COMPARATIVE ANALYSIS OF INTRUSION PREVENTION SYSTEM

BY

MD. NAZRUL ISLAM

ID: 151-15-5503

This Thesis Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Dr. Sheak Rashed Haider Noori

Associate Professor and Associate Head

Department of CSE Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY DHAKA, BANGLADESH

DECEMBER 2018

APPROVAL

This Thesis titled **"Comparative Analysis of Intrusion Prevention System**", submitted by Md. Nazrul Islam, ID No: 151-15-5503 to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 11th December 2018

BOARD OF EXAMINERS

Dr. Syed Akhter Hossain Professor and Head Department of Computer Science and Engineering Faculty of Science & Information Technology Daffodil International University

Narayan Ranjan Chakraborty Assistant Professor Department of Computer Science and Engineering Faculty of Science & Information Technology Daffodil International University

Md. Tarek Habib Assistant Professor Department of Computer Science and Engineering

Faculty of Science & Information Technology Daffodil International University

Dr. Mohammad Shorif Uddin Professor

Department of Computer Science and Engineering Jahangirnagar University

Chairman

Internal Examiner

Internal Examiner

External Examiner

DECLARATION

I hereby declare that, this thesis has been done by me under the supervision of **Dr. Sheak Rashed Haider Noori, Associate Professor and Associate Head, Department of CSE** Daffodil International University. I also declare that neither this thesis nor any part of this thesis has been submitted elsewhere for award of any degree or diploma.

Supervised by:

Dr. Sheak Rashed Haider Noori Associate Professor and Associate Head Department of CSE Daffodil International University

Submitted by:

Md. Nazrul Islam ID: 151-15-5503 Department of CSE Daffodil International University

ACKNOWLEDGEMENT

First I express my heartiest thanks and gratefulness to almighty Allah for His divine blessing makes me possible to complete this thesis successfully.

I fell grateful to and wish my profound my indebtedness to **Dr. Sheak Rashed Haider Noori, Associate Professor and Associate Head**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of my supervisor in the field of network security influenced me to carry out this. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior draft and correcting them at all stage have made it possible to complete this.

I would like to express my heartiest gratitude to **Dr. Sheak Rashed Haider Noori, Associate Professor and Associate Head**, for his kind help to finish my thesis and also to other faculty member and the staff of CSE department of Daffodil International University.

I would like to thank my entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

ABSTRACT

This thesis is on "**Comparative Analysis of Intrusion Prevention System.**". The main domain of this thesis is network security. Intrusion Prevention System is a well-known and important part of network security. Intrusion Prevention System provides critical infrastructures security by preventing intrusions in the network and computer systems. The aim of this thesis is to learn more about Intrusion Prevention System, know their implementation procedures, knowledge gathering on deep level packet inspection and find out the performance of most common open Intrusion Prevention Systems. In this study two most common Intrusion Prevention System (Snort and Suricata) is used to learn and experiment the performance on latest intrusion dataset named CICIDS2017. Performance are measured based on the CPU Utilization, Packet Processing speed, and on detection and prevention accuracy rate. Detection and prevention accuracy is measured using data mining techniques where different Machine Learning algorithms has been used.

TABLE OF CONTENTS

PAGE **CONTENTS** i Board of examiner ii Declaration Acknowledgements iii Abstract iv Table of contents v-vii viii List of tables List of figures ix-x

CHAPTER

CHAPTER 1: INTRODUCTION	1-7
1.1 Introduction	1
1.1.1 Snort	3
1.1.2 Suricata	4
1.2 Motivation	5
1.3 Research Questions	6
1.4 Expected Output	6
1.5 Thesis Layout	6

CHAPTER 2: BACKGROUND	8-9
2.1 Related Works	8
2.2 Research Challenges	9
CHAPTER 3: RESEARCH METHODOLOGY	10-12
3.1 Introduction	10
3.2 Research Design	10
3.3 Lab Architecture	10
3.4 Dataset Collection	11
	12.02
CHAPTER4:REQUIREMENTANALYSIS,INSTALLATION AND CONFIGURATION	13-23
4.1 Introduction	13
4.2 Requirements Analysis	13
4.3 Requirement Installation	14
4.3.1 Snort, DAQ, Barnyard2, PulledPork and WebSnort Installation	15
4.3.2 Suricata Installation	19
	24-30
CHAPTER 5: EXPERIMENTAL RESULTS AND DISCUSSION	24-30
5.1 Introduction	24
5.2 Experimental Results on CPU Utilization	24
5.3 Experimental Results on CICIDS2017 dataset	28

5.4 Descriptive Analysis and Results Comparison	29
5.5 Summary	30
	24
CHAPTER 6: CONCLUSION AND FUTURE WORKS	31
6.1 Conclusion	31
6.2 Future Work	31
APPENDIX	32-35
Appendix A: List of Abbreviation	32
Appendix B: Related Issues	33-35
REFERENCES	36-37

LIST OF TABLES

TABLES	PAGE NO
Table 3.1: Attack Types and flows in CICIDS2017	2
Table 4.1: Overview of hardware and software requirements	13
Table 4.2: Specific version of used software	14
Table 5.1: Experimental results of Suricata	28
Table 5.2: Experimental results of Snort	28
Table 5.3: Key difference between Suricata and Snort	30

LIST OF FIGURES

FIGURES	PAGE NO
Figure 1.1: Intrusion Prevention System Placement	2
Figure 1.2: Architecture of Snort	3
Figure 1.3: Architecture of Suricata	5
Figure 3.1: Research Design	10
Figure 3.2: Lab Architecture	11
Figure 4.1: nghttp2_install function	15
Figure 4.2: Running Snorter_IPS.sh script	16
Figure 4.3: Downloading DAQ and Snort with automated script	17
Figure 4.4: NFQ DAQ modules functions	17
Figure 4.5: Snort	18
Figure 4.6: PulledPork download IPS rules automatically	19
Figure 4.7: Installing Suricata dependencies	20
Figure 4.8: Suricata installation files	20
Figure 4.9: NFQueue support of Suricata	21
Figure 4.10: Suricata installation	21
Figure 4.11: Suricata detection and prevention rules	22
Figure 4.12: Suricata ICMP detection and prevention rules	22
Figure 4.13: Suricata configuration	23
Figure 5.1: Suricata I/O of packet	24
Figure 5.2: Snort I/O of packet	25

Figure 5.3: Interface of atop tool.	25
Figure 5.4: CPU usage of Suricata	26
Figure 5.5: CPU Usage of Snort	26
Figure 5.6: Analysis of UDP packet using Wireshark	27
Figure 5.7: Multi-thread, Multi-thread CPU affinity of Suricata	29
Figure A1: Atop output for experimenting CPU utilization	33
Figure A2: Checking Snort NFQ mode	33
Figure A3: Prove of Suricata's multi-threading capability	34
Figure A4: Netfilter Queue (NFQ)	34
Figure A5: Packet dropping in Suricata	35
Figure A6: Packet dropping in Snort	35

CHAPTER 1 INTRODUCTION

1.1 Introduction

The worldwide system named due to the fact the Internet has become a part and parcel of our existence. Consistently peoples have interaction with the Internet and plenty of them link their life with it. The Internet carried out numerous parts of life for example banking, shopping, learning, installments, business, payments and transactions. In this term due to the rapid growth of computer networks during the past two decades security has turn into a critical issue for the Internet. This quick growth has exposed computer networks to an increasing number of security threats. There are a variety number of security threats such as worms, viruses, adware, malware and approach to hack something on Internet developing every day. The threats don't seem to be solely to computers and hardware that we tend to connect with the Internet, however to the information and knowledge that resides among that infrastructure.

There are a lot of diverse ways and technique to increase the security of network and computer systems. However, in this study, I focus on Intrusion Prevention Systems (IPS). IPS are a hardware device or software system of network and computer security which detect and prevent intrusive activity both from insider network and outsider network. They cowl the large part of network security which allow us to manage major aspects. The aim of the Intrusion Prevention System is to prevent different kinds of intrusions and activities that are very dangerous for network and computer systems. Intrusions can be an attack against privilege escalation, unauthorized access to various sensitive files, network attacks against different critical vulnerable services, actions of harmful malware can be Trojans, viruses and worms. In general, IPS are placed either after or before the placement of firewall device in an organized network. In Figure 1.1, indicates general placement of Intrusion Prevention System in a pictorial format.

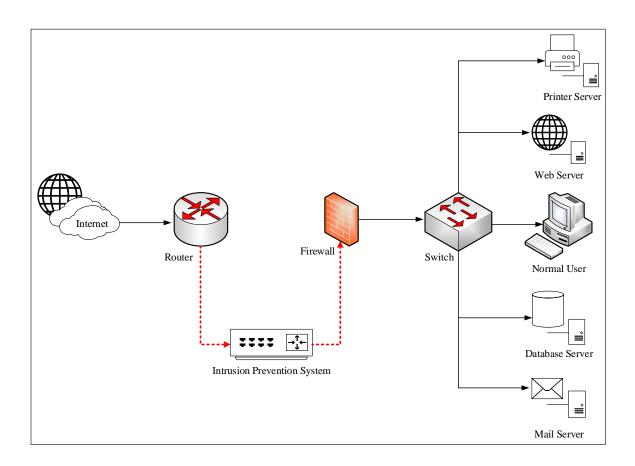


Figure 1.1: Intrusion Prevention System Placement

There are many valid ways to classify the Intrusion Prevention Systems. Scarfone et. al., [1] have used three types of IPS classification in a research. These are i) Host-based Intrusion Prevention Systems (HIPS), ii) Network-based Intrusion Prevention System (NIPS), and iii) Wireless Intrusion Prevention System (WIPS). Purpose of these IPS are given below.

- Host-based Intrusion Prevention System (HIPS): Host-based IPS detect and prevent intrusions that are generally affect end user. These type of IPS analyze traffics those are communicate with between the insider program and the internet or external network of a host. Host-based IPS must be installed to a host to make it workable.
- Network-based Intrusion Prevention System (NIPS): Network-based IPS monitors the network traffic and prevent suspicious data stream or packet. NIPS are work as a router also. All the traffics are passed over NIPS in network.

• Wireless Intrusion Prevention Systems (WIPS): Wireless IPS monitor actions in the wireless networks. Generally, it prevent the network from man-in-the-middle attacks, MAC address spoofing, wrong configured wireless access points and so on.

This study is conducted based on the performance, and prevention accuracy comparison of two most famous free and open source Network-based Intrusion Prevention called Snort and Suricata. These NIPS are helping the network security community way better.

