Infrastructure for generating new IDS dataset

Jana Uramová, Pavel Segeč, Marek Moravčík, Jozef Papán, Martin Kontšek, Jakub Hrabovský
Faculty of Management Science and Informatics, University of Zilina,
Univerzitna 8215/1, 010 26 Zilina
e-mail: {jana.uramova, pavel.segec}@fri.uniza.sk

Abstract— **This article describes the proposal of a system that was designed for effective network monitoring, analysis of network infiltration, and archiving of network flows for their later research. The article contains the design of an appropriate infrastructure for such a system under the conditions of the Department of Information Networks. The objective was to develop a methodology for creating a custom dataset that would contain normal network traffic and traffic with various types of attacks. Dataset can be used to test network attack detection methods that the research team is working on at the Department of Information Networks. We started with detailed analysis of the available datasets, which is an important source of information for creating a custom dataset, and to identify the imperfections of these datasets, and requirements, that a trusted dataset should meet. The article describes the course of implementation, testing and optimization of one of the possible solutions of such a system. The final part of the article informs about designing the appropriate infrastructure, and using the system for dataset creation. Articel describes also tools which were used to perform DoS, DDoS attacks and botnet architecture. The article follows the work presented in** [1].

Keywords: IDS dataset, Intrusion Detection, Cybersecurity, DoS, DDoS, Cyberattacks, Botnet, PCAP, Moloch, Suricata, Evebox, Elasticsearch.

I. Introduction

With every new day, the need to protect the computer network against potential security breaches increases, as increasingly sensitive data is transmitted to the network with new trends and innovations. Attackers and creators of defensive mechanisms are constantly rivaling. The gateway and the basic element of network security can be considered to be a firewall whose role is to manage communication by filtering or restricting flows based on defined rules.

A more sopfisticated application-layer flow analysis can be performed by either Intrusion Detection Systems (IDS) or Intrusion Prevention Systems (IPS), whose function is not only detection and warning but also blocking dangerous activity. Although the rules of these systems are updated on a daily basis, they only protect the system against known types of attacks.

Current trends in defence against cyber threats become elements of artificial intelligence [2], behavioral analysis [3], [4] and machine learning [5]. Very interesting approach how to increase the security of a distributed computer system rapidly is hybrid honeypot featuring antonomous operation, presented in [6].

So-called zero-day attacks, or those whose behavioral algorithm is not yet described, prevent sophisticated detection methods, which are the current research problem. The problem involves processing a large amount of network traffic and detecting anomalies. Network traffic is continuously compared with long-term trends and attention is focused on the discovery of differences. The need to test the effectiveness of IDS systems on the relevant data sample arises.

Testing the effectiveness of IDS systems is a complex process. Our department is focused on several network security topics and one of them is development and analyzation of innovative methods for detecting network attacks. There are many approaches how to obtain a relevant dataset that could be used to test these innovative methods, but none of which are clearly stated to be the most appropriate. Majority of researchers evaluate the performance of IDS systems using so-called benchmark datasets. By dataset we mean the collection of benign and malicious data flows. Existing datasets have their drawbacks, with the underlying problem being the lack of attention paid to their evaluation. Significant advances in this area were the definition of 11 criteria to be met by a reliable dataset [7]. This dataset was created in 2017, published in January 2018 as CICIDS2017 [8].

Our approach is inspired by the CICIDS2017 dataset and involves the creation of a complex system that would process, archive packets from the network flow and, in the case of detecting attack activity, identify attack flows with the type of attack. Such a system involves deploying the archivation and detection tool distributed on multiple nodes in the network so that at any time of the recent period a PCAP traffic record is available, as well as a description of the streams with the indication of attacks and the identification of the type of attack. The system can be used for forensic analysis using one of the most widely open-source IDS systems as well as for creating benchmark datasets - in this case, flows are not marked based on a particular IDS system but manually according to the implemented attack scenarios to be designed for the given dataset. The dataset thus obtained can be used to test detection methods.

An analysis of the tools we have chosen to solve such a complex archiving system is described in our previous article [1]. In this article we introduce the implementation of the system, especially the mutual cooperation of the Moloch archiving tool, and the Suricata intrusion detection system. In section III. A, we discuss the issues of this collaboration, and the solution proposed by us.

Following a pattern from [9], we sharpened the saw and proactively began with the end in our minds. Our intention was to solve partial problems in the research project "Security of Information and Communication Networks" at the Department of Information Networks at the Faculty of Management Sciences and Informatics at University of Zilina. The project focuses on four plains of security, with our research team addressing tasks from plane 2:

Plane 1. Security architecture – involves solution of security architecture relationship to relevant partial architectures, paradigms of security architecture [10].

