International Journal of Advance Research, IJOAR .org
Volume 1, Issue 7, July 2013, Online: ISSN 2320-9194

MALWARE BEHAVIOURAL TESTING IN ANDROID PLATFORM

Thiru. V. Kalyanasundaram
Thiru.v.23@yahoo.com

Abstract— The Android platform is that the quickest growing market in smart phone operating systems being the most viable target of security threats. Most anti-malware applications within the Market use static analysis for detection as a result of it, they are quick and comparatively easy. However, static analysis needs regular updates of threat databases and it is going to be circumvented by obfuscation techniques. As an answer to those issues, the study utilizes behavior analysis of applications as a basis for malware. As a primary step, features of non-benign and non-malicious applications are extracted for machine learning to provide baseline behavior datasets. The process of testing has been discussed in this paper.

Keywords—Android, Security, Behavior Analysis

Introduction

The Android platform utilizes an authorization-based security models to possess access to completely different functionalities of devices. This model provides data regarding the access and the privilege capability of an application which to a lot of technical users, may be used as a sign for malicious intent however to traditional users, this data is often neglected thus creating this model unreliable on its own. Static analysis may be a method adapted to discover malware however this method proves to be inadequate as malware will be unobserved by obfuscation. The time it takes to manually check the code provides a chance for malware to infect devices before they are detected.

To address these issues, an automated behavioural analysis system called AMDA is that the resolution. The AMDA system determines malicious behaviour from benign behaviour through the use of machine learning techniques. A behaviour model for Trojans, spyware, viruses and exploits are generated and used for classification of applications. The results are verified by forwarding them to skilled system, VirusTotal and from VirusTotal testing is carried out as described further.

Testing

The process of downloading applications, testing and selecting the rule, and classification of application is as follows:

Learning Data Acquisition

Known-malware applications with the kinds of Trojan, Virus, Spyware and Exploit are gathered manually from the skilled system VirusTotal which provides quality for the learning knowledge. VirusTotal uses discovered minimum of forty antivirus engines which scans the applications. Applications downloaded from VirusTotal are tagged as malware by a minimum of 10 anti-virus engines.

For the non-benign applications, they are gathered through the official Android Market employing a cell phone running on Gingerbread 2.3.3 Android OS. The applications are extracted from the cell phone through the use of a File Manager application called Astro. Each application extracted produces. apk file format of the applying and these files are forwarded to VirusTotal to verify their status as benign applications.

The table shows imbalanced distribution of applications downloaded for each classification from VirusTotal. Once the applications are processed through the API, most applications are classified as a Trojan, instead of the tagged classification

Table 1. Count of Applications Downloaded

Type	Number of Apps Collected from VirusTotal	Number of Apps used for Training
Benign	135	50
Trojan	2334	50
Virus	257	50
Spyware	87	50
Exploit	184	50

Application Acquisition

A web crawler is employed to transfer applications from different markets. Some sites involve javascript transfer links which cannot be accessed by the crawler. Owing to this, there are chosen markets where the web crawler could work since they need the direct transfer links. Each market includes a totally different webcrawler program to satisfy different settings.

Table 2. Count of Applications from Alt. Market

Alternative Android Market	Cell11	Appchina	Slideme
no. of applications downloaded	30	26	26

Application Virtualization

Applications are run within the emulator to collect for logs of the behavior of these applications to be employed by Weka. A tool, Strace, is employed to obtain the system calls made by the applying that's running. These system calls that are collected are then held on into a separate file along with its classification as benign or joined of the types of malware.

The systems calls to be collected are as follows: recv (), close (), bark (), open (), write (), msgget (), read (), lseek (), sigprocmask (), fork (), dup (), ioctl (), mprotect (), SYS_224.

Application Log Parsing

The application logs that are generated by Strace are collected and injected into the parser program. This parser program generates the ARFF (Attribute Relation File-Format) file to be employed by Weka in classifying the applications. The program searches for specific system calls made by the applying within the log file. The count of these system calls is taken so appended into the ARFF file. This is finished by collecting all desired system calls to be taken for all the applying logs.

Algorithm Testing

The ARFF file generated by the parser program is fed into Weka for the classification of the applications. Completely different rules are tested to envision whether which algorithm fares higher.

The metrics to be checked for being the following:

1) True Positive Rate 2) Kappa statistics 3) Receiver operative Characteristic (ROC)

Whilst the algorithms to be tested are the following: 1) J48 (J48graft) 2) Random Forest 3) Multinomial supplying Regression 4) Naive Thomas Bayes.

Application Log Parsing

The behavior logs of check applications are produced so processed through the use of the parser for check applications. The parser works with a set reference of relevant features based from the features of the behavior model of the coaching part. ARFF file for each check application is produced by the parser which is employed for comparison with the behavior-model of the most correct rule. The applications for this cake are downloaded from

completely different sources. These check applications came from VirusTotal and in addition from different Android markets namely: Slideme, Cell11 and Appchina.

