DATA VISUALIZATION OF NETWORK SECURITY SYSTEMS

Michal ENNERT, Branislav MADOŠ, Zuzana DUDLÁKOVÁ
*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: {michal.ennert, branislav.mados, zuzana.dudlakova}@tuke.sk

ABSTRACT

The goal of this article is to analyse, design, implement and test different ways of evaluating and visualizing logs from intrusion detection system (IDS). For the main objective of this work it was necessary to study the ways in which it will be possible to carry out data visualization recorded in the IDS database. It was necessary to design and implement evaluation, imaging and visualization logs of intrusions recorded in a computer network security system. The work will therefore bring more comfortable option to read large amounts of technical messages to humans. Such a person does not have to have knowledge of the system, which used to display visualized data in depth – the person should just be able to interpret the visualization displayed on a computer screen.

Keywords: *network security, data visualization, intrusion detection, graph*

1. INTRODUCTION

Nowadays due to informatization data is gradually moved from various records and archives to computer servers, which is accessible via computer network. Ascribable to growing number and importance of information there are more sophisticated attacks on the network, which is constantly increasing need to protect computer networks from diverse attacks. Few present systems are left without a firewall, but this protection begins to be insufficient for computers with sensitive information. Hence arose need for an intrusion detection system [1][2].

The aim of this work is to create the concept and design of network architecture and this system, system which has the task of monitoring and management stations for intrusion detection, which should contribute to an even greater degree of protection against possible network attacks. The first part is devoted to the issue of creating the concept and design of network architecture, the second is dedicated configuration Suricata (intrusion detection system) and its rules. Furthermore, this work has to familiarize with the potential for standardization of processed data. Subsequently, these data will be evaluated and visualized in an appropriate manner to the user.

2. ANALYSIS

2.1. Architecture of IDS

This work should bring opportunities to simplify and clarify the monitoring intrusion on the network with a large number of transmitted data.

The intrusion detection network analysis architecture used in this work is based on our previous research [3]. Architecture is familiarly named DINA and consists of number (at least two) of standalone network IDS stations which use the computing power of the graphics hardware. Cooperation between the IDS stations is managed 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 IDS 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 network IDS stations. The architecture of our solution (Figure 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.

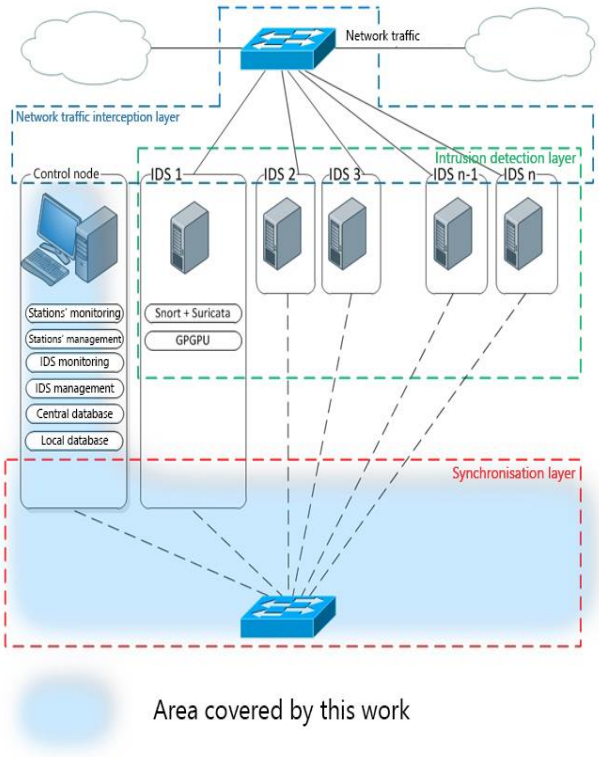

Fig. 1 Proposed DINA architecture [3]

The data analysis is carried by intrusion detection, layer which consists of a series of independently functioning IDS that operate based on predetermined rules. These rules are for each IDS may vary. To accelerate the evaluation of data from a network IDS system GPGPU technology can be used [3]. The data is transferred from the network traffic into different IDS layer to capture network traffic.