Plane 2. Infrastructure and data acquisition technology – involves creation of CC infrastructure (CC services and their security [11], [12]), collection of real network streams (monitoring [13], attack detection and archiving [1]), generate packet flows with available hardware and software generators.

Plane 3. Detection of attacks by analysis of network traffic flow – involves methods of modeling flows in the communications network, methods of detecting attacks on the communications network [14], analysis of signatures contained in packet headers, complexity of attack detection algorithms, new methods of attack detection, adaptation of methods to real-time detection algorithms, editing attack recognition methods

Plane 4. Implementing network traffic collection and new methods for real-time detection of attacks in FPGAs [15] and memristors [16], [17], [18], [19].

Our project team provides technological and infrastructural support for group 3 and 4 and gather required data and information about attacks and protocol weaknesses for group 1.

II. Analysis of existing datasets

We briefly list currently available datasets from previous years. By analyzing them, we tried to find inspiration on one side, but also to reveal their shortcomings and misapplied practices in their creation. The main source of inspiration was the article published by the Canadian Institute for CyberSecurity University of New Brunswick in January 2018 at the International Conference in Portugal [8].

A. *List of available datasets*

As part of the analysis of the current situation, we have devoted ourselves to the available datasets created by these organizations:

- DARPA (Lincoln Laboratory 1998-99)
- KDD 99 (University of California, Irvine 1998-99)
- DEFCON (The Shmoo Group, 2000-2002)
- CAIDA (Center of Applied Internet Data Analysis 2002-2016)
- LBNL (Lawrence Berkeley National Laboratory 2004-2005)
- CDX (United States Military Academy 2009)
- Kyoto (Kyoto University 2009)
- Twente (University of Twente 2009)
- UMASS(University of Massachusetts 2011)
- ISCX2012 (University of New Brunswick 2012)
- ADFA (Australian Defence Force Academy 2013)
- CICIDS 2017 (University of New Brunswick 2017)

Researchers from the Canadian Institute for CyberSecurity University of New Brunswick have created several datasets. Some cover a whole range of attacks, the most known ISCX2012, others focus on either the vulnerability of a single platform, Android Botnet [20] examining the vulnerability of Android software using Botnet, or using a single group of attacks, DoS dataset [21], Botnet dataset [22].

B. *Criteria for reliable dataset*

Since 1999, many researchers have been trying to design a framework for evaluating IDS datasets. The group of scientists mentioned in the previous paragraph sets out in its article eleven criteria that would inevitably have to be met by a comprehensive and credible dataset [7].

The dataset should meet the following criteria:

C1: Complete Network Configuration - is the basis for capturing the true effects of some attacks. In the network, it is essential to have a realistic configuration using all the features of firewalls, servers, routers and switches.

C2: Complete Traffic - is a sequence of packets from a source that may be a user's device, a router, or a switch to a destination that may be another host, multicast group, or broadcast domain. Generation techniques used should allow capture of all types of real, pseudo-real and artificial traffic.

C3: Labelled dataset - Labeling of streams is another important feature. If the correct tags are not available, the dataset is unusable and the results of the analysis are lost on credibility and validity. Captured traffic should not only contain common tags that characterize that stream (http, ssh, etc.), but also a flag indicating whether it is malicious or benign, as well as a tag describing the name and type of a particular attack.

C4: Complete Interaction - It is essential to capture all network interactions, both within the LAN and between the LAN, to interpret the anomalous behavior of the network correctly.C5: Complete Capture – Pre získanie relevantných výsledkov musí byť zaznamenaná i chybná a neoznačená prevádzka.

C6: Available Protocols - Operation should be generated by normal and less common transmissions of the most widely used protocols.

C7: Attack Diversity

C8: Anonymity – The ideal state is to publish a dataset containing the entire IP packet with payload and header. Most datasets contain packets whose body has been removed to protect privacy, which reduces their use for some detection mechanisms such as DPI (deep packet checking).

C9: Heterogenity) – In the infrastructure, it is possible to obtain information for creating a dataset from different sources, network traffic, files where the behavior of operating systems or other devices in the network is stored. A homogeneous dataset containing information from a single source can be used to analyze specific types of detection systems, while a heterogeneous dataset can be used for a complete test covering all aspects of the detection process.

C10: Feature Set – The dataset should also be useful for testing and analyzing the system or methods suggested by other researchers. The problem arises when calculating and analyzing related properties. Important features should be possible to extract if any of the sources mentioned in the previous criterion are used.