1.1.1 Snort

The Snort IDS and IPS system became a worldwide famous feature to protect network. Snort is built based on five import unique module. There are i) Packet capture, ii) Packet Decoder, iii) Preprocessor, iv) Detection Engine and v) Output module.

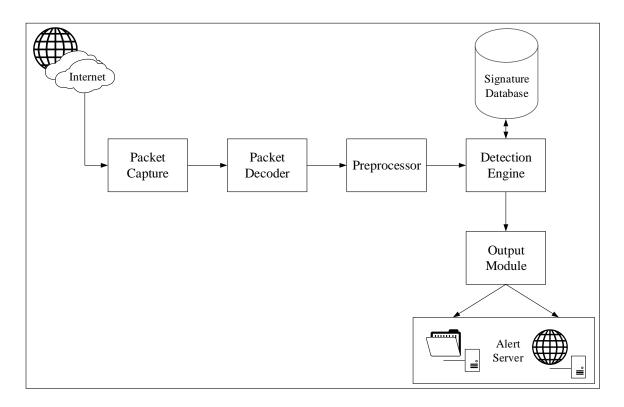


Figure 1.2: Architecture of Snort

Packet capture: In this module packets are captured using sniffer in the backend part of Snort. This module is responsible for capture the data transmitted over the network. For subsequent transmission to decoder with the help of a library named Data Acquisition (DAQ), it has done it job.

Packet decoder: Packet decoder deals with parsing the headers of captured packets. Decoding human readable information from raw packet by parsing them, the analysis of TCP flags, except for certain protocols of further analysis, finding anomalies and deviations from the RFC, and other similar work packet decoder done its job.

Preprocessor: The preprocessors of Snort are intended to do in-depth analysis and normalization protocols at each layer of TCP/IP model. Amongst most used preprocessor in Snort frag3, stream5, http_inspect, RPC2, sfPortscan are very popular. To work with fragmented traffic frag3 preprocessor is used. Similarly, for the reconstruction of TCP flows stream5, for normalizing HTTP traffic http_inspect preprocessor are used. To detect port scans in network sfPortscan preprocessor is used in Snort. And decoders for different types of protocol such as SSH, IMAP, SMTP, FTP, SIP, Telnet are also used in this module.

Detection engine: Detection engine of Snort consist of two parts. Of them one part is used to collect various signature from its database, and another is responsible for deep-level inspection where it match the signatures with the real-time network traffic.

Output module: Output module is responsible for alert to the administrator based on the detection of attacks, for logging the attacks, capture the network traffic for further analysis as pcap format and writing them in binary format on the base machine using Unified2.

1.1.2 Suricata

Suricata is a referred to as a free and open source, advanced, robust and fast network intrusion detection and prevention engine. It is capable of real-time intrusion detection and inline prevention (IDPS), monitoring network security and offline processing of captured pcap files. Suricata analyze network traffic with its powerful and sizable rules and signature

language, and has effective Lua scripting support for the detection of complicated modern intrusion.

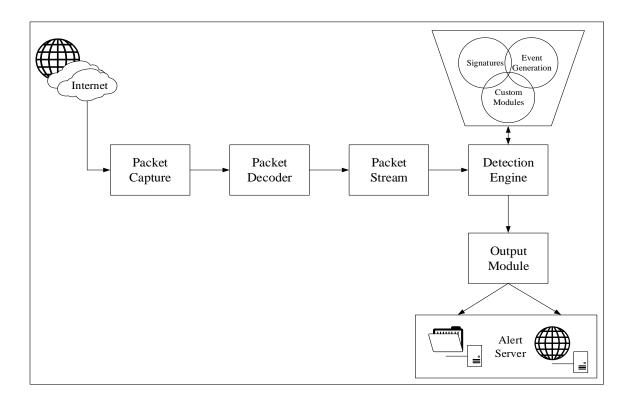


Figure 1.3: Architecture of Suricata

Architecture of Suricata is almost similar to Snort but has difference in some parts of its module. Suricata use PF_RING a high-speed packet processing framework which a new type of network socket that dramatically improves the packet capture speed [2] for capturing packet from the internet or other source. Packet stream is like preprocessor which is basically deals with network streams. Detection engine of Suricata support multi-threading techniques and that's why its processing speed is way better.

1.2 Motivation

Security threats are an alarming issue for the modern world. Attacks which are success in their motive called intrusion. In recent years from various study it is said that, cyber threats are increasing rapidly with modern techniques and tactics. Due to the increasing threat Cyber Crime is a big issue that hampers regular activity of our society and our systems.

Intellectual Property Theft and Cybercrime become commonplace during the 2000s. So, to protect our critical infrastructures, network and computer system necessary steps should be taken. Intrusion Prevention Systems are a solution to protect network and computer system from different threats and attack.

1.3 Research Questions

Research question of this study are as follows:

- Are Network-based IPSs are capable to protect network and computer systems, critical infrastructures from modern intrusion?
- ► Does IPS are enough to secure todays networks?
- ► Does Snort with single-thread processing capability better than Suricata?
- ► Does Suricata's CPU Utilization better than Snort?

1.4 Expected Output

From this study is expected to learn deep level packet analysis, know how to analyze real time network traffic in a structured way with well-known solutions. It is a great way to learn about intrusion, cyber threat, detection and prevention techniques and so on. Expected outcome would be identify the best solution to prevent modern threat in real world. Another would be to know about the way to secure critical infrastructures.

1.5 Thesis Layout

This study contains of six chapter in which have described the whole of the thesis. Thesis layout consists of the preview of all the chapters.

- 1. Chapter one covers introduction, motivation, research questions, expected output and thesis layout of the study.
- 2. Related research work have discussed in Chapter two.
- 3. Chapter three research methodology includes introduction, research design, lab architecture, and dataset collection procedure.

- 4. Chapter four of this study discussed on requirement analysis, requirement installation, and requirement configuration for the success of the experiment.
- 5. Experimental results on CPU Utilization, experimental results on CICIDS2017 dataset, descriptive analysis and result comparison are discussed in Chapter 5.
- 6. Finally, in chapter six have discussed about conclusion and future study.

CHAPTER 2 BACKGROUND

2.1 Related Works

Intrusion prevention has become significantly more important due to, with the increase in difficulty and regularity of Internet threats in recent years. Various tech companies and organizations working to develop the equipment and produces different product including open source and proprietary. One amongst the most well-known and widespread opensource intrusion detection and prevention system is Snort which works on signature-based detection and prevention. Snort was maintained by SourceFire Company, now acquired by Cisco Systems Limited. Martin Roesch developed Snort in 1998. It was mainly developed to monitor the network packet of layer 7 which is application layer of OSI model. But nowadays it is used in the backend part of most of the next-generation firewall and intrusion prevention systems. In 2009, after a decade another open source community named Open Information Security Foundation (OISF) announced another signature-based intrusion detection and prevention system called Suricata. The signification difference between Snort and Suricata is in their internal architecture. The advancement in Suricata is it's able to execute native multithreaded processes. Many research has been done in terms of testing and comparing different type intrusion prevention system in recent years. Researcher Sergey identified pros and cons of Snort and Security Onion in his thesis [3]. Ahmad Iftikhar, et al. recognized intrusion detection approached in their research with comparison [4]. Study on intrusion detection and prevention system are huge. Researcher B.Santos Kumar et al., identified type and prevention of intrusion detection system in their research [5]. A great thesis on analysis and comparison of Snort and Suricata was published in 2011 by Eugene [6]. Also many article has been published focused on intrusion detection and prevention system. Due to the rapid growth of Internet, need to be ensure its security first. And Intrusion prevention system can be a great technology in terms of its solution.

2.2 Research Challenges

Challenges of the study relies on the experiment part.

- Resources are limited and most of them not rich.
- ► A strong background on networking and OSI layer is must.
- In-depth knowledge on networking packet architecture is necessary to deploy the experiment.
- Previous basic knowledge on Intrusion Detection and Prevention Systems is also necessary.
- Hand-on working knowledge and experience on Linux is a must to fulfil the goal of this study.

CHAPTER 3 RESEARCH METHODOLOGY

3.1 Introduction

The research methodology discussed how the research has been done to complete the thesis. In-depth study on Intrusion Prevention System has been done prior to the experiment. This chapter includes the research design, lab architecture and dataset collection procedure and reason behind choosing the dataset.

3.2 Research Design

Research design shown in Figure 3.1 indicates how the whole research has been conducted.

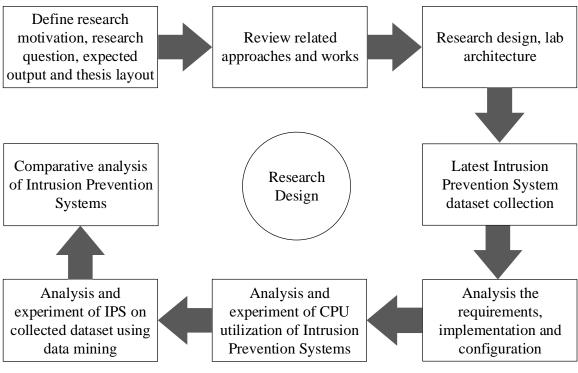


Figure 3.1: Research design

3.3 Lab Architecture

Lab architecture includes 4 PC (Attacker, Normal User, Victim PC1 and Victim PC2), 1 network switch, 1 network router PC (Intrusion Prevention System) and 2 logical Class B private networks which include 172.16.10.0/24 and 172.16.20.0/24 where each network

hosts are connected with the switch and IPS router. Figure 3.2 shows the lab architecture of this research.

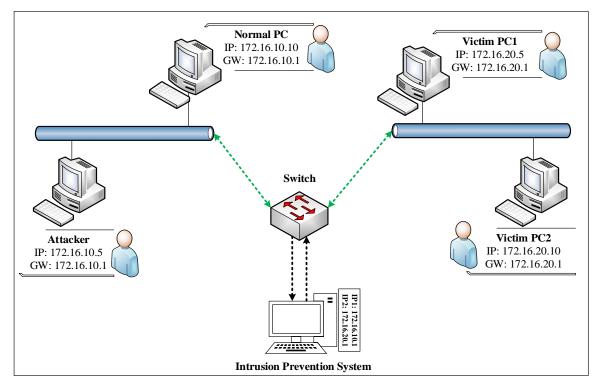


Figure 3.2: Lab Architecture

3.4 Dataset Collection

The experiment of this study has been conducted on one of the latest IPS dataset named CICIDS2017 collected from Canadian Institute for Cybersecurity (CIC) which is not publicly available on the Internet. Many researchers has been used this dataset for inventive research purpose. Among many researchers, Gobinath Loganath used this dataset for Real-time Intrusion Detection purpose [7]. Darya Lavrova et. al., also used CICIDS2017 dataset for "Wavelet-analysis of network traffic time-series for detection of attacks on digital production infrastructures" [8] purpose. The CICIDS2017 dataset contains benign and the most up-to-date common attacks, which resembles the true real-world data (PCAPs). Attack diversity and count of flows can be found on Table 4.1. This dataset also includes the results of the network traffic analysis using CICFlowMeter with labeled flows based on the time stamp, source and destination IPs, source and destination ports, protocols and

attack (CSV files) [9]. That's why CICIDS2017 dataset has been used in this study for experiment.

