JAM CM Journal of Applied Mathematics and Computational Mechanics 2015, 14(1), 55-62
www.amcm.pcz.pl p-ISSN 2299-9965
DOI: 10.17512/jamcm.2015.1.05 e-ISSN 2353-0588

TESTING OF IDS MODEL USING SEVERAL INTRUSION DETECTION TOOLS

Michal Ennert, Eva Chovancová, Zuzana Dudláková

Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics
Technical University of Košice, Slovakia
michal.ennert@tuke.sk, eva.chovancová@tuke.sk, zuzana.dudlakova@tuke.sk

Abstract. The aim of this work is to provide set of selected tests on IDS model that would enlarge the functionality of intrusion detection systems. Designed model is using several IDS, which allows it to investigate larger bandwidth and capture more attacks. This system consists of central master node and devices on which the intrusion detection systems are. The tests were designed with the attribute of repeatability and universality. They are divided into five categories which explore specific attributes of intrusion detection system.

Keywords: *intrusion detection systems, computer security, IDS testing*

Introduction

Computer security plays an important role in the present. Majority of computer attacks is used for network security breach and for their massive expansion for the acquisition of sensitive information. A large scale of security systems is being used as a protection against them. Among these programs there is also intrusion detection system (IDS). Their purpose is not to prevent the attacks, but to discover them. This work deals with the possibility to use several IDS at the same time and the application of tests for these systems.

With the usage of several IDS in the proposed system there is a problem with the central storage of data, in which all logs generated by the detection system are being saved. A master node was designed for this purpose. Its tasks are to centralize the management of individual IDS, as well as provide primary database to store records of identified threats and irregularities in the net. For the functionality of described node, it was necessary to design and implement unified system for recording of homogenized logs in the database on the IDS stations as well as to ensure access into individual devices.

This work is also devoted to design a testing model of systems intended for intrusion detection, test of applied IDS as well as the proposed system. When creating tests, it is necessary to keep certain features that ensure their repeatability and versatility. The tests were divided into several categories according to the type of

IDS feature, which were examined in the test. The intention was to design such a set of tests that would allow determining the environment in which it is appropriate to apply IDS as well as configuration of parameters that affect the performance of systems.

1. Intrusion Detection Systems

Network intrusion detection systems are divided into two categories. The difference is in the form how they examine the network traffic [1]:

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 between the Internet and a private network (Fig. 1). Internal IDS controls communication within the LAN. In some cases it is not necessary to monitor activity across the entire network, but only at 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.

Snort is the most widely deployed intrusion detection and prevention technology worldwide. It has the most numerous and active community in the open source network IDS field today. Snort is a type of IDS that uses for its operation set of rules, but can also monitor certain anomalies. Snort is logically divided into several modules: Packet Capture Module, Decoder, Preprocessors, Rules Files, Detection Plug-ins, Detection Engine and Output Plug-ins. These sections cooperate together to discover the individual attacks and to generate output in the required format [2].

Suricata belongs to the category of IDS that also uses rules to monitor and control network traffic. It uses several new innovative technologies that were first implemented in open source IDS. These technologies include support for multithreading processing of packets. This support is important due to current development of CPU. Improvement of the CPU performance is no longer aimed at increasing of the frequency and power of one core, but instead increasing the number of cores in the CPU itself. To increase performance and faster processing of examined data, CUDA GPU acceleration for pattern matching was added into Suricata IDS [3].

Network load is influenced by a variety of objects at different abstract level, whose complexity is further increased by advancing of new technologies and applications. If we would test IDS using only programs that allow us to simulate some types of attacks, we would never acknowledge the results that are reliable and comparable to the real network communication. For that reason communication in the background plays an important role in proper testing of IDS [4, 5].

There are many programs that allow generating communication in the background and attacks that can test the IDS. For the generation of communications in the background, D-ITG program can be used. It allows generating IPv4 and IPv6 data that accurately replicate the workload of actual Internet applications [6]. Set of programs, called TCPreplay, can be used to replay previously captured network communication. It allows classifying the traffic into server and client part. Pytbull program that contains over three-hundred tests can be used to generate attacks on IDS [7]. It is specifically designed for testing IDS Snort and Suricata.

