ISSN:2231-0711

David Ndumiyana et al | IJCSET(www.ijcset.net) | 2017 | Volume 7 Issue 10, 79-83

Behaviour -based Malware Classification on Mobile Phones using Support Vector Machines

[*]David Ndumiyana and Tarirayi Mukabeta
Bindura University of Science Education
Faculty of Science, Computer Science Department
P. Bag 1020, Bindura, Zimbabwe.
[*]Email: ndumiyanad@gmail.com

Abstract-The malware threats for mobile phones continues to rise as demand has prompted the development process to mainly focus on adding new attractive features. Unfortunately, this exponential growth of mobile devices is not keeping pace with design of new security solutions before these threats can inflict widespread damage. Many business systems and networks are the main victims of malicious attacks by worms, viruses, spyware and other intrusion activities to cripple even the most critical success services. There are reports suggesting that combining spyware as a malicious payload with worms as a delivery agent has generated malicious programs that can be used for industrial espionage and identity theft. In this paper we propose a new behaviour approach using machine learning algorithms for detecting existing and emerging malware targeting mobile phones. The approach is basically focusing on the concept of a generalised behaviour pattern with additional emphasis on detecting new classes of malware that integrate attributes or features from existing classes of malicious malware bodies. The evaluation experiments demonstrate that different levels of abnormal behaviour were accurately detected.

Keywords: **malicious malware, SVM, mobile malware, Behaviour-based malware, machine learning.**

I. INTRODUCTION

Mobile malware is defined as software that exhibits malicious behaviour and is categorised into worms, viruses, botnets and Trojan horses. The authors of malicious software are now using it to gain financial resources on a larger scale. Mobile phones are widely used because they are portable and easy to use even by computer illiterate individuals. We can use mobile phones to do business, surf the Internet, access bank account services such as money transfer, and paying goods and services. This development in the use of mobile devices has also attracted an increased number of criminals who want to exploit these actions for illegal financial gains. Today's malware has improved on functionalities to the extent of being capable of doing many undesirable things such as stealing and transmitting the list of contacts, locking the device completely, providing remote access to criminals, sending and receiving unsolicited SMS and MMS messages. Mobile malware is causing serious public concern as the population of these mobile devices continues to rise more than the number of PCs [4, 20, 21, 22]. The ability of smartphones to share programs and data with each other has worsened the situation by providing virus authors with more fertile environments to launch their attacks to a wider space. With reported high annual growth rate [4] of smartphones, they are bound to become the dominant and most favoured communication devices in the near future, there are possibilities of a virus explosion whose consequences could outweigh the disruption caused by traditional computer viruses [5]. The speed transmission of mobile viruses is facilitated mainly by two common communication protocols such as Bluetooth where a virus can attack and infect all Bluetooth-activated mobile devices confined within a range of 10m to 30m radius. The second communication protocol is the multimedia messaging service (MMS) which can infect all mobiles phones whose numbers are found in the infected phone's address book by sending a copy of itself. This is a replica of computer viruses [12,11] which exploited a wider spreading space to catch a number of mobile victims.

II. THE MOTIVATION FOR WRITING MOBILE MALWARE

2.1 **Selling Personal User Information**: Most mobile operating system APIs provide huge amount of individual information which is a great attraction to commercial organization. The user profiles are required to target the individuals for certain types of goods and services in what can be described as business intelligence. The software can be used to search mobile APIs to locate user's location, list of addresses, browsers and download history as well as a list of installed programs. We are very confident that malware distributors are making a lot of money by selling such information.

2.2 **Stealing Confidential Credentials**: Mobile smartphones are now widely used for shopping, banking, e-mail, buying airtime and other important activities which require passwords and payment details. Human beings by nature can easily forget their passwords so to avoid embarrassment, most users end up keeping authentication and payment credentials in text documents in their mobile devices. As a result of this development, smartphones have become a sudden target for credential theft. One good example is a three pieces of malware that target user credentials by intercepting SMS messages to capture bank account credentials [14]. A loss of these banking credentials

can bring huge repercussions to whosoever is targeted and become a victim of circumstances.

2.3 Commercial Spamming: SMS spam is used for commercial advertising by mostly organizations which save personal information in their databases when they do business with their customers such as opening an account with credit stores or registering a phone line with a telecommunications service provider. SMS spam is very tricky because users are forced to read them believing the message is coming from a trusted friend or relative. In addition, the sound of the messages announces its presence in your inbox especially at a time when you are waiting for a call or SMS from a dear friend or member of the family. Commercial spammers prefer sending SMS spam using malware to avoid litigation because in some countries sending spam has been banned by law.

