Automated Malware Detection Based on Novel Network Behavioral Signatures

Maros Barabas, Ivan Homoliak, Michal Drozd, and Petr Hanacek

Abstract—**In this paper we introduce the second generation of the experimental detection framework of AIPS system which is used for experimentation with detection models and with their combinations. Our research aims mainly on detection of attacks that abuse vulnerabilities of buffer overflow type, but the final goal is to extend detection techniques to cover various types of vulnerabilities. This article describes the concept of detection framework, updated set of network metrics, provides a design of model architecture and shows an experimental results with draft of framework on the set of laboratory simulated attacks.**

Index Terms—**Artificial intelligence, behavioral signatures, metrics, network security, security, security design.**

I. Introduction

During the last decade, anomaly detection has attracted the attention of many researchers to overcome the weakness of signature-based IDSs in detecting novel attacks, and 1999 KDD Cup is the mostly widely used dataset for the evaluation of these systems [1]. Although some intrusion experts believe that most novel attacks are variants of known attacks and the signature of known attacks can be sufficient to catch novel variants [2], the results of the 1999 KDD Cup “Classifier Learning Contest” shows the opposite [3]. Results of buffer overflow mining methods using the 1999 KDD Cup dataset are completely unsatisfactory even using most contemporary methods [4], [1], [5], [6]. Thus, we have decided to create a more representative dataset, similarly to the HoAH project [7], but at different levels of details.

In the previous article [8] we proposed an idea of framework architecture that would be used for detection of various network threats. The paper presented the novel Automated Intrusion Prevention System (AIPS) which uses honeypot systems for the detection of new attacks and the automatic generation of behavioral signatures based on network flow metrics. We have successfully experimented with the architecture of the AIPS system and we defined 112 metrics divided into five categories according to their nature. These metrics are used to describe properties of detected attack not upon the fingerprint of common signature, but based on its behavior.

During the experiments we found several limitations of the original idea and some parts of the architecture were changed. We extended the metric dataset to 169 metrics containing approximately 4000 parameters and changed the categories to reflect the nature of the new dataset. The main goals of this research is (a) to design the architecture of detection framework that will enhance the overall network security level with the ability to learn new behaviors of attacks without intervention of human by using the expert knowledge from Honeypot (or similar) systems; (b) to find the most suitable set of metrics that will successfully describe the behavior of attacks in the network traffic and will significantly higher the detection rate and lower the false positive rate.

In this article we introduce the second generation of the experimental detection framework of AIPS system which is used for experimentation with detection models and with their combinations. The fundamental principle of the detection is based on evaluation of metrics set, which describes the behavior of attack. These metrics are formally specified and extraction of them can be generally realized for each data flow. We could interpret the specification of metrics set as formally extended protocol *NetFlow* [9], which describes more than statistical properties of network communication. The metrics specification includes statistic, dynamic, localization and especially behavioral properties of network communication.

The paper is organized as follows. We describe the new idea of the framework architecture in Section 2. The novel network behavioral signatures we use for detection are briefly discussed in Section 3 and our experiments with framework are provided in Section 4. We give an overview of some limitations and challenges for future work in section 5 and concluded the paper with a summary of our work in Section 6.

II. AIPS Network Architecture

The schema in Fig. 1 includes an AIPS Network Detector (AIPS ND) working as a network probe capable of detecting intrusions using a knowledge base from the AIPS Attack Processor (AIPS AP). Further, the schema includes the Intrusion Detection and Prevention system (IDPS) for a real-time detection/prevention of attacks and a database (DB) for storing data and signatures (not included in the scheme). The last (optional) part of the AIPS architecture is a group

Manuscript received January 10, 2013; revised March 15, 2013. This article was created within the research plan of the Security-Oriented Research in Information (MSM0021630528). This work was supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070), by the project Advanced Secured, Reliable and Adaptive IT (FIT-S-11-1) and it has been realized with a financial support from the Czech Republic state budget through the Ministry of Industry and Trade by the research plan FR-TI1/037.

Authors are all with Faculty of Information Technology, Brno University of Technology, Czech republic (e-mail: ibarabas@fit.vutbr.cz, ihomoliak@fit.vutbr.cz, idrozd@fit.vutbr.cz, hanacek@fit.vutbr.cz).

DOI: 10.7763/IJET.2013.V5.552

of highly interactive Honeypot systems which are used to create expert knowledge of detected attacks. The knowledge is sent to AIPS AP where the knowledge base is concentrated. The AIPS AP is also responsible for learning the artificial intelligence of the AIPS ND.

