A CONTROL NODE FOR INTRUSION DETECTION SYSTEMS MANAGEMENT

Liberios VOKOROKOS[*], Michal ENNERT[*], Zuzana DUDLÁKOVÁ[*], Olympia FORTOTIRA[**]
[*]Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics,
Technical University of Košice, Letná 9, 042 00 Košice, Slovak Republic, tel. +421 55 602 4220, e-mail:
liberios.vokorokos@tuke.sk, michal.ennert@tuke.sk, zuzana.dudlakova@tuke.sk
[**]Gymnasio Limnis Evvoias, Limni, P. C. 34005, Greece, e-mail: olydi1212@gmail.com

ABSTRACT

Currently, several systems for Intrusion Detection System (IDS) management exist, however they are suffering from numerous downfalls; the fact they mostly focus on the visualization of gathered data and not on the management itself (which is, in fact, the critical part) being the main one. The goal of this work is to develop a solution for IDS management that would simplify the usage and provide greater efficiency when detecting intrusions, thus providing the overall improvement of the system security. This article concerns about the analysis of current IDS solutions and their management tools, the architecture of our solution and the evaluation of the solution based on this architecture. The expected results improve the efficiency of an IDS system and also of the whole system security itself.

Keywords: *Intrusion detection system, Management, Tool, Intrusion, Security*

1. INTRODUCTION

Increasingly larger amounts of information are being moved from numerous archives to computer servers every day, where they become accessible to a large amount of people connected to the network. The importance and value of these data is growing as well. Naturally, they become a target of increasingly sophisticated attacks of hackers and contemporary methods of their protection are becoming to fall short, especially of computers storing highly sensitive data. Therefore, an additional layer of protection is provided by Intrusion Detection Systems. However, as IDS systems evolved over time by refinement of their intrusion detection methods, new versions, etc., systems for their management had been largely left behind.

Several tools for IDS management exist nowadays, however they suffer from numerous deficiencies. One of their main shortcomings is the fact they focus on reading the log files and on the visualisation of the logged events, though they should be really focusing on the management and the monitoring of IDS's. This state of affairs thus creates a space for improvement.

2. IDS SYSTEMS AND THEIR MANAGEMENT

In this chapter, we present an analysis of IDS systems and their management tools.

2.1. Intrusion Detection Systems

Based on the resources [1] and [2], an intrusion can be considered as an attempt to violate the access rights to a file or to the whole computer system or to violate the integrity of a file. An attempt to overcome the security configuration of a network can also be considered an intrusion. These intrusions can be realized by exploiting a "hole" in the system security. The largest source of security holes are program and operation system errors, which are often created as a by-product of extending their functionality.

According to definitions in [3],[1],[4] and [5], we can define intrusion detection as the process of monitoring events and their analysis in order to find intrusions in an application, computer system or a network.

Authors of [6], [7], [8] and [9] define IDS's as a security software or hardware that automates the process of intrusion detection.

Sources [10] and [11] define the network-based IDS (NIDS) as a system monitoring the overall network traffic on the packet layer by intercepting and evaluating its packets.

Network intrusion detection systems are divided in to two categories. The difference is in the form how they examine the network traffic [12]:

1. System is based on signature in which the previous attacks and system vulnerabilities are recorded.
2. System is based on learned pattern that contains a behaviour of normal system activity to identify active intrusion attempts.

IDS placement depends on the topology of the network and the type of intrusion that should be detected, i.e. internal or external. When pursuing external threats, the IDS are placed in the network, where they monitor traffic be-tween the Internet and a private network. Internal IDS controls communication within the LAN. In some cases it is not necessary to monitor activity across the entire network, but only at a certain critical parts. An example for such part may be a demilitarized zone. Two systems that use signatures for testing the network traffic are Snort and Suricata.

Based on resources [13], [14], [6], [7], the principle of misuse detection (also called signature-based/knowledge-based detection) is based on comparison with patterns of already known attacks. Those are represented by pre-set rules. Main advantage of this method of detection is ability to provide a reliable way to detect attacks that are already known and also to effectively eliminate false-positive cases. On the other hand, the main disadvantage of this method (described in [15], [16] and [9]) is that the protection against new or modified types of attack is

problematic due to non-existent patterns or insufficient rules. For reliable IDS it is critical to have its rules database updated.

As a distributed Intrusion Detection System can be described as multiple IDS stations over a (large) network, all of which communicate with each other, or with a central control node that facilitates advanced network monitoring and analysis [17].

2.2. Tools for IDS management

IDS management tool is a software featuring graphical user interface (GUI). Its primary function is to simplify the work with IDS's. There are several solutions for IDS management, each of them differ in their support of different IDS's and in their capabilities. There are three most used tools - Snorby, IDSCenter a LogSiphon.