C11: Metadata – Most of the previous datasets had no documentation, or insufficient. If the documentation was available, it contained insufficient information about network configuration, victim operating systems, attackers and attack scenarios, which reduced the usability of the given datasets for research.

Based on the findings of previous research, they have created a dataset, which is the next generation of the ISCX2012 dataset. They tried to cover all 11 set criteria. This dataset was our starting point and main inspiration.

III. PROPOSAL FOR A SYSTEM FOR EFFECTIVE MONITORING AND ARCHIVATION

Our main goal was to design, deploy and evaluate the appropriate infrastructure to create a reliable dataset to compare different security breach detection methods as well as to effectively monitor and archive network flows so that they can be analyzed in detail later.

We divided the network infrastructure into two subnets. The first is a network of attackers, and the other is a victim network that includes a target server and legitimate users who are used as bots. The attack network is located in the CC system (OpenStack solution) and the victim network is placed in the KIS laboratory. Network infrastructure includes various devices such as L2 and L3 switch, firewall, servers and computers (Fig. 1). The archiving tool Moloch, running on the server connected to the switch, was responsible for capturing network traffic between the attackers' network and the victims' network. The switch to Moloch is configured with a SPAN port that provides mirroring of trafic. There is a problem with capturing network traffic within the victim network between the legitimate users (bots) and the target server, as this traffic does not pass through the switch with SPAN port and therefore can not be mirrored towards the Moloch tool, which we have solved by running tcpdump on the required target server. In the future, Moloch in distributed deployment would be a better solution.

A. Interconnection of Suricata, Moloch and EveBox

Our approach was to implement a solution that facilitates the collaboration of IDS Suricata [23], Moloch [24] and EveBox [25].

The implemented solution that enables Suricata, Moloch and Evebox tools to work together is not an officially issued solution. Therefore, there is no documentation describing the tasks of the individual tools, the installation procedure and the implementation of the solution or the required hardware requirements. The only source of information was the github repository of the user called hillar [26], who has developed an extension to allow traffic to be mapped on the basis of unified2 threat logs that both IDS Suricata and Snort have stored in the past [27]. Suricata, due to upgrades, stopped supporting logs storage in that format. Threat information is saving to the new JSON format. Snort did not stop supporting unified2 but the plugin was no longer developed. In the end, there is no support for newer versions of some packages, which were necessary for the proper functioning of other components of the proposed solution to work with the IDS. The author started working on developing a new JSON-based solution. Later, he has expanded cooperation with other users and together they provide an online training course dedicated to Moloch, Bro and Suricata in a single device. We estimate that this is also the reason why publicly available materials are not available to describe the solution in more detail. Our applied installation and deployment process is based on an analysis of the steps in the author's installation script, complemented by the knowledge gained in troubleshooting.

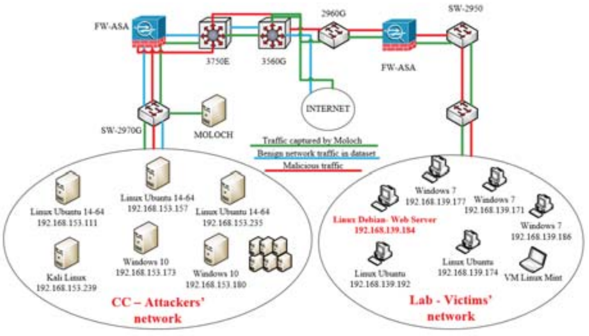


Figure 1. Our infrastructure and network flows

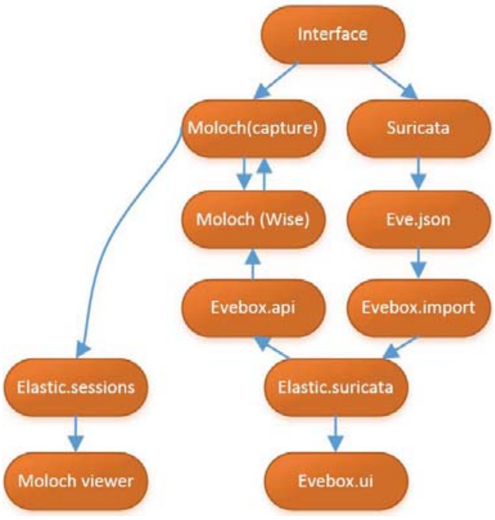


Figure 2. Scheme of collaboration between Moloch and Suricata

Fig. 2 describes the idea of this solution because Moloch and Suricata are working with packets from the same interface, thus working with data that share common characteristics.

