International Journal of Digital Communication and Analog Signals
Vol. 1: Issue 2
www.journalspub.com

A Literature Review of Intelligence Intrusion Detection Prevention Techniques for Unknown Malware Finding

S. Murugan [*] ***and K. Kuppusamy***
Department of Computer Science and Engineering, Alagapa University, Karaikudi, Tamil Nadu, India

Abstract

Intrusion detection system (IDS) has played a vital role as a device to guard our networks from unknown malware attacks. However, since it still suffers from detecting an unknown attack, the ultimate challenge in intrusion detection field is how we can precisely identify such an attack. For identifying known malware various tools are available but that does not detect Unknown malware exactly. It will vary according to connectivity and using tools and finding strategies what they used. Anyhow like known Malware few of unknown malware listed according to their abnormal activities and changes in the system. This paper will analyze the various unknown malware activities while networking, internet or remote connection.

Keywords: *malware attack, unknown malware attack, IIDS, known vs unknown, unknown threat*

[*]***Corresponding Author***
E-mail: murugan.sethu@gmail.com

INTRODUCTION

This paper surveys proposed solutions for the problem of Unknown Malware attack appearing in the computer security research literature. We distinguish between Known and Unknown as two distinct cases of attack. After describing the challenges of this problem and highlighting current approaches and techniques pursued by the research community for insider attack detection, we suggest directions for future research.[1]

Recent news articles have reported that every year to year time to time an enormous increase of known and unknown malware variants. This has made it even more difficult for the anti-malware vendors to maintain protection against the vast amount of Unknown threats. Various obfuscation techniques, such as reverse engineering, honeypot and intelligence intrusion detection prevention, contribute to this trend.[2–4] The ongoing battle between malware creators and anti-virus vendors causes an increasing signature, which leads to vulnerable end-systems for home users as well as in corporate environments. One of the major and serious threats on the Internet today is malicious software, often referred to as a malware. The malwares like polymorphic and metamorphic designed by attackers which have the capability of varying their code as they transmit. The communicative patterns acquired either statically or dynamically can be oppressed to identify and classify unknown malwares into their known families using machine learning methods.

IIDPS based on specific AI approach for unknown malware finding. The techniques that are being investigated includes artificial immune system and fuzzy logic with neural network profiling and

evolutionary computational methods, that uses simple data mining techniques to process the network data. Traditional signature-based anti-virus system fail to detect polymorphic or metamorphic and new, previously unknown malware. Any non- signature malware detection technology is only as effective as the size of the data set it processes. By leveraging its unique position at the Internet level. A deeper understanding of intrusion prevention and detection principles with intelligence may be responsible for acquiring, implementing or monitoring such systems in understanding the technology and strategies available.[5]

The malwares are unceasingly increasing in volume (growing threat landscape), diversity (innovative malicious methods) and velocity (fluidity of threats). These are developing, becoming more refined and utilizing novel ways to aim computers and mobile devices. McAfee catalogs over 100,000 new malware samples every day means about 69 new threats every minute or about one threat per second. The advanced malwares are targeted, unknown, stealthy, personalized and zero day as compared to the traditional malwares which were broad, known, open and one time.[6-11] Once inside, they hide, replicate and disable host protections. After installation, the calling of their command and control servers for further instructions is done which could be to steal data, infect other machines, and allow inspection. Attackers exploit vulnerabilities in web services, browsers and operating systems, or use social engineering techniques to make users run the malicious code in order to spread malwares. Features derived from analysis of malware can be used to group unknown malwares and classify them into their existing families. Here presents a review of techniques/approaches for analyzing and classifying the malware unknown Malware. Before creating the signatures for newly arrived malwares, these are required to be analyzed so as to understand the associated risks and intensions. The malicious program and its capabilities can be observed either by examining its code or by executing it in safe environment.[12-16]

DYNAMIC ANALYSIS

It discloses the malwares' natural behavior which is more resilient to static analysis. However, it is time intensive and resource consuming, thus elevating the scalability issues. The virtual environment in which malwares are executed is different from the real one and the malwares may perform in different ways resulting in artificial behavior rather than the exact one.[17-21] In addition to this, sometimes the malware behavior is triggered only under certain conditions (on specific system date or via a specific command) and can't be detected in virtual environment. Several online automated tools exist for dynamic analysis of malwares, e.g. Norman Sandbox, CW Sandbox, Anubis and TT Analyzer, Ether and Threat Expert. The analysis reports generated by these tools give in-depth understanding of the malware behavior and valuable insight into the actions performed by them. The analysis system is required to have an appropriate representation for malwares, which are then used for classification either based on similarity measure or feature vectors.