The synchronization layer facilitates communication layer intrusion detection with supervisory system. Surveillance system is using this layer to manage various IDS. Surveillance system is designed to:

- Perform remote management of IDS
- Change the configuration of IDS
- Collect results and logs,
- View current and previous
- Based on the results do a particular activity (inform the system administrator, send an email, etc.).

The concept of the system consists of creating basic system requirements.

The most important requirements for the system are authentication, secure communication over a network, remote control of clients, tracking their status, workload and attack detection.

Another important requirement is a graphical user interface (GUI), which should be clear and intuitive. This requirement, however, is not so important, because the system is not primarily designed for the everyday user, but the user who has the knowledge and skills in the areas of security and network administration of the Linux operating system.

2.2. Database schema analysis

In this part we specify database tables description of stored events:

- The table data contains data from a data packet that triggered the alarm.
- The table detail contains detail information about the items that were stored with the packet.
- The table encoding shows the encoding types used in logging packets.
- Table event contains a list of all events and the corresponding timestamp.
- Icmphdr table contains information about the ICMP packet headers that are logged into the database.
- Iphdr table contains all fields of logged IP packet headers. It stores information about source and destination address, IP protocol version, the length of the IP header TOS and TTL value and many others.
- Opt table contains the options.
- Table references and reference system contains information about websites on which there is a more detailed description of each vulnerability.
- The table schema contains information about the version of the database schema.
- Sensor table contains information about the various sensors that logs data to the database.
- Sig_class table contains information on the different classes of rules contained in Suricata.
- Table sig_reference contains web links to websites containing information about individual vulnerabilities.
- Signature table contains information about signatures that generated the alarm.
- Tcphdr table contains information about the TCP packet header.
- Udphdr table contains information about the UDP packet header.

3. VISUALIZATION OF DATA ACQUIRED BY INTRUSION DETECTION SYSTEM

Each of the layers of the application is used to hide implementation details of the functionality. The motivation for splitting applications into separate individual layers is a generalization of communication between the layers. Individual layers communicate with each other using a defined group of methods, regardless of the internal functionality. This approach achieves universality and closeness of functional units. Even though this approach requires more complex implementation, it is possible to reuse certain parts of the methods. If design changes and so should the implementation of one of the layers, this requires change only within the layer while maintaining public methods by which they communicate with a given layer, which means that, in this architecture, the individual layers are opaque from the perspective of other layers of abstraction and that ensures standardization of communication data.

In the event that the application needs should connect to another database type or to the same database with a different scheme it will be needed just to adjust only database layer. Other layers due to the opaque character of the database layer will not have to be changed.

Database layer separates the database with information about events from the rest of the application. Thus abstracts access to the actual data in an opaque manner.

After the application requests a new data, this data is retrieved from the database server through SQL requests. It normalizes the data retrieved and for each record creates an event object, which is then stored in the event list. This list of events is all located in the memory due to the speed of data processing. At the moment when the data is requested it causes increased demands on system memory size, but delivers faster loading and processing.

Part of this application is designed for loading and providing data on the geographical location of IP addresses. IP address location database is stored in a database csv file type, which is loaded into memory on start-up. The class is using design pattern "Singleton" has only one instance, a private constructor and static method for access to single instances. Again at this point there is approach that the entire database loaded into memory with the intention to increase the response speed applications while loading a large number of records. During a search, the location of the IP address is also used by "caching" - storing already retrieved records into a smaller list that is scanned before scanning the entire database, since the greater the chance that another search IP address has already been found and thus its next search will be faster because there is less items in the list [8]. DinaGraph is the main application class. Unofficially, this solution should

be named DINA, so therefore the name of the class. In this class, triggered by main method of application, which is using the same arguments as the main method started running a new object DinaGraph.

