The possibility of using LACP protocol in anomaly detection systems

Marek Bolanowski[1,*], *Piotr* Cisło[2]

[1]Deptartament of Complex Systems, Faculty of Electrical and Computer Engineering, Rzeszow University of Technology, al. Powstańców Warszawy 12, 35-959 Rzeszów, Poland
[2] piotrekcislo@gmail.com, Poland

Abstract. This article presents the use of the Link Aggregation Control Protocol (LACP) for detection of anomalies in network traffic. The idea itself is based on checking the representativeness of a single LACP link for the whole traffic transmitted by the aggregation. This approach allows to reduce the requirements for the performance of threat detection systems, and thus reduce their implementation costs and the gives a possibility of using probes (IDS or IPS) directly in the core of the network. The authors also examine the influence of hashing algorithms used for the particular LACP link on the possibility of using of developed method and on the level of intrusion detection.

1 Introduction

Detection of threats in modern information systems is an important issue both in science and engineering. This results in the use of more and more advanced filters in the communication channels that are capable to analyze traffic up to and including the seventh layer of ISO/OSI model. The bandwidth of the links in which they are used increases with the increase in the number of probes IDS and IPS. Unfortunately, increasing throughput significantly affects the performance requirements of the implemented solutions for threats detection. It often limits their use, in particular for core links and connection points at the edge of large computer networks. Therefore, recently the concept of rough identification of the suspicious traffic expands greatly, which leads consequently to the separation of the suspicious traffic and to passing it to further, more sophisticated inspection. An identification of suspicious traffic is often performed based on the analysis of statistical properties of the traffic [1-7]. In [1] the authors present a method that allows to identify anomalies in the network traffic based on the change of the selected set of parameters of network traffic. An important aspect of this case is not only the change of parameters but also the value of these changes. The authors propose the use of statistical signatures to detect the anomalies and they confirm their effectiveness by analyzing the results for the real network attacks. Of course, other statistical method can be used in anomaly detection process, which are dedicated to change detection in non-stationary time series [8]. In [9-10] the authors propose the network systems architectures for implementing the anomaly detection system in real network system.

[*] Corresponding author: marekb@prz.edu.pl

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (http://creativecommons.org/licenses/by/4.0/).

In [11] the authors examine the statistical properties of a traffic sent through each link of the LACP aggregation, and check whether they can be representative of the whole transferred traffic. The results obtained in this work show that the traffic on a single LACP link, from the viewpoint of anomalies detection, is representative of the entire channel. The results obtained in these works encouraged the authors to try to check the implementation possibilities of this class of systems using the tools available on the market and check their effectiveness. An important question that arose at the initial stage of the work was to check the influence of load balancing algorithms of individual links in the LACP on the effectiveness of threats detection.

2 The LACP protocol

Compliant with IEEE 802.3ad link binding offers a method of aggregating many Ethernet links into a single logic LAG channel. This function helps to improve the efficiency of the device, increasing the total throughput of a single channel without the need of a hardware upgrade. In addition, IEEE 802.3ad standard offers the ability to dynamically assign, manage and monitor various aggregated links and enables interoperability between different devices from different vendors.

LACP supports the automatic creation of aggregating LAG groups by exchanging LACP-PDU packets between the physical interfaces. LACP packets are exchanged only between ports in active mode and are received by the ports operating in passive mode. The LACP switch thus "learns" the possibility of aggregation of ports dynamically, simultaneously diagnosing the state of the established connection with the LACP-PDU frame. LACP identifies correctly dedicated Ethernet links, what greatly simplifies the grouping of the links into the LAG group. LAG logical interfaces are added to the spanning tree (STP) as a single logic interface of the LAG group.

The IEEE 802.3ad protocol redirects the flows to the particular link of the LACP aggregation using specially implemented hash algorithms for this purpose. We get two methods available. The first hashing method is using a packet header of the second layer and of the third of ISO/OSI model. The second method is using a packet header of the third layer and of the fourth. In the case of IPv4 traffic the hashing may for instance run as follows:

- In the case of “bridged” aggregation, the hash is built with the following variables: L2 MAC source/destination and L3 IP source/destination.
- In the case of “routed” aggregation, the hash is built as follows: L3 IP source/destination and L4 PORTS source/destination.

