World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:11, No:5, 2017

Malware Detection in Mobile Devices by Analyzing Sequences of System Calls

Jorge Maestre Vidal, Ana Lucila Sandoval Orozco, Luis Javier García Villalba

***Abstract*—**With the increase in popularity of mobile devices, new and varied forms of malware have emerged. Consequently, the organizations for cyberdefense have echoed the need to deploy more effective defensive schemes adapted to the challenges posed by these recent monitoring environments. In order to contribute to their development, this paper presents a malware detection strategy for mobile devices based on sequence alignment algorithms. Unlike the previous proposals, only the system calls performed during the startup of applications are studied. In this way, it is possible to efficiently study in depth, the sequences of system calls executed by the applications just downloaded from app stores, and initialize them in a secure and isolated environment. As demonstrated in the performed experimentation, most of the analyzed malicious activities were successfully identified in their boot processes.

***Keywords*—**Android, information security, intrusion detection systems, malware, mobile devices.

I. Introduction

DUE to the great capacities of connectivity, accessibility, and versatility of the current mobile technologies, in recent years there has been a significant growth in their popularity. As a result, every day more and more users rely on these technologies to carry out particular sensitive activities, such as e-commerce, asset sharing or management of personal information. As the European Network and Information Security Agency (ENISA) warned in its latest annual threat report, this makes the mobile devices a highly desired target for cybercriminals [1]. Within this problem, the importance of the migration of conventional attacks to the mobile infrastructure has to be underlined, hence being the adaptation of the classical malware to the new scenarios one of the most habitual practices. According to the European Police Office (Europol), behind this laborious task are hidden complex networks of organized crime [2]. From among their most habitual propagation strategies the use of the official application distribution markets predominates. In this case, criminals offer variations of originally legitimate products, but manipulated enough to accommodate the infection vector that allows the installation of the malicious contents. Given the ineffectiveness of the defensive methods offered by these markets against certain adversarial attacks, as well as the frequent overconfidence of their users (usually encouraged by lack of knowledge), criminals are able to spread specimens very quickly and indiscriminately, posing the need to deploy more accurate security elements.

With a view to contribute to the mitigation of these threats, this paper introduces a novel malware recognition system for mobile devices. Our main motivation is to prevent malicious third-party software from being installed on the protected devices. To this effect, the applications downloaded from official distribution media are analyzed and evaluated in a secure and isolated execution environment, prior to their deployment on the real system. The analysis tasks involve building sequences of system calls gathered from the application startup processes. These are compared to collections of legitimate samples by sequence alignment algorithms. In order to determine their degree of similarity, the Wilcoxon signed-rank test is applied. By using this method it is possible to recognize malware before it acts against the protected device. Both features have been demonstrated in the performed experimentation. It considered samples from the public domain collections Genome [4] and Drebin [5], where good precision when dealing with true attacks was shown, and a low false positive rate was observed.

The rest of the paper is structured as follows: Section II reviews the related works; Section III introduces the detection system; Section IV describes the experimentation and results; and finally, Section V presents the conclusions.

II. Background

Over the last decade several states of the art about security on mobile devices have been published, some of them focused on general purpose approaches [3], [6] and others dealing with more specific issues, where the Android operating system is the most frequent monitoring environment [7]. An aspect of particular relevance of the previous proposals is how the features to be analyzed by the intrusion detection systems are selected. As is indicated in [8], the features on metrics are crucial to ensure the success of the proposals. Because of this, it is comprehensible that these characteristics were considered as the main distinction element between intrusion detection systems for mobile devices, as it is illustrated in [9]. In accordance with this criterion, the proposals can be distinguished into four broad groups: analysis of static, dynamic or mixed features and metadata.