The AIPS ND works as a network probe capable of detecting intrusions using a knowledge base from the AIPS AP. The mirrored traffic (from a backbone router/firewall) is captured by a *tcpdump* probe and separated to individual flows. From the pre-processed traffic flow all individual connections are extracted and further processed to create a signature vector by predefined metrics.

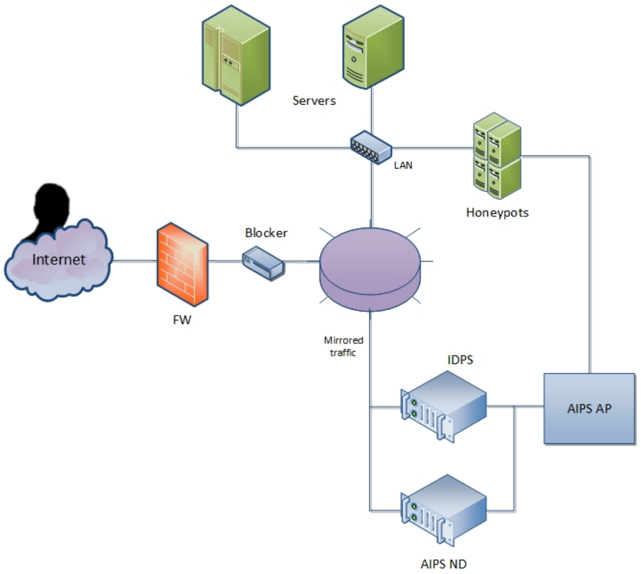


Fig. 1 AIPS Network Architecture

The architecture of framework is designed by modular principles to allow the flexibility of exchanging or enhancing each part of framework model to cover all potential use cases. The part with deployed honeypots is designed to be either the regular instance of the honeypot system or a part of real operating system within a DMZ (demilitarized zone) as a service. We have experimented with Windows XP, Windows 2000 and Linux systems, all with successful deploy and successful detection of tested attacks. For interaction between the parts of framework we use PostgreSQL database. The use of database also fastened the processes working with high amount of data.

AIPS Use Case

In the first step the group of highly interactive Honeypots is used for attraction of an attacker to attack a vulnerable service. After an attacker exploits the vulnerability on the Honeypot system, all information about the attack vector (virtual address space, registers and traffic) are sent to the AIPS AP. The malicious network traffic is divided to separate flows further preprocessed to create the set of values representing set of behavioral metrics (described in the next chapter). This set of values is a Behavioral Signature used for description of the network behavior of analyzed attack. Each time a new attack is detected, the new behavioral signature is created (with expert knowledge provided by honeypot) and the model is updated as part of the learning process. The AIPS ND engine with the network node is processing the traffic by learned model and detecting known attacks by comparison of traffic parameters with the behavioral signatures identified with reasonable confidence (e.g., exceed the threshold of maliciousness decision), or by expert knowledge provided by shadow honeypot. The expert knowledge confidence of maliciousness decision is based on characteristics of shadow honeypot systems which are optimized for detection attacks abusing the vulnerabilities of buffer overflow type based on taint analysis [10].

An attack can be detected using technique of tainting the memory of each process and when the process uses the memory that shouldn't be normally accessible, honeypot claims the process malicious. We have reprogrammed the Argos [11] shadow honeypot to store all crucial data of detected attack into a database. Our system depends on expert knowledge of honeypot systems and the confidence that each detection alert raised from honeypot signifies an attack. This part of our concept provides a room for more enhancements of any form of analysis that can provide sufficient confidentiality to detection alerts. However, these enhancements are reliant on real-time and performance limitations of network traffic analysis to maintain the possibility to stop an attacker on next defense perimeter of IPS systems and firewalls. Actually we experiment with more detection techniques and we try to enhance the process of taint analysis and honeypot self-defense mechanisms to avoid potential compromise of these systems.

The signatures created by AIPS are specific and unique for their behavioral nature. Each signature is a vector composed of dozens of numbers, each corresponding to a value of a specific metric. Each metric is a characterization of the network flow and could be specified as an extension of the *NetFlow* protocol describing not only statistical properties of the network flow, but it also includes dynamic, localization and behavioral specifics. These metrics are briefly described in the next chapter.

III. Network Behavioral Signatures

In our previous article [8] we introduced 112 metrics (ASNM) which are able to describe properties of attacks and legitimate communications based on their behavior. These metrics suppose source data which are directly extracted from network traffic. These metrics are mostly simple network parameters excluding packets content, which can be in the most cases encrypted. We extended the previous metric set by adding new metrics to approximate communication progress in time by Gaussian curves and other goniometric functions (Fourier series). We also added new packets distribution metrics and other behavioral metrics. We also change the scale of all time-parameterized metrics. Now we use the scale by power of two because of better performance of used data-mining tools. The exact interpretation of this step is behind scope of this paper. Actual count of all proposed metrics together with the new dataset is 169. These metrics are in many cases results of convenient parameterization. The definition of metrics in current set will be published in a future article.