In simple terms, Moloch's role lies in archiving received packets and analyzing their headers. Header information separates text strings in key_word = value. In this format, they are later send to the Elasticsearch database. The user environment of Moloch works only with the text information stored in the database, which provides quick data browsing in the User Interface.

In the event of a threat statement, Suricata writes the information that characterizes both the specific operation and the threat information associated with it in a text file *Eve.json*. Essential information for us is the source, destination IP addresses, source, destination port numbers, and time data. Less important for us are GeoIP location data for a registered public IP address (country coordinates, code, country name, and much more), protocol used and other information related to the Suricata instance (name, interface, alert ID, etc.).

The most important data describing the identified threat are:

- Category - uniquely identifies the threat category

- Severity - identifies the severity of the threat by the number (1-3).
- Signature - description of the rule by which the traffic is classified as a suspect.
- Signature_id - the number identifying a specific rule.
- Flow_id - a unique flow identifier

1) Evebox

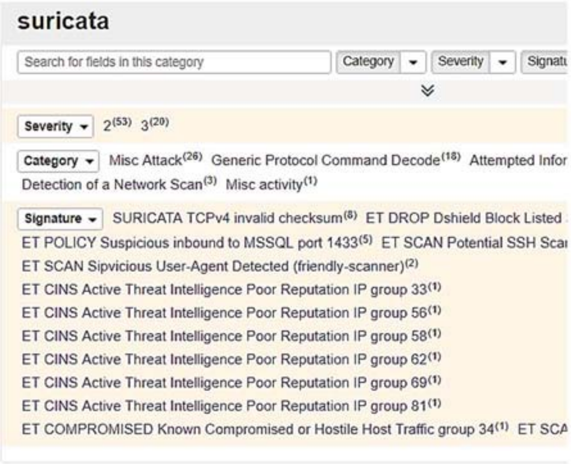
From the eve.json log file, threats are constantly red by Evebox [11]. This tool runs in two modes. The Evebox_import process has the role of importing threat warnings, alerts and index them into the Elasticsearch database. The second Evebox Server process provides management and access to a web-based graphical interface that helps to simplify archived threats. Threats can be filtered, displayed in different charts, archived, deleted, labeled as specific, or analyzed for detailed inforamtion. These threat notifications are automatically added to the user environment at the time of indexing. At the same time, the Evebox Application Interface creates a request to the extension of Moloch - Wise.

2) Moloch WISE

Moloch WISE activates the plugin source_suricata.js whose task is to unify the information format describing a a flow from Suricata with information about the same flow that has been indexed using the Moloch Capture modul to the Elasticsearch database. Both tools store all relevant information, such as IP addresses, as well as time data, in differente format. Plugin also takes care of cases where Moloch and Suricata have the opposite view of target and source IP addresses and ports for the same stream. Based on the match found, the plugin adds tags for the detected threat, including the category, severity, rule name, flow and signature, to the flow information in Moloch. This allow us to search for data according to given tags in the Moloch Viewer, which ensures the full functionality of the web interface to access captured data and statistics. Figure 3 shows a section of the user environment in Moloch Viewer illustrating the tags added by the successful integration of Moloch and IDS Suricata.

B. Export to CSV file

In order to meet C11, it was necessary to provide a captured PCAP file with a CSV file containing header information that uniquely characterizes the flow, but especially information about whether the stream is part of the attack or is a benign communication. As a solution, we chose to export directly from the Moloch tool, where we can add tags to streams based on the filter we specify - for example, with respect to the types of attacks we generated in the dataset when it was created.



suricata

Severity $2^{(53)}$ $3^{(20)}$

Category Misc Attack$^{(26)}$ Generic Protocol Command Decode$^{(18)}$ Attempted Infor
Detection of a Network Scan$^{(3)}$ Misc activity$^{(1)}$

Signature SURICATA TCPv4 invalid checksum$^{(8)}$ ET DROP Dshield Block Listed
ET POLICY Suspicious inbound to MSSQL port 1433$^{(5)}$ ET SCAN Potential SSH Scai
ET SCAN Sipvicious User-Agent Detected (friendly-scanner)$^{(2)}$
ET CINS Active Threat Intelligence Poor Reputation IP group 33$^{(1)}$
ET CINS Active Threat Intelligence Poor Reputation IP group 56$^{(1)}$
ET CINS Active Threat Intelligence Poor Reputation IP group 58$^{(1)}$
ET CINS Active Threat Intelligence Poor Reputation IP group 62$^{(1)}$
ET CINS Active Threat Intelligence Poor Reputation IP group 69$^{(1)}$
ET CINS Active Threat Intelligence Poor Reputation IP group 81$^{(1)}$
ET COMPROMISED Known Compromised or Hostile Host Traffic group 34$^{(1)}$ ET SCA