In the case of IPv6 traffic hashing is carried out as follows:

- In the case of “bridged” aggregation, the hash is built with the following variables: L2 MAC source/destination and L3 IPv6.
- In the case of “routed” aggregation, the hash is built with the following variables: L3 IPv6 source/destination and L4 IPv6 source/destination.
- For Non-IP traffic, the hash is built with the following variables: L2 MAC source/destination.

Depending on the particular vendor of network switches we very often have different options of the configuration. Let us consider an example of a hash value configuration for the switch [12]:

```
forwarding-options {
    hash-key {
        family inet {
            layer-3;
            layer-4; }}}
```

We can choose in a detailed way the dedicated values of the hash calculation:

```
forwarding-options {
    hash-key {
       family inet {
            layer-3 {
                destination-address;
                incoming-interface-index;
                protocol;
                source-address;
            }
            layer-4 {
                destination-port;
                source-port;
                type-of-service; }}}}
```

The tests have shown that the more configuration possibilities of hashing algorithms we have - the more efficient the use of LAG is. If you choose only two or more hashing components you get bigger saturation of the individual interfaces of the aggregating group. If you choose a single method, for example “source-mac” or “source-ip”, a particular device connected to the LAG unit will take into account only the single features of the packets of the second layer ISO / OSI (MAC address) or of the third (the IP address of the machine) so that all packets will be sent by a single interface of the aggregation group. However, if you choose layer 4 additionally, and the source port method, it is possible for a single device to use the full capabilities of all aggregation interfaces. Depending on the implementation and configuration in a given type of switch or in a given operating system, you can receive different performance.

3 The physical and virtual structure of test environment

The environment for network anomaly detection has been created in accordance to the concept of universal hardware-network environment, with high capability of scaling, taking into account the idea of future migration to more efficient hardware environment. The guiding idea was to use opensource tools, and tools based on "for non - commercial use" licenses. In order to manage flexibly the server and network infrastructure on the server, the virtualization environment by using of VMware was installed. Within the virtual environment the following virtual switches have been created:
VS-IF-1 to **VS-IF-4** with a transmission capacity of one gigabit per second, which included the virtual cards of threat detection systems, aggregation hubs, and the physical interface used in case of mirroring LAG-IF1 to LAG-IF4 (respectively) traffic from an external switch.
VS-STREAM-IN - with a transmission capacity of one gigabit per second, which included the virtual cards of network traffic generators, and physical interfaces for transparent forwarding the traffic coming from the physical network devices.
VS-STREAM-OUT - with a transmission capacity of one gigabit per second, which included the virtual cards of network traffic generators, and physical interfaces for transparent forwarding the traffic coming from the physical network devices.
PENTEST-LAB - with a transmission capacity of one gigabit per second, isolated network created to manage virtual machines used for the network attacks.
VM-MGMT - with a transmission capacity of one gigabit, used for VMware system managing.

VM-Kernel - with a transmission capacity of one gigabit, used to perform the migration processes of the VMware system.

In the next step the following virtual machines have been created:

IDS-0 - the system of network threats detection, having six virtual cards, four virtual processors and four gigabytes of RAM. Additionally, resources was reserved, ie. a critical level of access to the processor time, memory, and a required space in data store resources.

LACP-0 and **LACP-1** aggregation traffic hub, having six virtual cards, two virtual processors and two gigabytes of RAM.

TXGEN-0 - network traffic generator, having sixteen virtual cards, two virtual processors and two gigabytes of RAM.

RXGEN-0 - network traffic generator, having sixteen virtual cards, two virtual processors and two gigabytes of RAM.

PEN-0 - network threats generating system having seventeen virtual cards, two virtual processors and two gigabytes of RAM.

Resources have been allocated to each of the machines depending on the planned load level and the tasks to be performed. Fig. 1 presents the structure of the test virtual environment taking into account interconnections and names of the machines, switches and network interfaces.