The analysis of static traits inspects applications before their execution, looking for malicious content. With this purpose, several sources of information, such as their binary code, privileges requested, demand of hardware resources or connectivity are analyzed. For example, in [10] these

Jorge Maestre Vidal, Ana Lucila Sandoval Orozco y Luis Javier García Villalba, Grupo de Análisis, Seguridad y Sistemas (GASS, http://gass.ucm.es), Departamento de Ingeniería del Software e Inteligencia Artificial (DISIA), Facultad de Informática, Despacho 431, Universidad Complutense de Madrid (UCM), Calle Profesor José García Santesmases, 9, Ciudad Universitaria, 28040 Madrid, España (e-mail: jmaestre@ucm.es, asandoval@fdi.ucm.es, javiergv@fdi.ucm.es).

characteristics are considered to explore different distribution markets in search of instances of a particular specimen. In [11] the source code of applications at the Android Package Management Service (PMS) is studied in pursue of malware with privilege escalation capabilities. In general, static analysis has the advantages of its simplicity and efficiency in the tasks involved in data extraction. However, it is a method susceptible to be deceived by code obfuscation techniques. Another important drawback is that due to its lack of capability when defining the behavior of applications at runtime, it often does not reach the expected accuracy. On the other hand, the dynamic malware recognition monitors the protected system behavior at runtime, and extracts multiple features from the activity of the installed applications. As shown in [9], the most frequent dynamic analysis strategy is the study of the system calls involved in the software execution. A typical example of this approach is Crowdroid [12], where the frequency of occurrence of the monitored system calls is considered for malicious content detection. Another interesting contribution is [13], where their relationship with the running thread scheduling system is analyzed. This group of strategies usually collects data from the executions in isolated and safe monitoring environments, which are commonly known as sandboxes [14]. In general terms, the analysis of dynamic features is highly accurate. But it requires the use of a significant amount of resources, situation that may render its deployment unfeasible, and failing that, may demand the availability of additional infrastructure.

The more heterogeneous and sophisticate monitoring environments, usually those with greater susceptibility of become jeopardized, often face the identification of intrusions through the combination of the previous both techniques, which is known as mixed feature analysis. An easy example of this approach is illustrated in [15], where the static analysis of the AndroidManifest file and the source code of the suspicious applications are carried out, at the same time as several dynamical features are studied, among them phone call logs and network traffic. These hybrid proposals offset the benefits and drawbacks of the combined methods. This implies giving up some of their advantages, in order to strengthen their weaknesses.

The last group of approaches is based on the analysis of metadata. It was defined in [9] as the information users see prior to the download and installation of the applications, such as their software description, their requested permissions, their rating and information regarding developers. An illustrative example of this strategy is observed in [16], where information from the distribution markets related to the permissions requested by the applications is considered. To this end, natural language processing methods are implemented, which study in detail the reasons that authors justify the authorization of each one of them. The main advantage of the malware recognition based on metadata analysis is that it allows harmful content to be identified before it is downloaded. However, its success depends on information heavily editable by attackers, a situation that facilitates their evasion and often entails the emission of classification errors.

III. Malware Detection System for Mobile Devices

The malware detection system performs the dynamic analysis of the behavior of monitored applications. To do this, the system calls executed during their boot process are extracted. Unlike the similar proposals contemplated in the bibliography, our approach takes into account the temporal relationships between these actions, and therefore, the order in which they are executed. This is illustrated in Fig. 1, where the suspicious applications arrive at the protected system by different distribution markets. They are executed in a safe and isolated region (sandbox); therefore they are not able to make changes in the victim device. on this secure environment, the system call sequences are captured and transmitted to the analysis module. The extracted information is preprocessed, sequenced and aligned with samples of legitimate actions. If their similarity score differs representatively from the set of scores calculated in previous studies of legitimate samples, alerts are issued. The following describes the main elements of this approach.

A. Monitoring and Preprocessing

In the monitoring stage, and within a sandbox, the malware detector captures system calls of the application boot processes. As in the previous works [12], the monitoring implements the diagnostic tool *strace*, which is present in most of the GNU/Linux systems, as is the case of the current Android versions. In order to extract all the activities of a particular program from its initialization, including those processes derived from it, the parent process (Zygote) is monitored. The observed actions are preprocessed, so that each type of system call is associated with a symbol. Because of this, given that the operating system offers a predefined system calls repertoire of length l, the alphabet that identifies the actions will have size l. Note that in order to bypass the intrinsic characteristics of the device, consecutive repetitions of actions are simplified into a single action. This also allows reducing the problems derived from errors in capture processes, frequent in the motorization of software downloaded from application stores.

