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

BEHAVIOUR ANALYSIS MODEL AND DECISION TREE CLASSIFICATION OF MALWARE IN ANDROID

Orhan Pamuk, Ha Jin

Abstract

The Android platform with its growing market in smart phone operative systems is the most viable target of security threats. The dependence of the Android Market Security Model on its reactive anti-malware system offers an opportunity for malware to be present within the Official Android Market and does not include applications outside the official market. This allows applications to mask as harmless applications which result in the loss of credentials if precautions are not taken. Most anti-malware applications within the Market use static analysis for detection as a result of it, they are fast and relatively easy. However, static analysis needs regular updates of threat databases and it is going to be circumvented by obfuscation techniques. As a response to those matters, the study utilizes behavior analysis of applications as a basis for malware.

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 information 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 behavioral analysis system called AMDA is that the resolution. The AMDA system determines malicious behavior from benign behavior through the use of machine learning techniques. A behavior 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.

Behavior-based Analysis Module

The behavior-based Analysis Module is liable for classifying humanoid applications as either benign or malicious. This is done by applying machine learning algorithms for the generation of behavior models of malicious and benign applications. A coaching part, break away the system, is that the one which identifies the behavior of the applications. This module identifies Android applications into four classifications namely: Virus, Trojan, Spyware, Exploit or Benign.

For the coaching part, behavioral models for each and every sort of Android application are generated by sampling a variety of applications per each classification to run on entirely different algorithms. Features of applications extracted from the previous module are translated into an .Arff file format for Weka to enable it to amass knowledge. Currently, the module solely generates the accuracy results of the chosen algorithms given feature sets from each sort of malware and of the benign applications.

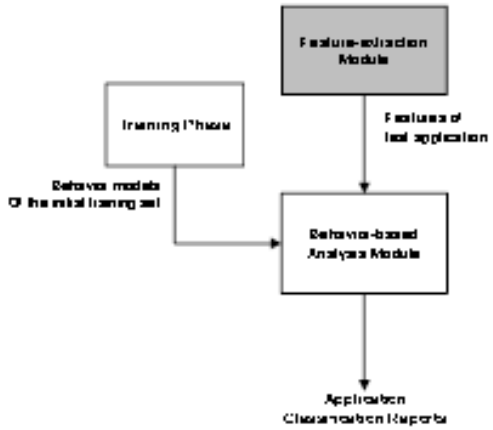

Fig: Behavioural Analysis

The patterns applied for this module embody the Naïve Thomas Bayes algorithm for high bias in small data sets, the Decision Tree Algorithm for its low bias 5 and also the logistic Regression algorithm to accommodate for adjustments within the features. Based on studies which used the same system setup for malware detection, the same algorithms performed best based on the garnered False Positive Ratings and True Positive magnitude relation from the tests. The most efficient algorithm based on percentage of correctly classified instances, kappa statistics, precision, true positive rate and false positive rate

```
=== Summary ===

Correctly Classified Instances          32               64      %
Incorrectly Classified Instances        18               36      %
Kappa statistic                          0.5114
Mean absolute error                      0.18
Root mean squared error                  0.4234
Relative absolute error                 48.4208 %
Root relative squared error             97.9931 %
Total Number of Instances               50
```

Figure 4. Weka Statistics Result

In the statistics outline above (Fig. 4), the portion of the right classified instances is 64 and for the inaccurate is 365 days. The correctly and incorrectly classified instances, usually called accuracy or sample accuracy, are the share of the check instances that were correctly and incorrectly classified. These also seek advice from the case where the illustrations are used as check knowledge. Once it involves classification, correctly and incorrectly classified instances are the most authoritative figures and can be applied in the field. With these figures, correctness of the categorization of the applications for the various grade levels will be determined. The numbers of applications and also the classification are shown within the

