VOL. 11, NO. 5, MARCH 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences


www.arpnjournals.com

AN EVALUATION OF N-GRAM SYSTEM CALL SEQUENCE IN MOBILE MALWARE DETECTION

M. Z. Mas'ud, S. Sahib, M. F. Abdollah, S. R. Selamat and R. Yusof
Faculty of Information Technology and Communication, Univeristi Teknikal Malaysia, Melaka,Ayer Keroh, Melaka, Malaysia
E-Mail: zaki.masud@utem.edu.my

ABSTRACT

The rapid growth of Android-based mobile devices technology in recent years has increased the proliferation of mobile devices throughout the community at large. The ability of Android mobile devices has become similar to its desktop environment; users can do more than just a phone call and short text messaging. These days, Android mobile devices are used for various applications such as web browsing, ubiquitous services, social networking, MMS and many more. However, the rapid growth of Android mobile devices technology has also triggered the malware author to start exploiting the vulnerabilities of the devices. Based on this reason, this paper explores mobile malware detection through an n-gram system call sequence which uses a sequence of system call invoked by the mobile application as the feature in classifying a benign and malicious mobile application. Several n-gram values are evaluated with Linear-SVM classifier to determine the best *n* system call sequence that produces the highest detection accuracy and highest True Positive Rate (TPR) with low False Positive Rate (FPR).

Keywords: mobile malware detection, n-gram, machine learning, linear SVM.

INTRODUCTION

Technological advancement had produced many mobile operating systems such as iOS from Apple, Blackberry, Symbian, Windows mobile and Android by Google. Of all the systems mentioned, the most popular platform is the Android system by Google as it has controlled over 80% of the overall mobile devices market sales in 2013 [1]. Despite the high market demand, Android-based mobile devices are also exposed to mobile malware threat. This is shown in the 2013 Kaspersky's Lab report which reveals almost 98 % of the mobile malware found in 2013 is targeting the Android platform [2]. A mobile device infected by malware can expose the user to information theft, activity and location sniffing, overbilling of sending random SMS and MMS to contacts, being exploited as denial of services attack source and can cause the mobile device resources such as memory, battery and storage overloaded by unknown processes [3].

To date, several mitigation processes to overcome the mobile malware infection have been introduced. For instance, software companies have introduced their mobile version of antivirus, yet they still detect malware using the signature approach and works as a cleaning up service after the mobile devices have been infected. Since the signature based mobile malware detection only detects known malware, a new malware on the market can easily evade this approach. Furthermore, solely depending on antivirus is not enough. Based on Zhou et al. [4], in the best case, only 79.6% antivirus can detect the mobile malware variant they have collected and even worse, some existing antivirus only detected 20.2% of the malware variant. Hence, there is a need for an effective approach in detecting mobile malware.

Another detection approach that can be used to mitigate mobile malware is the anomaly-based detection which has the ability to monitor regular activities in the devices and look for any behavior that deviates from the normal pattern. Anomaly-based detection is effective in detecting a known and unknown malware, yet it has a drawback of generating false alert which indicates an incorrectly classified benign application to be malicious or vice versa. This research also uses the anomaly-based detection approach through the machine learning classifying technique in revealing the benign and malicious method. On the contrary, this research attempts to improve the drawback of the anomaly-based detection by improving the detection TPR, FPR and accuracy using the n-gram system call sequence as a feature in classifying the mobile malware. The n-gram system call sequence is an *n* number of system call sequence invoked by the mobile application. The n-gram system call sequence can be used to represent a set of system call processes invoked by the malicious application. The basic concept of this approach is discussed in the next section.

RELATED WORK

Malicious software or called malware is written for the purpose of exploiting the weaknesses found in a computer system. The rapid changes and evolution of malware have made it difficult to stop any malware in any platform. The first step in mitigating the mobile malware is to understand how mobile malware behaves in mobile devices. This can be done by analyzing the malware sample itself. Malware analysis can enable researchers to observe and obtain the real behavior of a malware in action [5]. Based on the work done by [6] and [7] most Android-based malware has the following malicious intentions.

- Changes and accessing file system, e.g. creation, modification or deletion of files.
- Attempts on root access, e.g. creation or modification root files.

- Infection of running processes, e.g. to insert malicious code into other processes, or updating packages.
- Acquiring of sensitive data, e.g. IMEI, GEO Location or SMS and call log.
- Network activity and transfer, e.g. HTTP connections.
- Starting and stopping OS or application, e.g. restart OS.