B. Sequence Alignment

All the extracted sequences are transferred to a dedicated server, where they are analyzed by sequence alignment methods. The analysis of the monitored data is driven by sequence alignment processes, which compare the received sequences with a collection of samples of legitimate application executions. The more completeness this dataset presents, the greater precision the system offers. It is important to bear in mind that this approach does not cover the decision of when the reference set is enough representative to be accepted. This is a well-known machine learning issue out of the scope of this paper, but which could be studied in depth in future works. The alignment algorithm implemented is an adaptation of the global alignment method proposed by Needleman-Wunsch [17].

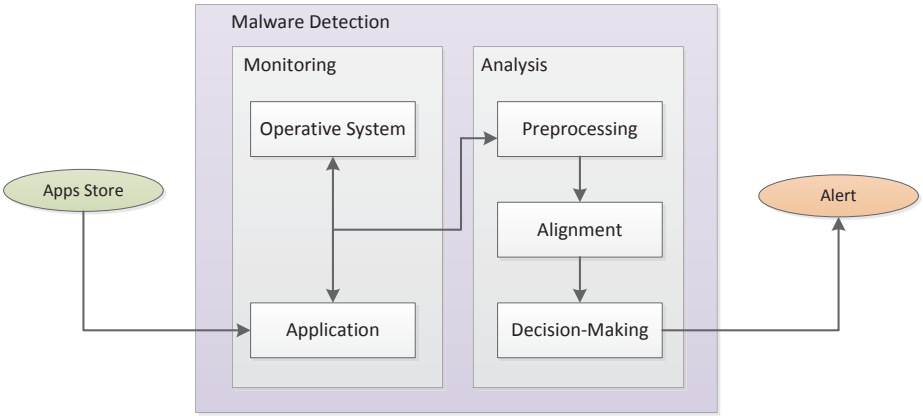

Fig. 1 Architecture for malware detection on mobile devices

C. Labeling and Decision Making

The labeling step is the vector of the scores resulted from aligning the monitored sequences of system calls, with the collection of legitimate samples. It also considers the vector scores obtained from aligning all legitimate sequences with each other. With this data, the non-parametric Wilcoxon signed-rank test [18] is applied. This evaluation operates on paired data vectors, so that the elements with the same index are compared. It assumes that there are l pairs of observations (x_i, y_i). The goal of the Wilcoxon signed-rank test is verify that every couple of values x_i and y_i are equivalent. In our approach, this data is provided by the previously calculated scoring vectors. The test is passed if $p < I$, where I, $0 \leq I \leq 1$ is a previously defined confidence interval (for example 0.1 or 0.05). This may be interpreted as the difference between scoring vectors is significant, and hence there is strong evidence that the monitored sequence does not match with the system call sequences of the legitimate collection of samples.

IV. Experiments

In the performed experimentation, our approach has been implemented on 30 different mobile devices with distinct versions of the Android operating system. However, relevant variations on their evaluation have been not found, so in this context it is possible to state that the nature of the Android kernel has not impact on the performance of the intrusion detection system. The evaluation methodology is based on the analysis of the behavior of our proposal when it acts on application samples provided by legitimate and malicious collections. The training samples were extracted from the public domain datasets Genome [4] and Drebin [5]. Note that despite the large dimension of these public collections, it has been especially difficult to find applications compatible with all the devices and their Android versions. Because of this, a cross-validation scheme was implemented, where 570 legitimate samples were divided into four groups of similar length. To heighten the realism, the false positive rate was calculated by inserting malware into the legitimate applications, hence allowing to consider of a total of 5130 malicious samples.

At implementation stage, the system has been calibrated to minimize the false positive rate; therefore, a confidence interval of 0.001 was applied. The best results were achieved considering samples of the first 2000 system calls gathered when initializing the applications within the sandbox. The true positive rate is 98.61% and the false positive rate is 6.88%. In view of these results it is possible to state that our approach behaves with enough precision to complement most of the detection methods on the bibliography, but with the added advantage of only require the study of the first monitored actions, hence improving their performance. This also demonstrates that selecting only the first actions performed is a good measure of computational resource optimization. On the other hand, Figs. 1 and 2 display the variation on the true positive rate and false positive rate when the length of the boot sequences is recalibrated. To simplify their visualization, the scores obtained have been grouped into three clusters: Max, Min and $Average$, where Max is the highest value obtained throughout the test, Min is the lower value and average is the arithmetic mean of the observations. As it is observed, the precision obtained grows as the length of the sequence increases. However, there is a point where its evolution ceases to be significant, which we have called saturation length. This is because in very short sequences, noise is smoothed worse. It is also the main reason of selecting sequences of at least 2,000 actions when calculating the final true positive rates and false positive rates.