Snorby is a web application focused on displaying statistical data about events detected by separate IDS's. One of its main advantages is simple and intuitive web user interface featuring an ability to send reports by e-mail and also an automatic update of rules database. Relatively limited sensor management capabilities can be considered as the main disadvantage.

IDS Center is a front-end program for Snort. Its task is to simplify the configuration and management of Snort IDS. Its main advantages include interactive configuration of Snort, activity logging, simple and intuitive GUI and the possibility to change the IDS configuration directly from the application, while its greatest disadvantage is the support for only one IDS.

Log Siphon is program capable of collecting and analysing events in real time. Among its advantages are: web GUI, report sending by e-mail and real-time event monitoring. Its main disadvantages are missing management and sensor (devices) monitoring options, confusing GUI and the high cost.

None of these solutions offers the possibility of remote IDS management on a sufficient level (an interactive remote IDS configuration for example), which creates the space for improvement.

3. ARCHITECTURE OF THE PRESENTED SOLUTION

The architecture proposed consists of number (at least two) of standalone NIDS stations which use the computing power of the graphics hardware. Cooperation between the NIDS stations is covered by the central control node – mentor. All these stations are connected to a node where the whole network communication is mirrored and sent the same way by parallel into every NIDS station. Whole distributed intrusion detection system from the performance and process side of view is based on the distribution of workload cyber-attacks and interruptions into the elementary NIDS stations. The architecture of our solution (Fig.1) is diversified into three layers according to their main function:

- network traffic interception layer,
- intrusion detection layer,
- synchronization layer.

It is designed for the purpose of supplement the network security statement, which is ensured by systems designed to detect intrusions.

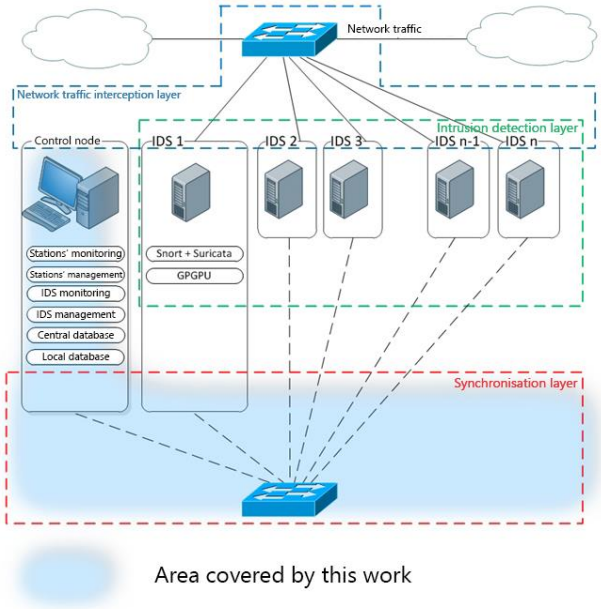

Fig. 1 Proposed architecture

The network traffic interception layer mirrors the traffic over the network using switches for each type of IDS.

Intrusion detection layer detects intrusions using rules. This layer can consist from one or from several IDS's. In the case of the latter, each of them may have:

1. Different set of rules, which enables more effective detection by preventing packet ditching which can occur when IDS is overloaded. This also enables IDS to apply different rules on the same kind of attack, improving its detection.
2. The same set of rules, which guarantees the higher level of security if IDS is shut down either by some failure or by an attack, or when it is restarting after a configuration change.

By combining these features, we can use the advantages of both methods, although hardware costs with this solution would rise as well.

The task of **the synchronisation layer**, consisting of the master node and stations connected to each other by a computer network, is to manage and check the stations, and thus the overall management of the intrusion detection layer.

IDS Snort and Suricata were implemented to such designed model. For the proper functionality of IDS, it was necessary to add program Barnyard2. It cares for reading logs and writing them to the database which is in the control node. For the administration of rules, it was necessary to add program Pulledpork with IDS Snort and program Oinkmaster with IDS Suricata. The whole model has been proposed with the intention of using multiple IDS for intrusion detection, which should lead to the following enhancements:

1. This type of model allows to process large data stream. Each IDS has only a certain set of rules, which minimizes the risk of overloading the IDS.

2. When using rules from different makers on several IDS that process the same data, comparison statement of truth or false can be achieved.

Goal of this proposed architecture is to improve the intrusion detection and the management of IDS stations and thus to improve the overall security of a system.

4. EVALUATION OF SOLUTION AND RESULTS

Testing of our solution was conducted on five devices. Two devices: a communication generator and an attack generator, simulated the network traffic. Intrusion detection was provided by two IDS's, intercepting network traffic by a pair of mirrored ports. Synchronisation layer consisted of master node and the both IDS's. The developed solution was deployed on the master node.