Metrics extraction process considers communications data stored in *libpcap format*[1]. For each TCP flow we extract all metrics in correct order within each category and also in correct order among categories. The order is necessary due to the fact some metrics depend on other metrics. Metrics extraction dependability is used in maximum amount, because of fast extraction. We will have to consider it in the phase of extraction parallelization and hardware acceleration process and some dependencies may change.

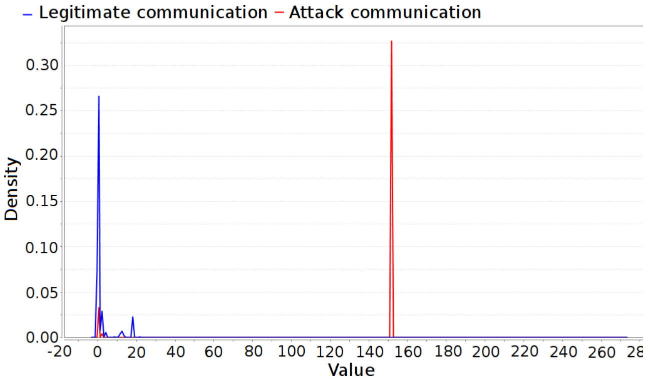


Fig. 2. Value density distribution for standard deviation of time differences between arrived packets.

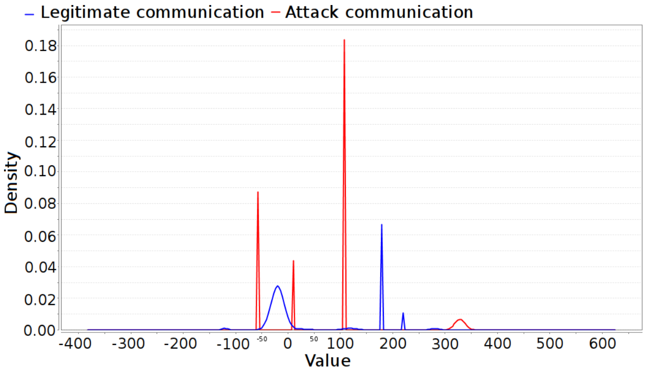


Fig. 3. Value density distribution for one of polynom approximation metric

In our demonstration we simulated buffer overflow attacks with *Metasploit framework*. Captured and preprocessed data were analyzed in knowledge mining application *RapidMiner*. We explored potential of individual metrics by creation value density distribution graphs. In the Fig. 3 Fig. 3is depicted value distribution for metric coefficient of third order polynomial which approximates output communication in output direction from the side of attacked machine. In the Fig. 2 Fig. 2we depict simple metric - standard deviation of time differences between two consecutive packets in the input direction from the side of attacked machine. At both figures we can see value differences between attack behavior and legitimate communication behavior.

The next step of our analysis phase was experiments with classification method optimization. For finding optimal parameters of each classification method we used grid combination components of mining tool. We found rough values at first then we tried to optimize them by lower scales. We have also experimented with data preprocessing phase: we used discretization of ordinal attributes and principal component analysis method for finding principal attributes. Results of these experiments are showed in Table ITABLE , where the methods are horizontally ordered by classification accuracy.

TABLE I: SUMMARY RESULT OF USED CLASSIFICATION METHODS

Classification method	SVM with radial kernel	Decision tree with gini index criterium of attribute selection	Naive Bayess classificator and PCA with automatic count of components	Naive Bayess classificator with discretization of ordinal attributes	Naive Bayess classificator	Naive Bayess classificator and PCA with fixed count of components
recall	41.67%	25.00%	16.67%	25.00%	8.33%	8.33%
specificity	96.09%	94.97%	94.48%	94.97%	96.13%	96.73%
presicion	41.67%	25.00%	16.67%	25.00%	14.29%	16.67%
accuracy	89.85%	87.82%	87.82%	87.82%	76.14%	75.63%

From the perspective of classification accuracy, we achieved the best results in the case of SVM method. We also compared classification methods by ROC method, however, best results were achieved by SVM method (with neutral ROC bias) and by the decision tree (with optimistic ROC bias).

In the Fig. 4 we can see ROC diagram for neutral ROC bias and in Fig. 5 for optimistic ROC bias.