2.4 Demanding Ransom: Some disgruntled people may use malware as a tool to settle personal scores. Blackmail was used for example by the desktop Trojan Kenzero to steal the victim's browser history, have it published on the Internet together with the victim's name and demanded a ransom of 1500 yen to remove the browser's history. Although the cases are not many, however, a Dutch worm locked iPhones screens and demanded 5 euros to unlock screens of infected phones [14] in a suspected case of ransom.

III Related Literature Survey

Behaviour-based analysis and detection for malware had become the most preferred technique to the traditional signature-based approach. We want to recognise a variety of techniques proposed by previous authors and distinguish our method from current solutions. Research published by Forest et al.[41] was designed specifically for host-based anomaly detection. Their method observed the application behaviour in the form of system call sequences and they created a database of all consecutive system calls from normal application software. Potential intrusion in the network was discovered by looking for any call sequence that did not appear in the database. The authors later improved behaviour-based profile by applying advanced mining techniques on the call sequences such as rule learning algorithms [9] and hidden Markov model [16] among others but they all suffered simple obfuscation attacks [19]. The use of machine learning algorithms in anomaly detection has received wide acceptance particularly the use of Support Vector Machine (SVM) had been found effective in this regard [13]. Recent reports done by Vapnik [17] and Joachims [15] on statistical learning theory have registered a number of successes on many classification problems. A report published by Abhijit and Hu et al. [1, 18] was a proposed framework mobile for worms, viruses and Trojan horse detection. The technique used support vector machine classifier to distinguish malware and normal software. They presented a time domain sequence based on logical order of program behaviour, where an effective representation of malware behaviour was well articulated. In a related development, not to be outdone was the effort of Shabtai et al. [23] who proposed a methodology to detect suspicious temporal patterns as malicious behaviour, widely recognised as knowledge-based temporal abstraction. Their knowledge-based analysis system is different from ours in that we are proposing a behaviour based classifier. The research done by Burguera and Zurutuza et al.[2] gives a framework to detect malware on Android platform. Their work monitors system call in Linux level and generate software behavioural patterns and classifying these patterns using a cluster algorithm. The technique is effective for detecting malware that can be observed from a Linux kernel but suffered one limitation of failing to detect behaviour of many kinds malware from a Linux level. Specific examples of many kinds of malware that cannot be observed from a Linux level include send malicious SMS malware and malicious call malware to mention just a few.

3.1 Behaviour Classification for Mobile Malware

Apart from malware, there are other two groups of mobile threats namely personal spyware and grayware. Spyware is known for collecting information such as user location, SMS messages and call history without knowledge of the victim [3]. We cannot label spyware illegal because it does not send information to the application's author, but installs personal spyware on a mobile phone without the device owner's authorization, could be considered unethical. The second group is grayware which is simply annoying but less risk compared to other malware. Grayware can change user's font colours or install irritating pop-ups. Many smart phones contain grayware applications as they are closely regarded as legal [6].

3.2 Provides Novelty and Amusement.

AndroidWalkinwat is a very good example of malware developed by its authors simply to showcase their technical expertise and primarily designed for fun. It is not known to pose any risk to its victims.

3.3 Sells user Information

DroidDreamLight is an example of malware that secretly captures user details such as location, installed applications, download history and contact lists. Thereafter these details are cheaply sold to advertisers and marketers.

3.4 Steals User Credentials

The malware captures user credentials such as bank account details, by secretly snooping on text messages, capturing keystrokes by key logging, scanning documents and launching phishing attacks [7]. A common example is Ikee.B

3.5 Send SMS Spam

Geinimi is responsible in some cases for sending multiple messages to mobile phones that usually contain advertisements and phishing links.

3.6 Manipulates content delivery

The malware is known for generating premium-rate phone calls as well as sending text messages, probably to deliver content such as technical support and adult services. FakePlayer is a known example of this type of malware.

IV METHODOLOGY

Behaviour-based malware analysis and detection technique proposed in this paper trains a classifier using support vector machine learning algorithm (SVMs) to distinguish an infected program and normal application behaviours. This detection system is capable of recognising any malware that keeps on changing its features to fool the classifier. Our behaviour-based technique monitors behaviour of an application and compare against a set of malicious and/or normal behaviour profiles. Our behaviour-based classifier is more than able to deal with problems caused by code obfuscation since it evaluates the effects of an application based on more than just specific payload signatures. If we consider the fact that a new malware variant can be created by adding new functionality to existing malware, this abstraction, therefore becomes effective for detecting previously-unknown malware variants that share common behaviours exhibited by previous-known malware.