MACHINE LEARNING FOR DETECTING AND CLASSIFYING MALWARES

A literature review is discussed in this section. Schultz *et al*.[19] were the first to make known to the concept of data mining for detection of malwares. They custom three dissimilar static features for malware cataloging: Portable Executable (PE), strings and byte sequences. In the PE approach, the features are extracted from DLL information inside PE files. Strings are extracted from the executables based on the text strings that are encoded in program files. The byte sequence approach

uses sequences of n bytes extracted from an executable file. They used a data set consisted of 4266 files including 3265 malicious and 1001 benign programs. Later on improvement of results was done by Kolter *et al.*[6] They used n-gram as a replacement of non-overlapping byte sequence and data mining technique for detection of malicious executables. They used different classifiers including Naive-Bayes, Support Vector Machine, Decision Tree and their boosted versions. They concluded that boosted decision tree gives the best classification results.

Kong *et al.*[7] proposed a framework for automated malware cataloging based on structural information of malwares with a collective learning approach to detect unknown malwares. It is a type of semi-supervised learning that presents the method for optimizing the classification of partially-labeled data. Collective classification algorithms are used to build different machine learning classifiers using a set of labeled and un-labelled instances. It is validated that the labeling efforts are lower than when supervised learning is used while maintaining the high accuracy rate.

Zolkipli *et al.*[22] presented an approach for malware behavior analysis. Clustering is used to identify the novel classes of malware with similar behavior. Assigning unknown malware to these discovered classes is done by classification. Based on both, clustering and classification, an incremental approach is used for behavior-based analysis, capable of processing the behavior of thousands of malware binaries on daily basis.

LSH can be used to perform an approximate clustering while computing only a small fraction of the *n*2/2 distances between pairs of points. The authors demonstrate the scalability of their approach by clustering a set of 75,000 malware samples in three hours.

Firdausi *et al.*[5] presented a proof of concept of a malware detection method. Firstly the behavior of malware samples is examined in sandbox environment using Anubis. The preprocessed reports were generated into sparse vector models for sorting using machine learning. Network traces were used as input to the framework in the form of pcap files from which the network flows are mined.

Lee *et al.*[8,9] proposed a method that clusters the malicious programs by using machine learning method. All the samples of data set are executed in a virtual environment and system calls along with their arguments are monitored. A behavioral outline is formed on the base of information recorded concerning sample's interaction with system resources like registry keys, writing files and network activities. After completing the training process, the new and unknown samples are assigned to the cluster having method closer to the sample *i.e.* nearest neighbor.

It is distinctive that a single view either static or dynamic is not sufficient for classification of malicious programs efficiently and accurately because of the complication and execution-stalling techniques. So, researches have improved a hybrid technique which integrates both static and dynamic features concurrently for better malware detection and classification.

Santos *et al.*[17,18] proposed a hybrid unknown malware detector called OPEM, which utilizes a set of features obtained from both static and dynamic analysis of malicious code. The static features are obtained by modeling an executable as a sequence of operational codes and dynamic features are acquired by monitoring system calls, operations and

raised exceptions. The approach is then validated over two different data sets by considering different learning algorithms for classifiers Decision Tree, K-nearest neighbor, Bayesian network, and Support Vector Machine and it has been found that this hybrid approach enhances the performance of both approaches when run separately. The machine-learning technologies that are being used in detecting and classifying malwares are not adequate to handle challenges arising from the huge amount of dynamic and severely imbalanced network data. These should be transformed so that their potential can be leveraged to address the challenges posed in cyber security.