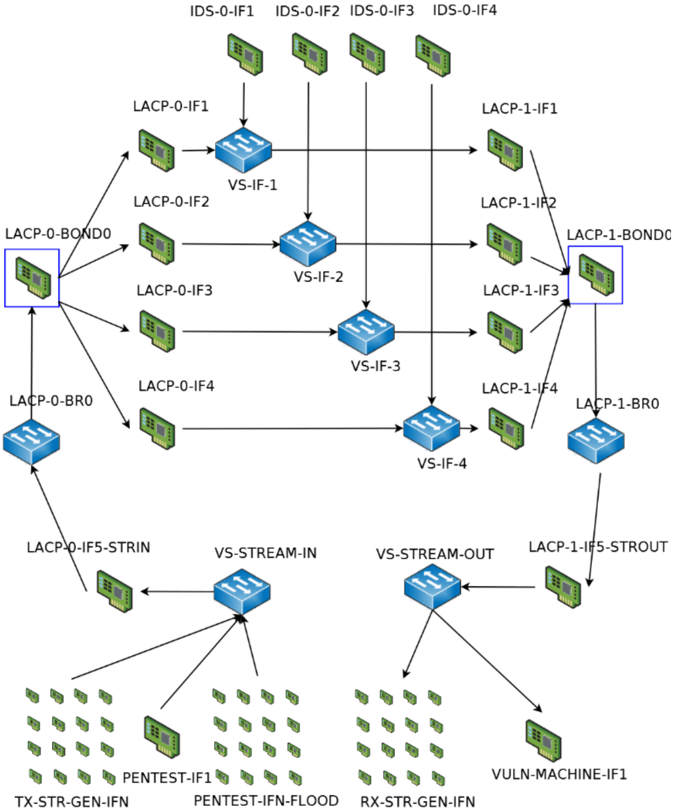

Fig. 1. Virtual structure of test environment.

The whole path leading from the virtual VS-STREAM-IN switch to the virtual VS-STREAM-OUT switch had the VLAN-s service at every point, so it was possible to transfer the TRUNK traffic (802.1q), and the normal ACCESS traffic. To the virtual VS-IF-1,2,3,4 switches the virtual IDS cards have been connected. Virtual and physical server interfaces have been connected to virtual VS-STREAM switches. The last step was the test of the configuration, which clearly confirmed the operation of the whole system. The virtual switch VS-STREAM-IN and VS-STREAM-OUT are connected with two physical interfaces of the server with a bandwidth of one gigabyte, and then the traffic from the ISP router was redirected to them. The traffic from the ISP had following attributes:

- The clients connected to PPPoE authentication server used the dedicated VLAN.
- IP-Multicast (TV) and VOIP used the dedicated VLANs.
- The average load of the client router was fifty megabits.
- Default VLAN (MGTVLAN) was not tagged with the standard identifier "1"

As a result of the performed test, the following parameters were changed: MTU to 9000 bytes and a dedicated resource disk was created to serve the queues and output files (tcpdump). The environment built this way enabled to run tests using both real traffic and traffic coming from the program generator.

4 The logical structure of test environment

In order to carry out the necessary statistical analysis of IEEE 802.3ad standard and of LACP protocol, it was necessary to run tests with previously prepared network and server environment, with particular traffic hashing methods. The study was divided into two stages. The first stage was to configure bound interfaces of the network switches so as to forward network traffic using the hashing algorithm of the second and the third layer. The second stage was to configure the network switches so that they serve the forwarding of network traffic using a hashing algorithm of the third and the fourth layer.

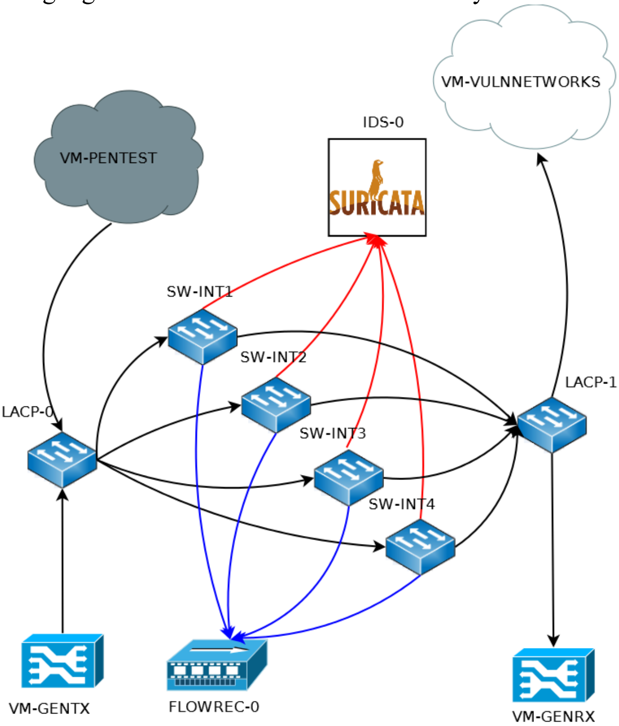

Fig. 2. Logical structure of test environment.