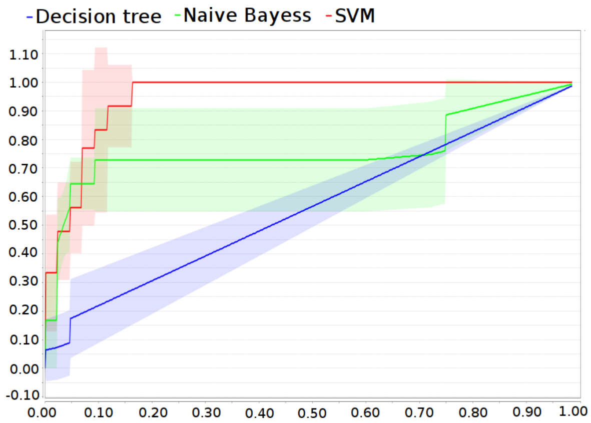


Fig. 4. Diagram for neutral ROC bias

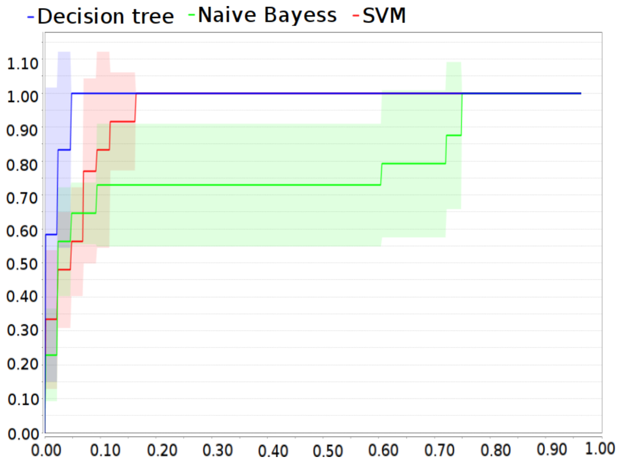


Fig. 5. Diagram for optimistic ROC bias

[1] basic format to save captured network data used by tcpdump and similar tools

All experiments were performed in laboratory conditions; therefor there can be some differences from real environment. Laboratory conditions of experiments may differ mainly in context-dependent metrics, where the context was generated only by two laboratory hosts (the attack machine and the vulnerable machine). The second set of metrics depends on transmition time of packets in the analyzed traffic. In a real traffic more nodes are present within the route between the attacker and the detector, and this path can be dynamically selected according to actual network conditions, but in laboratory conditions these parameters are constant. Other influence relates on errorness of communication channel and therefore with TCP retrasmition of packets.

IV. Future Work

There are some limitations we are aware of in a time we are writing this article. We experiment with several options that could bring the solutions.

The first limitation is the dependency on expert knowledge of honeypot that can be compromised or can produce false positives by nature of buffer overflow vulnerabilities that could be triggered accidentally with no intent to attack. However, honeypot systems we use are shadow instances of virtualized operating systems that have no legitimate traffic and all connections from outside are a priori considered malicious. The ability of successful detection of buffer overflow attacks and possibility of compromising such heavily exposed systems remain the challenge for future work. We identified several other limitations based on the nature of Honeypot systems:

- Ability to detect only buffer overflow based vulnerabilities.
- Time-consuming system and traffic analysis.
- Delay between the attack detection and its blocking.
- All production systems are vulnerable before one of the deployed honeypot systems is attacked.
- The necessity of deployment of honeypot systems in the production environment.
- Difficult simulation of ARM instructions in taint analysis.

Our future work is aimed on finding solutions for previously outlined limitations and problems. We have currently 6 Honeypot systems deployed on the real network collecting the data which are crucial for further research. For our next step we have prepared experiments with the dataset of metrics and various data-mining techniques to optimize the dataset. We also plan to focus on other types of attacks, such as Denial of Service, Remote Access Trojans communication, etc. and optimize the process between the detection and blocking the attack.

V. Conclusion

In this paper we updated the current status of AIPS detection model aimed on detection new previously unknown buffer overflow attacks. It focuses on detection results by using chosen data-mining techniques used for attack recognition. The detection framework was described in terms of fundamental principles together with detection metrics that describe the behavior of a network traffic from different aspects. Emphasis was placed on the presentation of results in the detection of cross validations of training and testing set. The testing set includes the metrics generated from the captured attacks and valid communication.

In the result we were able to capture unknown attacks with a 96% success rate using all set of metrics. If the time consumption was optimized to a minimum of mathematical operations, it was not possible to perform the analysis in real time. When we used metrics with maximum entropy (45 parameters) it was possible to maintain the quality of detection and the classification was done in real time.

