A Chronological Evaluation of Unknown Malcode Detection

Robert Moskovitch[1], Clint Feher[1], Yuval Elovici[1]

[1] Deutsche Telekom Laboratories at Ben Gurion University
Ben Gurion Univsersity of the negev, Beer Sheva 84105, Israel
{robertmo, clint, elovici}@bgu.ac.il

Abstract. Signature-based anti-viruses are very accurate, but are limited in detecting new malicious code. Dozens of new malicious codes are created every day, and the rate is expected to increase in coming years. To extend the generalization to detect unknown malicious code, heuristic methods are used; however, these are not successful enough. Recently, classification algorithms were used successfully for the detection of unknown malicious code. In this paper we describe the methodology of detection of malicious code based on static analysis and a chronological evaluation, in which a classifier is trained on files till year k and tested on the following years. The evaluation was performed in two setups, in which the percentage of the malicious files in the training set was 50% and 16%. Using 16% malicious files in the training set for some classifiers showed a trend, in which the performance improves as the training set is more updated.

Keywords: Unknown Malicious File Detection, Classification.

1 Introduction

The term malicious code (malcode) commonly refers to pieces of code, not necessarily executable files, which are intended to harm, generally or in particular, the specific owner of the host. Malcodes are classified, based mainly on their transport mechanism, into five main categories: worms, viruses, Trojans, and a new group that is becoming more common, which comprises remote access Trojans and backdoors. The recent growth in high-speed internet connections has led to an increase in the creation of new malicious codes for various purposes, based on economic, political, criminal or terrorist motives (among others). A recent survey by McAfee indicates that about 4% of search results from the major search engines on the web contain malicious code. Additionally, Shin et al. [12] found that above 15% of the files in the KaZaA network contained malicious code. Thus, we assume that the proportion of malicious files in real life is about or less than 10%, but we also consider other options.

Current anti-virus technology is primarily based on signature-based methods, which rely on the identification of unique strings in the binary code, while being very precise, are useless against unknown malicious code. The second approach involves heuristic-based methods, which are based on rules defined by experts, which define a malicious behavior, or a benign behavior, in order to enable the detection of unknown malcodes [4]. The generalization of the detection methods, so that unknown malcodes can be detected, is therefore crucial. Recently, classification algorithms were employed to automate and extend the idea of heuristic-based methods. As we will describe in more detail shortly, the binary code of a file is represented by n-grams, and classifiers are applied to learn patterns in the code and classify large amounts of data. A classifier is a rule set which is learnt from a given training-set, including examples of classes, both malicious and benign files in our case.

Over the past five years, several studies have investigated the option of detecting unknown malcode based on its binary code. Schultz et al. [11] were the first to introduce the idea of applying machine learning (ML) methods for the detection of different malcodes based on their respective binary codes. This study found that all the ML methods were more accurate than the signature-based algorithm. The ML methods were more than twice as accurate, with the out-performing method being Naïve Bayes, using strings, or Multi-Naïve Bayes using byte sequences. Abou-Assaleh et al. [1] introduced a framework that used the common n-gram (CNG) method and the k nearest neighbor (KNN) classifier for the detection of malcodes. The best results were achieved using 3-6 n-grams and a profile of 500-5000 features.

Kolter and Maloof [6] presented a collection that included 1971 benign and 1651 malicious executables files. N-grams were extracted and 500 were selected using the information gain measure [8]. The authors indicated that the results of their n-gram study were better than those presented by Schultz and Eskin [11]. Recently, Kolter and Maloof [7] reported an extension of their work, in which they classified malcodes into families (classes) based on the functions in their respective payloads.

