carried out by testing the proposed classification with the real mobile apps from the Google Play store. 500 random mobile apps were downloaded to the emulator platform and analyzed by using the static and dynamic analyses. Permission and system call for each of app was captured and tabulated. The results were then being analyzed and compared with the new proposed classification. The percentage of similarity was taken as the result of the experiment.

IV. RESULTS AND DISCUSSION

Based on the experiment conducted, 58 system calls and 40 permissions have been extracted from 5560 dataset. These system calls and permissions work together for possible camera exploitation in the Android smartphone. The extracted and the nominal representation of the 58 system calls can be referred in Table I. While the extracted and the nominal representation of the 40 permissions can be referred in Table II. The data representation is important so later the formation of the classification can be done systematically by using the covering algorithm. Once this data representation is completed, the classification has been developed based on the combination of the extracted system calls and permissions. As a result, 32 new classification for mobile malware attacks based on system calls and permissions have been developed as summarized in Table III.

ISBN: 978-988-14047-5-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE I.
DATA REPRESENTATION FOR THE EXTRACTED SYSTEM CALLS.

System call name	Nominal data representation	System call name	Nominal data representation
clock_gettime()	sc1	socket()	sc30
epoll_wait()	sc2	bind()	sc31
recvfrom()	sc3	getsockname()	sc32
sendto()	sc4	unlinkat()	sc33
futex()	sc5	madvise()	sc34
gettimeofday()	sc6	pwrite64()	sc35
writev()	sc7	setsockopt()	sc36
getuid32()	sc8	lseek()	sc37
read()	sc9	nanosleep()	sc38
ioctl()	sc10	getrlimit()	sc39
write()	sc11	brk()	sc40
close()	sc12	fchown32()	sc41
open()	sc13	getpid()	sc42
mmap2()	sc14	gettid()	sc43
mprotect()	sc15	lstat64()	sc44
dup()	sc16	recvmsg()	sc45
fcntl64()	sc17	recv()	sc46
epoll_ctl()	sc18	stat64()	sc47
munmap()	sc19	sigprocmask()	sc48
pread()	sc20	select()	sc49
sched_yield()	sc21	umask()	sc50
getsockopt()	sc22	getpaid()	sc51
clone()	sc23	pread64()	sc52
access()	sc24	rename()	sc53
fstat64()	sc25	fdatasync()	sc54
chmod()	sc26	mkdir()	sc55
fsync()	sc27	uname()	sc56
connect()	sc28	rt_sigreturn()	sc57
sendmsg()	sc29	_llseek()	sc58

TABLE II.
DATA REPRESENTATION FOR THE EXTRACTED PERMISSIONS.

Permission name	Nominal data representation	Permission name	Nominal data representation
Access_Course_Location	pm1	Install_Packages	pm21
Access_Fine_Location	pm2	Install_Shortcut	pm22
Acess_Location_Extra_Commands	pm3	Internet	pm23
		Kill_Background_Process	pm24
Access_Network_State	pm4		
Access_Wifi_State	pm5	Modify_Audio_Setting	pm25
Battery_Stat	pm6	Read_Calendar	pm26
Bluetooth	pm7	Read_Call_Log	pm27
Bluetooth_Admin	pm8	Read_Contact	pm28
Call_Phone	pm9	Read_External_Storage	pm29
Camera	pm10		
Change_Network_State	pm11	Read_Logs	pm30
Change_Wifi_Multicast_State	pm12	Read_Phone_State	pm31
		Read_Settings	pm32
Change_Wifi_State	pm13	Read_Sms	pm33
Clear_App_Cache	pm14	Receive_Boot_Complete	pm34
Control_Location_Updates	pm15		
Delete_Packages	pm16	Receive_Mms	pm35
Disable_Keyguard	pm17	Receive_Sms	pm36
Expand_Status_Bar	pm18	Record_Audio	pm37
Get_Accounts	pm19	Restart_Packages	pm38
Get_Tasks	pm20	Write_External_Storage	pm39
		Write_Settings	pm40

TABLE III. DATA REPRESENTATION FOR THE DEVELOPED CLASSIFICATION.