Confusion Matrix below, where a, b, c and d are the class labels which within the study's case, the malware varieties. There have been 50 samples, thus when you add up, a + b + c + d = fourteen + fourteen + fourteen + 8

```
=== Confusion Matrix ===

  a  b  c  d   <-- classified as
 10  0  4  0 |  a = trojan
  1  8  4  1 |  b = virus
  2  2  9  1 |  c = exploit
  0  2  1  5 |  d = spy
```

Figure 5. Weka Confusion Matrix

Kappa statistics (see Fig. 4) measures the agreement of prediction between genuine categories and also the classifications. A figure higher than 0.0 means the classifier is doing higher than the chance and a figure of 1.0 demonstrates complete or excellent agreement. However, the error rates are used for numeric prediction instead of classification tasks which aren't relevant within the study.

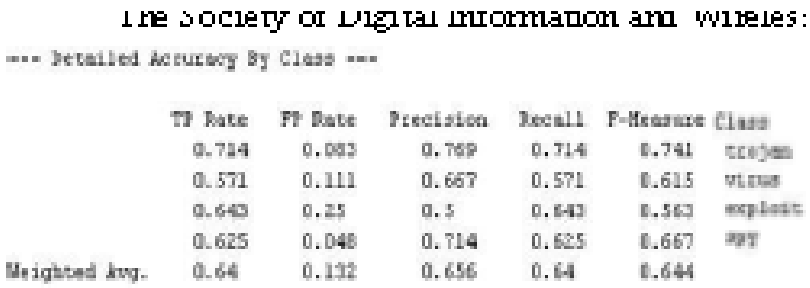

Figure 6. Weka Detailed Accuracy Result

The True Positive (TP) rate is that the proportion of applications which were actually categorized to an exact class and the way a lot of part of the class was captured. It is also equal to the Recall. The share of Trojan-labeled applications that are classified as Trojans

might be determined utilizing TP. False Positive (FP) rate is that the proportion of examples which were categorized to an exact category however belongs to a varied class. With FP, the share of the applying classified as Trojans however are Virus-labelled will be generated. The correctness is that the proportion of the cases which actually belong to a class among those were classified to a specific category. The F-Measure is solely (2*Precision*Recall/(Precision + Recall), a combined calculation for Recall and precision. This could really be interpreted to be the weighted average of precision and Recall. ROC, on the opposite hand, is that the calculation of certainty of the algorithm/rule with the classification made.

Machine Learning

Naïve Bayes

Naïve Thomas Bayes is that the simplest type of Bayesian Network whereby given a categorical variable, all properties are assumed to be independent. The rule/algorithm is in a position to categorize by calculating the utmost likelihood of the attributes belonging to an exact year. Even with the interaction of certain attributes, the Naïve Thomas Bayes assumption does not lose prophetical accuracy even if the actual probabilities are completely dissimilar.

A perceiving of the Bayer classifier (1) is required to also understand the Naïve Bayes classifier. C is that the category of the unobserved stochastic variable to be learned. X denotes a feature vector variable while x denotes the value of the variable. Given the Thomas Bayes Classifier

$$h^*(x) = \arg\max P(X = x | C = i) P(c = i)$$

,

Equation 1. Bayes Classifier which determines the maxmim a posteriori probability (MAP) given example x, proves difficult in providing direct estimation when there is high-dimensionality in feature space. This is because the Bayes classifier considers a class-conditional probability distribution (CPD) defined in $P(X = x|C = i)$ which relies on the dependence of each feature vector to another. Equation (2) describes a simplified assumption of the independence of features given the class.

$$f_i^{NB}(x) = \Pi_{j=1}^{n} P(X_j = x_j | C = i) P(C = i)$$

Equation 2. Naive Bayes Classifier

Detection

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 examined are the following: 1) J48 (J48graft) 2) Random Forest 3) Multinomial supplying Regression 4) Naive Thomas Bayes