In order to obtain a homogeneous comparative data four tests have been planned, based on the same sets of attacks, using the same methods and the same virtual machines. A supplement for the statistical research was a detection of network threats by IDS system based

on Suricata software[13]. The Suricata system worked in inline mode, connected to virtual network switches operating between LACP switches. The logic diagram is shown in Fig.2. The VM-GENTX and VM-GENRX machines were responsible for traffic generated through LACP switches. The generated TCP traffic had the task to increase network load artificially between switches, in order to generate a traffic close to the real load in high-throughput networks. The task for the FLOWREC-0 machine was to copy periodically the traffic recorded in the particular time cycles from incoming attacks, from all connected interfaces. The collected data were then used to detect anomalies using statistical analysis.
The task of IDS-0 (2) machine was to detect network threats based on standard emergency signatures [14]. In the area of VM-PENTEST network, responsible for generating of sequent network attacks, there were machines based on Kali distributions, and there was a machine based on Debian distributions with installed and configured pytbull software. In the VM-VULNNETWORKS network there were standard Debian machines with running services of ftp, ssh, mysql, www, which were the objects of further attacks.

5 Results

In the first study the LACP network environment has been configured so that the hashing algorithm used only the properties of the second and the third layer. In order to determine the effectiveness of the proposed solution, a number of attacks was carried out. The results will be presented on the basis of four examples, it is: Brutaforce attack, Serverside attack, ClientSide attack, DoS attack. The test results have been shown in Fig. 3 Using hashing algorithms for L2 and L3 layers, all the traffic is directed to one interface and only on it the traffic analysis is possible - both in the case of detection using the statistical signatures, or in the case of detection using the Suricatta program. Samples of the traffic on particular interfaces are not statistically representative for the total flow, and their use for the anomaly detection is not possible.

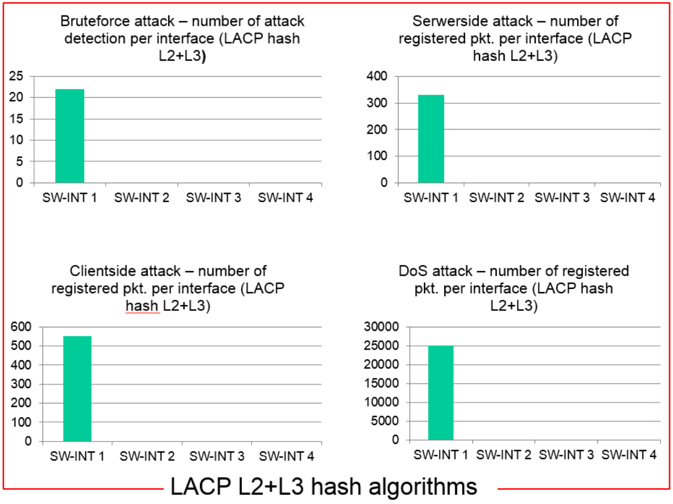

Fig. 3. Anomaly detection or number of packet per interface in case of using the LACP L2+L3 hash algorithms.

In next study the LACP network environment was configured so that the hashing algorithm used only the features of the third and the fourth layer. The test results are shown in Fig 4. Thanks to the use of hash algorithms L3 + L4 for LACP protocol, the traffic spread evenly on particular interfaces. Such a situation allows to carry out the detection using the statistical signatures, and using the Suricatta program on any interface. Number of successfully verified attacks on a single interface was in the range of 60% to 80%.