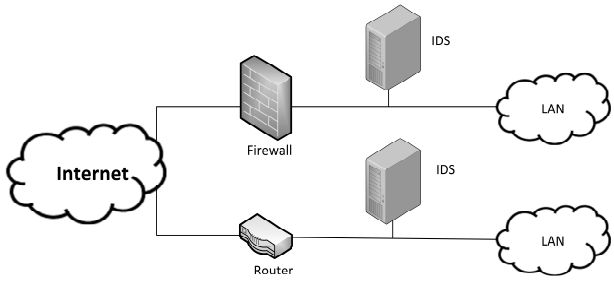

Fig. 1. IDS network

2. System design and model for testing IDS

In the Figure 2 below there is a model of system that is designed for the purpose of supplementing the network security statement, which is ensured by systems designed to detect intrusions.

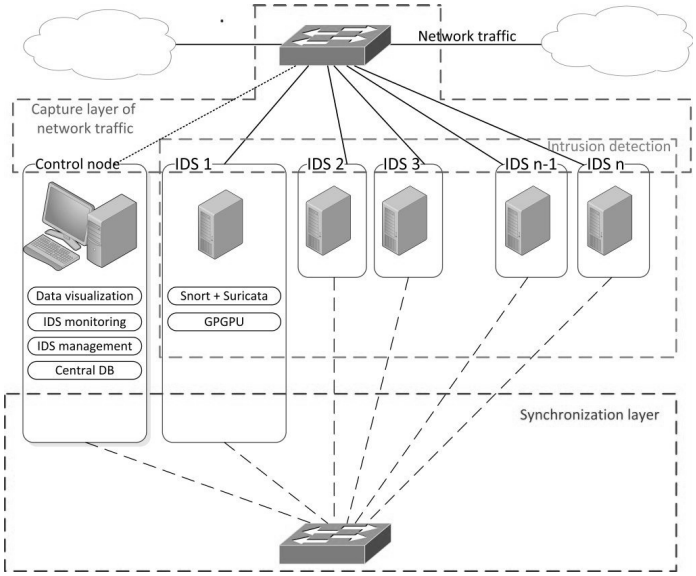

Fig. 2. Model of IDS architecture

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 true or false can be achieved.

For testing of IDS, five types of tests were designed and implemented and logs regarding workload of system resources were recorded. In the first test the PCAP file is being processed that is located on the same device as IDS. The test is not oriented to determine which IDS detects more attacks but the speed with which it can process the file. Second test focuses on generating attacks against IDS. The accuracy of intrusion detection systems is being monitored while detecting attacks. Program Pytbull and TCPreplay were used to generate attacks. TCPreplay was used to replay PCAP files containing malware into the network. The third test is focused on determining the system resources used by individual IDS. In this test only one CPU core is solely dedicated to IDS that processes network traffic at a defined speed. CPU load and RAM usage is monitored. The fourth test is similar to the second test but with one difference. The background communication of network is being generated. Test designed like this simulates real-network traffic. The program D-ITG is used to generate communication in the background. Last, fifth test was designed to test the proposed system that uses multiple IDS. In this test the different load of the systems and detection of attacks is monitored with the usage of single and multiple IDS.

3. Tests and results

In the first test a PCAP file was used that contains 10 874 809 packets and has a size of 1150 megabytes. The graph in Figure 3 shows the time periods that were recorded for each IDS. Based on the obtained data, the result is that the fastest process of file was done by Suricata IDS, but the increase in performance that was expected from the usage of the GPU was not achieved.

When testing IDS, it is important to find out how accurate the detection of potential threats and attacks can be. The second test focuses exactly on this attribute of systems designed to detect intrusions. The results of this study can be found in the Table 1. The difference between scored points among tested IDS is minimal; however Suricata scored six points more. This difference refers to various rules that IDS used.