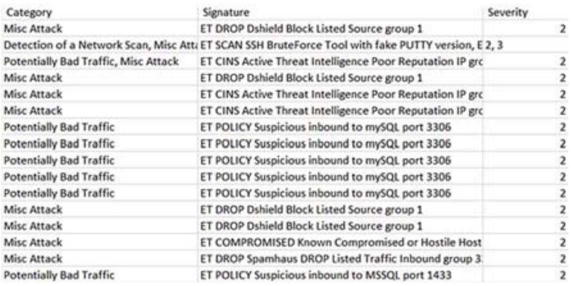
Figure 3. A sample of records in Moloch with the information provided by Suricata

Moloch offers the ability to export filtered data to PCAP file and also CSV format. In the case of CSV, all the columns that the user has activated in the UI on the Sessions panel are recorded in the file. Moloch allows us to add any tags in the form of text to any stream. One of the advantages offered by Molocha with Suricata is the possibility of automated inclusion of the tag glued from Suricata to the flow directly into the CSV file and thus we have not only the information that the flow is an offensive but also a description of the attack category or the name of the rule. For each official rule, it is possible to look at the detailed form of the signature and the full description of the rule.

Export has caused many problems from the very beginning. Although we added the associated columns in the UI in the resulting file, only column headers appeared, but not specific values. Looking at the SPI metadata stored in the database, we found that the missing information in the resulting file is located in a given stream in the database. To resolve the issue, we were looking for help on the Slack channel of the tool [12]. The authors recorded the error and was removed in the latest version of 0.50.1, which we subsequently updated the software. Figure 4 shows a section of the resulting CSV file, which also contains the threat information taken with Suricata.

We encountered problems trying to export data for a longer period of time. Defining what means a larger interval in this context is problematic. It is better to talk about the number of flows that Moloch will try to access when exporting. The same number of streams may be a completely different time interval because it depends on the number of sessions per minute and their lifetime. The MaxPackets parameter specifies the maximum number of packets captured in a single session. The default value is 10,000 packets, and if it is exceeded, other packets are recorded as a new session even though the original session was not actually completed.

Our second recommendation is to use Moloch Viewer on a more powerful server if we plan to use it for exportating of large files (PCAP, or CSV). In our case, the Elasticsearch database is also on this server, and the Capture module is paused.



Category	Signature	Severity
Misc Attack	ET DROP Dshield Block Listed Source group 1	2
Detection of a Network Scan, Misc Atta	ET SCAN SSH BruteForce Tool with fake PUTTY version, E	2, 3
Potentially Bad Traffic, Misc Attack	ET CINS Active Threat Intelligence Poor Reputation IP grc	2
Misc Attack	ET DROP Dshield Block Listed Source group 1	2
Misc Attack	ET CINS Active Threat Intelligence Poor Reputation IP grc	2
Misc Attack	ET CINS Active Threat Intelligence Poor Reputation IP grc	2
Potentially Bad Traffic	ET POLICY Suspicious inbound to mySQL port 3306	2
Potentially Bad Traffic	ET POLICY Suspicious inbound to mySQL port 3306	2
Potentially Bad Traffic	ET POLICY Suspicious inbound to mySQL port 3306	2
Potentially Bad Traffic	ET POLICY Suspicious inbound to mySQL port 3306	2
Potentially Bad Traffic	ET POLICY Suspicious inbound to mySQL port 3306	2
Misc Attack	ET DROP Dshield Block Listed Source group 1	2
Misc Attack	ET DROP Dshield Block Listed Source group 1	2
Misc Attack	ET COMPROMISED Known Compromised or Hostile Host	2
Misc Attack	ET DROP Spamhaus DROP Listed Traffic Inbound group 3	2
Potentially Bad Traffic	ET POLICY Suspicious inbound to MSSQL port 1433	2

Figure 4. Exported CSV file with Suricata tags

IV. Description of Malicious Traffic in Our Dataset

To meet the criterion C7, our effort was to cover a sufficient number of different attacks in our datasheet.

Our dataset captures a 10-hour network traffic of one day starting at 08:00 in the morning and ending at 18:00 in the evening. Malicious traffic is generated from 12:26 with minor breaks until 15:40. In Table 1 we can see the list of attacks we generate.