The comparison between the malicious intention with the mobile malware behavior experiment and the observation made in the previous work [8] reveals that the behavior of mobile malware can be traced via the system call invoked by the application. Figure 1 shows a sample of the system call log file captured in the experiment run in [8].

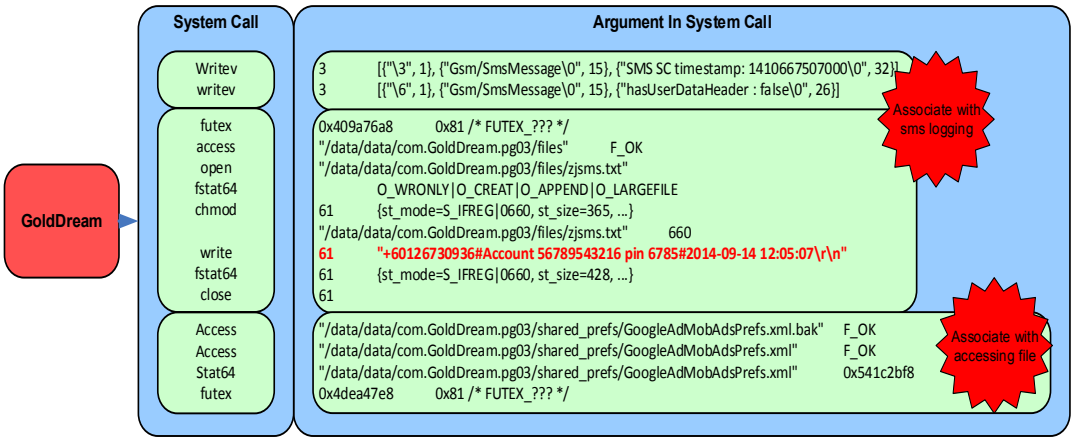

Figure-1. A GoldDream system call log for capturing and logging user SMS.

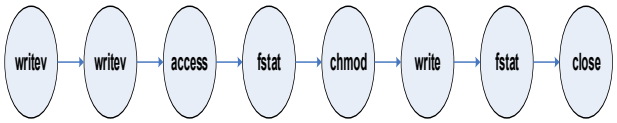

Figure-2. Sequence of system call used to capture and log sms received.

SYSTEM CALL AND N-GRAM ANALYSIS

A set of system call invoked by a mobile application through the kernel interface in an operating system is able to provide accurate information on the behavior of a mobile application. This is based on the reason that all requests made from the mobile application such as network communication, file management or process related operations have to pass through to the kernel using the system call interface before they are executed. This system has been used by these people in their work; Crowdroid [9], Isohara et al. [10], Azteni et al. [13], AMDA [11] and MADAM [12]. Crowdroid captures the whole system call invoked by application and use K-mean clustering algorithm to distinguish the benign and malicious application. Meanwhile, Isohara et al. applies a based signature approach in detecting mobile malware by filtering list of system call against a database of malicious system called regular expressions signature. System call dependency graph of an app execution and supervised learning machine learning is used by Azteni *et al.* to classify the application. MADAM and AMDA use machine learning classifier on a set of selected system call. Similarly, these researches used the anomaly-based detection approach and system call as features to distinguish a benign and malicious mobile application, yet our approach attempts to improve the detection rate accuracy and reduce the false alert using the n-gram system call sequence. The classification accuracy achieved by AMDA and Azteni et al. using their approach are 71.15% and 86.80%, respectively.

RESEARCH METHODOLOGY

This paper proposes an n-gram system call sequence as a feature to be used in machine learning classifying algorithm for classifying the benign and malicious android application. For the purpose of this research, the study has captured the system call invoked from 100 normal Android applications acquired from Google play and 102 infected Android applications acquired from the MalGenome Project [4]. Each Android application used in this experiment is scanned with these antiviruses, Bitdefender [20], eseT [21] and VirusTotal [22] for verification whether it is truly a malicious or benign application. Every Android application is executed on a tablet and stimulated with user interaction such as web browsing and SMS for 10 minutes. After each execution and simulation, the tablet is wiped out clean to its factory setting before another Android application is installed. This manual approach is used for evading any anti emulator or virtual machine evasion mechanism that might be included in the malicious application. The research methodology used in this research consists of four phase; data collection phase, n-gram extraction phase, machine learning classifiers phase and performance evaluation phase. The research methodology is depicted in Figure-3.