The detection method using behavioral signatures have been proven to detect unknown attacks, but the efficiency of the detection was tested only on a small number of attacks. In the near future, we plan to create a public detection set that would create a challenge in the development of detection algorithms to detect unknown attacks.

VI. References

[1] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” in *Proceedings of the Second IEEE international conference on Computational intelligence for security and defense applications*, pp. 53–58, 2009.

[2] S. Hettich and S. D. Bay, “KDD Cup 1999 Data,” The UCI KD Archive, Irvine, CA: University of California, Department of Information and Computer Science, 1999.

[3] C. Elkan, “Results of the KDD'99 classifier learning,” *ACM SIGKDD Explorations Newsletter,* roč. 1, pp. 63–64, Led. 2000.

[4] R. G. M. Helali, “Data Mining Based Network Intrusion Detection System: A Survey,” in *Novel Algorithms and Techniques in Telecommunications and Networking*, T. Sobh, K. Elleithy, a A. Mahmood, Ed. Springer Netherlands, 2010, pp. 501-505.

[5] A. A. Ghorbani, W. Lu, M. Tavallaee, A. A. Ghorbani, W. Lu, a M. Tavallaee, “Evaluation Criteria,” in *Network Intrusion Detection and Prevention,* roč. 47, Springer US, 2010, pp. 161-183.

[6] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos. "Feature selection and classification in multiple class datasets: An application to KDD Cup 99 dataset." *Expert Systems with Applications* 38.5 (2011): pp. 5947-5957.

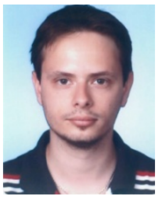
[7] NoAH FP6 EU Project. [Online]. Available: URL http://www.fp6-noah.org/index.html, 2008.

[8] M. Barabas, M. Drozd, and P. Hanáček, “Behavioral signature generation using shadow honeypot,” in *World Academy of Science, Engineering and Technology*, Tokyo, JP, WASET, 2012, pp. 829-833.

[9] NetFlow. Cisco Systems, Inc. [Online]. Available: URL www.cisco.com/go/netflow. 2011.

[10] X. Zhang, L. Zhi, a D. Chen, “A Practical Taint-Based Malware Detection,” *in Proc. Apperceiving Computing and Intelligence Analysis International Conference*, pp. 73 -77, 2008

[11] G. Portokalidis, A. Slowinska, a H. Bos, “Argos: an Emulator for Fingerprinting Zero-Day Attacks,” in *Proc. ACM SIGOPS EUROSYS'2006*, 2006



Maros Barabas obtained his master’s degree from the Faculty of Information Technology, Brno University of Technology in 2009 in the field of computer security. He is currently a Ph. D. student under the supervision of doc. Petr Hanacek. In 2006, he joined the emerging Czech branch of Red Hat, from 2008 he worked in the team aimed at security of Linux systems up to 2011. Since the beginning of 2012, he works as it security consultant at AEC, part of Cleverlance group. In 2009 he joined the Security@FIT security research group at Brno

University of Technology. His research is centered on development of malware detection, computer and network security.

Ivan Homoliak was born at Rimavska Sobota, Slovak republic in July 1987. He obtained his master's degree from Faculty of Information Technology, Brno University of Technology in 2012 in the field of network security and now he is a Ph. D. student under the supervision of doc. Petr Hanacek.

In 2012 he joined the Security@FIT security research group at Brno University of Technology. The current research interests are aimet to network security breach detection by statistic and behavioral flow-based classification.

Michal Drozd Michal Drozd is currently a Ph. D. student under the supervision of doc. Petr Hanacek, his research is focused on advanced malware detection based on network behavior.

Since 2006 he works as a senior it security consultant in AEC with focus on penetration testing, advanced malware incidents and their detection. He is a member of Security@FIT security research group at Brno University of Technology.

Petr Hanacek heads the Department of Intelligent Systems and works as associate professor on Faculty of Information Technology, Brno University of Technology, Czech republic. He leads the security research group Security@FIT, which focuses on research in the field of computer and network security. Dr. Hanacek obtained Master's degree in computer engineering in 1988, Ph. D. in computer science in 1997 and habilitation at Faculty of Information Technology, BUT in 2003.

Since 1987 to 2001, he worked at Department of Computer Science at Faculty of Electrical Engineering and Computer Science, BUT, since 2002 he works at Faculty of Information Technology.

Doc. Hanacek is member of Czech and Slovak Information Society (CIS), Czech & Slovak Simulation Society (CSSS) and member of Special Interest Group on Security, Audit and Control (ACM – SIGSAC).