Henchiri and Japkowicz [5] presented a hierarchical feature selection approach which makes possible the selection of n-gram features that appear at rates above a specified threshold in a specific virus family, as well as in more than a minimal amount of virus classes (families). Moskovitch et al [9], who are the authors of this study, presented a test collection consisting of more than 30,000 executable files, which is the largest known to us. A wide evaluation consisting on five types of classifiers, focused on the imbalance problem in real life conditions, in which the percentage of malicious files is less than 10%, based on recent surveys. After evaluating the classifiers on varying percentages of malicious files in the training set and test sets, it was shown to achieve the optimal results when having similar proportions in the training set as expected in the test set.

In this paper we investigate the need in updating the training set, through a rigorous chronological evaluation, in which we examine the influence of the updates of the training set on the detection accuracy. We start with a survey of previous relevant studies. We describe the methods we used to represent the executable files. We present our approach of detecting new malcodes and perform a rigorous evaluation. Finally, we present our results and discuss them.

2. Methods

2.1 Data Set Creation

We created a data set of malicious and benign executables for the Windows operating system. After removing obfuscated and compressed files, we had 7688 malicious files, which were acquired from the VX Heaven website. The benign files set contained 22,735, including executable and DLL files, were gathered from machines running Windows XP operating system on our campus. The Kaspersky anti-virus program was used to verify that these files indeed contain malicious code, or don't for the benign files.

2.2 Data Preparation and Feature Selection

We parsed the files using several *n-gram* lengths moving windows, denoted by *n*. Vocabularies of 16,777,216, 1,084,793,035, 1,575,804,954 and 1,936,342,220, for 3-gram, 4-gram, 5-gram and 6-gram respectively were extracted. Later each n-gram term was represented using its Term Frequency (TF), which is the number of its appearances in the file, divided by the term with the maximal appearances. Thus, each term was represented by a value in the range [0,1]. To reduce the size of the vocabularies we first extracted the top features based on the Document Frequency (DF) measure. We selected the top 5,500 features which appear in most of the files, (those with high DF scores). Later, three feature selection methods: *Gain Ratio (GR)* [8] and *Fisher Score (FS)* [3], were applied to each of these two sets. We selected the top 50, 100, 200 and 300 features based on each of the feature selection techniques. More details on this procedure and results can be found in [9], in which we found that the optimal settings were top 300 features selected by Fisher score where each feature is 5-gram represented by TF from the top 5500 features, which we used in this study.

3 Evaluation

We employed four commonly used classification algorithms: *Artificial Neural Networks* (ANN) [Bishop, 1995], *Decision Trees* (DT) [10], *Naïve Bayes* (NB) [2]. We used the Weka [13] implementation for the Decision Trees and the Naïve Bayes and the ANN tool box in Matlab.

To evaluate the importance of and need for updating the training set, we divided the entire test collection into the years from 2000 to 2007, in which the files were created. Thus, we had 6 training sets, in which we had samples from year 2000 till year 2006. Each training set was evaluated separately on each following year from 200k+1 till 2007. Obviously the files in the test were not presented in the training set. We present two experiments which vary in the Malicious Files Percentage (MFP) in the training set, having 50% which is commonly used and 16% which is expected to maximize the performance, which was the same in the test set (16%) to reflect real life conditions (in both cases).

Decision Trees

Figure 3 presents the results of the chronological evaluation, for the 50% MFP in the training set. Training on 2000 results below 0.9 accuracy, while training on the next years improved the accuracy. However, generally a significant decrease in performance was seen when testing on 2007.

Figure 4 presents the results of the chronological evaluation, in which the MFP in the training set was 16%. In Figure 4 we see generally a higher performance than in figure 3. 2004 introduced a significant challenge for the training sets of until 2000 and 2001. In this set of results there is a clear trend which shows that the more the training set is updated the higher the accuracy in the following years, and even when testing on 2007 the accuracy was above 0.9, when trained on till 2004, 2005 and 2006.