ZASMIN

The false rate of the detection methods which are based on abnormal traffic behavior is a little high and the accuracy of the signature generation is relatively low. Moreover, it is not suitable to detect exploits and generate its signature. ZASMIN provides early warning at the moment the attacks start to spread on the network and to block the spread of the cyber-attacks by automatically generating a signature that could be used by the network security appliance such as IPS. . Even if these vulnerabilities which the attacks used were released long time ago, these kinds of attacks still exist in the public domain with polymorphic form. Through this case study has convinced that new attack or polymorphic Authorized known attack can be detected by the ZASMIN system. It's hard to evaluate the exact system-level false positive rate in the real environment, but we can say that the ZASMIN system has relatively low false rate with this case study. And also need to focus on reducing its false rate as the further study.

Even if two-day analysis is not enough long to detect various unknown attacks, researcher could find some attacks without any well-known signature through the case study. Even if these vulnerabilities which the attacks used were released long time ago, these kinds of attacks still exist in the public domain with polymorphic form. Through this case study, researcher have convinced that new attack or polymorphic known attack can be detected by the ZASMIN system.

MalTRAK

MalTRAK, a framework for tracking and eliminating known and unknown malware, allows the user to run any program without requiring policies or rules to be places a priori, while guaranteeing the capability of restoring the system to a clean state in case of an infection. Furthermore, it does so with minimal runtime overhead and by minimizing the amount of clean data lost during disinfection.

The framework achieves these goals by establishing different logical views of the system during runtime and by maintaining a relationship between the views depending upon the system operations. It can then switch to a clean system state upon infection by switching to the appropriate view before the infection took place.

The framework monitors system operations at the lowest possible level ensuring that it is very difficult (almost impossible) to bypass. Implemented MalTRAK on Windows and tested our prototype on 8 real world malware and compared it with two popular commercial antivirus tools. With minimal overhead (both disk space and runtime latency) were able to completely remove their effects on the system while the commercial tools, on an average were only able to restore 36% of all their effects put together. Among one of the malware samples, the commercial tools could only sense it but incapable of repairing any of its damage. Additionally, for two of the malware samples, the commercial tools were totally

incompetent to identify or restore any of their belongings.

MALWARE FORENSICS– DETECTING THE UNKNOWN

The increasing speed of new malware strains being written and released means that security professionals are more likely than ever before to see new malware. This means new malware which is not detected by the anti-malware solutions they have deployed in their infrastructure, be it workstation, server, PDA or at the gateway. Imagine this scenario: An end-user calls the helpdesk and reports that their system is running very sluggishly when it wasn't a week ago and that they can't access the Windows 'Task Manager' or open a command prompt any more. The virus scanner is right up to date and active, and it says the system is clean; the personal firewall is active too. It will focus on a step by step approach of what tools to use, what to look for and what to do with any suspicious files. It will also discuss the use of forensic tools in such a scenario, as a last port of call.[5,6]

As with other security threat, especially malware related ones, deploy a multi-layered approach to minimize the chance of malware getting onto your computers. This means not only do you need good technological solutions, and overlapping technologies at that, but these need to be backed up with good security policies, procedures, education and constant vigilance.

MALCIOUS EXECUTABLE APPROACH

Active Learning

The concept of detecting unknown computer worms based on a host behavior,[7–9] using the SVM classification algorithm based on several kernels. Based on the results shown in this study, the use of support vector machines in the task of detecting unknown computer worms is possible. A feature-selection method which enabled to identify the most important computer features in order to detect unknown worm activity, currently performed by human experts. Based on the initial experiment (*e1*), the Gain Ratio feature selection measure was most suitable to this task. On average the *Top20* features produced the highest results and the RBF kernel commonly outperformed other kernels. In the detection of unknown worms (*e2*), the results show that it is possible to achieve a high level of accuracy (exceeding 80% on average); as more worms were included in the training set the accuracy improved. To reduce the noise in the training set and improve the learning researcher argued that the use of the active learning approach as a selective method would improve the performance, which actually happened, increasing the accuracy after selecting 50 examples to above 90% accuracy and 94% when the training set contained four worms. When selected 100 and 150 examples no improvement was observed above the performance after selecting 50 examples.
These results are highly encouraging and show that unknown worms, which commonly spread intensively, can be stopped from propagating in real time. The advantage of the suggested approach is the automatic acquisition and maintenance of knowledge, based on inductive learning. Currently in the process of extending the amount of worms in the dataset, as well as extending the suggested approach to other types of malicious code using temporal data mining.

