Marek BOLANOWSKI [1], Bogusław TWARÓG [2], Rafał MLICKI
[1] RZESZOW UNIVERSITY OF TECHNOLOGY, DEPARTAMENT OF DISTRIBUTED SYSTEMS, 12 Powstańców Warszawy Ave., 35-959 Rzeszow
[2] UNIVERSITY OF RZESZOW, FACULTY OF MATHEMATICS AND NATURAL SCIENCES, DEPARTMENT OF COMPUTER ENGINEERING, 1 Pigonia St., 35-959 Rzeszow

Anomalies detection in computer networks with the use of SDN

Abstract

In the paper, the authors present a method of anomalies detection and identification in network traffic using statistical signatures. There is also shown a new system architecture based on the Software-Defined Networking (SDN) which allows for application of statistical anomaly detection in computer networks. With the proposed hardware-software model, it becomes possible to implement custom algorithms for the threats detection with the use of recognized and secure communications standards. The proposed architecture has been built based on an open-source solutions and can be used directly in production environments.

Keywords: anomaly detection, statistical signature, computer network, software defined network.

1. Introduction

Modern computer systems and networks are the environment concentrated on processing large sets of information in an environment of heterogeneous applications and data convergence. The processed data are sensitive and must be protected. The challenge is the threats detection in the environment of high speed computer networks. Due to the large amount of transmitted data, their current, accurate analysis (Deep Packet Inspection) in the upper layers of the ISO/OSI model is not possible.Three main issues should be considered in current models of data protection:

- Detection point of threats shifts from a single point of protection for the entire network (DMZ, the firewall) in the direction of devices deployed in the access layer (DPI switches) or even on terminals.
- The way in which the modern IT infrastructure elements are managed is changing. Network Management System sees network devices as well as virtual machines, servers and end station. Some of NMS based on the transferred data analysis are capable to change the access policy to network resources (BYOD). So it is natural to place in the NMS infrastructure probes like IDS.
- Changing the network management from a reactive to an active network control, through the introduction of applications that on the fly can modify the parameters of the network environment in which they operate (self-adaptation and self-organization)[1].

In the paper, a model of a threat detection system using SDN is presented. It allows returning to the rough threat detection model in a single DMZ point, but also allows for ongoing analysis of threats in the access layer of network devices. There is a lot of works devoted to this subject, in which the authors using the SDN implement their own proprietary solutions for threat detection [2,3]. Large individualism of these solutions does not allow for their widespread use in production networks. In this paper, the authors focus on the detailed presentation of the model based on the open source solution that enables implementations of custom anomaly detection algorithms in the environment of high speed computer networks.

2. Anomaly detection

The use of statistical methods for roughing identification of threats in computer networks allows monitoring core networks links without the need of analyzing threats patterns on each ISO/OSI layer separately. This holistic approach enables the creation of statistical signatures of threats that can generate alarms or initiate pre-programmed actions [2, 4]. Unfortunately, the learning phase of algorithms for statistical anomaly detection is a very complex process and a set of statistical patterns is different for different types of threats. In the test environment shown in Fig. 1, the sensitivity of selected statistical parameters for the specific types of attacks was determined. A set of computers (PC 1 to PC n) generates the network traffic to and from the Internet. For this movement there were determined the values of selected statistical parameters using a copy of all traffic sent from a switch to a PC n.

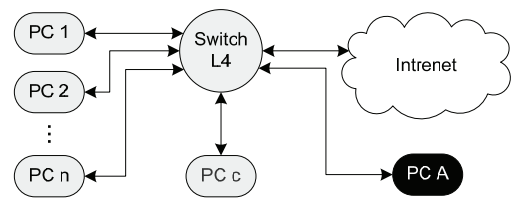

Fig. 1. Experimental stand for determining the statistical signature

Then, the selected attacks from the computer PC A (with the use of a Linux Backtrack system) were performed. Tab. 1. shows some examples of the changes of network traffic statistical parameters in the case of two attacks. Note that different parameters have different sensitivity to the ongoing attacks.