Mobile malware classification content	Nominal data representation
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26	CLASS1
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26+sc7	CLASS2
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26+sc7+sc28	CLASS3
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26+sc7+sc28+sc30	CLASS4
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26+sc7+sc28+sc30+sc31	CLASS5
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26+sc7+sc28+sc30+sc31+sc46	CLASS6
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26+sc7+sc28+sc30+sc46	CLASS7
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26+sc7+sc28+sc31	CLASS8
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26+sc7+sc28+sc31+sc46	CLASS9
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26+sc7+sc28+sc46	CLASS10
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26+sc7+sc30	CLASS11
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26+sc7+sc30+sc31	CLASS12
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26+sc7+sc30+sc31+sc46	CLASS13
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26+sc7+sc30+sc46	CLASS14
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26+sc7+sc31	CLASS15
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26+sc7+sc31+sc46	CLASS16
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26+sc7+sc46	CLASS17
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26+sc28	CLASS18
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26+sc28+sc30	CLASS19
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26+sc28+sc30+sc31	CLASS20
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26+sc28+sc30+sc31+sc46	CLASS21
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26+sc28+sc30+sc46	CLASS22
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26+sc28+sc31	CLASS23
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26+sc28+sc31+sc46	CLASS24
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26+sc28+sc46	CLASS25
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26+sc30	CLASS26
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26+sc30+sc31	CLASS27
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26+sc30+sc31+sc46	CLASS28
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26+sc30+sc46	CLASS29
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26+sc31	CLASS30
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26+sc31+sc46	CLASS31
pm10+pm19+pm29+pm31+pm39+sc2+sc9+sc10+sc24+sc26+sc46	CLASS32

Though there are 40 permissions that have been extracted, there are 5 permissions that are most related with the camera exploitation. These are pm10, pm29, pm39, pm19 and pm 31 as displayed in Table II. So these 5 permissions have been used as the attributes for the developed classification in Table III. As for system call, same like permission the 5

ISBN: 978-988-14047-5-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

most related with camera exploitation have been identified which are sc2, sc9, sc10, sc24 and sc 26. Based on our analysis, apart from these 5 most related system call, there are another 5 system calls though the percentage of occurrence were very low in the testing dataset. But still they are important to complement and to make the camera exploitation successful. The system calls are sc7, sc28, sc30, sc31and sc46. All the representation can be referred in Table I. For the new classification proposed as displayed in Table III, the combination of above 15 attributes have been used as the basis.

Then, these 32 possible classification for camera exploitation have been tested with 500 mobile apps that are randomly picked from different categories in the Google play store. The results showed that 105 of the mobile apps matched with our 32 classification. The results can be referred in Table IV. Though 105 apps matched, still we rerun the evaluation on real time to consolidate and to make sure only genuine apps that exploit the camera with malicious intention were being selected. As a result, only 19 genuine apps have been identified that are capable to exploit camera for malicious purposes. This can be referred in Table V.

TABLE IV. EVALUATION RESULT.

Classification	Percentage	Classification	Percentage
			2.80%
CLASS1	3.00%	CLASS17	0.40%
CLASS2	2.80%	CLASS18	0.40%
CLASS3	0.40%	CLASS19	0.20%
CLASS4	0.40%	CLASS20	0.20%
CLASS5	0.20%	CLASS21	0.40%
CLASS6	0.20%	CLASS22	0.20%
CLASS7	0.40%	CLASS23	0.20%
CLASS8	0.20%	CLASS24	0.40%
CLASS9	0.20%	CLASS25	0.60%
CLASS10	0.40%	CLASS26	0.20%
CLASS11	0.60%	CLASS27	0.20%
CLASS12	0.20%	CLASS28	0.60%
CLASS13	0.20%	CLASS29	0.20%
CLASS14	0.60%	CLASS30	0.20%
CLASS15	0.20%	CLASS31	2.80%
CLASS16	0.20%	CLASS32	

TABLE V. 19 APPS WITH CAMERA EXPLOIT FEATURES.