Collective Classification

The obtained results validate our initial hypothesis that building an unknown malware detector based on collective classification is feasible. The classifiers achieved high performance in classifying unknown malware, improving our

previous results using LLGC (Santos et al., 2011),[17,18] which achieved an 86% of accuracy in its best configuration. Therefore, researcher believe that our results will have a strong impact in the area of unknown malware detection, which usually relies on supervised machine learning (Schultz et al., 2001; Kolter and Maloof, 2004).[17,6] Training the model through supervised machine-learning algorithms can be a problem itself because supervised learning requires each instance in the dataset to be properly labeled. This demands a large amount of time and resources. In this way, researcher tried to find among our results the number of labeled malware that is needed to assure a certain performance in unknown malware detection.

Classification Technique on Op–Code patterns

In earlier studies sorting algorithms were engaged successfully for the finding unknown malicious code. Further most of these studies mined features based on byte n-gram patterns for representing the inspected files. In this study researcher signify the inspected files using Op–Code n-gram patterns which are extracted from the files after disassembly. The Op–Code n-gram patterns were used for the classification process as features. The main goal of classification process is to identify unknown malware among the set of suspected files which will later be included in antivirus software as signatures. A laborious assessment was accomplished using a test collection comprising of more than 30,000 files, in which several settings of Op–Code n-gram patterns of numerous size illustrations and eight types of classifiers were assessed.

In this study researcher used Op–Code n-gram patterns generated by disassembling the inspected executable files to extract features from the inspected files. Op–Code n-grams are used as features during the classification process with the aim of identifying unknown malicious code. Researcher performed an extensive evaluation using a test collection comprising more than 30,000 files.

Comparative Analysis

The following table shows the comparative analysis of IIDPS with CPC-CPS models and other Intrusion Detection Systems and Techniques reviewed in the literature survey in predicting and classifying the Unknown malware in the network.

Table 1. *Comparative Analysis of IIDPS and IDS.*

Unknown Malware detection	**TP**	**FP**	**DR**	**Class**
KNN	0.948	0.051	0.95	Malware
ANN	0134	0.033	0.80	Malware
NB	0.069	0.382	0.15	Malware
NN Back Propagation	0.864	0.183	0.83	Malware
SVM Normalized (poly kernel)	0.986	0.025	0.98	Malware
DT	0.90	0.10	0.90	Malware
Voted Perceptron	0.95	0.05	0.95	Malware
ZASMIN	0.94	0.023	0.98	Malware
Data mining using PE	0.99	0.01	0.99	Malware
CPS model	0.998	0.00	0.997	Malware
ICPC model	1	Undefined	1	Malware
II CPC model	0.991	0.07	0.9971	Malware

As shown in Table 1 the detection accuracy of CPS and CPC classifier outperforms the rest of the data mining classifiers in most of the cases. From the table it is observed that the classifiers using the ANN, NN feature had the lowest

detection rate. Further, Naïve-Bayes gives the worst detection accuracy in most cases. The classifiers using CPC and CPS models give better detection accuracy than the classifiers using other system features.

CONCLUSION

This paper describes in depth many of the popular Computational Intelligence techniques found in malware detection research. Several existing intelligence techniques show promise in the malware detection problem. Many of the machine learning techniques has application to both continuous and discrete datasets. The results obtained from IIDPS are compared with existing IDS in the literature and are tabulated with implemented method. As the result it is concluded that the proposed IIDPS produces good results in worm detection and produces perfect result with accuracy of 99.96% in detecting the presence of worm in the network even for unknown worms.