The first tool we've used is Bonesi. Using this tool, we have simulated a flood of UDP datagrams with source devices that appear in our topology as regular users but become part of the botnet.

We also used CnC-Botnet-in-Python, which runs on our device in our OpenStack Cloud at the Department with IP address 192.168.153.235. We have a list of bots that we control with the Command and Control server and with common commands. Using the Command and Control server, we installed the Slowhttptest tool on a regular device with IP address 192.168.139.192 in the laboratory. Then, using commands on Command and Control, the server started the slowhttptest attack tool.

After a short break, the DDoS attack was launched using GoldenEye.

Scanning methods were then performed to detect open ports and device vulnerabilities using the Nmap, Nikto, Vmap, and Nessus tools. At this stage, a brute force attack was performed to guess the SSH password using the Patator tool.

In the penultimate phase, several DDoS attacks were performed using various tools. First we used Slowloris, which was very effective, after ten seconds the web server did not respond to legitimate users' requests. Only when the tool launched on both devices was shut down, the server was available again. In addition, the HULK tool was launched on two attacking devices, but the server still ran without side effects. Subsequently, LOIC was launched on Windows devices with different variations, firstly a flood of TCP packets, then a flood of UDP packets, and finally an HTTP attack. Last but not least, we used the Scapy tool, with which we made a flood of ICMP packets and so-called SYN Flood.

Botnets were used in the last phase. The first botnet was Ares, where we downloaded the agentAres.exe from the website. When we ran it, there was a connection between these bots and the Command and Control server, where we saw the attached bots. This Botnet was used to screen the activity of current bots. The second botnet we used was the Athena HTTP botnet. We also downloaded the agentAthena.exe file from the website. After the file was started, bots joined the Command and Control server but did not execute the commands.

In our approach, we focused on the tools available to create botnets so that we can cover a wider range of attacks. However, a major problem was the acquisition of specific botnets, which were very often mentioned and could be used for our purposes.

TABLE I.
LIST OF ATTACKS IN OUR DATASET

	Attack	Attacker	Victim
Thursday 23.5.2018 (8:00-16:00)	Bonesi (12:26:44 – 12:30:15)	192.168.139.177 192.168.139.171 192.168.139.186 192.168.139.192 192.168.139.174	WebServer Apache 192.168.139.184
	Slowhttptest (13:18:48 - 13:20:54)	Linux Ubuntu 192.168.139.192	WebServer Apache 192.168.139.184
	GoldenEye (13:32:57 - 13:36:13)	Linux Ubuntu 192.168.153.157	WebServer Apache 192.168.139.184
	GoldeEye (13:33:24 - 13:36:24)	Linux Ubuntu 192.168.153.235	WebServer Apache 192.168.139.184
	Nmap (13:50:24 – 13:50:42)	Kali Linux 192.168.153.239	WebServer Apache 192.168.139.184
	Nmap (13:54:25 – 13:54:34)	Kali Linux 192.168.153.239	Router 158.193.138.291
	Nikto (13:58:19 – 13:58:38)	Kali Linux 192.168.153.239	WebServer Apache 192.168.139.184
	Vmap (14:03:58 – 14:04:25)	Kali Linux 192.168.153.239	WebServer Apache 192.168.139.184
	Nessus (14:10:41 – 14:13:05)	Kali Linux 192.168.153.239	WebServer Apache 192.168.139.184
	SSH-Patator (14:17:23 - 14:25:15)	Kali Linux 192.168.153.239	WebServer Apache 192.168.139.184
	Slowloris (14:33:39 – 14:36:14)	Linux Ubuntu 192.168.153.157	WebServer Apache 192.168.139.184
	Slowloris (14:34:42 – 14:36:56)	Linux Ubuntu 192.168.153.235	WebServer Apache 192.168.139.184
	HULK (14:45:13 – 14:52:59)	Linux Ubuntu 192.168.153.111	WebServer Apache 192.168.139.184
	HULK (14:45:29 – 14:51:37)	Linux Ubuntu 192.168.153.235	WebServer Apache 192.168.139.184
	LOIC (15:00:01 – 15:02:44)	Linux Ubuntu 192.168.153.173	WebServer Apache 192.168.139.184
	LOIC (15:01:07 – 15:04:28)	Linux Ubuntu 192.168.153.180	WebServer Apache 192.168.139.184
	LOIC (15:03:12 - 15:05:14)	Linux Ubuntu 192.168.153.173	WebServer Apache 192.168.139.184
	Scapy (15:10:23 – 15:11:32)	Linux Ubuntu 192.168.153.111	WebServer Apache 192.168.139.184
	Scapy (15:12:05 – 15:12:59)	Linux Ubuntu 192.168.153.111	WebServer Apache 192.168.139.184
	Botnet Ares (15:23:46 – 15:24:40)	Linux Ubuntu 192.168.153.111	Win7 192.168.139.177 Win7 192.168.139.171 Win7 192.168.139.186
	Botnet Athena HTTP (15:27:00 – 15:40:55)	Linux Ubuntu 192.168.153.235	Win7 192.168.139.177 Win7 192.168.139.171 Win7 192.168.139.186