The goal of the testing was to evaluate the efficiency of the proposed architecture, i.e. to confirm if the number of ditched packets could be reduced by dividing the set of rules between several IDS's and also, if their efficiency could be increased using several types of them or their different configurations.

Both IDS's were configured with the same settings. At first, only one IDS was tested with all rules enabled and number of ditched packets was monitored. After the end of this test, around half of the rules were disabled on this IDS and another IDS was added with these rules enabled instead. Then, the number of ditched packets was monitored on this configuration.

Number of ditched packets when testing only one Suricata IDS is displayed in Figure 2.

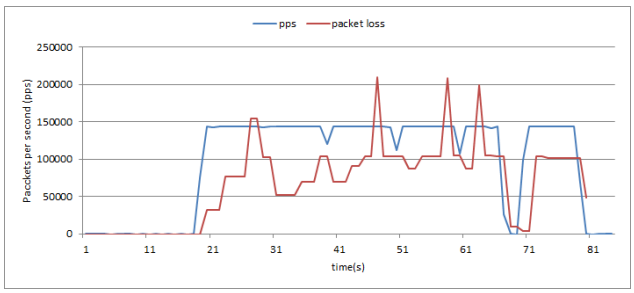

Fig. 2 Overall amount of sent and ditched packets when using one Suricata IDS

Amount of ditched packets when testing two Suricata IDS's is being shown in Figure 3.

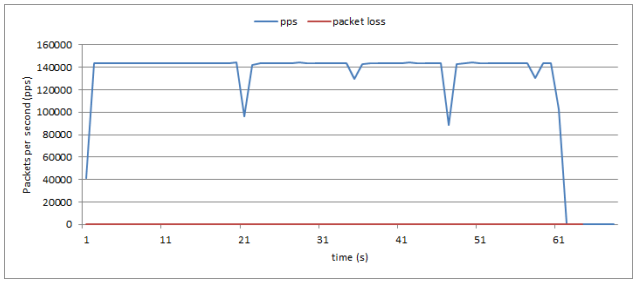

Fig. 3 Overall count of sent and ditched packets when using two Suricata IDS's

Based on the graphs presented in figures 2 and 3, we can state that in the case of dividing the rules between two IDS we achieved the minimum number of ditched packets and thus the overall system security was improved.

Snort IDS was tested with the same conditions and achieved results were also the same.

In the second test we generated attacks on a configuration of one Suricata IDS and one Snort IDS. Several attacks got detected by both systems, while some of them were detected only by one. Reason for this is that they were applying different rules. If some attack was not detected, it would be because of the rule being missing or insufficient.

Based on these test we can state that using several IDS's or their different configurations improves system security.

5. CONCLUSIONS

The goal of this work was to design and implement a system simplifying the work with IDS and improving the intrusion detection efficiency and system security.

Our solution is able to operate with several IDS's and currently it supports two types of IDS: Snort and Suricata. Each of them can have different configurations. The usage of different IDS's or their different configurations can lead to more effective or faster detection. That can also be achieved by using a different set of rules for the same type of attack. By using different IDS's with an identical configuration of rules, we can greatly improve the system security in case of failure of the one of IDS's. This solution offers a combination of the both of these approaches, which also enables to combine their benefits.

ACKNOWLEDGMENTS

This work was supported by the Slovak Research and Development Agency under the contract No. APVV-0008-10 and KEGA 008TUKE-4/2013 Microlearning environment for education of information security specialists.

REFERENCES

[1] LIAO, Hung-Jen et al.: Intrusion detection system: A comprehensive review. In: Journal of Network and Computer Applications. v. 36, 2013, issue 1, pp. 16-24.

[2] MARINOVA-BONCHEVA, Vera: A Short Survey of Intrusion Detection Systems. In: Problems of Engineering Cybernetics and Robotics, 2007, issue 58, pp. 23-30.

[3] SCARFONE, Karen -- MELL, Peter: Guide to Intrusion Detection and Prevention Systems (IDPS). National Institute of Standards and Technology, 2007. [online] : [cited 19.10.2013]. Available on the internet: http://csrc.nist.gov/publications/nistpubs/800-94/SP800-94.pdf

[4] SANS Institute: Understanding Intrusion Detection Systems. SANS Institute, 2001. [online]: [cit. 19.10. 2013]. Available on the Internet:http://www.sans.org/reading-

room/whitepapers/detection/understanding-intrusion-detection-systems-337

[5] MEISAM, S. A. Najjar - MOHAMMAD, Abdollahi Azgomi: A distributed multi-approach intrusion detection system for web services. In: Proceedings of the 3rd international conference on Security of information and networks. 2010, pp. 238-244.