Consequently, in this class by calling method initRefresher thread is initialized used for periodic loading new records from a database of intrusion detection system to the local list of intrusions stored in main memory. By calling methods showNewGraph graphs are initialized and then displayed.

RealTimeGraph visualization is aimed to showing the current added events to the database of events. This chart is suitable for frequent renewal of the content. The vertical axis shows the number of recorded events and the horizontal axis is the time stamp, shown in Figure 2.

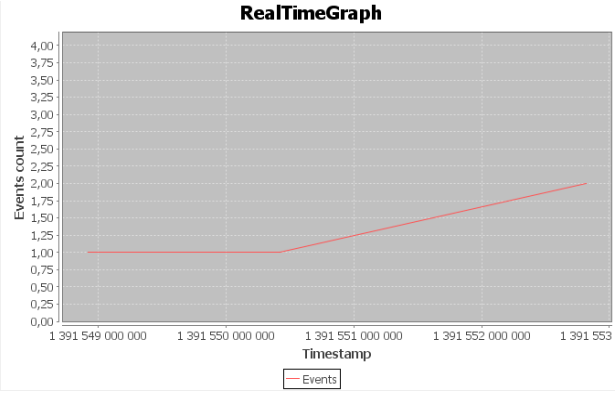

Fig. 2 RealTimeGraph

AttackNodes is a graph, where nodes are displayed and oriented edge, see Figure 3. The nodes represent network elements identified by their IP addresses. Node size represents the number of events coming from the IP address. Edges represent events recorded in the database.

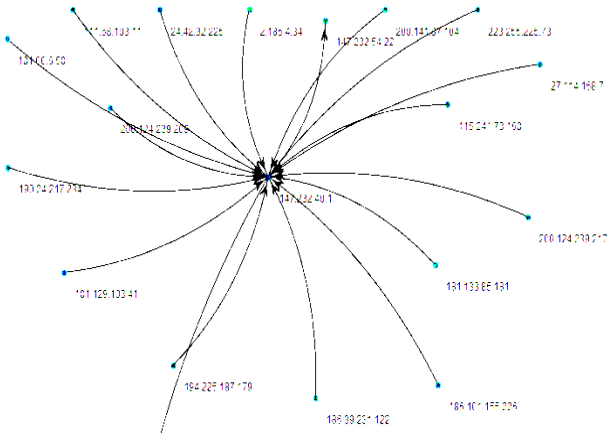

Fig. 3 AtackNodesGraph

WorldMap is a graph, where nodes are displayed and oriented edge, see Figure 4. The nodes represent network elements identified by their geographic location specified by IP address. Edges represent events recorded in the database. After placing the mouse cursor over a particular node it shows the IP address associated with the node as a tooltip near the mouse cursor.

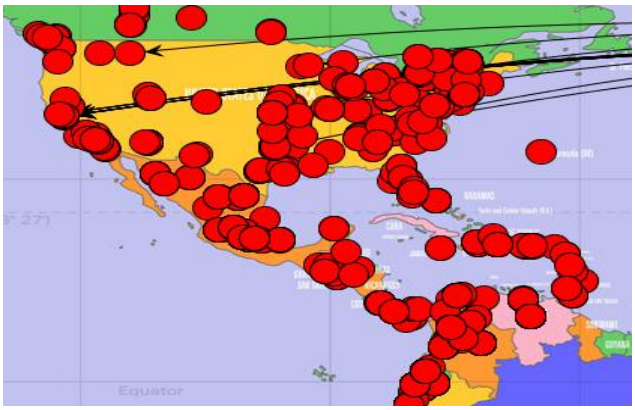

Fig. 4 WorldGraph

4. EXPERIMENTAL RESULTS

Logging part of the solution has been deployed on the internal computer network of the Technical University in Kosice. The central router of computer network has one port set up in a way to forwarded all traffic on the network for the selected port, where there was server listening to all network packets. In this way it ~ 700000 records were recorded for the first three months of operation. This sample was sufficient to test this application.