V. Dataset evaluation

In terms of criteria' fulfillment, our dataset fulfill 8 criteria, and we will work to address issues related to criteria C6, C8 and C11 in the future.

- C1: Complete Network Configuration – a variety of facilities are part of the infrastructure, see Fig. 1.
- C2: Complete Traffic – dataset contains bening and malicious traffic. Bening traffic is generated by 300 virtual machines in our IaaS cloud. We are missing normal traffic from the guest who worked as a victim and the web server we attacked. This will be our goal to complement in our second attempt to create a dataset. Packet header data is unchanged; payload is not removed.
- C3: Labelled dataset – normal traffic and malicious taffic are tagged in CSV file.
- C4: Complete Interaction, C5: Complete Capture –we captured traffic between the bot and the target server using tcpdump. Moloch captured all other network traffic. Duplicate records have been deleted in the datasheet.
- C7: Attack Diversity – the most common attacks were used, such as DoS, DDoS, scanning, brute force attack, and Botnet.
- C9: heterogenita – dataset contains information from different sources in the infrastructure
- C10: Feature Set – it is possible to extract information from any source.

In the future, we plan to work on these criteria, which require some improvements from our side:

- C6: Available Protocols – our dataset contains common protocols like HTTP, HTTPS, ICMP, DNS, NTP, RDP, SIP, SNMP, SSH. We plan to extend this set, considering that variety of benign user traffic is one of the basic criteria that almost any from actually available datasets does not meet entirely. The exception is the CICIDS 2017 dataset, where the traffic was generated by ten agents who were profiled using 25 users' behaviour with the help of machine-learning. We will continue to work on improvements in this area in the near future.
- C8 Anonymity.
- C11: Metadata – socumentation will be modified after the previous two criteria have been incorporated and will be available with full information.

VI. conclusion

We presented a design of such infrastructure that will allow effective monitoring of network streams from multiple locations in the network, with high-performance detection using multi-core processors, network flows archivation for later analysis, and provides a full range of capabilities for further processing of the captured and archived records. We used interconnection of available open-source tools Moloch, Suricata and EveBox.

We used presented infrastructure to create a custom dataset that anyone can use to test attack detection methods. In order to designate our dataset as reliable dataset, it will still be necessary to incorporate three criteria from the 11 criteria defined by [7] as the basic criteria for a complex, reliable and widely usable IDS dataset. In the future, we plan to create a new dataset that will meet all 11 criteria and make it publicly available.

VII. Acknowledgement

This paper is supported by Faculty of Management Science and Informatics of University of Žilina, funded by research grant number FVG/52/2018 and FVG/37/2018.

VIII. References

[1] J. Uramova, P. Segec, M. Moravcik, J. Papan, T. Mokos, and M. Brodec, "Packet capture infrastructure based on Moloch," in *ICETA 2017 - 15th IEEE International Conference on Emerging eLearning Technologies and Applications, Proceedings*, 2017.

[2] E. Tyugu, "Artificial intelligence in cyber defense," *2011 3rd Int. Conf. Cyber Confl. proceedings, 7-10 June, 2011, Tallinn, Est.*, 2011.

[3] K. N. Mallikarjunan, S. Prabavathy, K. Sundarakantham, and S. M. Shalinie, "Model for cyber attacker behavioral analysis," in *2015 IEEE Workshop on Computational Intelligence: Theories, Applications and Future Directions (WCI)*, 2015, pp. 1–4.

[4] W. Hurst, M. Merabti, and P. Fergus, "Big Data Analysis Techniques for Cyber-threat Detection in Critical Infrastructures," in *2014 28th International Conference on Advanced Information Networking and Applications Workshops*, 2014, pp. 916–921.

[5] Y. Xin *et al.*, "Machine Learning and Deep Learning Methods for Cybersecurity," *IEEE Access*, vol. 6, pp. 35365–35381, 2018.