V. Conclusion

A malware detection system for mobile devices has been presented. Their inputs are the sequences of system calls observed at the initialization of Android applications downloaded from the different distribution markets, and analyzed in a a safe and isolated environment. The performed experimentation demonstrates good accuracy, so it is possible to state that the proposed approach is an efficient complement for current intrusion detection systems. It is also important to highlight that throughout the paper have been proposed

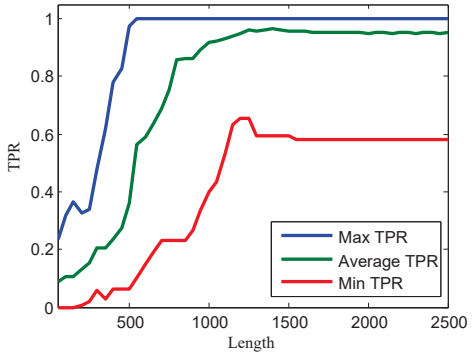

Fig. 2 True Positive Rate evolution

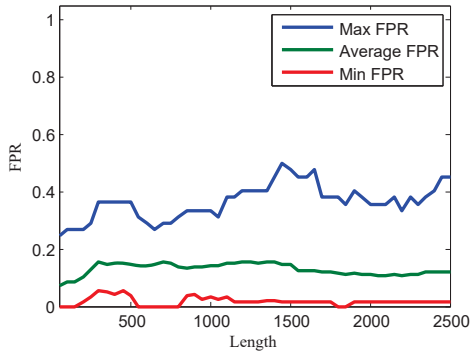

Fig. 3 False Positive Rate evolution

different lines of future work, among them the selection and definition methods for identifying when the reference datasets are enough representative, or how saturation lengths may be efficiently calculated.

Acknowledgment

This work was funded by the European Commission Horizon 2020 Programme under Grant Agreement number H2020-FCT-2015/700326-RAMSES (Internet Forensic Platform for Tracking the Money Flow of Financially-Motivated Malware).