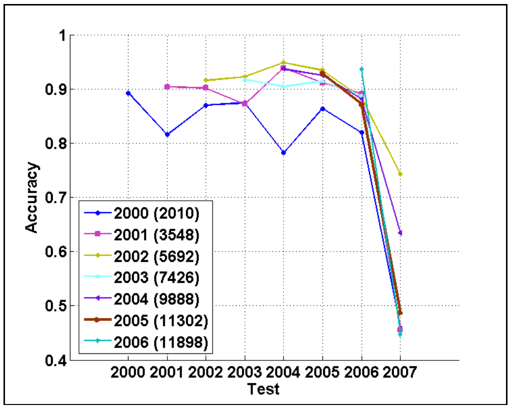

Figure 3. For the 50% MFP training set the more updated the higher the accuracy. A significant decrease is seen in 2007, while training on 2006 outperforms.

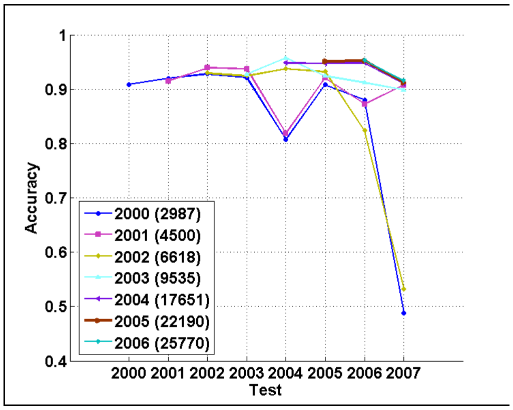

Figure 4. For the 16% MFP training set, the more updated the higher the accuracy. Testing on year 2004 presented a challenge for the 2000 and 2001 training sets.

Naïve Bayes

Figure 5 and 6 present the results of the chronological evaluation using the Naïve Bayes classifier, for the 50% MFP (fig 5) and 16% MFP (fig 6) training sets. In 2003 there is a significant drop in the accuracy in both MFPs, which appears only with this classifier. The results in general are lower than the other classifiers. The results with the 16% MFP are slightly better. However, in both figures the accuracy drops for the last years, especially for 2007.

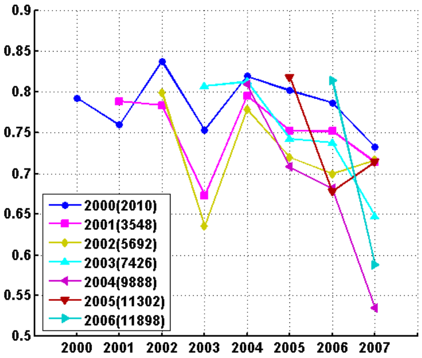

Figure 5. For the 50% MFP a decrease is in 2003 and generally the results are low.

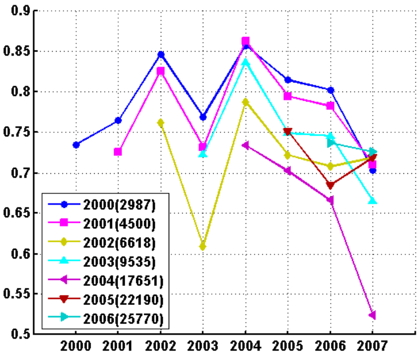

Figure 6. For the 16% MFP there is a slight improvement in comparison to the 50% MFP.

Artificial Neural Networks

Figures 7 and 8 present the chronological results for the ANN classifier. The results seem better than the Naïve Bayes, especially for the 16% MFP results. In Figure 8 the results seem to perform very well along most of the years, out of a significant drop for the training set from 2005, especially with the test set of 2007.

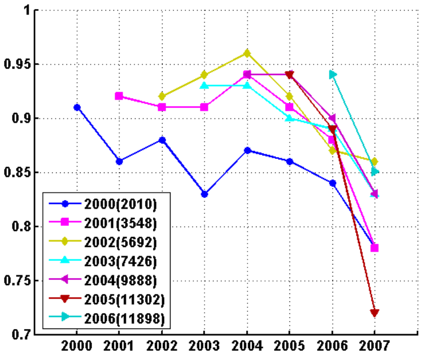

Figure 7. For the 50% MFP training set, training on 2000 performed very low, unlike the others.