[6] E. Chovancová *et al.*, "Securing Distributed Computer Systems Using an Advanced Sophisticated Hybrid Honeypot Technology," *Comput. INFORMATICS*, vol. 36, no. 1, pp. 113–139, 2017.

[7] I. Sharafaldin, A. Gharib, A. H. Lashkari, and A. A. Ghorbani, "Towards a Reliable Intrusion Detection Benchmark Dataset," *Softw. Netw.*, vol. 2017, no. 1, pp. 177–200, Jan. 2017.

[8] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, 2018, vol. 1, pp. 108–116.

[9] M. Sedliacikova, M. Minarova, and D. Mala, "Emotional Intelligence of Manageres," *Procedia Econ. Financ.*, vol. 26, pp. 1119–1123, 2015.

[10] M. Drozdova, S. Rusnak, P. Segec, J. Uramova, and M. Moravcik, "Contribution to cloud computing security architecture," in *ICETA 2017 - 15th IEEE International Conference on Emerging eLearning Technologies and Applications, Proceedings*, 2017.

[11] M. Moravcik, P. Segec, J. Uramova, and M. Kontsek, "Teaching cloud computing in cloud computing," in *ICETA 2017 - 15th IEEE International Conference on Emerging eLearning Technologies and Applications, Proceedings*, 2017.

[12] M. Moravcik, P. Segec, J. Papan, and J. Hrabovsky, "Overview of cloud computing and portability problems," in *2017 15th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, 2017.

[13] F. Hock and P. Kortis, "Design, implementation and monitoring of the firewall system for a DNS server protection," in *2016 International Conference on Emerging eLearning Technologies and Applications (ICETA)*, 2016, pp. 91–96.

[14] A. Smieško, Juraj; Kováč, "Detection of DDoS attack by Hurst coefficient," *Elektrorevue*, vol. 19, no. 1, pp. 32–38, 2017.

[15] J. Hrabovsky, P. Segec, M. Moravcik, and J. Papan, "Systolic-based 2D convolver for CNN in FPGA," in *2017 15th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, 2017, pp. 1–7.

[16] M. Klimo, P. Tarábek, O. Šuch, J. Smieško, and O. Škvarek, "Implementation of a deep ReLU neuron network with a

memristive circuit,” *Int. J. Unconv. Comput.*, vol. 12, no. 4, pp. 319–337, 2016.

[17] M. Klimo, O. Such, O. Skvarek, and M. Fratrik, “Memristor-based pattern matching,” *Semicond. Sci. Technol.*, vol. 29, no. 10, Oct. 2014.

[18] M. Fratrik, S. Badura, M. Klimo, and O. Skvarek, “Memristor measurements and simulations,” in *CAS 2013 (International Semiconductor Conference)*, 2013, pp. 243–246.

[19] M. Klimo and O. Šuch, “Memristive implementation of fuzzy logic for cognitive computing,” in *Future computing 2017*, 2017.

[20] A. F. A. Kadir, N. Stakhanova, and A. A. Ghorbani, “Android Botnets: What URLs are Telling Us,” in *International Conference on Network and System Security*, 2015, pp. 78–91.

[21] H. H. Jazi, H. Gonzalez, N. Stakhanova, and A. A. Ghorbani, “Detecting HTTP-based application layer DoS attacks on web servers in the presence of sampling,” *Comput. Networks*, vol. 121, pp. 25–36, Jul. 2017.

[22] E. Biglar Beigi, H. Hadian Jazi, N. Stakhanova, and A. A. Ghorbani, “Towards effective feature selection in machine learning-based botnet detection approaches,” in *2014 IEEE Conference on Communications and Network Security*, 2014, pp. 247–255.

[23] B. Brumen and J. Legvart, “Performance analysis of two open source intrusion detection systems,” in *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2016, pp. 1387–1392.

[24] “Moloch - large scale, open source, full packet capturing, indexing, and database system.” [Online]. Available: https://molo.ch/. [Accessed: 12-Oct-2018].

[25] “EveBox - web based alert and event management tool for events generated by the Suricata network threat detection engine.” [Online]. Available: https://evebox.org/. [Accessed: 12-Oct-2018].

[26] Hillar, “Cyber Defence Monitoring Course Site.” [Online]. Available: https://github.com/hillar/CDMCS. [Accessed: 12-Oct-2018].

[27] Hillar, “Pigsty-Moloch plugin,” 2014. [Online]. Available: https://github.com/hillar/pigsty-moloch-plugin. [Accessed: 12-Oct-2018].