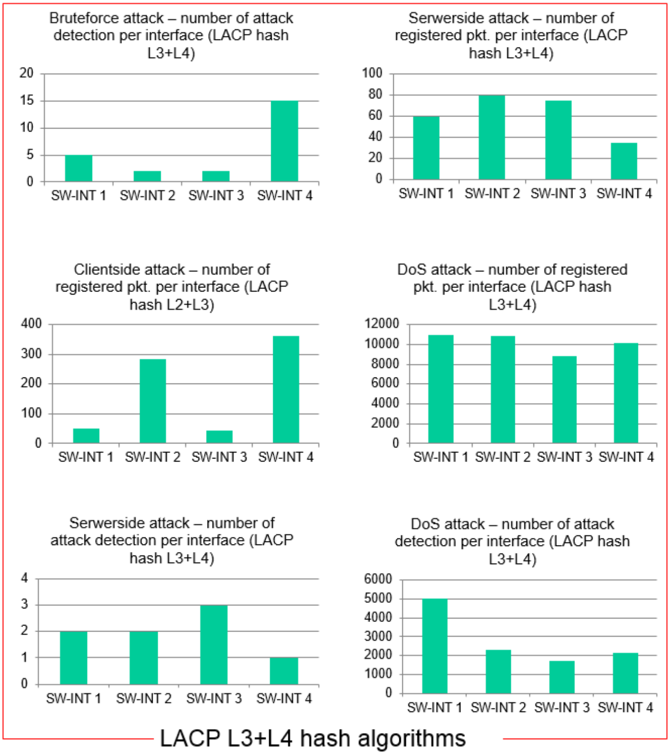

Fig. 3. Anomaly detection or number of packet per interface in case of using the LACP L3+L4 hash algorithms.

6 Conclusion

Detection of threats in computer networks is an important and current subject. A large number and intensity of attacks on modern information systems forces the producers of protection systems to a multistage process of detecting and blocking threats. In the study, the authors propose a method of rough identification of the attack based on a separate part of the flow. The LACP standard is an open standard and allows to divide a single communication channel into maximum eight logic channels. The tests have shown that with the use of hashing algorithms of L3 + L4 layer it is possible to identify 60 to 80% of attacks based on a single logic link. Thanks to this solution financial spendings for detection systems can be reduced, especially in the area of hardware performance. It should also be noted that the whole proposed solution was built on the basis of existing and widely implemented protocols and open source system. The described solution can also be easily implemented in a network environment of SDN type.

References

1. M. Bolanowski, A. Paszkiewicz, *The use of statistical signatures to detect anomalies in computer network,* Lecture Notes In Electrical Engineering, **324**, 251-260, (2015).
2. F. Simmross-Wattenberg, J. Asensio-Perez, P. Casaseca-de-la-Higuera, M. Martin-Fernandez, I. Dimitriadis, C. Alberola-Lope, *Anomaly Detection in Network Traffic*

Based on Statistical Inference and alpha-Stable Modeling, IEEE Transactions on Dependable and Secure Computing, **8**(4), 494-509, (2011).

3. W. Zhang, Q. Yang, Y. Geng, *A Survey of Anomaly Detection Methods in Networks*, Computer Network and Multimedia Technology, CNMT 2009, International Symposium on, 1-3, (2009).
4. S. Oshima, T. Nakashima, *Computational Complexity of AnomalyDetection Methods,* Seventh International Conference on Broadband, Wireless Computing, Communication and Applications, 664-649, (2012).
5. P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, E. Vázquez, *Anomaly-based network intrusion detection: Techniques, systems and challenges*, Computers & Security, **28**(1-2), 18-28, (2009).
6. W. Han, W. Xiong, Y. Xiao and others, *A class of Non-statistical Traffic Anomaly Detection in Complex Network Systems*, 32nd International Conference on Distributed Computing Systems (ICDCSW), 640-646, (2012).
7. F. Palmieri: *Network anomaly detection through nonlinear analysis*, Computers & Security, **29**(7), 737-755, (2010).
8. T. Pełech-Pilichowski, J.T. Duda, *Low-frequency signal reconstruction and abrupt change detection in non-stationary time series by enhanced moving trend based filters*, Studies in Computational Intelligence **579**, 111–125, (2015).
9. M. Bolanowski, B. Twaróg, R. Mlicki, *Anomalies detection in computer networks with the use of SDN*, Measurement Automation Monitoring **61**, 443-445, (2015).
10. M. Bolanowski, A. Paszkiewicz, *Nowy model detekcji zagrożeń w sieci komputerowej*, Przegląd Elektrotechniczny, **89**, 308-311, (2013).
11. M. Bolanowski, A. Paszkiewicz, M. Wroński, R. Żegleń, *Representativeness analysis and possible applications of partial network data flows*, Measurement Automation Monitoring, **62**(01), 29-32, (2016).
12. http://www.juniper.net/us/en/.
13. https://suricata-ids.org/.
14. https://rules.emergingthreats.net/open/suricata/rules.