Tab. 1. Attacks and corresponding changes of statistical parameter

Name	Parameter	Without the attack	During the attack
1. Internet Explorer DHTML Behaviors Use After Free	Hurst	**0.5593**	**0.53953**
	average	629.3	620.9
	Median	**352**	**85**
	Standard deviation	632.88	658.60
	Kurtosis	−1.54	−1.63
	Slant	0.48	0.49
	Range	1454.00	1472.00
	Minimum	60.00	42.00
	Maximum	1514.00	1514.00
	P/s	**276.4**	**343.5**
	UDP/s	0.05	0.06
	TCP/s	**253.55**	**286.61**
	Average Packet Size /s	190.13	244.07
2. Windows ANI LoadAniIcon() Chunk Size Stack Buffer Overflow (HTTP)	Hurst	0.558538	0.545627
	average	629.37	536.12
	Median	**352.00**	**62.00**
	Standard deviation	632.88	605.08
	Kurtosis	**−1.54**	**−1.14**
	Slant	**0.48**	**0.77**
	Range	1454.00	1472.00
	Minimum	60.00	42.00
	Maximum	1514.00	1514.00
	P/s	**276.48**	**585.66**
	UDP/s	**0.05**	**0.47**
	TCP/s	**253.55**	**488.42**
	Average Packet Size /s	**190.29**	**254.47**

The analysis of such cases is not easy and requires the use of a variety of techniques and algorithms that are specific to a particular network system and the nature of the user traffic. Consequently, the traffic analysis of the statistical parameters required quite large computational power and an open development environment allowing administrators to independently analyze the data from the switches and use them to prepare their own anomaly detection algorithms. As it can be seen, interpretation of the received signatures is difficult, and finding a universal algorithm that will classify signals for all the risks and

decide whether the alarm should be generated is not feasible. Tab. 1. shows that the attack 1 had the significant influence on the change of the following parameters: median P/s, TCP/s, while the attack 2 changed: the median, kurtosis, skewness, P/s, UDP/s, TCP/s, Average Packet Size/s. By checking which values have changed and in what percentage we are able to determine the type of the attack.

The introduction of the presented signatures not only allows for anomaly detection, but also for determining, with a certain probability, its type or its exact identification. Unfortunately, in the case of implementing such solutions, it is necessary to use dedicated network devices or software platforms. Both solutions are not acceptable: first because of the cost, second, because of poor performance. Moreover, until now anomaly detection systems have acted as IDS systems, i.e. have detected attacks and generated an alarm. A feedback communication interface between the IDS and network devices in an environment containing hardware from different vendors is very limited. Therefore, it is necessary to develop an architecture that will allow for the current complex statistical analysis without disturbing the operation of the network equipment and will not provide an additional delay to the network traffic.

3. SDN network

The authors propose the use of the SDN architecture as a hybrid hardware-software network that will provide an environment to implement statistical signatures to detect anomalies. Conventional hierarchical networks fulfill their role when the model client-server is dominant. They have a distributed control structure, which does not allow for (makes it very difficult) a dynamic response to the needs of services and hindering the implementation of new solutions. While many areas of computer science have experienced a rapid development in the area of virtualization and remote access, the networking branch has focused mainly on increasing the capacity and improving the quality for specific services. After a time, this approach has resulted in a situation in which the network solutions are less well developed and do not meet the modern requirements, especially in the area of control and virtualization.