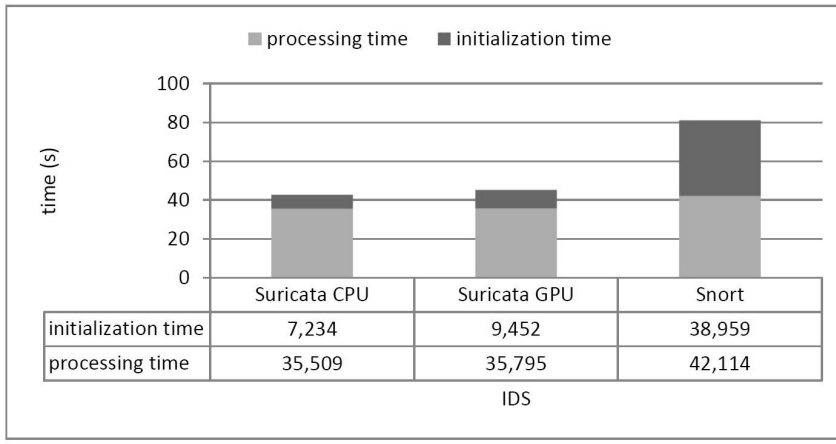

Fig. 3. Comparison of processing times of each IDS

Table 1

Results of tests with Pytbull and PCAP files

Category	Suricata	Snort
Rule testing	14	10
Bad traffic	6	2
Fragmented packets	2	6
Brute Force Attacks	3	3
Evasion techniques	28	26
Shell codes	27	27
Denial of service	3	3
PCAP TCPreplay	27	27
Total	**110**	**104**

The third test was aimed at detecting workload of IDS system resources. Graph in Figure 4 shows the workload progress of system resources consumed by IDS Snort with particular configurations of the network traffic speed. Graph with workload progress of the system resources used by Suricata IDS is shown in Figure 5. The results of this test indicate that the Snort IDS with hardware configuration in which only one CPU core is reserved, can process network traffic at higher speed without causing the drop of packets.

Fourth test had a similar course as the second one with the only difference being that IDS had to process the communication in the background. Results of this test were compared to the results from the second test. Both tested systems, Suricata and Snort IDS, had managed to capture the same attacks.