#	Attack Type	Total flow
1	Heartbleed	11
2	Web Attack: SQL Injection	21
3	Infiltration	36
4	Web Attack: XSS	652
5	Web Attack: Brute Force	1507
6	Botnet	1966
7	DoS Slowhttptest	5499
8	DoS Slowloris	5796
9	SSH Patator	5897
10	FTP Patator	7938
11	DoS GoldenEye	10293
12	DDoS	41835
13	Port Scan	158930
14	DoS Hulk	231073
15	BENIGN	2358036

Table 3.1: Attack Types and flows in CICIDS2017

Next chapter will discuss the requirement analysis, installation and configuration.

CHAPTER 4

REQUIREMENT ANALYSIS, INSTALLATION AND CONFIGURATION

4.1 Introduction

This chapter discussed on requirement analysis, installation and environment configuration for the experiment. Requirement analysis includes hardware and software requirements which are the most important part for the experiment of this research.

4.2 Requirement analysis

Both hardware and software requirements are necessary to study the experiment. Table 4.1 shows the overview of both hardware and software requirements. Requirements are needed to be ready before the experiment. PCs of victim network and Attacker network both have 4GB of RAM. Intrusion Prevention System has 4 GB of RAM. After successful implementation of hardware requirements, software requirements was implemented where different software were installed and configured for the experiment.

Hardware Requirements			Software	
	Machine	Operating System	IP Address	Requirements
Victim Network	Victim PC1 Victim PC2	Windows 10 x64 Ubuntu 16.04.5	172.16.20.5 172.16.20.10	XAMPP, Mysql, Apache2, DVWA
	Intrusion Prevention System (IPS)	Ubuntu 16.04.5	172.16.10.1 172.16.20.1	Snort, DAQ, Barnyard, Pulledpork, Mysql,

				Suricata, WebSnort, Wireshark, Atop
Attacker Network	Attacker Normal User	Kali Linux Windows 10 x64	172.16.10.5 172.16.10.10	Tcpreplay, Wireshark, Nmap, Atop

4.3 Requirements Installation

To make the environment ready for the experiment, firstly hardware requirements were setup properly. According to the lab architecture can be found on Chapter 3 Switch, Router and PCs were connected with necessary network cables. And two logical private network 172.16.10.0/24 and 172.16.20.0/24 has been configured and tested on Router and PCs prior to the installation of software requirements. It is mentioned that Internet connection was ensured to download necessary software for the experiment. Table 4.2 shows the specific version of software which were used in this research.

Software	Version
XAMPP	7.2.10
MySQL	5.7.16
Apache2	2.4.34
DVWA	1.9
Snort	2.9.11.1
DAQ	2.0.6
Barnyard2	2-1.14
PulledPork	0.7.4
Suricata	4.0.5

Software	Version
Websnort	0.8
Wireshark	2.6.4
Atop	2.3.0
Tcpreplay	4.2.5
Nmap	7.70

4.3.1 Snort, DAQ, Barnyard2, PulledPork and WebSnort installation

Here for installing Snort, DAQ, Barnyard2, PulledPork and WebSnort have used an interactive automated script named Snorter_IPS.sh developed by Joan Bono along with one of the contributor named Md. Nazrul Islam [10]. This script was taken from open source platform GitHub and then modified. Function of the script includes-

```
function main(),
function update_upgrade(),
function nghttp2_install(),
function snort_install(),
function snort_edit(),
function snort_test(),
function barnyard2_ask(),
function pulledpork_ask(),
function service_create(),
function websnort_ask(),
function last_steps(),
function system reboot()
```

The script start from the function main() and step by step and install and configured NGHTTP2, Snort, DAQ, Barnyard, PulledPork and WebSnort along with their dependencies. Figure 4.1 shows nghttp2_install a function of Snorter_IPS.sh script which install and configure NGHTTP2. NGHTTP2 is necessary for Snort to run as IPS mode.

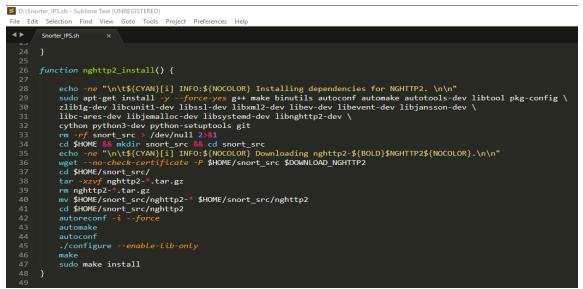


Figure 4.1: nghttp2_install function

Dependencies for Snorter_IPS.sh script:

jq, curl

Dependencies for NGHTTP2:

cython libxml2-dev python3-dev binutils libevent-dev git libev-dev libssl-dev libjansson-dev zlib1g-dev python-setuptools automake libjemalloc-dev pkg-config libnghttp2-dev libc-ares-dev autotools-dev g++ make autoconf libtool libcunit1-dev libsystemd-dev

Dependencies for Snort:

gcc libpcre3-dev libnghttp2-dev openssl libdnet bison zlib1g-dev libpcap-dev
libssl-dev libdumbnet-dev flex

Dependencies for Barnyard2:

mysql-server libmysqlclient-dev mysql-client autoconf libtool libdnet checkinstall yagiuda libdnet-dev locate

Dependencies for PulledPork:

libcrypt-ssleay-perl liblwp-useragent-determined-perl

Figure 4.2 shows the running script where options –i indicates the interface of the machine and –o indicates the oinkcode (A unique code for snort individual user).

ridoy@anips: ~/Downloads	t, En	■))	4:40 PM	₩
rldoy@antps:-/Downloads\$ sudo ./Snorter_IPS.sh -t enp2s0 -o 86aee43ca84134204b2435f071fdfa13c8896150				
Image: Constraint of the second se				
<pre>[i] INFO: Updating and Upgrading repositories [i] INFO: Updating and Upgrading repositories Hit:1 http://mirror.xeonbd.com/ubuntu-archive xental InRelease Hit:3 http://mirror.xeonbd.com/ubuntu-archive xental-updates InRelease Hit:3 http://mirror.xeonbd.com/ubuntu-archive xental-backports InRelease Hit:3 http://mirror.xeonbd.com/ubuntu-archive xental-backports InRelease Reading package lists Done</pre>				
Building dependency tree Reading state information Done Calculating upgrade Done 0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded. (1) INFO: Installing dependencies for NGHTTP2.				
Reading package lists Done Building dependency tree Reading state information Done autocomf is already the newest version (2.69-9). automake is already the newest version (1:1.15-4ubuntu1). autotools-dev is already the newest version (20150820.1). gent is already the newest version (4:5.3.1-1ubuntu1). libtool is already the newest version (2.4.6-0.1).				

Figure 4.2: Running Snorter_IPS.sh script

Figure 4.3 shows that daq-2.0.6 and snort-2.9.11.1 is downloading automatically. It is mentioned that the script always find the latest version of required softwares. At the time of the experiment daq-2.0.6 and snort-2.9.11.1 was the latest version.

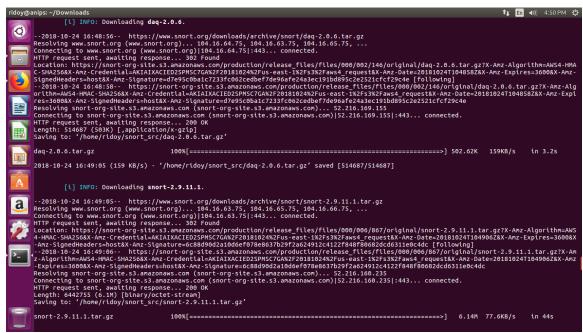


Figure 4.3: Downloading DAQ and Snort with automated script

To run Snort software as intrusion prevention mode nfqueue is necessary. So it must be needed to ensure that nfqueue is enable in DAQ module before compiling.

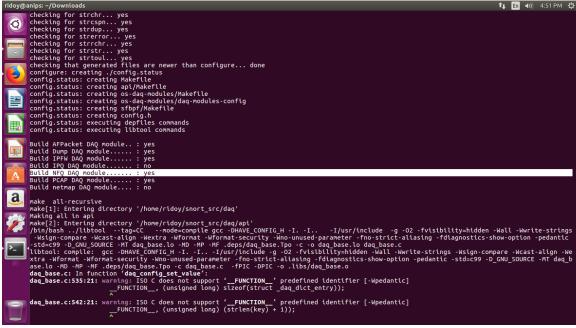


Figure 4.4: NFQ DAQ modules functions

Figure 4.4 shows DAQ modules where NFQ DAQ module is successfully enabled with yes notation. After downloading and DAQ and Snort, DAQ module was compiled before Snort installation. Because DAQ module is a must pre-requirement module of Snort software. Figure 4.5 indicates Snort successfully installed and configured. Snort installation can also be verified using the command-

sudo /usr/bin/snort -T -c /etc/snort/snort.conf

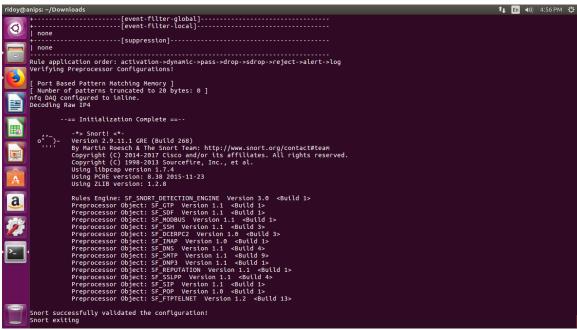


Figure 4.5: Snort

If the command returned successful indication without any error means that snort installation and its configuration is ok. Snort works based on detection and prevention rules. Everyday new intrusion are discovered and new rules are generated against them to prevent propagation in the world. So it is necessary to have the latest detection and prevention rules. Using pulledpork an open source software automatically download the latest rules every day at a scheduled time. Figure 4.6 shows pulledpork is downloading latest community, opensource, emerging-rules and snort-snapshot rules. Snort-snapshot rules are especially for snort user. These rules are identified and download based on the unique oinkcode.