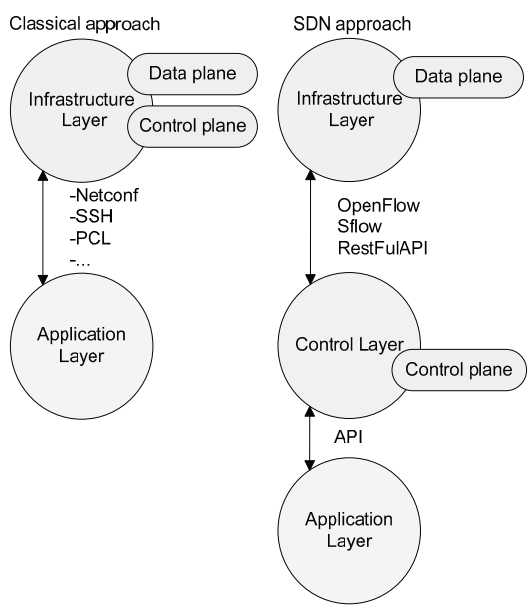

Fig. 2. Classical and SDN approach to the network infrastructure management

In response to the need for change, in 2011 there was established the foundation Open Networking Foundation (SDN) [4] in order to promote solutions software-defined networking and OpenFlow protocol. SDN, under the premise, will replace the traditional network architecture. This solution is possible due to ensuring its five main features: programmability, flexibility, centralize management, software configurability and open license. In Fig. 2, there is presented the classic approach to interactive managing of network devices. In classical architecture of network equipment, we can distinguish two main layers:

- Control Plane (CP): the device layer responsible for logic data service, routing protocols, switching and filtering.
- Data Plane (DP): the layer directly responsible for switching and sending data.

Because of the lack of possibility to make complex calculations on a network device processor, (e.g. in the area of security) threat detection is often made on separate dedicated servers (arbiter). After receiving the results, the arbiter can automatically send to CP a request to block the unwanted traffic, or change the routing policy. Unfortunately, for this purpose in a heterogeneous environment of network devices, custom communication protocols generally based on SSH, or NETCONF have to be used. End customers are afraid to implement such solutions because of potentially low security of the communication channel between the arbiter and the network device. In the SDN architecture layer, CP is transferred from the network device directly to the Control Layer and a complex system responsible for the entire logic of the device is replaced with a much simpler, whose task is to communicate and execute commands of the controller placed in the network architecture. A specialized application installed in the resources of the arbiter performs functions of the Control Layer. By introducing an additional layer (Control Layer) in the SDN architecture administrators gain the ability to use the standard (Sflow, OpenFlow) to change the configuration of network devices on the fly, according to the calculation results obtained from the application or the current network analysis. In the Control Layer, controllers communicate with devices to identify and adequately control packets. In this layer, there are also implemented: network services such as routing, multicast, security, access control, bandwidth management, traffic management, quality of service, optimization of resource consumption, etc. Services run mostly as additional software or plugin in the SDN controller [5]. Using the API implemented on the SDN controller it is possible to create an application that cooperates with the controller, which may offer great opportunities to control the network flows. At present, mostly used SDN controllers are as follows: OpenDayLight [6], Floodlight [7], Onos [8], the HP Virtual Application Networks.

4. System architecture

The architecture of the proposed system in the control and application layer consists of three virtual machines running appropriate software:

- Floodlight – the SDN controller that supports OpenFlow 1.0. With its help certain network flows will be injected into the switch ALU OS OS6860.
- sFlow-RT - a tool for sampling the network traffic. It will be used to detect anomalies.
- SDN-APP - a machine on which the custom application will be launched. it will be responsible for all the functionality for detecting and blocking attacks and anomalies.

The network topology was built with these elements. In it there is clearly marked the division of the whole structure into respective SDN layers: data, control, application (see Fig. 3.). The program creates a flow definition on the server sFlow-RT and continually determines the value of statistical signature based on its analysis. A variety of traffic views may be created, e.g.: Pts/s TCP/s, UDP/s, etc. If the pre-programmed alarm occurs, the application obtains the list of registered switches from the Floodlight controller, compares the IP addresses with the address received from the event and on this basis obtains the DPIP of the switch on which there is an undesirable flow. The DPIP is the

MAC address of the switch used in communication with the controller. The program sends to the controller a command to create flow with the specified parameters on the switch with a specific DPIP. The controller sends to the switch the parameters of the flow to block the unwanted traffic.