Mobile Malware App No	Classification No	Type of Malware Mobile App
M1	CLASS1, CLASS2, CLASS17, CLASS32	Game
M2	CLASS1, CLASS2, CLASS17, CLASS32	Game
M3	CLASS1	Browser
M4	CLASS1	Game
M5	CLASS1, CLASS2, CLASS17, CLASS32	Music
M6	CLASS1, CLASS2, CLASS17, CLASS32	Communication
M7	CLASS1, CLASS2, CLASS17, CLASS32	Browser
M8	CLASS1, CLASS2, CLASS17, CLASS32	Game
M9	CLASS1, CLASS2, CLASS3, CLASS4, CLASS5, CLASS6, CLASS7, CLASS8, CLASS9, CLASS10, CLASS11, CLASS12, CLASS13, CLASS14, CLASS15, CLASS16, CLASS17, CLASS18, CLASS19, CLASS20, CLASS21, CLASS22, CLASS23, CLASS24, CLASS25, CLASS26, CLASS27, CLASS28, CLASS29, CLASS30, CLASS31, CLASS32	Wallpaper
M10	CLASS1, CLASS2, CLASS3, CLASS4, CLASS9, CLASS10, CLASS11, CLASS14, CLASS17, CLASS18, CLASS19, CLASS22, CLASS25, CLASS26, CLASS29, CLASS32	Wallpaper
M11	CLASS1, CLASS2, CLASS11, CLASS14, CLASS17, CLASS26, CLASS29, CLASS32	Game
M12	CLASS1, CLASS2, CLASS17, CLASS32	Game
M13	CLASS1, CLASS2, CLASS17, CLASS32	Antivirus
M14	CLASS1, CLASS2, CLASS17, CLASS32	Communication
M15	CLASS1, CLASS2, CLASS17, CLASS32	Travel
M16	CLASS1	Communication
M17	CLASS1, CLASS2, CLASS17, CLASS32	Game
M18	CLASS1	Monitoring Tool
M19	CLASS1	Graphic

Based on the experimental results conducted, it can be concluded that cybercriminal can exploit system calls and permissions for camera functionality in Android smartphone without the owner's consent. Therefore, a good solution to detect the malicious intention is needed to overcome such challenge. Furthermore, based on the current trend where social engineering technique has been integrated in the malicious mobile apps by the cybercriminal, therefore user should be educated on how to detect, respond and prevent from malicious mobile app infection.

IV. CONCLUSION

In this paper 32 new classification to detect mobile malware attacks via camera exploitation based on system calls and permissions have been presented. Cybercriminal can easily exploit system calls and permissions for camera functionality without users knowing it at their Android smartphones. Furthermore, based on the evaluation result it can be concluded that users have to be very careful in installing any third party apps. Apart from the technology and technical aspect that have been discussed in this paper, human factor which involves social engineering technique should not be ignored. Therefore for future work, a good solution that combines technical aspect and human aspect must be taken into account. Furthermore, automatic detection of malicious system calls and permissions is recommended to ease the malware detection job. Lastly, the finding in this paper can be used as guidance and input for the formation of mobile malware detection for camera exploitation.

ACKNOWLEDGMENT

The authors would like to express their gratitude to Ministry of Higher Education (MOHE), Malaysia and Universiti Sains Islam Malaysia (USIM) for the support and facilities provided.

REFERENCES

[1] Smith,C., Hackers have a new way to steal your banking login using WhatsApp. BGR Media, LLC, http://bgr.com/2017/01/02/android-malware-whatsapp-threat/, last accessed 2017/8/8.

[2] Stefanko, L., Turn the light on and give me your passwords!. Eset, https://www.welivesecurity.com/2017/04/19/turn-light-give-passwords/, last accessed 2017/8/8.

[3] Koriat,O. and Polkovnichenko, A., Charger Malware Calls and Raises the Risk on Google Play. Check Point Software Technologies Ltd, http://blog.checkpoint.com/2017/01/24/charger-malware/, last accessed 2017/8/8.