Application Log Parsing

The behavior logs of check applications are produced so processed through the use of the parser for check applications. The parser works with a set reference of relevant features based from the features of the behavior model of the coaching part. ARFF file for each check application is produced by the parser which is employed for comparison with the behavior-model of the most correct rule

The applications for this cake are downloaded from completely different sources. These check applications came from VirusTotal and in addition as form different humanoid markets namely: Slideme, Cell11 and Appchina.

AMDA application classification results hold on within the info are compared to the classification report from VirusTotal. AMDA collates the results and uses a numeration mechanism as to what percentage tags of applications are made by the AV engines as Trojans, Spyware, Exploit, Virus and Unclassified.

Again, the performance of the system is measured by true Positive Rate, Kappa statistics and also the Receiver operating Characteristic (ROC Curve).

Algorithm And Classification Results

The coaching/training support for the system undergoes a rigorous method for being able to generate the most effective behavior model for the system. There' are total of 80 varieties of tests done. As mentioned within the earlier section, each rule is tested with three completely different feature choice strategies and without a feature choice method used. This is done 5 times and for each check, the variety of applications used varied in number

Table 3. Number of Applications per Training Phase

Training Phase	Number of Applications
Test 1	50
Test 2	100
Test 3	150
Test 4	200
Test 5	250

For the coaching set, the Random Forest rule in check one with Gain magnitude relation because the feature choice method and check 5 with no feature choice method (See Figure. 8) Garnered the most effective accuracy through the measure by True Positive Rate. Both tests achieved 78 accuracy. Additionally, the Random Forest rule consistently outperformed the opposite algorithms. The behavior model of coaching part check 5 is chosen to be used for the system since it performed well even with a bigger variety of applications used and also the check garnered a higher rate of Roc which means that the rule is defined with its classifications.

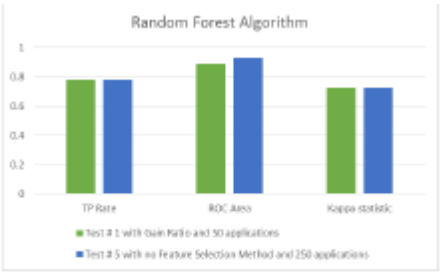

Figure 8. Summarized Results of Machine Learning Algorithms

After knowing the most effective rule for classification through the coaching part, gathering and processing of check applications follow

Table 4. Applications Classified by AMDA

Type of Android Application	Number of Applications Classified
Benign	35
Trojan	41
Spyware	54
Exploit	81
Virus	13

There is a total of 224 applications parsed by the system. The results are compared to the classification report from VirusTotal.

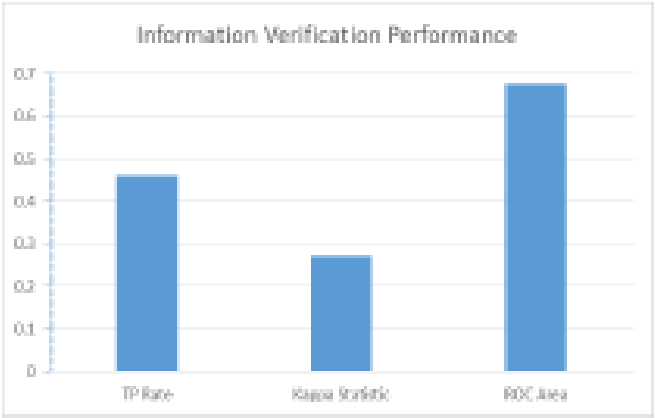

Figure 9. Information Verification Results

When the results of the AMDA System are valid to the results garnered through VirusTotal, TP Rate measure exacted to 46.2%, Kappa statistics to 27.17% and also the Roc space measured 67.5%. The results above represent quite a low accuracy for classification of the types of Android applications. The Roc space, being above the 500 mark, means the system is mostly certain of its classifications. An occasional liver was garnered by the Kappa statistics which means that the system encountered a dataset with principally random attributes.

Further checking deep into the system calls is made to spot different measure of analysis and issues. It is found that 14of the supervisor call instruction features exhibit identical characteristics for pairs of malware varieties. Virus and Exploit applications generally live identical for these system calls. Trojan and Spyware applications are paired for the mentioned system calls.

With that data, it will be derived that malware applications exhibit identical behaviors which explains why the results of the classification is low. Instead of having 4 classifications for Malware, it's simplified into simply Malware versus Benign classifications.