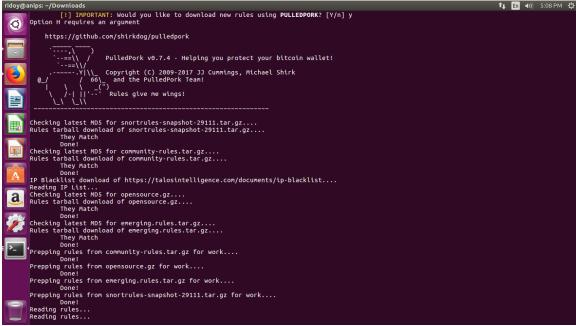


Figure 4.6: PulledPork download IPS rules automatically

Snort ruled are located at /etc/snort/rules/ .When rules download was complete, websnort was installed and configured successfully.

4.3.2 Suricata installation

Suracata also known as open-source network based IPS developed by Open Information Security Foundation (OISF). Suricata also capable to capture real-time network packet and able to identify network intrusion and protect them using inline prevention mode. Suricata use NetfilterQueue a.k.a NFQ for performing inline functionality [11].

Suricata dependencies:

autoconf libjansson-dev libcap-ng-dev libjansson4 libnet1-dev libpcre3-dbg libmagic-dev libtool libpcre3-dev automake libpcap-dev libyaml-dev zlib1g-dev

Suricata dependencies for IPS:

libnetfilter-queue-dev libnetfilter-queue1 libnfnetlink-dev

Figure 4.7 shows the installation process of Suricata dependencies. After that Suricata was downloaded, installed and configured.

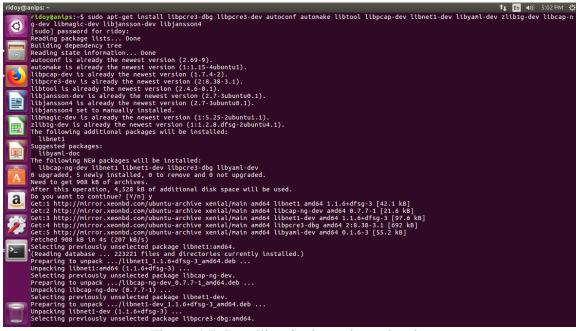


Figure 4.7: Installing Suricata dependencies

Suricata latest version (suricata-4.0.5) is downloaded using command-

wget https://www.openinfosecfoundation.org/download/suricata-current.tar.gz

doy@a	anips: ~/thesis/suricata-4.0.5								🏚 En 🜒) 5:18 PM
	suricata-4.0.5 suricata-c								
O)	ridoy@anips:~/thesis\$ cd s								
	ridoy@anips:~/thesis/suric						· · ·		
		fig.guess		depcomp	libhtp	Makefile.am Makefile.in	reference.config		
_			configure.ac	doc etc	LICENSE ltmain.sh		rules	suricata.yaml.in	
	classification.config con compile con	fig.sub	COPYING	install-sh		qa	scripts	threshold.config	
	compile con ridoy@anips:~/thesis/suric			install-sn	M4	qa	scripts		
	total 1700	aca-4.0.55	us -ua						
5	drwxr-xr-x 12 ridoy ridoy	4096 701	17 20:43 .						
	drwxrwxr-x 3 ridoy ridoy		24 17:18						
	-rw-rr 1 ridoy ridoy			cal.m4					
	-rw-rr 1 ridoy ridoy		17 20:43 Char						
	-rw-rr 1 ridov ridov		17 20:43 clas		onfig				
	-rwxr-xr-x 1 ridoy ridoy		17 20:43 comp						
	-rwxr-xr-x 1 ridoy ridoy		17 20:43 conf						
±1	-rw-rr 1 ridoy ridoy	16029 Jul	17 20:43 conf	ig.h.in					
	-rw-rr 1 ridoy ridoy	0 Jul	17 20:43 conf	ig.rpath					
	-rwxr-xr-x 1 ridoy ridoy	36144 Jul	17 20:43 conf	ig.sub					
	-rwxr-xr-x 1 ridoy ridoy								
_	-rw-rr 1 ridoy ridoy		17 20:43 conf	igure.ac					
	drwxr-xr-x 4 ridoy ridoy		17 20:43 cont						
A	-rw-rr 1 ridoy ridoy								
-	-rwxr-xr-x 1 ridoy ridoy			:omp					
	drwxr-xr-x 3 ridoy ridoy		17 20:43 doc						
a	drwxr-xr-x 2 ridoy ridoy		17 20:43 etc						
<u> </u>	-rwxr-xr-x 1 ridoy ridoy								
	drwxr-xr-x 6 ridoy ridoy		17 20:43 lib						
-19	-rw-rr 1 ridoy ridoy								
	-rw-rr 1 ridoy ridoy			iin.sn					
	drwxr-xr-x 2 ridoy ridoy		17 20:43 m4	611					
	-rw-rr 1 ridoy ridoy -rw-rr 1 ridoy ridoy		17 20:43 Make						
	-rwxr-xr-x 1 ridoy ridoy		17 20:43 Make 17 20:43 miss						
	drwxr-xr-x 3 ridoy ridoy		17 20:43 MUSS	ing					
	-rw-rr 1 ridoy ridoy		17 20:43 qa 17 20:43 refe	rence confid					
	drwxr-xr-x 2 ridoy ridoy		17 20:43 rele						
	drwxr-xr-x 6 ridoy ridoy		17 20:43 ruce						
	drwxr-xr-x 3 ridoy ridoy		17 20:43 rust						
	drwxr-xr-x 3 ridoy ridoy		17 20:43 SFC						
	-rw-rr 1 ridoy ridoy		17 20:43 suri	cata.vaml.ir					
	-rw-rr 1 ridoy ridoy		17 20:43 thre						
	ridoy@anips:~/thesis/suric			e on e o					

After download tar file was extracted and Figure 4.8 show the insider installation files.

Figure 4.8: Suricata installation files

Before installation of Suricata software configure file is needed to be compiled first with enabling necessary module such as nfqueue module. The command –

ridoy@anips: ~/thesis/suricata-4.0.5		▲ En A)) 5:20 PM 公
checking that generated files are newer the	pan configure done	
checking that generated files are newer the second seco		
configure: creating ./config.status		
config.status: creating htp/htp version.h		
config.status: creating Makefile		
config.status: creating htp.pc		
config.status: creating htp/Makefile		
config.status: creating test/Makefile		
config.status: creating docs/Makefile		
<pre>config.status: creating htp_config_auto_get</pre>		
<pre>config.status: executing depfiles command</pre>		
config.status: executing libtool commands		
AF_PACKET support:	yes	
PF RING support: NEQueue support:	no	·
	yes	
NFLOG support: IPFW support:	no no	
Netmap support:	no	
DAG enabled:		
Napatech enabled:		
Unix socket enabled:	ves	
Detection enabled:	yes	
Libmagic support:		
Libmagic support:	yes	
Libnss support:		
libnspr support:		
libjansson support:	yes	
hiredis support:	no	
hiredis async with libevent:	no	
Prelude support:	no	
PCRE jit: LUA support:	yes	
libluajit:	no no	
libgeoip:	no	
Non-bundled htp:	no	
Old barnyard2 support:	no	
CUDA enabled:	no	
Hyperscan support:	no	
Libnet support:	yes .	

sudo ./configure --enable-nfqueue --prefix=/usr --sysconfdir=/etc --localstatedir=/var

Figure 4.9: NFQueue support of Suricata

Figure 4.9 shows NFQueue module is enabled and supported notation as yes. So Suricata can be run as Intrusion Prevention mode. After that Suricata is installed and configured.

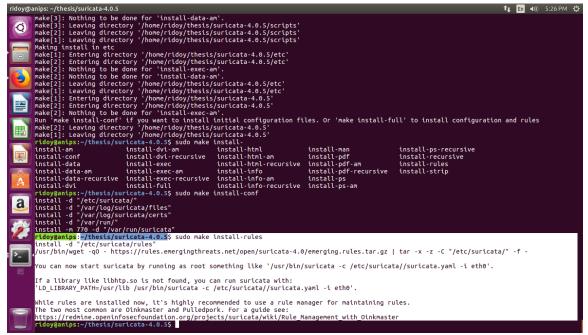


Figure 4.10 Suricata installation

Figure 4.10 shows the installation of Suricata where firstly installed configuration and then Suricata rules. Surcata detection and prevention rules are located at /etc/suricata/rules directory. Figure 4.11 shows Suricata detection and prevention rules.

ridov@a	nips: /etc/suricata/rules		↑』 En (4)) 5:29 PM (4)
, e	ridoy@anips:/etc/suri	insta/sulast is la	
	total 18216		
(O)	drwxr-xr-x 2 root roo	ot 4096 Oct 24 17:25 .	
	drwxr-xr-x 3 root roo		
	-rw 1 root roo		
	-rw-rr 1 root roo		
		1 168444 Oct 24 03:41 botcc.rules	
	-rw-rr 1 root roo		
		t 109665 Oct 24 03:41 ciarmy.rules	
	-rw-rr 1 root roo		
	-rw-rr 1 root roo		
	-rw 1 root roo		
	-rw 1 root roo	ot 1221 Oct 24 17:25 dnp3-events.rules	
	-rw 1 root roo	ot 1777 Oct 24 17:25 dns-events.rules	
	-rw-rr 1 root roo	ot 26497 Oct 24 03:41 drop.rules	
	-rw-rr 1 root roo	ot 2667 Oct 24 03:41 dshield.rules	
E C	-rw-rr 1 root roo	ot 364717 Oct 24 03:41 emerging-activex.rules	
	-rw-rr 1 root roo		
	-rw-rr 1 root roo		
		ot 1608058 Oct 24 03:41 emerging-current_events.rules	
		ot 1270871 Oct 24 03:41 emerging-deleted.rules	
	-rw-rr 1 root roo		
A	-rw-rr 1 root roo		·
		ot 367806 Oct 24 03:41 emerging-exploit.rules	
	-rw-rr 1 root roo		
a	-rw-rr 1 root roo		
	-rw-rr 1 root roo		
	-rw-rr 1 root roo		
1722	-rw-rr 1 root roo -rw-rr 1 root roo		
		1 20590 Oct 24 03:41 emerging-info; tate; ottes	
		t 47186 Oct 24 03:41 emerging-tallorates	
The second second		t 2/179 Oct 24 03:41 emerging-misc.rules	
∎P_ I		t 22055 Oct 24 03:41 emerging-mobile malware.rules	
		1 330472 Oct 24 03:41 emerging notbios.rules	
	-rw-rr 1 root roo		
		33188 Oct 24 03:41 emerging-policy.rules	
	-rw-rr 1 root roo		
	-rw-rr 1 root roo		
	-rw-rr 1 root roo	ot 13230 Oct 24 03:41 emerging-scada.rules	
	-rw-rr 1 root roo	ot 122269 Oct 24 03:41 emerging-scan.rules	
A COLUMN A	-rw-rr 1 root roo		
	-rw-rr 1 root roo	ot 10059 Oct 24 03:41 emerging-smtp.rules	

Figure 4.11: Suricata detection and prevention rules

Rules extension is .rule and can be open through any text editor software such as vi, vim, gedit, nano and so on. Figure 4.12 shows the rules for ICMP packet.