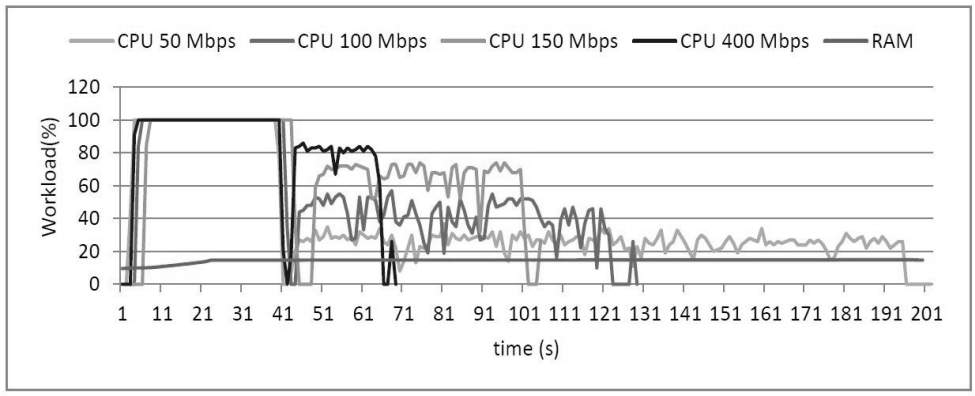

Fig. 4. Workload of system resources for IDS Snort

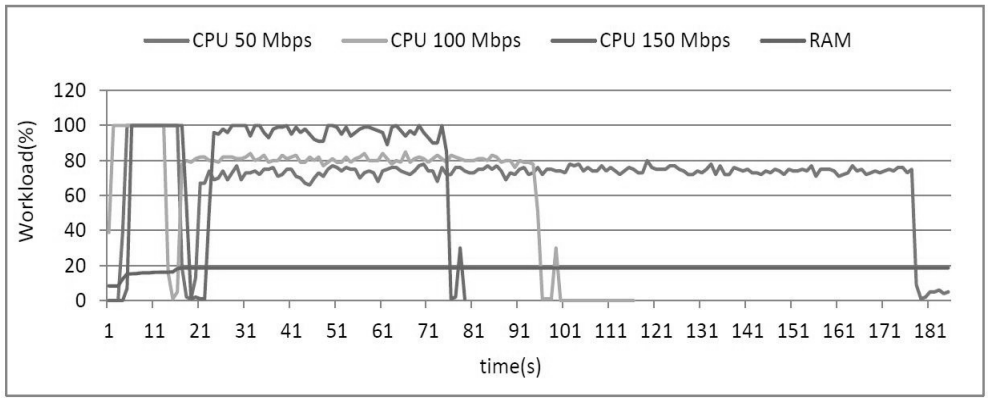

Fig. 5. Workload of system resources for IDS Suricata

The last, fifth test was focused on testing the designed model. While testing, the results of using one and then two IDS were compared. When using two IDS the rules were evenly distributed between them. The first tested system was Suricata. At first, it was launched independently and had all the rules activated. Subsequently after this test, one more detection unit Suricata was added to the system and took over half of the rules. The graph in Figure 6 displays development of network communication where the quantity of packets which came to IDS per second was recorded. In the same chart it is also shown the amount of packets that was missed by detection units when using one or two IDS.

The same testing procedure as with the Suricata was applied also for IDS Snort. The first test was done using only one IDS Snort. Thereafter a second detecting unit Snort was added and the rules were divided between the two intrusions detection systems. Progress of network communication is recorded in graph in Figure 7. Decreased amount of dropped packets is visible when using two identical IDS. Decreased tendency of dropped packets was on average 8.6%.