References

[1] ENISA (2016), "Threat Landscape 2015". Available: https://www.enisa.europa.eu/
[2] European Police (2015), "The Internet Organised Crime Threat Assessment (iOCTA)". Available: https://www.europol.europa.eu
[3] G. Suarez-Tangil, J.E Tapiador, P. Peris-Lopez, A. Ribagorda, "Evolution, Detection and Analysis of Malware for Smart Devices", in *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 961-987, 2014.
[4] Y. Zhou, X. Jiang, "Dissecting Android Malware: Characterization and Evolution", in *Proceedings of the 33rd IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, US, 2012, pp. 95-109.
[5] D. Arp, M. Spreitzenbarth, M.H. Hubner, H. Gascon, K. Rieck, "Drebin: Effective and Explainable Detection of Android Malware in your Pocket", in *Proceedings of the 21th Annual Symposium on Network and Distributed System Security (NDSS)*, San Diego, CA, US, 2014, pp. 1-12.
[6] M. La Polla, F. Martinelli, D. Sgandurra, "A Survey on Security for Mobile Devices", *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 446-471, 2013.
[7] P. Faruki, A. Bharmal, V. Laxmi, "Android Security: A Survey of Issues, Malware Penetration, and Defenses", *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 998-1022, 2015.
[8] P. Garca-Teodoro, J. Daz-Verdejo, G. Maci-Fernndez, E. Vzquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges", *Computers & Security*, vol. 25, no. 1-2, pp. 18-28, 2009.
[9] A. Feizollah, N. B. Anuar, R. Salleh, A.W.A. Wahab, "A Review on Feature Selection in Mobile Malware Detection", *Digital Investigation*, vol. 13, pp. 23-37, 2015.
[10] M. Lindorfer, S. Volanis, A. Sisto, M. Neugschwandtner, E. Athanasopoulos, F. Maggi, C. Platzer, S. Zanero, S. Ioannidis, "AndRadar: Fast Discovery of Android Applications in Alternative Markets", in *Proceedings of the 11th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*, Egham, UK, 2014. *lecture Notes in Computer Science*, vol. 8550, pp. 51-71, 2014.
[11] L. Xing, X. Pan, R. Wang, K. Yuan, X. Wang, "Upgrading your android, elevating my malware: privilege escalation through mobile OS updating", in *Proceedings of the 35th IEEE Symposium on Security and Privacy*, San Jose, CA, US, 2014, pp. 393-408.
[12] I. Burguera, U. Zurutuza, S. Nadjm-Tehrani,"Crowdroid: Behavior-Based Malware Detection System for Android", in *Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, Chicago, IL, US, 2011, pp. 15-26.
[13] Y.D. Lin, Y.C. Lai, C.H. Chen, H.C. Tsai, "Identifying android malicious repackaged applications by thread-grained system call sequences", *Computers & Security*, vol. 39, pp. 340-350, 2013.
[14] Z.C. Schreuders, T. McGill, C. Payne, "The state of the art of application restrictions and sandboxes: A survey of application-oriented access controls and their shortfalls", *Computers & Security*, vol. 32, pp. 219-241, 2013.
[15] X. wei, L. Gomez, I. Neamtiu, M. Faloutsos, "ProfileDroid: multi-layer profiling of android applications", in *Proceedings of the 18th annual international conference on Mobile computing and networking (Mobicom)*, Istambul, Turkey, 2012, pp. 137-148.
[16] R. Pandita, X. Xiao, W. Yang, W. Enck, T. Xie, "WHYPER: Towards Automating Risk Assessment of Mobile Applications", in *Proceedings of the 22nd USENIX Conference on Security*, Washington, D.C, US, 2013, vol. 13, pp. 527-542.
[17] S. B. Needleman, C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins", *Journal of Molecular Biology*, vol. 48, no. 3, pp. 443-453, 1970.
[18] F. Wilcoxon, "Individual Comparisons by Ranking Methods", *Biometrics Bulletin*, pp. 80-83, 1945.

Jorge Maestre Vidal He received a Computer Science Engineering degree from the University Complutense of Madrid (UCM) in 2012 and a M.Sc. in Research in Computer Science in 2013. In 2016 he is Visiting Research at Instituto de Telecomunicaes (IT), Aveiro, Portugal. He is currently a Ph.D. student at the UCM and a member of the research group GASS (http://gass.ucm.es). His main research interests are Artificial Intelligence, Pattern Recognition and Information Security.

Ana Lucila Sandoval Orozco She received a Computer Science Engineering degree from the Universidad Autnoma del Caribe (Colombia) in 2001. She holds a Specialization Course in Computer Networks (2006) from the Universidad del Norte (Colombia), and a M.Sc. in Research in Computer Science (2009) and a Ph.D. in Computer Science (2014), both from the Universidad Complutense de Madrid (Spain). She is currently a Research Assistant at the research group GASS. Her main research interests are Computer Networks and Computer Security.

Luis Javier García Villalba He received a Telecommunication Engineering degree from the Universidad de Mlaga (Spain) in 1993 and holds a M.Sc. in Computer Networks (1996) and a Ph.D. in Computer Science (1999), both from the Universidad Politcnica de Madrid (Spain). Visiting Scholar at COSIC (Computer Security and Industrial Cryptography, Department of Electrical Engineering, Faculty of Engineering, Katholieke Universiteit Leuven, Belgium) in 2000 and Visiting Scientist at IBM Research Division (IBM Almaden Research Center, San Jose, CA, USA) in 2001 and 2002, he is currently Associate Professor of the Department of Software Engineering and Artificial Intelligence at the Universidad Complutense de Madrid (UCM) and Head of Complutense Research Group GASS (Group of Analysis, Security and Systems) which is located in the Faculty of Information Technology and Computer Science at the UCM Campus. His professional experience includes research projects with Hitachi, IBM, Nokia and Safelayer Secure Communications. His main research interests are Computer Networks and Computer Security.