Decision Trees

Decision Trees base the classification of cases by sorting feature vectors. In the very call tree, a node represents a feature to be classified and a branch represents following potential value of a node. Call trees may be interpreted as a set of rules for each path from the basics to every leaf of the tree. The foundations utilized by call trees define how a split is formed and the way cases are classified as to what animation is attained. These rules may additionally be

derived from coaching knowledge to be used for actual testing. 11

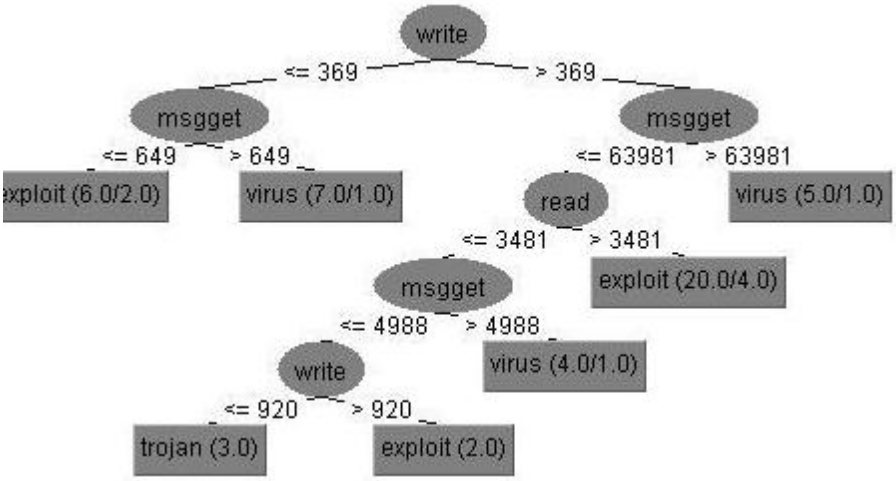

Figure 7. Decision Tree Sample

F. J48

J48 is open source Java-based implementation of the C4.5 call Tree rule. The rule splits the detail set to create an exact node in a tree. The info with the highest data gain would be the one that the majority effectively splits the information set onto one class or another thus this certain data is preferred. Once selecting the info, a call node is formed to split based on the info chosen. The sublist obtained by breaking up the data with the best information gain is that the recursed and added as children of the decision node.

Random Forest

Random Forest utilizes several classification trees to be ready to classify object based on the majority vote of classification generated by the trees. A tree is mature by 1st sampling a random variety of N cases within the coaching set. For each input variable M, variety value m is employed for each guest to pick out every which way from the input variable to be accustomed to split a node. Afterwards, the generated tree is fully grown as deep as possible.

Multinomial Logistic Regression

Since the study classifies over two kinds of an Android application, Multinomial logistic Regression (MLR) has to be used to offer polychotomous results over logistic Regression (LR) which solely produces a divided result. In this note, MLR is an extension of LR which provides regression models by comparison of arbitrary reference class two categories of a

unordered response variable. Simply put, MLR utilizes multiple logistic regressions on x multi-categorical response variable that's unordered. Equation (3) illustrates a general multinomial supplying regression model where j is identified variants and j' : is reference variable and X is informative variable affects the resulting model.

$$\log \frac{Pr(Y = j)}{Pr(Y = j')} = \alpha + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k$$

Equation 3. General Equation for MLR

Conclusion

The system, given the potential to classify unknown applications based from its knowledge, will be accustomed to reason completely different Android applications within the marketplace. With the online crawler at hand, the scheme holds the potential to automatically transfer and classify new applications uploaded to the various different markets. Aside from this, the system has the ability to classify malware to different types using 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 quick to classify applications as from once the applying was 1st classified or as a result of there are a lot of malware families being placed. With this, there's a clear lack of standards within the categorization scheme of applications. This lack of standards contributes to the inutility of classifying malware into completely different classifications aside from simply classifying it as malware. Another consequence would be that malware families would have a diversity of different malware families which makes it even harder 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 such as 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: Aug 30, 2013]

[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] 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] 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%20 ehavioral%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] 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