In data collection phase (Phase I), each application runs on a real device in an experimental test bed environment [23]. The system call generated by each of the application is captured in a log file using a tool called strace. Next, an extraction module is introduced in Phase II. The module starts by extracting all the system call sequence and

dropping all the parameter from the system call log. The sequence of system call is then encoded into a Unicode UTF-8 representation before it goes through the N-gram Generator. The encoding of system call will represent each system call with a unique character; hence it reduces the size of data stored for processing. Finally, the output of the N-gram Generator is a data set consists of n-gram system call sequence and its occurrence frequency, fngram. In order to evaluate the optimum n value, this research only generates 6 different dataset comprise of 1-gram, 2-gram, 3-gram, 4-gram, 5-gram and 6-gram system call sequence for evaluation. Each of n-gram system call sequence subset is then applied to the classifier in Phase III.

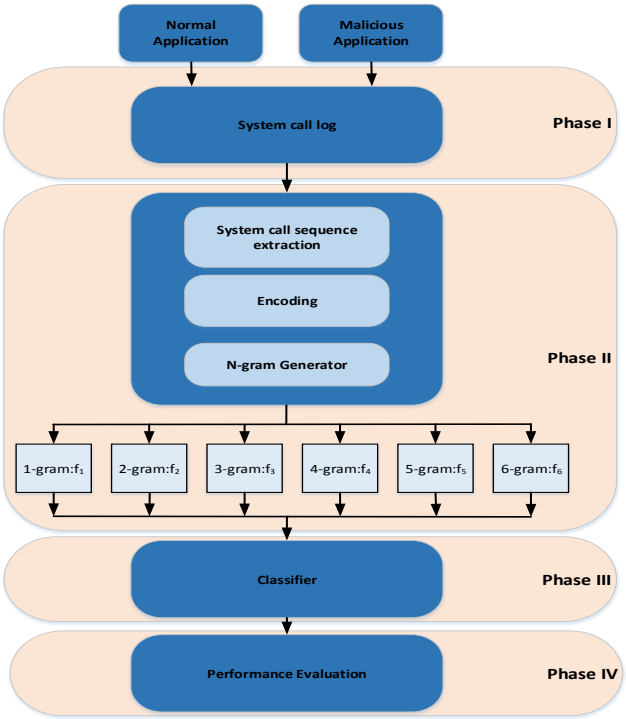

Figure-3. Research methodology.

The evaluation for finding the optimum n-value in the n-gram system call sequence for this research uses the linear-SVM classifier provided in the LibLinear Package introduced by [24]. The approach is chosen based on the fact that Linear SVM is suitable for machine learning applications that deal with large instances and features such as in text classification and bioinformatics [25] and with the number of features increased as the value of n-gram is increased, the Linear SVM classifier is the appropriate classifier algorithm for this evaluation. Given a data set of instance (xi, yi), i = 1,..., l, xi ∈ Rn, yi ∈ {−1, +1}, Linear SVM type L2-SVM can be used to solve the optimization problem for

$$\min_{\omega} \frac{1}{2}\omega^{T}\omega + C\sum_{i=1}^{l}\left(\max\left(0,1-y_i\omega^{T}x_i\right)\right)^2 \quad (1)$$

Where C > 0 is a penalty parameter and the loss functions is $\left(\max\left(0,1-y_i\omega^{T}x_i\right)\right)2$

Finally, Phase IV evaluates the performance of the classifier based on the True Positive Rate (TPR) that indicates the rate of correctly detecting an instance as malware, False Positive Rate (FPR) that indicates the rate of false detection of benign application as malware, and Accuracy is the percentage of correctly classified benign or malicious mobile application [26], [27], [28]. The best n value is chosen based on the number of system call sequence that can generate the highest detection Accuracy and TPR while the FPR is low. The equation of all matrices represented as below:

$$\text{True Positive Rate, } TPR = \frac{TP}{TP+FN} \quad (2)$$

$$\text{False Positive Rate, } FPR = \frac{FP}{FP+TN} \quad (3)$$

$$ACCURACY = \frac{TP+TN}{TP+FP+TN+FN} \quad (4)$$

Where
TP = number of malware cases correctly classified (true positives)
FN = number of malware cases misclassified as legitimate software (false negatives)
FP = number of benign software cases incorrectly detected as malware
TN = number of legitimate executables correctly classified.