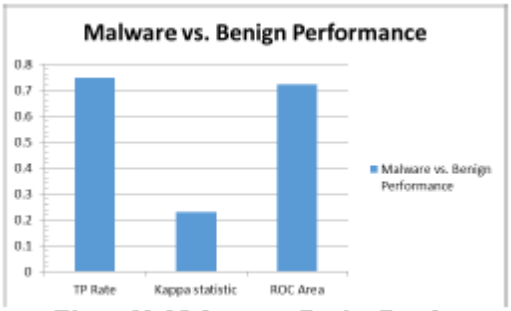

Figure 10. Malware vs. Benign Results

The TP Rate measure inflated to 74.7%, the Kappa statistics to twenty three. 17% and also the Roc space as 721%. The TP Rate achieved a considerably higher value share compared to the previous result which indicates that the system is in a position to correctly classify the applications. The Kappa statistics measured is sort of identical because the previous check. This is expected since identical dataset is employed as with the previous check. The Roc space still achieved a high measure which indicates that the system is mostly certain of the classifications made.

Conclusion

The system, given the potential to classify unknown applications based from its knowledge, will be accustomed reason completely different humanoid applications within the market. With the online crawler at hand, the system has the potential to automatically transfer and classify new applications uploaded to the various different markets. Aside from these, the system has the power to classify malware to different type victimization behavior-based analysis. With this at hand, the system can act as antivirals that would easily provide classification results to users.

However, skilled systems or completely different classification sources modification classifications from time to time. This happens once a lot of antivirus engines are ready to classify applications as from once the applying was 1st classified or as a result of there are a

lot of and a lot of malware families being identified. With this, there's a clear lack of standards within the classification scheme of applications. This lack of standards contributes to the in utility of classifying malware into completely different classifications aside from simply classifying it as malware. Another issue would be that malware families would have a variety of different malware families which makes it even tougher to differentiate between malware varieties.

Future Work

Further work to be done is that the ability to discover advanced malware attacks like Zero-day attack. Implementation of Behavior-based analysis with permission-based also can be done to work out malicious Android applications. Administrative program an AMDA humanoid Application will allow easier analysis and access of the system.

References

[1] Srivastava, H., Choudhury, T., & Vashisht, V. Counter Strike: Prompt Gaming Time on Android and iPhone. International Journal of Scientific & Engineering Research.

[2] Sharma, R. Study of Latest Emerging Trends on Cyber Security and its challenges to Society. International Journal of Scientific & Engineering Research.

[3] Gharehchopogh, F. S., Abbaspour, F., Tanabi, M., & Maleki, I. Review and Evaluation of Performance Measures in the Mobile Operating Systems.

[4] Gharehchopogh, F. S., Rezaei, R., & Maleki, I. Mobile Cloud Computing: Security Challenges for Threats Reduction. International Journal of Scientific & Engineering Research.

[5] Garg, P., & Sharma, V. Secure Data Storage in Mobile Cloud Computing. International Journal of Scientific & Engineering Research.

[6] Chauhan, A., Mishra, G., & Kumar, G. (2012). Survey on Data mining Techniques in Intrusion Detection. Lap Lambert Academic Publ. International Journal of Scientific & Engineering Research

[7] Ibrahim, H. E., Badr, S. M., & Shaheen, M. A. (2012). Phases vs. Levels using Decision Trees for Intrusion Detection Systems. arXiv preprint arXiv:1208.5997.

[8] Singh, M., Singh, G., & Sharma, S. (2012). Human Protein Function Prediction from Sequence Derived Features using See5. International Journal of Scientific & Engineering Research

[9] Ali, K. B., & Gosain, A. (2012). Predicting the quality of object-oriented multidimensional (OOMD) model of data warehouse using decision tree technique. Int J Sci Eng Res, 1-5.

[10] Kulkarni, M. S. V. (2011). Mining knowledge using Decision Tree Algorithm.International Journal of Scientific and Engineering Research, 2(5), 131-136

[11] Pitale, V. V. K. R. R., & Tajane, K. PERFORMANCE IMPROVEMENT USING INTEGRATION. International Journal of Scientific & Engineering Research

[12] Bhoria, M. P., & Garg, K. An Imperial learning of Data Mining Classification Algorithms in Intrusion Detection Dataset. International Journal of Scientific & Engineering Research

[13] Padmavathi, J. (2012). Logistic regression in feature selection in data mining.International Journal of Scientific & Engineering Research, 3(8).

[14] Bhoria, M. P., & Garg, K. An Imperial learning of Data Mining Classification Algorithms in Intrusion Detection Dataset. International Journal of Scientific & Engineering Research.

[15] Firdhous, M., Hassan, S., & Ghazali, O. A Comprehensive Survey on Quality of Service Implementations in Cloud Computing. International Journal of Scientific & Engineering Research.