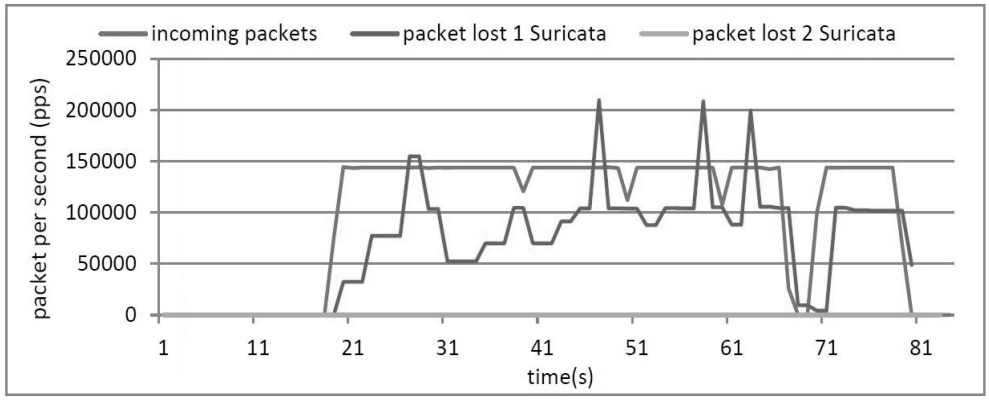

Fig. 6. Packet drop IDS Suricata

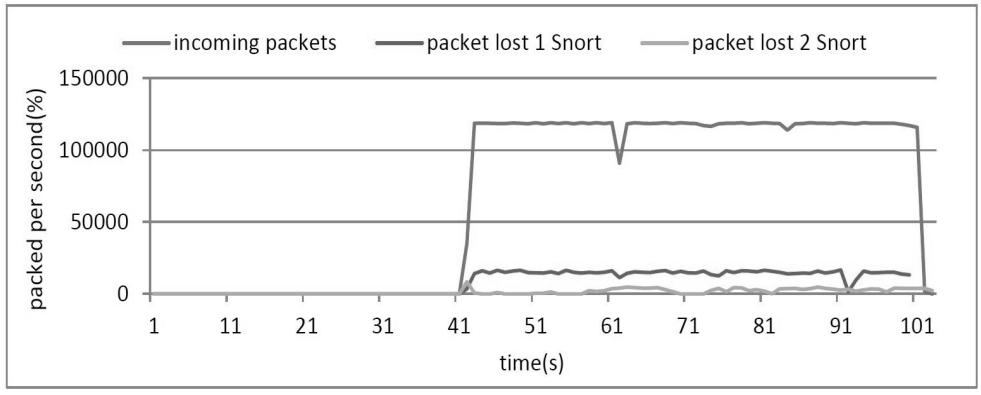

Fig. 7. Packet drop IDS Snort

Conclusion

This work has set several goals to contribute to the increase of the security of computer networks, simplify the simultaneous use of multiple intrusion detection systems and to create a testing model for IDS. Therefore a model was proposed that consists of several IDS and one central control node. Such a structure should contribute to an increase in computer network security. This type of proposed system will be capable of processing data stream at a higher speed as an individual IDS and it can detect a wider range of attacks and suspicious communications. Testing model of IDS has been divided into five categories. These categories are focused on the characteristic attributes of intrusion detection systems. The first test was used to optimize the configuration of tested IDS Snort and Suricata. After optimizing the configuration of the various systems, PCAP file was processed fastest by Suricata IDS without activated GPU. This result can be mainly attributed to the multithreaded architecture of Suricata IDS which has managed to utilize the

CPU power. Second test was aimed to determine the accuracy of the IDS while detecting attacks. In this test Suricata IDS scored 6 points more. The third test pointed out the different architecture of the tested IDS. Single-threaded architecture of Snort IDS managed to make a better use of system resources available to it. During the fourth test both tested IDS managed to detect the same attacks as by the second one. The communication in the background did not cause any problems to IDS regarding detecting the attacks. The last fifth test was aimed to determine whether the proposed architecture is relevant. From the results of tests it is obvious that this type of connection of IDS processes network traffic with a larger bandwidth. This causes decreased probability of packet drop which means higher success rate in detecting of attacks.

References

[1] Pintello T., Introduction to Networking with Network, John Wiley & Sons, New Jersey 2013, 142-143.

[2] Calvo Moya M.A., Analysis and evaluation of the Snort and Bro network intrusion detection systems, Proyecto de fin de carrera, Universidad Pontificia Commillas, Madrid, Sep. 2008.

[3] White J.S., Fitzsimmon T.T., Matthews J.N., Quantitative analysis of intrusion detection systems: SPIE Procedings 8757, Cyber Sensing 2013, 1-13.

[4] Antonatos S., Anagnostakis K.G., Markatos E.P., Generating realistic workloads for network intrusion detection systems, ACM SIGSOFT Software Engineering Notes, January 2004, 29, 1, 207-215.

[5] Evans D.L., Bond P.J., Bement A.L., An Overview of Issues in Testing Intrusion Detection Systems, U.S. Department of Commerce, Technology Administration National Institute of Standards and Technology. Gaithersburg 2003, 2-18.

[6] Botta A., Dainotti A., Pescapé A., A tool for the generation of realistic network workload for emerging networking scenarios, Computer Networks 2012, 56, 15, 3531-3547.

[7] Damaye S., Pytbull. [Online] [2013-12-6] http://pytbull. sourceforge.net/index.php?page=home

[8] Rehman Rafeeq: Intrusion Detection Systems with Snort. Pearson Education, New Jersey 2003, 0-13-140733-3, 8-23.

[9] Vokorokos L., Ennert M., Čajkovský M., Turinská A., A distributed network intrusion detection architecture based on computer stations using GPGPU, IEEE 17th International Conference on Intelligent Engineering Systems: proceedings: June 19-21, 2013, Budapest: IEEE, 2013, 323-326.