ANALYSIS AND DISCUSSION

The Linear SVM applies in this experiment is the L2-SVM classifier from the Liblinear package and the experiment is performed using Weka 3.7.10 [26]. The experiment is done on Windows 7 that runs on a desktop computer with Pentium Dual Core CPU and 2GB of RAM. The classifier evaluation is validated using k-fold cross-validation which can estimate how well the learned model generalizes. For this implementation, the value of *k* is set to be 10. The 10 fold cross validation divided the data into 10 subsets and the holdout method is repeated for 10 times. In every fold, the subset is divided into 9 training sets and one testing set. The average value of the performance metric from each fold result is taken as a single estimation result for the overall implementation performance. The empirical results of the experiment are shown in Table-1.

Table-1. The classifier performance evaluation result.

N-gram	TPR (%)	FPR(%)	Accuracy (%)
1-gram	96.07	22.00	87.08
2-gram	96.07	9.00	93.51
3-gram	97.06	5.00	96.19
4-gram	100	15.00	92.33
5-gram	100	29.00	85.59
6-gram	100	42.00	79.16

Table-1 shows the TPR, FPR and the Accuracy of 6 different n-gram system call sequence and it apparently shows 3-gram system call sequence has the highest detection accuracy which is 96.19% and FPR which is 5%. Even though the TPR is only 97.06% and not the highest value, 3-gram system call sequence provide a classification between benign and malicious application with high detection accuracy and low FPR. Comparatively, the 3-gram system call sequence provides better classification accuracy with an acceptable TPR and the lowest FPR value compared to the 1 gram system call. Interestingly, when n-gram is higher than 3-gram, the accuracy is decreasing. This is due to the higher the number of n-gram sequence, the fewer occurrences of the system call sequence will be invoked hence, most of the system call sequence value will be 0. This causing the system call sequence for the higher n-gram generates a sparse vector, resulting in lower detection accuracy.

CONCLUSIONS

Mobile malware evolution was triggered due to the rapid development in mobile device technology. It has multiplied greatly in recent years and causing adverse effects towards mobile users. Thus, there is a need in finding more effective measure to handle this issue as this paper explores the used of n-gram system call sequence as a feature in classifying benign and malicious android application. The experiment and evaluation of the n-gram show that the n-gram can improve the accuracy for the classification made and the false alarm rate which solved the issues in anomaly-based malware detection. The 3-gram system call sequence returns the highest detection accuracy with a well-balanced TPR and FPR values even though it cannot achieve 100% for the TPR value compared to 4, 5 and 6 gram system call sequence. For future works, the proposed method may be potentially applied in developing Android malware detection. However, there is still an issue to be addressed especially in the limitation and constrain of mobile devices' environment especially on the storage and memory consumption such as the large number of features generated as the n-gram value increases.

ACKNOWLEDGEMENTS

The authors would like to express their gratitude to InsForsNet research group of Universiti Teknikal Malaysia Melaka (UTeM) for the invaluable support in encouraging the authors to publish this paper. This work is supported by UTeM's Short Grant funding (PJP/2013/FTMK(9D)/S01167).

REFERENCES

[1] R. van der Meulen and J. Rivera. 2013. Gartner Says Smartphone Sales Accounted for 55 Percent of Overall Mobile Phone Sales in Third Quarter of 2013, (Gartner Newsroom), [online] Retrieved on December 2012 from http://www.gartner.com/newsroom/id/2623415.

[2] C. Funk and M. Garnaeva. 2013. Kaspersky Security Bulletin 2013. Overall statistics for 2013", (Securelist), [online] Retrieved on December 2013 http://www.securelist.com/en/analysis/204792318/Kasper.sky_Security_Bulletin_2013_Overall_statistics_for_2013

[3] M. La Polla, F. Martinelli, D. Sgandurra. 2013. A Survey on Security for Mobile Devices. Communications Surveys & Tutorials, IEEE. 15(1): 446-471.

[4] Y. Zhou, J. Xuxian. 2012. Dissecting android malware: Characterization and evolution. In Security and Privacy (SP), 2012 IEEE Symposium on. 95-109.

[5] D. Farmer, W. Venema. 2005. Forensic discovery, Vol. 6, Upper Saddle River: Addison-Wesley.

[6] A. P. Felt, M. Finifter, E. Chin, S. Hanna, D. Wagner. 2011. A survey of mobile malware in the wild. In Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices. pp. 3-14.

[7] H. Pieterse, M. S. Olivier. 2012. Android botnets on the rise: Trends and characteristics. In Information Security for South Africa (ISSA). pp. 1-5.