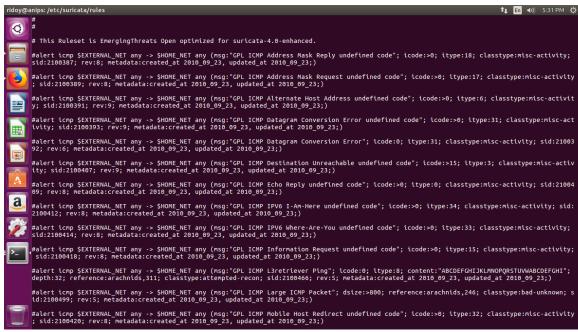


Figure 4.12: Suricata ICMP detection and prevention rules

While rules and other necessary configuration was complete, Suricata main configuration file was configured to make ready for run.

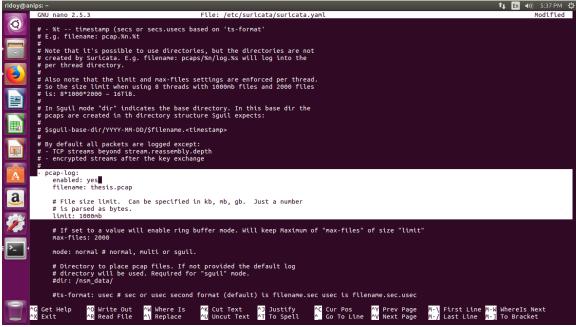


Figure 4.13: Suricata configuration

Suricata configuration file is located /etc/suricata/suricata.yaml. Configuration has several parts like network setup, output setup, and log setup and so on. Figure 4.15 indicates the Suricata configuration file where pcap-log is enabled with filename as thesis.pcap. In the next chapter experimental results will be performed and discussed.

CHAPTER 5

EXPERIMENTAL RESULTS AND DISCUSSION

5.1 Introduction

The experiment tested and compared with Snort and Suricata Intrusion Prevention system in performance and accuracy of detection and prevention in a real setup environment. Performance evaluated by measuring the percentage of memory usage, network usages and CPU Utilization in this experiment. Accuracy was measured and compared based on the generated alert of detection and prevention of each prevention system using machine learning and data mining technique on CICIDS2017 dataset.

5.2 Experimental Results on CPU Utilization

The experiment was conducted in two stage. Where in first stage Packet Processing and CPU Utilization of both Intrusion Prevention System (Snort and Suricata) was measured and calculated. And in another stage detection and prevention of Suricata and Snort was analyzed and measured. Packet processing was logged and calculated using Wireshark. From Wireshark packets I/O value was taken as a csv file. Figure 5.1 and Figure 5.2 shows the packets processing graphs of Suricata and Snort respectably. Suricata Intrusion Prevention System processed 351.70 packet/s on an average with a high value 4295 packet

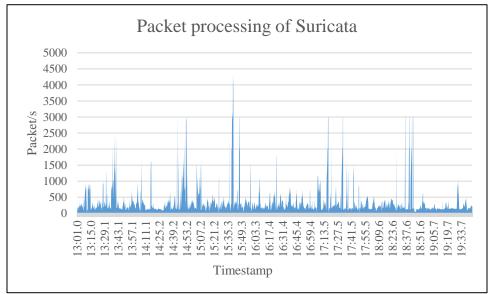


Figure 5.1: Suricata I/O of packet

in a second. Suricata was running for 6 minutes during the experiment. On the other hand Snort was running for 10 minutes shows in Figure 5.2 processed 327.22 packets/s on an average with a high value 5578 packet in a second during the experiment.

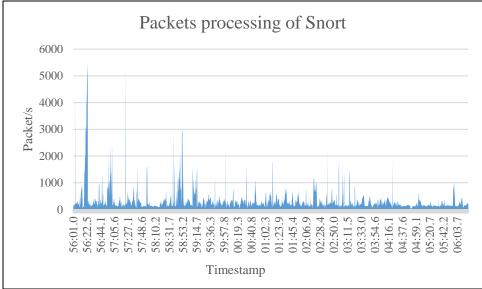


Figure 5.2: Snort I/O of packet

CPU Utilization of Suricata and Snort was measured using atop a tool that is capable of reporting the activity of all processes like CPU utilization, memory growth, disk utilization, priority, username, state, exit code and so on. Figure 5.3 shows the interface of atop tool.