[6] LOUVIERIS, Panos et al.: Effects-based feature identification for network intrusion detection. In: Neurocomputing. v. 121, 2013, pp. 265-273.

[7] PAULINS, Nauris: An agent-based hybrid intrusion detection system. In: esearch for Rural Development - International Scientific Conference. v. 1, 2011 pp. 191-195.

[8] SHAMSHIRBAND, Shahaboddin et al.: An appraisal and design of a multi-agent system based cooperative wireless intrusion detection computational intelligence technique. In: Engineering Applications of Artificial Intelligence. v. 26, 2013, issue 9, pp. 2105-2127.

[9] AMZA, Cristina et al.: A Hybrid network Intrusion Detection. In: Intelligent Computer Communication and Processing (ICCP). 2011, pp. 503-510.

[10] LIN, Ying et al.: The Design and Implementation of Host-Based Intrusion Detection System. In: Intelligent Information Technology and Security Informatics (IITSI). 2010, pp. 595-598.

[11] SHIRI, F. Izak et al.: A parallel technique for improving the performance of signature-based network intrusion detection system. In: Communication Software and Networks (ICCSN). 2011, pp. 692-696.

[12] PINTELLO, T.: Introduction to Networking with Network. New Jersey: John Wiley & Sons, 2013. 9780470487327, pp. 142-143.

[13] ANEETHA, S. A et al.: Hybrid network intrusion detection system using expert rule based approach. In: Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology. 2012, pp. 47-51.

[14] CORONA, Igino et al.: Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues. In: Information Sciences. 2013, issue 239, pp. 201-225.

[15] BIN HAMID ALI, F. A. -- YEE YONG LEN: Development of host based intrusion detection system for log files. In: Business, Engineering and Industrial Applications (ISBEIA). 2011, pp. 281-285.

[16] DING, Yu-Xin et al.: Research and implementation on snort-based hybrid intrusion detection system. In: Machine Learning and Cybernetics. v. 3, 2009, pp. 1414-1418.

[17] ZHANG, Yichi et al.: Distributed Intrusion Detection System in a Multi-Layer Network Architecture of Smart Grids – 2011. In: Smart Grid, IEEE Transactions, vol.2, no.4, pp.796 – 808.

[18] FANFARA, Peter et al.: Usage of Proposed Autonomous Hybrid Honeypot for Distributed Heterogeneous Computer Systems in Education Process – 2013. In: ICETA 2013 : 11th IEEE International Conference on Emerging eLearning Technologies and Applications : proceedings : October 24-25, 2013, Stary Smokovec. - Danvers: IEEE, 2013 P. 83-88. - ISBN 978-1-4799-2161-4

[19] VOKOROKOS, L. et al.: A Distributed Network Intrusion Detection System Architecture Based on Computer Stations Using GPGPU - 2013. In: INES 2013: IEEE 17th International Conference on Intelligent Engineering Systems: proceedings: June 19-21, 2013, Costa Rica. - Budapest: IEEE, 2013 P. 323-326. - ISBN 978-1-4799-0828-8

Received October 20, 2014 , accepted November 17, 2014

BIOGRAPHIES

Liberios Vokorokos (prof., Ing., PhD.) was born on 17.11.1966 in Greece. In 1991 he graduated (MSc.) with honours at the Department of Computers and Informatics of the Faculty of Electrical Engineering and Informatics at Technical University in Košice. He defended his PhD. in the field of programming device and systems in 2000; his thesis title was "Diagnosis of compound systems using the Data Flow applications". He was appointed professor for Computers Science and Informatics in 2005. Since 1995 he is working as an educationist at the Department of Computers and Informatics. His scientific research is focusing on parallel computers of the Data Flow type. In addition to this, he also investigates the questions related to the diagnostics of complex systems. Currently he is dean of the Faculty of Electrical Engineering and Informatics at the Technical University of Košice. His other professional interests include the membership on the Advisory Committee for Informatization at the faculty and Advisory Board for the Development and Informatization at Technical University of Košice.

Michal Ennert (Ing.) was born on 4th August 1987 in Revúca, Slovakia. In 2011 he graduated at the Department of Computers and Informatics of the Faculty of Electrical Engineering and Informatics at the Technical University of Košice and received the engineering degree. Since 2011 he is PhD. student. He is doing research and experiments mainly in the field of computer security with usage of GPGPU technology and in the field of distributed software architecture.

Zuzana Dudláková (Ing.) was born on 9th July 1988 in Košice, Slovakia. In 2012 she graduated at the Department of Computers and Informatics of the Faculty of Electrical Engineering and Informatics at the Technical University of Košice and received the engineering degree. Since 2012 she is PhD. Student.