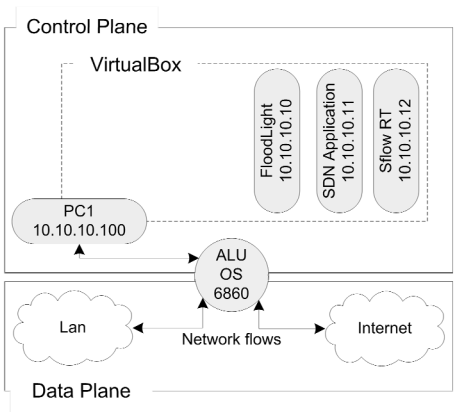

Fig. 3. The proposed system architecture

They may be taken other actions such as for example: redirecting traffic. It should be noted that at any time, as a result of the algorithm calculation, the program can send a request to the controller to remove the flow from the switch that blocks the traffic.

Fig. 4. Example of anomaly detection

Fig. 4 shows an example of DDoS attack detection and automatic flow blocking for 10 seconds. In a similar way, network administrators can implement their own algorithms and automate the management of network security.

5. Conclusions

The paper presents a method for detecting anomalies in network traffic based on the use of statistical signatures. The authors also show the hardware and software architecture model that supports the detection of anomalies using the SDN network. Basing on the results obtained, one can state that the proposed solution can be used in production environments of large network systems. It should be noted, however, that at the moment the SDN is under rapid development, and many network equipment vendors after implementation OpenFlow version 1.0 will wait with implementation of the next version until the protocol stabilizes. The architecture can be used for anomaly detection, however, it is not free from the vulnerability [9] or unreliability of programmers writing control applications. In the further research, the authors want to focus on analyzing the impact of the proposed solution on the stability of the network and load of the controller resources.

6. References

[1] Grabowski F., Bolanowski M., Paszkiewicz A.: Self organization and self adaptation of computer networks in nonextensive approach, vol.59, z.10, pp.1049-1053, PAK 2013.
[2] Mehdi S.A., Khalid J., Khayam S. A.: Revisiting Traffic Anomaly Detection Using Software Defined Networking. RAID 2011, LNCS 6961, pp. 161–180, Springer-Verlag Berlin Heidelberg 2011.
[3] Isolani P., Wickboldt J., Both C., Rochol J., Granville L.: Interactive monitoring, visualization, and configuration of OpenFlow-based SDN. Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on, pp. 207 – 215, IEEE 2015.
[4] Bolanowski M., Paszkiewicz A.: Nowy model detekcji zagrożeń w sieci komputerowej. T.89, s.308-311, z.11, Wydawnictwo Sigma-Not Sp. Z O.O., Przegląd Elektrotechniczny 2013.
[5] What is ONF https://www.opennetworking.org/images/stories/downloads/about/onf-what-why.pdf,
[6] http://www.opendaylight.org/software
[7] http://www.projectfloodlight.org/floodlight/,
[8] ONOS Architectural Progress, http://onos.wpengine.com/wp-content/uploads/2014/11/ONOS-architecture.pdf
[9] Cabaj K., Wytrębowicz J., Kukliński S., Radziszewski P., Truong Dinh K.: SDN Architecture Impact on Network Security. Position papers of the 2014 Federated Conference on Computer Science and Information Systems pp. 143–148, ACSIS, Vol. 3, PTI 2014.

Received: 26.06.2015 *Paper reviewed* *Accepted: 03.08.2015*

Marek BOLANOWSKI, PhD, eng.

He received a PhD degree in Computer Science from Lodz University of Technology in 2009. His current research interests focus on computer system and network design and interconnection network performance. He works at the Department of Distributed Systems Rzeszow University of Technology as an assistant professor.

e-mail: marekb@prz.edu.pl

Bogusław TWARÓG, PhD, eng.

He defended his doctoral thesis entitled "Information technologies in the assessment and prediction of states of electromechanical devices using artificial neural networks". Actually assistant professor at University of Rzeszow. Head of Real Time Diagnostic Systems Laboratory. He is the author of several scientific papers. His research interests are focused on the neural networks, artificial intelligence, robotics and industrial process control.

e-mail: btwarog@ur.edu.pl

Rafał MLICKI, B, eng.

Since 2010, a student of Information Technology on Rzeszow University of Technology. Graduated from a BEng Programme in 2014. His research interests focused on the computer network security.

e-mail: rmlicki@stud.prz.edu.pl