Product sys 4% juser 33% irq 0% idle 35% istel 0% guest 0% curf 3.20GHz curred 1 curred 2 <	root@a	nips: ~/	PycharmP	rojects	;														t,	En 🜒) 6:1	4PM ∰
Yeu sys 4% user 33% irq 0% idle 35% istel 0% guest 0% curf 3.20GHz curred 1 curred 2 curred 1 curred 2 cu		ATOP	- anips				201	8/10/24	18:13	3:59					-					2h37m13s e	lapsed
Picture CPU sys 4 % juser 33% ire 0 % idle 35% istel 0 % guest 0 % curf 3.29CHz curred a psych CPU sys 1% user 9% ire 0 % curf 3.29CHz curred a psych		PRC 1	svs	4m59s	l user	34m19s	#ргос	281	l #trun	2	#tslpi	824	#tslp	u 0	#z	ombie	Θ	<pre>l clones</pre>	s 64734	#exit	0
Cpu sys 1% user 9% 1 (rig 0% Cpu000 w 1% steal 0% Guest 0% Curf 3.290/z Curscal 99% Cpu sys 1% user 8% 1rg 0% (1dle 89% cpu008 w 3% steal 0% guest 0% curf 3.290/z curscal 99% Cpu sys 1% user 7% trq 0% (1dle 91% cpu001 w 1% steal 0% guest 0% curf 3.290/z curscal 99% CPL avg1 1.44 avg5 1.57 avg15 1.18 L Curf 3.290/z curscal 9% numcpu 4 Inder isca fora fora <td></td> <td></td> <td></td> <td></td> <td>user</td> <td>33%</td> <td></td> <td></td> <td>idle</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>i au</td> <td>est</td> <td>0%</td> <td>curf</td> <td>3.29GHz</td> <td>curscal</td> <td>99%</td>					user	33%			idle						i au	est	0%	curf	3.29GHz	curscal	99%
Cpu Sys 1% User 9% Lfc 09% Cpu009 w 1% steal 0% Guest 0% Curf 3.290/z Curscal 99% Cpu Sys 1% user 7% Lfc 0% Ldle 99% Steal 0% Guest 0% Curf 3.290/z Curscal 99% Cpu Sys 1% user 7% Lfc 0% Ldle 91% Steal 0% Guest 0% Curf 3.290/z Curscal 99% Cpu Sys 1/4 avgl LiAid LiAid LiAid LiAid LiAid <t< td=""><td></td><td>сри і</td><td></td><td>1%</td><td>user</td><td>9%</td><td></td><td>0% i</td><td>idle</td><td>89%</td><td>CDU002</td><td>w 1%</td><td>i steal</td><td>0%</td><td></td><td></td><td>0%</td><td>curf :</td><td>3.29GHz</td><td></td><td>99% i</td></t<>		сри і		1%	user	9%		0% i	idle	89%	CDU002	w 1%	i steal	0%			0%	curf :	3.29GHz		99% i
Cru Sys 1% User BX Ird OX Couldol W Steal OX Guest OX Currscal 99X CFL avg1 1.44 avg5 1.57 avg15 1.18 uncpu 4 steal OX intr Steal OX currscal 99X Intr ord steal OX steal OX funct Steal OX Variant Steal OX Steal OX Steal OX Steal OX Steal OX Steal OX Steal																	0%				99%
Cpu Sys 1% User 7% Irc 0% ith Steal 0% Curf 2.29GHZ curscal 9% MER tot 37.64 Free 182.0M cache 1.26 dirty 0.4M buff 29.2M stab 142.0M nuncpu 4 PAG scan 619327 stall 0 stall 0 stab 142.0M stab 142.0M stab 142.0M stab 142.0M stab 142.0M stab 147.0 1 tcpn 34 avto 4.37 ns tudplp 0 1 tcpa 230 udplp 0 1 tcpa 230 udplp 0 1 tcpa 1 tcpa 30 tcpa 230 udplp 0 1 tcpa 1 tcpa 1 tcpa 1 tcpa 30 tcpa 1 tcpa 1 tcpa 1 1 1 1				1%	user	8%	ira	0%	idle	89%			steal	0%			0%	curf	3.29GHz	curscal	99%
CPL avg1 1.44 avg5 1.75 avg15 1.76 free 1.77 1.76 1.77 1.76 1.77 <th< td=""><td></td><td>cou i</td><td></td><td>1%</td><td>luser</td><td>7% i</td><td>ira</td><td>0% i</td><td>idle</td><td>91%</td><td>CDU001</td><td>w 1%</td><td>i steal</td><td>0%</td><td>i au</td><td>iest</td><td>0%</td><td>i curf :</td><td>3.29GHz</td><td>i curscal</td><td>99% i</td></th<>		cou i		1%	luser	7% i	ira	0% i	idle	91%	CDU001	w 1%	i steal	0%	i au	iest	0%	i curf :	3.29GHz	i curscal	99% i
MEM tot 3.7G free 182.0M coche 1.2G dirty 0.4M buff 29.2M slab 142.6M varcas 9.4G varcas 9.2G varcas 9.2G varcas 9.2G varcas 9.2G varcas 9.2G varcas varcas 9.2G varcas varcas 9.2G varcas varcas 9.2G varcas varcas varcas varcas varcas varcas <t< td=""><td></td><td>CPL I</td><td></td><td>1.44</td><td>i avo5</td><td>1.57</td><td>avo15</td><td>1.18</td><td></td><td></td><td></td><td></td><td>i csw 1</td><td>9464573</td><td>i ír</td><td>tr 554</td><td>6438</td><td>i</td><td></td><td></td><td>4 İ</td></t<>		CPL I		1.44	i avo5	1.57	avo15	1.18					i csw 1	9464573	i ír	tr 554	6438	i			4 İ
Image Image <th< td=""><td></td><td>мем і</td><td></td><td>3.7G</td><td></td><td>182.0M</td><td></td><td>1.2G</td><td>dirtv</td><td>0.4M</td><td>buff</td><td>29.2M</td><td>slab</td><td>142.6M</td><td></td><td></td><td></td><td>i</td><td></td><td></td><td></td></th<>		мем і		3.7G		182.0M		1.2G	dirtv	0.4M	buff	29.2M	slab	142.6M				i			
DSK sda busy 5% read 71462 write 3330 K1B/r 30 K1B/r 76 MBF/s 0.28 avio 4.37 ns NET I network I tip 127660 I tcpo 114456 ipfrw 0 deliv 216721 I Ipropo 0 Itcmpi 787 Itcmpi 0 dfpp 0 NET enebroid pck 1438 si 0 KBF so 7 Kbps errit 0 error 0 dfpl 0 dfpp 0 NET enebroid ridoy ridoy tids 131 2 KDP so 7 Kbps errit 0 error 0 dfpl dfpl 0																					
NET transport tcpi 15916 tcpi 149722 udpt 610 udpi 5833 tcpio 220 tcpio 11 tcpirs 220 udpip 6 NET 1 network tpi 127660 tpo 1143656 tpfr 0 defive 10 rtcpio 0 drpi drpi <td></td> <td>PAG I</td> <td></td>		PAG I																			
NEI network Tpc 114350 TprW 0 0etV 210/21		DSK I		sda	busv	5%	read	71462	write	35362	KiB/r	30	KiB/w	76	I ME	r/s	0.22	MBw/s	0.28	avio 4.3	7 ms
NEI network Tpc 114350 TprW 0 0etV 210/21			transpo																		0
NET 1 to j pckt 4483 j pcko 4483 j st 0 % kbps j errt 0 j drpt 0 j drpt <th< td=""><td></td><td></td><td></td><td></td><td></td><td>217660</td><td></td><td>14456</td><td></td><td>0</td><td>deliv</td><td>216721</td><td></td><td></td><td></td><td></td><td></td><td></td><td>787</td><td></td><td>30</td></th<>						217660		14456		0	deliv	216721							787		30
PID PUT FUT FUT <td></td> <td>NET</td> <td>enp2s0</td> <td>0%</td> <td>pcki</td> <td>220482</td> <td>pcko 1</td> <td>10262</td> <td>si 19</td> <td>92 Kbps</td> <td>so</td> <td>7 Kbps</td> <td>erri</td> <td>0</td> <td>] er</td> <td>го</td> <td>0</td> <td>drpi</td> <td>1</td> <td>drpo</td> <td>Θİ</td>		NET	enp2s0	0%	pcki	220482	pcko 1	10262	si 19	92 Kbps	so	7 Kbps	erri	0] er	го	0	drpi	1	drpo	Θİ
PID RUTD EUID THR SYSCPU USRCPU VGROM RGOM RDDSK MRDSK ST EXC S CPUNR AMEH CMD 1/ 5320 ridoy ridoy 28 16.19s 64.14s 1.9C 270.7M 20960k 0K N - 5 0 7% Web Content 5274 ridoy ridoy 23 13.69s 57.31s 1.8C 246.18M 1344K 0K N - 5 1 5% Web Content 5274 ridoy ridoy 23 61.65 35.70s 1.8C 208.8M 1346K 0K N - 5 1 5% barnyard2 6457 ridoy ridoy 83.22s 2m55s 1.46 169.1M 8968K N - 5 2 4% Heb Content 16701 snort snort 1 0.565 6m53s 204.3M 172.3M 139698K		NET	10		pcki	4483	pcko	4483	si	0 Kbps	SO I	0 Kbps	erri	0	i er	го	0	drpi	0	drpo	Θİ
520 ridoy ridoy 28 16.19s 64.14s 1.90c 270.7M 20966K 0K N- - S 0 7% Web Content 5115 ridoy ridoy ridoy 23 13.69s 57.31s 1.86 226 227.44 4732X 366.3M N- - S 2 7% Web Content 5205 ridoy ridoy 23 13.69s 57.31s 1.86 226.1M 14572X 0K N- - S 1 6K Web Content 5205 ridoy ridoy 23 6.16s 35.70s 1.86 228.8M 13464K 0K N- - S 1 %K Web Content 16701 snort 10.555 647.30s 136.95 136.1M 1396K N N - S 1 %K Complz 4 Complz 2 Xorg	Шũ.							**													
Image: Sits ridoy ridoy <td></td> <td>PID</td> <td>RUID</td> <td></td> <td>EUID</td> <td>THR</td> <td>SYSCP</td> <td>U US</td> <td>SRCPU</td> <td>VGROW</td> <td>RGRO</td> <td>W RD</td> <td>DSK</td> <td>WRDSK</td> <td>ST</td> <td>EXC</td> <td>S</td> <td>CPUNR</td> <td>AMEM</td> <td>CMD</td> <td>1/12</td>		PID	RUID		EUID	THR	SYSCP	U US	SRCPU	VGROW	RGRO	W RD	DSK	WRDSK	ST	EXC	S	CPUNR	AMEM	CMD	1/12
5274 ridoy ridoy 23 13.695 57.315 1.86 246.1M 14572K 0K N- - 5 1 6K Web Content 16701 snort snort 1 0.565 6m535 204.3M 172.3M 1396K 0K N- - 5 3 5% barnyard2 16701 snort snort 1 0.565 6m535 204.3M 172.3M 1396K 0K N- - 5 3 5% barnyard2 9504 ridoy ridoy 23 7.175 35.905 1.8G 136.4K 0K N- - 5 2 4% Web Content 906 mysqld 090 rysql 29 2.555 4.708 13.6 134.7M 25124K 1472K N- - 5 2 4% Web Content 4411 ridoy ridoy ridoy 4 0.895 1.795 1.36 924K 1472K 1472K N-<		5320	ridoy		ridoy	28	16.19	IS 64	4.14s	1.9G	270.7	M 209	60K	0K	N -				7%	Web Conte	nt
5274 ridoy ridoy 23 13.695 57.315 1.86 246.1M 14572K 0K N- - 5 1 6K Web Content 16701 snort snort 1 0.565 6m535 204.3M 172.3M 1396K 0K N- - 5 3 5% barnyard2 16701 snort snort 1 0.565 6m535 204.3M 172.3M 1396K 0K N- - 5 3 5% barnyard2 9504 ridoy ridoy 23 7.175 35.905 1.8G 136.4K 0K N- - 5 2 4% Web Content 906 mysqld 090 rysql 29 2.555 4.708 13.6 134.7M 25124K 1472K N- - 5 2 4% Web Content 4411 ridoy ridoy ridoy 4 0.895 1.795 1.36 924K 1472K 1472K N-<	122				ridoy				4m03s	2.4G	252.2			06.3M	N -				7%	firefox	
16701 snorf 1 0.565 60533 204.3M 172.3M 1396K 0K N- - 5 3 5% barnyard2 4507 ridoy ridoy 8 38.222 2n555 1.46 166.1H B968K 8K N- - 5 1 4% compiz 906 mysql 29 2.555 4.765 1.36 156.6M 2184K 0K N- - 5 2 4% Web Content 906 mysql visit 1.725 35.995 1.36 156.6M 2184K 0K N- - 5 2 4% Web Content 906 mysql visit 1.725 1.32 147.4M 14792K N- - 5 2 2% gnore-softwar 3749 root root 2 61.633 2044s 413.34 146.6 8068K 1816K 228K N- - 5 3 2% compiz 1734 nadir nadir 0.035 1.6345 <t< td=""><td></td><td>5274</td><td></td><td></td><td>ridoy</td><td></td><td>13.69</td><td>s 57</td><td>7.31s</td><td>1.8G</td><td>246.1</td><td>M 145</td><td>72K</td><td>ΘK</td><td>N -</td><td></td><td></td><td></td><td></td><td>Web Conte</td><td>nt</td></t<>		5274			ridoy		13.69	s 57	7.31s	1.8G	246.1	M 145	72K	ΘK	N -					Web Conte	nt