4.1 System Architecture

Sources codes of worms and normal applications are not freely and readily available, so to evaluate the proposed behavioural detection system, we borrowed tried and tested emulated applications of known worms. We were also able to modify certain application software to incorporate the current behaviours of malware. These were tested against real-world worms whose source codes are accessible by the authors of this paper. Most of the software we used emulated mostly known Symbian worms such as Cabir and Lasco. In order to get effective results we included the variants of each malware basing on a review of malware family reported most anti-virus manufacturers. Very important parameters were also considered among them were malware lifetime, number and contents of messages, type of attachment, size of attachment and malware payload. A total of 8 applications (3 legitimate and 5 worms) containing many loops corresponding to different malware behaviours that can be captured by run-time monitoring was used. The source that created malware behaviours from the monitoring system, we obtained full signatures of various lengths. Next we collected unique signatures generated from the sample runs to create a training dataset and a test dataset that was eventually used for our evaluation. We went on to generate several training and test datasets by repeating the procedure described in this section and calculating averages of classification accuracy, false positive and negative rates. After this, we used the training data to train the SVM model and classify each signature in the test data to determine the classification accuracy of our proposed system.

V. EXPERIMENTAL RESULTS AND DISCUSSION

In order to adequately evaluate the effectiveness of SVM in capturing previously known worms, we had to vary the size of training dataset. Table 5.1 shows the classification accuracy, number of false positives and false negatives for a test set of 905 varied signatures and different training data sizes.

Results coming from tests carried out indicated that SVM almost never falsely classified a legitimate application signature to be malicious whereas for small training set sizes, the number of false negatives was high. We also observed that as the training data size increased, the classification accuracy increased rapidly reaching nearly100% detection of malicious signatures.

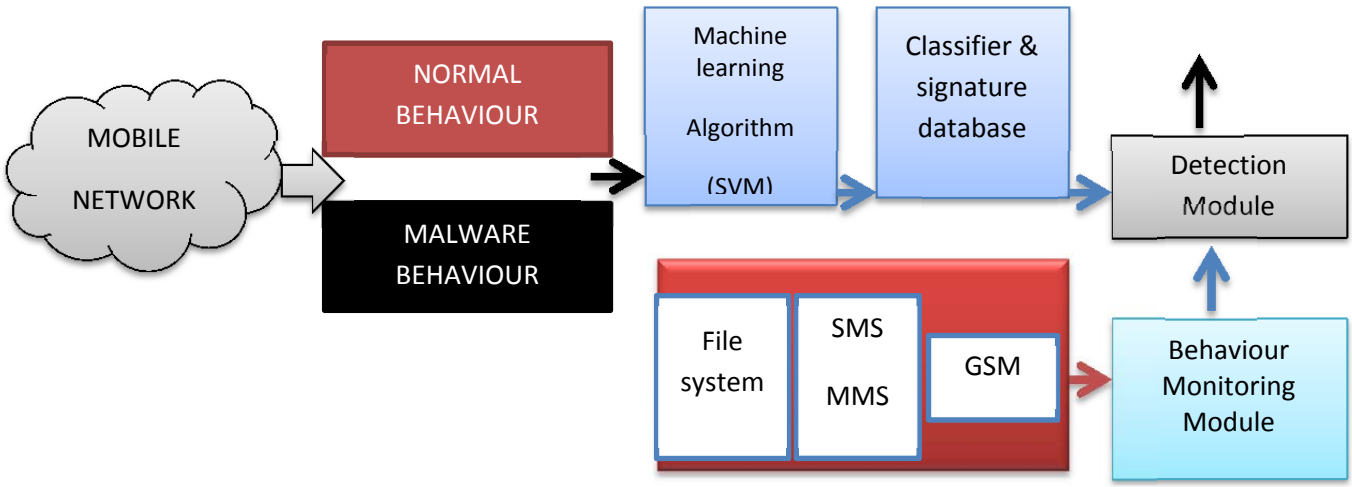

Fig. 4.1 Detailed Design of System

Training Set Size	Total of Support Vectors	Accuracy	False Positive	False Negative
22	21	82.1%	0	16
47	22	97.9%	1	18
56	20	97.5%	0	22
74	34	98.4%	0	14
92	29	99.4%	0	5
122	30	99.5%	0	4
142	51	99.2%	0	7
153	38	99.6%	0	3
256	48	100%	0	0
356	82	99.7%	0	2
462	61	100%	0	0
547	95	99.8%	0	1
628	106	99.8%	0	1
720	68	100%	0	0
798	186	99.8%	0	1

Table 5.1 Shows classification Accuracy, False Positive and False Negative results.