[8] M. Z. Mas'ud, S. Sahib, M. F. Abdollah, S. R. Selamat, R. Yusof, R. Ahmad. 2013. Profiling mobile malware behaviour through hybrid malware analysis approach. Information Assurance and Security (IAS). 9th International Conference on. pp. 78-84.

[9] I. Burguera, U. Zurutuza, S. Nadjm-Tehrani. 2011. Crowdroid: behavior-based malware detection system for android. In: Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices. pp. 15-26.

[10] T. Isohara, K. Takemori, A. Kubota. 2011. Kernel-based Behavior Analysis for Android Malware Detection. Computational Intelligence and Security (CIS), Seventh International Conference on. pp. 1011-1015.

[11] Abela, K. Joshua, J. R. D. Alas, D. K. Angeles, R. J. Tolentino, M. A. Gomez. Automated Malware Detection for Android AMDA. In: The Second International Conference on Cyber Security, Cyber Peacefare and Digital Forensic (CyberSec2013). 180-188.

[12] G. Dini. , F. Martinelli, A. Saracino, D. Sgandurra. 2012. MADAM: a multi-level anomaly detector for android malware. In Computer Network Security. 240-253.

[13] W. B. Cavnar, J. M. Trenkle. N-gram-based text categorization. Ann Arbor MI 48113.2. 161-175.

[14] D. Jurafsky, J. H. Martin. 2000. Speech & language processing. Pearson Education India.

[15] S. Zhou, and G. Jihong. 2002. Chinese documents classification based on N-grams. Computational Linguistics and Intelligent Text Processing. Springer Berlin Heidelberg. pp. 405-414.

[16] J. Arpith and M. Gokhale. 2007. Language classification using n-grams accelerated by FPGA-based Bloom filters. Proceedings of the 1st international workshop on High-performance reconfigurable computing technology and applications: held in conjunction with SC07.

[17] R. Moskovitch, D. Stopel, C. Feher, N. Nissim, N. Japkowicz , Y. Elovici. 2009. Unknown malcode detection and the imbalance problem. Journal in Computer Virology. 5(4): 295-308.

[18] T. Abou-Assaleh, V. Keselj, R. Sweidan. 2004. N-gram based detection of new malicious code. Proc of the 28th Annual International Computer Software and Applications Conference, IEEE Computer Society. pp. 41-42.

[19] S. Asaf, R. Moskovitch, C. Feher, S. Dolev, Y. Elovici. 2012. Detecting unknown malicious code by applying classification techniques on opcode patterns. Security Informatics. 1(1): 1-22.

[20] BitDefender, [online] Retrieved on December 2014 http://www.bitdefender.com/

[21] eseT, [online] Retrieved on December 2014 http://www.eset.com/my/

[22] Virus Total, [online] Retrieved on December 2014 https://www.virustotal.com/

[23] M. Z. Mas'ud, S. Sahib, M. F. Abdollah, S. R. Selamat, R. Yusof. 2014. Analysis of Features Selection and Machine Learning Classifier in Android Malware Detection. Information Science and Applications (ICISA). pp. 1-5.

[24] R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, C. J. Lin. 2011. LIBLINEAR: A library for large linear classification. The Journal of Machine Learning Research. 9: 1871-1874.

[25] C. Chang, and C. Lin. 2011. LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST). 2(3): 27.

[26] Feizollah, Ali, N. B. Anuar, R. Salleh, F. Amalina, R. R. Ma'arof, and S. Shamshirband. 2014. A Study of Machine Learning Classifiers for Anomaly-Based Mobile Botnet Detection. Malaysian Journal of Computer Science. 26 (4).

[27] B. Sanz, I. Santos, X. Ugarte-Pedrero, C. Laorden, J. Nieves, y P.G. 2013. Bringas Instance-based Anomaly Method for Android Malware Detection. En Proceedings of the 10th International Conference on Security and Cryptography (SECRYPT). Reykjavik (Iceland). 387-394, ISBN: 978-989-8565-73-0

[28] Firdausi, Ivan, C. Lim, A. Erwin, and A. S. Nugroho. 2010. Analysis of machine learning techniques used in behavior-based malware detection. In Advances in Computing, Control and Telecommunication Technologies (ACT).

[29] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten. 2009. The WEKA data mining software: an update. ACM SIGKDD explorations newsletter. 11(1): 10-18.