4597 ridoy ridoy 8 38.225 20555 1.46 169.1M 8968K 8K N- - 5 1 44 compiz 5504 ridoy ridoy 23 7.175 35.905 1.86 166.8M 2184K 0K N- - 5 2 4K Web Content 900 mysql 29 2.555 4.705 1.32 134.7M 25124K 14792K N- - 5 2 4K Web Content 900 mysql 29 2.555 4.705 1.33 95480K 2184K 0K N- - 5 2 2K gnome-softwar 7441 ridoy ridoy 4 0.895 1.795 1.33 93080K 1816K 228 K N- - 5 1 2% gnome-softwar 71796 nadir nadir 9.095 4.81s 708.0M 73016K 122K N- 5 3 2% update-nanage 857 root root 2 5.205 <td></td> <td>5205</td> <td>ridoy</td> <td></td> <td>ridoy</td> <td>23</td> <td>6.16</td> <td>s 35</td> <td>5.70s</td> <td>1.8G</td> <td></td> <td></td> <td>64K</td> <td>ΘK</td> <td>N -</td> <td></td> <td></td> <td></td> <td></td> <td>Web Conte</td> <td>nt</td>		5205	ridoy		ridoy	23	6.16	s 35	5.70s	1.8G			64K	ΘK	N -					Web Conte	nt
1507 1100y 1100y 38.245 2.3335 1.40 105.17 35.905 1.80 105.17 35.905 1.40 105.17 35.905 1.40 105.17 35.905 1.40 105.17 35.905 1.80 105.17 35.905 1.80 105.17 35.905 1.47 25124K 107.21X N- 5 0 4% mysqld mysqld 29 2.555 4.705 1.34 71.77 35.905 1.36 95486K 276K 92K N- - 5 0 4% mysqld mysqld mysqld mysqld 105.17 136.956.864K 23676K 92K N- - 5 0 2% Xorg Xorg Xorg Xorg 1734 nadir nadir 1.245 1.855 1.36 90606K 23676K 0K N- - 5 3 2% Compiz Xorg Xorg Xorg 2% 2% <td>Λ</td> <td>16701</td> <td>snort</td> <td></td> <td>snort</td> <td></td> <td>0.56</td> <td>s 6</td> <td>5m53s</td> <td>204.3M</td> <td>172.3</td> <td></td> <td></td> <td>ΘK</td> <td>N -</td> <td></td> <td></td> <td></td> <td>5%</td> <td>barnyard2</td> <td></td>	Λ	16701	snort		snort		0.56	s 6	5m53s	204.3M	172.3			ΘK	N -				5%	barnyard2	
900 mysqi 29 2.555 4.785 1.36 134.7% 25124K 14792K N- - 5 0 4% mysqid 3141 ridoy ridoy 4 0.895 1.795 1.36 95488K 276K 92K N- - 5 0 2% Xorg 3749 root root 2 61.635 2m94s 413.3% 938806K 23676K 0K N- - 5 0 2% Xorg 3749 root root 2 61.635 2m94s 413.3% 938806K 23676K 0K N- - 5 0 2% Xorg 3747 root rodtr nadir 1.466 808081K 11408K 8K N- - 5 3 2% comptz 2% comptz 2% comptz 2% comptz 2% comptz 2% comptz 33536K 4968K 288K N- - 5 3 1% Xorg 4428 ridoy ridoy 0.03		4507							2m55s	1.4G			68K		N -				4%		
4431 rídoy 4 0.895 1.79s 1.336 95488K 276K 92K N- - 5 2 2% grome-softwar 3749 root root 2 61.63s 20045 413.3M 93800K 18116K 228K N- - 5 0 2% Xorg 1796 nadtr nadtr 4 1.245 1.85s 1.32 90664K 23676K 0K N- - 5 0 2% Xorg gnome-softwar 1774 nadtr nadtr 4 1.245 1.85s 1.32 90664K 23676K 0K N- - S 3 2% complz gnome-softwar 1734 nadtr nadtr 4 0.095 4.81s 7006.0H 73016K 12K 0K N- - S 3 2% complz complz complz S 2% complz complz S 3 0K N N - S 3 1% Xorg complz S <td< td=""><td></td><td></td><td>ridoy</td><td></td><td>гidoy</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>nt</td></td<>			ridoy		гidoy																nt
3749 root root 2 61.635 20045 413.3M 93800K 1816K 228K N- - 5 0 2% Xorg 1796 nadir nadir 4 1.245 1.85s 1.36 90664K 22676K 0K N- - 5 0 2% Xorg 1796 nadir nadir 9 3.535 16.34s 1.0 00664K 22676K 0K N- - 5 1 2% gnome-softwar 5088 nadir nadir 4 0.095 4.81s 700.60 73016K 12K 0K N- - 5 0 2% update-nanage 57 rot rot 70 2.50.9 3.634 369.2M 53364 4968K 28K N- - 5 0 1% evolution-cal 4520 ridoy ridoy 5 0.035 0.67s 850.0M 46786K 0K N- - 5 1 % evolution-cal 1779 nadir na		900	mysql		mysql	29	2.55	s 4	4.70s	1.3G	134.7			4792K	N -				4%	mysqld	
1796 nadir nadir 4 1.24s 1.85s 1.36 90664K 23676K 0K N- - 5 1 2% gnome-softwar 1734 nadir nadir 9 3.53s 16.34s 1.46 80608K 1848K N- - 5 3 2% compt: 1734 nadir nadir 9 3.53s 16.34s 1.46 80608K 1848K N- - 5 3 2% compt: 1734 nadir nadir 4 0.095 4.61s 700.6M 73016K 12K 0K N- - 5 3 2% compt:	<u>e</u> ,	4431	ridoy		ridoy		0.89	s 1	1.79s		95488	K 2	76K		N -					gnome-sof	tware
1734 nadir 9 3.53s 16.34s 1.46 80088K 18488K 8K N- - 5 3 2% compiz 1714 nadir nadir 4 0.09s 4.81s 700.64 73016K 12% 0K N- - 5 3 2% compiz 857 root root 2 5.20s 9.54s 369.24 4968K 288K N- - 5 3 1% Xorg 1779 nadir nadir 5 0.64s 0.11s 850.04 46764K 4096K 0K N - 5 0 1% evolution-cal 1779 nadir nadir 5 0.64s 0.14s 850.04 45764K 4096K 0K N - 5 0 1% evolution-cal 1272 nadir nadir 5 0.64s 0.14s 850.95.44 4524K 7248K 7248K N - 5 1 % evolution-cal 4529 ridoy ridoy 0							61.63							228K							
5988 nadir nadir 4 0.095 4.81s 700.6M 73016K 12K 0K N- - 5 0 2% update-nanage 857 root root 2 5.20s 9.54s 369.2M 53536K 4968K 288K N- - S 3 1% Xorg 4520 ridoy ridoy 5 0.03s 0.67s 850.0M 46780K 0K N- - S 2 1% evolution-cal 1779 nadir nadir 5 0.04s 0.14s 850.0M 46780K 0K N- - S 0 1% evolution-cal 1872 nadir nadir 5 0.04s 0.09s 1.02c 42248K 742.0K N- - S 1 % evolution-cal 1872 nadir nadir 0.02s 0.66s 729.2M 4182X 48K 8K N- - S 1 %<									1.85s												tware
857 root root 2 5.20s 9.54s 369.2M 53536K 4968K 228K N- - S 3 1% Xorg 117 nadir nadir 5 0.60% 859.0M 46790K 0K 0K N- - S 3 1% zorg 117 nadir nadir 5 0.64% 0.14% 859.0M 46764K 4096K 0K N- - S 0 1% evolution-cal 1272 nadir nadir 5 0.64% 0.14% 859.0M 46764K 4096K 0K N- - S 0 1% evolution-cal 1872 nadir nadir 5 0.04% 0.09% 1.06 42248K 32K 0K N- - S 1% evolution-cal 4539 ridoy ridoy 7 0.02% 0.60% 796.1H 4172K 4K N N																					
4520 ridoy 5 0.03s 0.07s 850.0M 46780K 0K N- - S 2 1% evolution-cal 1779 nadir nadir 5 0.04s 0.14s 850.0M 46780K 0K N- - S 2 1% evolution-cal 1779 nadir nadir 5 0.04s 0.14s 850.0M 46764K 4096K 0K N- - S 1% evolution-cal 1872 nadir nadir 5 0.04s 0.09s 1.04 42248K 32K 0K N- - S 1 % evolution-cal 4537 ridoy ridoy 0.02s 0.06s 729.2M 41852K 48K N- - S 1 % evolution-cal 4527 ridoy ridoy 7 0.02s 0.66s 796.1M 41732K 0K N- - S 1 % evolution-cal 32085 </td <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>inager</td>																					inager
1779 nadir 5 0.04s 0.14s 850.0M 46764K 4096K 0K N- - S 0 1% evolution-cal 1872 nadir nadir 5 0.04s 0.09s 1.02c 8.58s 955.4M 45396K 724.8K N+ - 5 1 1% nautius 1872 nadir nadir 5 0.04s 0.09s 1.02c 42249K 32K 0K N+ - 5 1 1% evolution-cal 4527 ridoy ridoy 7 0.02s 0.06s 779.2M 41852K 48K 8K N+ - 5 0 1% evolution-cal 4527 ridoy ridoy 7 0.02s 0.06s 796.1H 41722K 0K 0K N- - 5 1 1% evolution-cal 1856 nadir nadir 7 0.02s 0.08s 652.1H 41720K <																					
4429 ridoy 8 1.12s 8.58s 955.4M 45396K 7248K 7428K N- - 5 1 1% nautilus 1872 nadir nadir 5 0.04s 0.09s 1.0G 42248K 7428K 7428K N- - 5 1 1% nautilus 1872 nadir nadir 5 0.04s 0.09s 1.0G 42248K 32K 0K N- - 5 1 1% evolution-cal 4539 ridoy ridoy 7 0.02s 0.66s 796.1M 41732K 0K N- - 5 1 1% evolution-cal 1856 nadir nadir 7 0.02s 0.66s 868.1M 41722K 144K 0K N- - 5 1 1% evolution-cal 32085 ridoy ridoy 4 0.28s 1.93s 652.1M 34656K 546K N- -																					
1872 nadir 5 0.04s 0.09s 1.06 42248K 32K 0K N- - 5 1 1% evolution-cal 4539 ridov ridov 5 0.04s 0.09s 729.2M 41852K 48K N- - 5 1 1% evolution-cal 4527 ridov ridov 7 0.02s 0.06s 796.1M 41732K 0K N- - 5 1 1% evolution-cal 1856 nadir nadir 7 0.02s 0.08s 868.1H 41720K 14K 0K N- - 5 0 1% evolution-cal 1856 nadir nadir 7 0.02s 0.08s 868.1H 41720K 144K 0K N- - 5 0 1% evolution-cal 1856 nadir nadir 0.02s 0.09s 552.1H 34050K 544K 0K N- -																					-cale
4539 ridoy ridoy 5 0.02s 0.06s 729.2M 41852K 48K 8K N- - S 0 1% evolution-cal 4527 ridoy ridoy 7 0.02s 0.06s 796.1M 41732K 0K 0K N- - S 1 # evolution-cal 1856 nadir nadir 7 0.02s 0.06s 868.1M 41720K 144K 0K N- - S 0 1% evolution-cal 32085 ridoy ridoy 4 0.28s 1.93s 652.1M 34656K 540K 0K N- - S 0 1% gnome-termina 1785 nadir nadir 4 0.19s 1.10s 850.3M 31944K 8048K 20K N- - S 0 1% nultius 1785 nadir nadir 4 0.62s 5.92s 637.9M 30972K <td< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></td<>																					
4527 ridoy 7 0.02s 0.06s 796.1M 41732K 0K 0K N- - S 1 # evolution-cal 1856 nadir nadir 7 0.02s 0.08s 868.1M 41720K 14K 0K N- - S 0 1% evolution-cal 2085 ridoy ridoy 4 0.28s 1.39s 652.1M 34656K 540K 0K N- - S 0 1% gonce-term 1785 nadir nadir 4 0.19s 1.10s 850.3M 31944K 8048K 20K N- - S 0 1% nutlus 1785 nadir nadir 4 0.62s 5.92s 637.9M 30972K 260K N - S 0 1% nutlus																					
1856 nadir nadir 7 0.02s 0.08s 868.1M 41720K 144K 0K N S 0 1% evolution-cal 32085 ridoy ridoy 4 0.28s 1.93s 652.1M 34656K 540K 0K N S 0 1% gnome-termina 1785 nadir nadir 4 0.19s 1.10s 850.3M 31944K 8048K 20K N S 0 1% nautilus 4270 ridoy ridoy 4 0.62s 5.92s 637.9M 30972K 260K 0K N S 3 1% unity-panel-s																					
32085 ridoy ridoy 4 0.285 1.935 652.1M 34656K 540K 0K N 5 0 1% gnome-ternina 1785 nadir nadir 4 0.195 1.105 850.3M 31944K 8048K 20K N S 0 1% nautilus 7270 ridoy ridoy 4 0.625 5.925 637.9M 30972K 260K 0K N S 3 1% unity-panel-s																					
1785 nadir nadir 4 0.19s 1.10s 850.3M 31944K 8048K 20K N S 0 1½ nautilus 4270 ridoy ridoy 4 0.62s 5.92s 637.9M 30972K 260K 0K N S 3 1½ unity-panel-s																					
4270 ridoy ridoy 4 0.62s 5.92s 637.9M 30972K 260K 0K N S 3 1% unity-panel-s																					minal
																					iel-se
1849 root root 5 0.07s 0.72s 500.2M 30012K 6732K 376K N S 1 1% fwupd		1849	root		root	5	0.07	s e	0.72s	500.2M	30012	K 67	32K	376K	N -		Ś		1%	fwupd	