[16]T. Vennon, "Threat Analysis of the Android Market," 2010. [Online]. Available:http://www.globalthreatcenter.com/wp-content/uploads/2010/06/Android-Market-Threat-Analysis-6-22-10-v1.pdf [Accessed: October 30, 2012]

[17] K. Elish, D. Yao, and B. Ryder, "User-Centric Dependence Analysis For Identifying Malicious Mobile Apps," in Proceedings of the IEEE CS Security and Privacy Workshop, 2012. San Francisco, CA.

[18]T. Isohara, K. Takemori and A. Kubota, "Kernel-Based Behavior Analysis for Android Malware," in proceedings of the 2011 Seventh International Conference on Computational Intelligence and Security. Saitama, Japan. 2011.

[19]I. Burguera., U. Zurutuza and S Tehrani, "Crowdroid: Behavior-Based Malware Detection System for Android," in Proceedings of the 18th ACM Conference on Computer and Communications Security, 2011. Chicago, IL. 17 October 2011.

[20] T. Blasing and et al, "An Android Application Sandbox System for Suspicious Software Detection," in Proceedings of the 5th IEEE International Conference on Malicious and Unwanted Software, 2010.

[21] Oracle, 2008. "Data Mining Concepts: Regression," 2008.[Online].Available:http://docs.oracle.com/cd/B28359_01/datamine.111/b28129/regress.htm#DMCON005 [Accessed: October 30, 2012]

[22] B. Sans., "On the Automatic Categorisation of Android Applications," 2012. [Online]. Available: http://paginaspersonales.deusto.es/isantos/publications/2012/Sanz_2012_CCNC_Android_Apps_Categorisation.pdf. [Accessed: October 25, 2012]

[23] A. Shabtai and C. Glezer, " "Andromaly" a behavioral malware detection framework for android devices," 2010. [Online]. Available: http://posgrado.escom.ipn.mx/biblioteca/%E2%80%9CAndromaly%E2%80%9D%20a%20behavioral%20malware%20detection.pdf

[24] P. Flach and N. Lachiche, "Naïve Bayesian Classification of Structured Data," [Online]. Available: http://www.cs.bris.ac.uk/~flach/papers/mlj04-1BC-final2.pdf [Accessed: November 1, 2012]

[25] H. Zhang, "The Optimality of Naïve Bayes," [Online]. Available: http://courses.ischool.berkeley.edu/i290-dm/s11/SECURE/Optimality_of_Naive_Bayes.pdf [Accessed: November 1, 2012]

[26] S. Kotsiantis, I. D. Zaharakis and P. E. Pintelas, "Supervised Machine Learning: A Review of Classification and Combining Techniques," [Online]. Available: www.cs.bham.ac.uk/~pxt/IDA/class_rev.pdf [Accessed: November 2, 2012]

[27] J. Chan, K. Chan, and A. Yeh, "Detecting the Nature of Change in an Urban Environment: A Comparison of Machine Learning Algorithms," American Society for Photogrammetry and Remote Sensing, , vol. 67, No. 2, pp. 213-225, February 2001.

[28] L. Breiman and A. Cutler, "Random Forests," [Online]. Available: http://stat-www.berkeley.edu/users/breiman/RandomForests/cc_home.htm#intro [Accessed: November 3, 2012]

[29] L. Moutinho and G.D. Hutcheson, "Dictionary of Quantitative Methods in Management," [Online]. Available: http://www.research-training.net/addedfiles/READING/MNLmodelChapter.pdf [Accessed: November 4, 2012]

[30] I. Rish, "An Emprical Study of the naïve Bayes Classifer," [Online]. Available: www.cc.gatech.edu/~isbell/reading/papers/Rish.pdf [Accessed: November 5, 2012]

[31] Weka, 2008, "Weka: Primer," 2012. [Online]. Available: http://weka.wikispaces.com/Primer [Accessed: November 5, 2012]

[32]. J. Tiedemann, "Interpreting Weka Output," [Online]. Available: http://www.let.rug.nl/tiedeman/ml06/InterpretingWekaOutput [Accessed: November 5, 2012]

[33] J. He, "Linux System Call Quick Reference," [Online]. Available: http://www.digilife.be/quickreferences/qrc/linux%20system%20call%20quick%20reference.pdf [Accessed: November 7, 2012]

[34] T.Borovicka, M.Jirina Jr., P. Kordik and M. Jirina, "Selecting Representative Data Sets," [Online]. Available: http://cdn.intechopen.com/pdfs/39037/InTech-Selecting_representative_data_sets.pdf [Accessed: December 2, 2012]

[35] ESET Labs, 2013. "Trends for 2013: Astounding growth of mobile malware.," [Online]. Available: http://go.eset.com/us/resources/white-papers/Trends_for_2013_preview.pdf