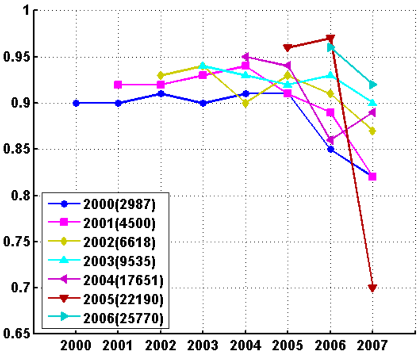

Figure 8. For the 16% MFP there is a significant improvement in comparison to the 50% MFP.

4 Discussion and Conclusions

We presented the problem of unknown malicious code detection using classification algorithms. We described the use of n-grams for the representation where feature selection methods are used to reduce the amount of features. We presented the creation of our test collection, which is 10 times larger than any previously presented. In a previous study [9], we investigated the aspects of the percentage of malicious files in the training set to maximize the accuracy in real life conditions.

In this study we referred to the question of the importance of updating the training set with the new malicious codes in a yearly time granularity and whether it is important to keep samples of old files in the training set from few years ago. Our results indicate that when having 16% MFP in the training set which corresponds to the test set we achieve a higher level of accuracy, and also a relatively clear trend that as the training set is more updated the accuracy is higher. However, this varies according to the classifier and one should be aware of this influence in deployment, as sometimes it decreases the accuracy. Moreover, it seems to be better to have also files which are from several years earlier and to incrementally update the database.

References

[1] Abou-Assaleh, T., Cercone, N., Keselj, V., and Sweidan, R. (2004) N-gram Based Detection of New Malicious Code, Proceedings of the International Computer Software and Applications Conference (COMPSAC'04).

[2] Domingos, P., and Pazzani, M. (1997) On the optimality of simple Bayesian classifier under zero-one loss, Machine Learning, 29:103-130.

[3] Golub, T., Slonim, D., Tamaya, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C., and E. Lander, E. (1999) Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring, Science, 286:531-537.

[4] Gryaznov, D. Scanners of the Year 2000: Heuristics, Proceedings of the 5th International Virus Bulletin, 1999

[5] Henchiri, O. and Japkowicz, N., A Feature Selection and Evaluation Scheme for Computer Virus Detection. Proceedings of ICDM-2006: 891-895, Hong Kong, 2006.

[6] Kolter, J.Z. and Maloof, M.A. (2004) Learning to detect malicious executables in the wild, in Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 470–478. New York, NY: ACM Press.

[7] Kolter J., and Maloof M. (2006) Learning to Detect and Classify Malicious Executables in the Wild, Journal of Machine Learning Research 7 2721-2744.

[8] Mitchell T. (1997) Machine Learning, McGraw-Hill.

[9] Moskovitch R, Stopel D, Feher C, Nissim N and Elovici Y,(2008) Unknown Malcode Detection via Text Categorization and the Imbalance Problem, IEEE Intelligence and Security Informatics (ISI08), Taiwan, 2008.

[10] Quinlan, J.R. (1993) C4.5: programs for machine learning. Morgan Kaufmann Publishers, Inc., San Francisco, CA, USA.

[11] Schultz, M., Eskin, E., Zadok, E., and Stolfo, S. (2001) Data mining methods for detection of new malicious executables, in Proceedings of the IEEE Symposium on Security and Privacy, 2001, pp. 178-184.

[12] Shin, S. Jung, J., Balakrishnan, H. (2006) Malware Prevalence in the KaZaA File-Sharing Network, *Internet Measurement Conference (IMC)*, Brazil, October

[13] Witten, I.H., and Frank, E. (2005) Data Mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann Publishers, Inc., San Francisco, CA, USA.