REFERENCES

1. Anderson. B., Storlie C., Lane T. Improving Malware Classification Bridging the Static/Dynamic Gap. *Proceedings of 5th ACM Workshop on Security and Artificial Intelligence (AISec)*. 2012; 3–14p.
2. Anderson. D., Lunt H., Javitz A. *et al.* Safeguard Final Report: Detecting Unusual Program Behavior Using the NIDES Statistical Component. *Computer Science Laboratory*. SRI International. Menlo Park, CA, Technical Report. 1993.
3. Bayer U., Moser A., Kruegel C. *et al.* Dynamic Analysis of Malicious Code. *Journal in Computer Virology*. 2006; 2: 67–77p.
4. Biley M., Oberheid J., Andersen J. *et al.* Automated Classification and Analysis of Internet Malware. *Proceedings of the 10th International Conference on Recent Advances in Intrusion Detection*. 2007; 4637: 178–197p.
5. Firdausi I., Lim C., Erwin A. Analysis of Machine Learning Techniques Used in Behavior Based Malware Detection. *Proceedings of 2nd International Conference on Advances in Computing, Control and Telecommunication Technologies (ACT)*. Jakarta; 2010 December 2-3: 201–3p.
6. Kolter J. Maloof M. Learning to Detect Malicious Executable in the Wild. *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2004; 470–8p.
7. Kong D., Yan G. Discriminant Malware Distance Learning on Structural Information for Automated Malware Classification. *Proceedings of the ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems*. 2013; 347–8p.
8. Lee W., Stolfo S.J. Data Mining approaches for intrusion detection. *In Proc. Seventh USENIX Security Symposium*. San Antonio; TX: 1998.
9. Lee W., Stolfo S.J., Kwok K.W, Mining audit data to build intrusion detection models. *In Proc. Fourth International Conference on Knowledge Discovery and Data Mining*. New York; 1998.
10. Moser A, Kruegel C., Kirda E. Exploring multiple execution paths for malware analysis. *In Proceedings of the 2007 IEEE Symposium on Security and Privacy*. 2007.
11. Moser A., Kruegel C. Kirda E. Limits of static analysis for malware detection. *In Proceedings of the 23rd Annual Computer Security Applications Conference (ACSAC)*. 2007.

12. Nataraj L., Karthikeyan S., Jacob G. Malware Images: Visualization and Automatic Classification. *Proceedings of the 8th International Symposium on Visualization for Cyber Security*. 2011; 4.
13. Nataraj L., Yegneswaran V., Porras P. *et al.* A comparative assessment of malware classification using binary texture analysis and dynamic analysis. *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence*. 2011; 21–30p.
14. Ning P., Cui Y., Reeves D.S. Constructing Attack Scenarios through Correlation of Intrusion Alerts. *Proc. ACM Computer and Communications Security Conf*. 2002.
15. Park Y., Reeves D., Mulukutla V. *et al.* Fast Malware Classification by Automated Behavioral Graph Matching. *Proceedings of the 6th Annual Workshop on Cyber Security and Information Intelligence Research*. 2010; 45.
16. Rieck K., Trinius P, Willems C. and Holz T. *Automatic Analysis of Malware Behavior Using Machine Learning. Journal of Computer Security*. 2011; 19: 639–68p.
17. Santos I., Devesa J., Brezo F. *et al.* A Static-Dynamic Approach for Machine Learning Based Malware Detection. *Proceedings of International Conference CISIS'12-ICEUTE'12, Special Sessions Advances in Intelligent Systems and Computing.* 2013; 189: 271–80p.
18. Santos I., Nieves J. Bringas P.G. Collective Classification for Unknown Malware Detection. *Proceedings of the International Conference on Security and Cryptography.* Seville, 18–21 July 2011, 251–6p.
19. Schultz M., Eskin E., Zadok E. *et al.* Data Mining Methods for Detection of New Malicious Executables, *Proceedings of the IEEE Symposium on Security and Privacy*. 2001; 178–84p.
20. Siddiqui M., Wang M.C., Lee J. Detecting Internet Worms Using Data Mining Techniques. *Journal of Systemic Cybernetics and Informatics*. 2009; 6: 48–53p.
21. Tian R., Batten L., Versteeg S. Function length as a tool for malware classification. *Proceedings of the 3rd International Conference on Malicious and Unwanted Software*. Fairfax. 2008 October 7–8. 57–64p.
22. Zolkipli M.F., Jantan A. An approach for malware behavior identification and classification. *Proceeding of 3rd International Conference on Computer Research and Development*. Shanghai. 2011 March 11–13. 191–4p.