	Testing Set for (unknown) Worms				Overall
Training Set (Known) Worms	Cabir	Mabir	CW	Lasco	
CABIR	100	17	35	72.5	56
MABIR	100	100	51	27	69.5
CW	100	30.5	100	69.5	75
LASCO	64.5	17.5	38.5	100	55.1
CABIR, MABIR	100	100	42	54	74
CABIR, CW	100	45	100	100	86.3
CABIR, LASCO	100	27	50.5	100	69.4
MABIR, CW	100	100	100	100	100
MABIR, LASCO	100	100	100	100	100
CABIR, MABIR, CW	100	34.5	100	100	86.3
CABIR, MABIR, LASCO	100	100	100	76.5	94.1
CABIR, CW, LASCO	100	100	100	100	100
MABIR, CW, LASCO	100	99.5	100	100	99.9

Table 5.2: Showing detection accuracy (%) for unknown worms

The detection rates for different combinations of known and unknown worms were tabulated as shown in Fig. 5.2 below. According to these results it can be safely ascertained that SVC methodology was able to detect previously unknown worms, particularly for malware that share similar behaviour with existing malware. We also kept the size of a malicious signature database to small because new strains of worms targeting mobile phones kept on rising up. It was necessary to confirm the effectiveness of our behaviour-based detection system by testing it against real-world mobile malware. We were able to access source codes of two Symbian worms, Cabir and Lasco. By considering the fact that dynamic analysis of results may depend on the run-time environment, we had to run each malware sample 10 times with different environmental factors such as running time and number of neighbouring mobile phones. For example in one of the settings, the number of neighbouring phones was zero, thus motivating the worm to keep on searching for new devices infection. This development generated a variety of signature sizes that described the worm behaviour in each of the indicated working environment. We also applied the trained classifier with a training set size of 92 (Table 5.1) on each captured signature. It was observed that SVC achieved a 100% detection success on all worm instances.

CONCLUSION

We were able to successfully identify malicious behaviour from complete and partial signatures. We used SVM to train a classifier from normal and malicious data. The evaluation of both emulated and real-world malware demonstrated that behaviour detection not only results in high detection rates but also goes onto detect new malware which share certain behavioural patterns with existing patterns stored in the created database.

REFERENCES

[1] Abhijit, B. & Xin, H. Behavioural Detection of Malware on Mobile Handsets MobiSys'08, June 17 – 20, 2008.

[2] Burguera, I. & Zurutuza, U. Crowdroid: Behaviour-Based Malware Detection System for Android. SPSM'11, October 17, 2011, Chicago, Illinois, USA, 2011.

[3] http://www.infoq.com/articles Last visited 30 November, 2014.

[4] Cheng, J., Wong, S., Yang, H. and Lu, S. SmartSiren: Virus Detection and Alert for Smartphones. Proceedings of the 5th ACM International Conference on Mobile System Application Services. (ACM, New York, 2007).

[5] Zucker, D.F., Uematsu, M. and Kamada, T. Markup-based SmartPhone User Interface using the Web Browser Engine. Proceedings XTech 2005 (2005).

[6] Felt, A.P. et al., "A Survey of Mobile Malware in the Wild", Proceedings of ACM Workshop on Security and Privacy in Mobile Devices, ACM, 2011, pp. 3 -14

[9] Cohen, W.W. Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning*, 1995.

[11] Balthrop, J., Forest, S., Newman, M.E.J. and Williamson, M.M. Technological networks and the spread of computer viruses. Science 304, 527 – 529 (2004).

[12] Pastor-Satorras, R. and Vespignani, A. Epidemic spreading in scale-free networks. Phys. Rev.Lett. 86, 3200-3203 (2001).

[13] Gosh, A. K., Schwartzbard, A. & Schatz, M. Learning program behaviour profiles for intrusion detection. In ID'99: *Proceedings of the 1st conference on Workshop on Intrusion Detection and Network Monitoring*, 1999.

[14] Adrienne, P.F., Matthew, F., Erika, C., Steven, H. & David, W. "A Survey of Mobile Malware in the Wild", 1st ACM workshop on Security and privacy in smartphones and mobile devices, October 2011.

[15] Joachims, T. Making large-scale support vector machine practical, 1998.

[16] Sekar, R., Bender, M., Dhurjati, D. & Bollineni, P. A fast automaton-based method for detecting anomalous program behaviours. In SP'01: *Proceedings of the 2001 IEEE Symposium on Security and Privacy*, Washington, DC, USA, 2001. IEEE Computer Society.

[17] Vapnik, P. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[18] Xie, L. & Zhang, X. pBMDS: A Behaviour-based Malware Detection System for Cellphone Devices Wisec' 10, March 22-24, 2010, Hoboken, New Jersey, USA, 2010

[19] Warrender, C., Forest, S. & Pearlmutter, B.A. Detecting intrusions using system calls: Alternative data models. In IEEE Symposium on Security and Privacy, pp133 – 145, 1999.

[20] A. Bose, X. Hu, K. Shin and T. Park. Behaviour detection of malware in mobile handsets. In Proceedings of MobiSys, Breckenridge, CO, 2008.

[21] A. Bose and K. Shin. Proactive security for mobile messaging networks. In Proceedings of WiSe, 2006

[22] A. Bose and K. Shin. On mobile virus exploiting messaging and Bluetooth services. In Proceedings of Securecomm, 2006.