[4] Kaspersky Lab, Mobile Malware Evolution 2016, https://securelist.com/files/2017/02/Mobile_report_2016.pdf, last accessed 2017/8/8.

[5] Y. Zhou and X. Jiang, Dissecting Android Malware: Characterization and Evolution, IEEE Symp. Security and Privacy 2012, 2012, pp. 95–109.
[6] L. Wu, X. Du and X. Fu, Security threats to mobile multimedia applications: Camera-based attacks on mobile phones, in IEEE Communications Magazine, vol. 52, no. 3, pp. 80-87, March 2014.
[7] Bartere,M. and Pore,M.A, Preventions and Features of Camera Based Attacks on Smart Phones.International Journal of Emerging Trends & Technology in Computer Science, Vol.4, No4, 2015,pp. 9-12.
[8] Xu,N., Zhang,F., Luo,Y., Jia,Y., Jia,W., Xuan,D. and Teng,J. , Stealthy Video Capturer: A New Video Based Spyware in 3g Smartphones, Proc. 2nd ACM Conf. Wireless Network Security, 2009, pp. 69–78.
[9] F. Maggi, Gasparini,S., Boracchi,G., A Fast Eavesdropping Attack against Touchscreens, 7th Int'l. Conf.Info. Assurance and Security, 2011, pp. 320–25.
[10] R. Raguram,M.W.,Andrew, Goswami,D., Monrose, F. and Frahm,J.M., Ispy: Automatic Reconstruction of Typed Input from Compromising Reflections, Proc. 18th ACM Conf. Computer and Commun. Security, 2011, pp. 527–36.
[11] Kundu, A., Lin, Z. and Hammond, J., 2017. Energy Attacks on Mobile Devices. arXiv preprint arXiv:1704.04464.
[12] Kynigos, C., Glisson, W.B., Andel, T. and McDonald, T., 2016, January. Utilizing the Cloud to Store Hijacked Camera Images. In System Sciences (HICSS), 2016 49th Hawaii International Conference on (pp. 5497-5506). IEEE.
[13] Henry B Wolfe, The Insecurity of Mobile Phones, Proceedings of Informing Science & In-formation Technology Education Conference, 2010, 119-131.
[14] Deshpande, S. and Dharmadhikari, S.C., 2016. Analysis on Camera Attacks and their De-fenses on Android Smartphones. European Journal of Advances in Engineering and Tech-nology, 3(3), pp.26-29.
[15] Chouhan, J. , Singh, N. , Modi, P. , Jani, K. and Joshi, B. (2016) Camera and Voice Control Based Location Services and Information Security on Android. Journal of Information Security, 7, 195-205.
[16] D. Geneiatakis, I. N. Fovino, I. Kounelis, and P. Stirparo, A Permission verification ap-proach for android mobile applications, Comput. Secur., vol. 49, pp. 192–205, 2015.
[17] D. Arp, M. Spreitzenbarth, H. Malte, H. Gascon, and K. Rieck, Drebin: Effective and Ex-plainable Detection of Android Malware in Your Pocket, Symp. Netw. Distrib. Syst. Secur., no. February, pp. 23–26, 2014
[18] M. Dimjasevic, Atzeni,S., Ugrina,I. and Rakamaric,Z., Android Malware Detection Based on System Calls, Uucs, vol. 11, no. 1, pp. 209–216, 2015.
[19] Y. D. Lin, Y. C. Lai, C. H. Chen, and H. C. Tsai, Identifying android malicious repackaged applications by thread-grained system call sequences, Comput. Secur., vol. 39, pp. 340–350, 2013.
[20] A. Reina, a Fattori, and L. Cavallaro, A system call-centric analysis and stimulation tech-nique to automatically reconstruct android malware behaviors, ACM Eur. Work. Syst. Se-cur. (EuroSec), pp. 1–6, 2013.

ISBN: 978-988-14047-5-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)