Testing took place on the grounds of a given network Technical University where it was running without interruption all the time, so that new events were recorded during each time of the day. During business hours 8:00 to 15:00, the load was ~ 3 times higher than outside these hours. This makes it possible to follow normal operation during working hours and also less busy hours during the night. Periodic changes can be clearly monitored on charts:

- load of the central processing unit,
- the use of random access memory,
- load of local network adapter,
- percentage of dropped received network packets.

Increased load can be seen in the number of recorded events displayed according to the time of recording.

The application itself has been tested on the local computer outside of the test network. Access to the server, where the data was stored was conducted over the internet. In this way data was downloaded from the database, which simulated real possibility of deployment. Deployed surveillance system in full operation would be monitored outside the network due to the architecture of the proposed system and also because the inaccessible inside network would render that server unavailable.

During testing events were recorded from five continents of the world (Antarctica has refused to participate in testing).

5. CONCLUSION

The goal was to create the concept and design of the future system architecture. This concept, in this unique time system monitoring and management stations

intrusion detection, was created based on the requirements for the system and architecture design based on client-server architecture and network protocol (SSH and SCP protocol extensions). It was necessary to select the most promising tool for analysing network traffic detection and penetrate. We decided to use Suricata IDS, which currently represents the most powerful open source solution. Fair architecture design was necessary to study and understand the operation Suricata and its rules to detect anomalies in network traffic. Another problem we have had to cope with was to move Suricata information generated by the system for further processing and evaluation. To solve this problem, we used the log output of Suricata which is further processed by applying Barnyard2 and embedded SQL database.

In the future, this solution allows incorporation into another project using the methods available for another project. Parametric running applications allow easy integration with other solutions. Additionally there remains scope for providing additional visualizations to predicting system load, the analysis of combinations of the metrics and their application to early warning before a possible system overload. The combination of these amendments, together with the existing solution would be a suitable solution for comprehensive management and warnings of attacks on computer networks monitored using IDS.

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] 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.

[3] 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

[4] DISEM: Cisco ASA Syslog Linechart [online]. 2013. http://secviz.org/content/cisco-asa-syslog-linechart>

[5] GRAPHSTREAM TEAM: A random walk on a graph [online]. 2013. <http://graphstream-project.org/doc/Algorithms/Random-walks-on-graphs_1.0/>

[6] UNITED STATES PATENT AND TRADEMARK OFFICE: Trademark Status & Document Retrieval [online]2013. <http://tsdr.uspto.gov/#caseNumber=75263259&caseType=SERIAL_NO&searchType=statusSearch>

[7] THE JUNG FRAMEWORK DEVELOPMENT TEAM: Overview [online]. 2010. <http://jung.sourceforge.net/>

[8] PEKÁR, A. et al.: Issues in the Passive Approach of Network Traffic Monitoring - 2013. In: INES 2013: IEEE 17th International Conference on Intelligent Engineering Systems: proceedings: June 19-21, 2013, Costa Rica. - Budapest: IEEE, 2013 P. 327-332. - ISBN 978-1-4799-0828-8

Received January 18, 2015 , accepted February 23, 2015

BIOGRAPHIES

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 distributed software architecture.

Branislav Madoš (Ing., PhD.) was born on 20th May 1976 in Trebišov, Slovakia. In 2006 he graduated (Ing.) at the Department of Computers and Informatics at the Faculty of Electrical Engineering and Informatics of the Technical University of Košice. He defended his PhD. in the field of Computers and computer systems in 2009; his thesis title was "Specialized architecture of data flow computer". Since 2010 he is working as an Assistant Professor at the Department of Computers and Informatics. His scientific research is focused on the parallel computer architectures and control flow computer architectures.

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.