Figure 5.3: Interface of atop tool.

During the experiment CPU Utilization of Suricata was 44% on overage while Suricata ran for 6 minutes. Figure 5.4 indicates the utilization of CPU by Suricata Intrusion Prevention System.

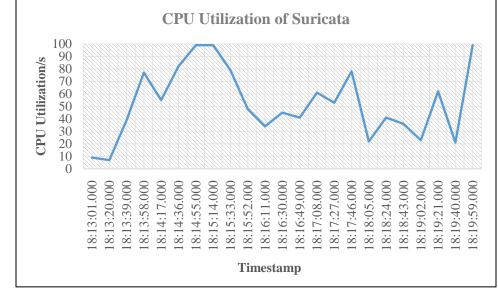


Figure 5.4: CPU Utilization of Suricata

On the other hand in Figure 5.5 shows the CPU Utilization of Snort where average CPU Utilization was 59% during Snort ran for 10 minutes.

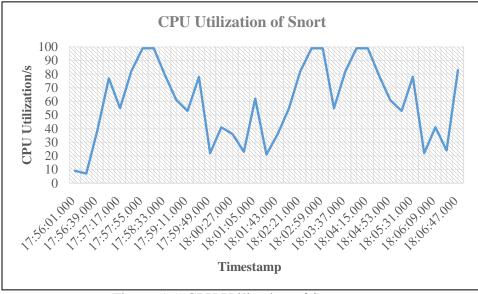


Figure 5.5: CPU Utilization of Snort

Deep level network packet analysis was conducted with most well-known and powerful packet analyzer tool named Wireshark [12]. With this powerful tool packet pattern has

been identified and inspected the malicious and suspicious pattern on a packet. During the experiment most of the network packet was under TCP protocol and less was ICMP and UDP, DNS, HTTP and other protocols network packet. It was identified that same packet was sent from one network to another network during the attacks in several times. And most of them were fragmented and aimed to make denial-of-service of the victim server. Figure 5.6 indicates the UDP packet analysis using Wireshark network packet analyzer tool.

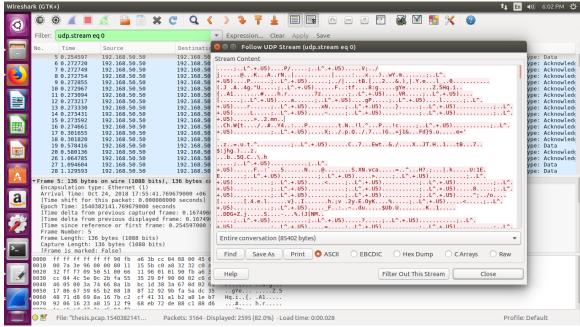


Figure 5.6: Analysis of UDP packet using Wireshark

At the same time during the experiment while attacks were launched from attacker network to victim network both Intrusion Prevention System Snort and Suricata generated prevention notification based on their rules against malicious and suspicious packet called intrusion. Suricata generated alert against 31427 enabled rules and they were downloaded and configured with PulledPork during Suricata installation. Then again Snort were generated alerts against 29471 enabled rules and also downloaded via PulledPork during the installation of Snort. After that generated logs of Suricata and Snort were collected for calculating their accuracy and performance using machine learning algorithms and data mining techniques.

5.3 Experimental Results on CICIDS2017 dataset

Data mining techniques was applied to calculate the prevention accuracy using five machine learning algorithm J48, IBk, MLP (Multilayer Perceptron), BayesNet and Naïve Bayes on 810 data for Suricata and 673 data for Snort of CICIDS2017 Intrusion Prevention System dataset. Table 5.1 indicates the results for Suricata and it is seen that overall classification accuracy of five machine learning algorithm J48, IBk, MLP, BayesNet and Naïve Bayes, J48 is performs better with 97.65% overall classification accuracy.

Algorithms	TPR (%)	FPR (%)	FNR (%)	Pr. (%)	F-1 (%)	OA (%)
J48	97.70	0.50	2.30	97.80	97.60	97.65
IBk	92.70	1.30	7.30	93.00	92.70	92.71
MLP	91.00	3.60	9.00	90.80	90.60	90.99
BayesNet	82.20	1.70	17.80	85.60	82.70	82.22
Naïve Bayes	68.10	3.60	31.90	84.50	72.10	68.15

Table 5.1: Experimental results of Suricata

*** TPR = True Positive Rate, FPR = False Positive Rate, FNR = False Negative Rate, Pr. = Precision, F-1 = F-measure, OA = Overall Accuracy [13]

From the Table 5.2 show results for Snort and among five machine learning algorithm J48 classification accuracy is better with 97.33% accuracy.

Algorithms	TPR (%)	FPR (%)	FNR (%)	Pr. (%)	F-1 (%)	OA (%)
J48	97.30	0.40	2.30	97.30	97.20	97.33
IBk	93.20	1.10	7.30	93.30	93.10	93.16
MLP	90.30	1.30	9.00	90.70	89.90	90.34
BayesNet	85.90	1.60	17.80	86.10	85.10	85.88
Naïve Bayes	81.90	2.10	31.90	86.80	82.80	81.87

Table 5.2: Experimental results of Snort

*** TPR = True Positive Rate, FPR = False Positive Rate, FNR = False Negative Rate, Pr. = Precision, F-1 = F-measure, OA = Overall Accuracy

5.4 Descriptive Analysis and Results Comparison

From experimental result it is found that, Suricata processed 351.70 packet/s on an average in 6 minutes where Snort processed 327.22 packets/s on an average in 10 minutes. So, at this point Suricata performs better than Snort. In terms of CPU Utilization, Suricata used 44% CPU on an average where Snort CPU Utilization was 59% on an average. It is also identified that, in intrusion prevention part Overall accuracy of Suricata is slightly better than Snort. So, after the experiment on results it is proved that, in all cases Suricata performs better than Snort. It is also identified that Suricata perform well due to its multi-thread architectural design and multi-CPU affinity capability where Snort can deal with single-thread process. Figure 5.7 shows the multi-thread architectural design and multi-CPU affinity of Suricata in pictorial format.

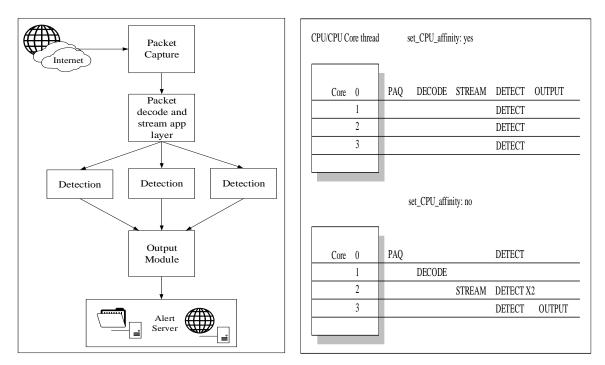


Figure 5.7: Multi-thread, Multi-thread CPU affinity of Suricata [14]

Key difference of both Intrusion Prevention System (Suricata and Snort) also identified and Table 5.3 indicates the key comparison of Suricata and Snort.

Parameter	Suricata	Snort
Intrusion Prevention Feature	Yes	Yes
VRT rule support	Yes	Yes
Emerging threat rules support	Yes	Yes
SO rule support	No	Yes
Multi-thread support	Yes	No
IPv6 support	Yes	Yes
Capture accelerator support	Yes	No
Ease of installation	No	Yes
Configuration filename	suricata.yaml	snort.conf

Table 5.3: Key difference between Snort and Suricata

5.5 Summary

The experimental results shows that Suricata performed well than Snort in terms of latest threats or intrusion on CICIDS2017 dataset. Day by day zero day exploits, malware, ransomware are made to interrupt the network and computer systems. It is necessary to improve existing Intrusion Prevention Systems like Suricata and Snort, make them more efficient to protect from modern intrusions.

CHAPTER 6 CONCLUSION AND FUTURE WORK

6.1 Conclusion

The study conducted with two most common and well-known open-source network-based intrusion prevention system. CPU utilization and performance accuracy was evaluated and compared of both systems. In same structured lab environment experiment was deployed to fulfil the goal of the study. Performance was evaluated based on the latest intrusion prevention system dataset to test their ability for preventing modern intrusions. Both system performed very well during the experiment. But in some cases Suricata's performance was really noteworthy. Due to the difference in their internal structure like multi-thread detecting engine and multi-affinity CPU capability, performance was varied. It was also identified that Suricata used more RAM than Snort for multi-processing functionality. After the experiment it is stated that existing Intrusion Prevention Systems are capable to work against modern known threats. And it is also recommended to use Intrusion Prevention System in Internet-based companies and organization to protect critical infrastructures and to improve data security.

6.2 Future work

Future work of the study could be develop an enhance Intrusion Prevention System that will be capable to identify and protect unknown intrusion in both network and computer systems. As existing IPS are very much dependent on their rules. So they are only capable to protect known threats. Zero-day attack is on the rise. So, it is necessary to improve existing Intrusion Prevention Systems or develop an enhance system.

APPENDIX

APPENDIX A: LIST OF ABBREVIATION

IDS	Intrusion Detection System
IPS	Intrusion Prevention System
HIPS	Host-based Intrusion Prevention System
NIPS	Network-based Intrusion Prevention System
WIPS	Wireless Intrusion Prevention System
IP	Internet Protocol
ТСР	Transmission Control Protocol
UDP	User Datagram Protocol
PC	Democrael Commuter
ic	Personal Computer
CPU	Central Processing Unit
	-
CPU	Central Processing Unit
CPU RAM	Central Processing Unit Random Access Memory
CPU RAM MAC	Central Processing Unit Random Access Memory Media Access Control
CPU RAM MAC OISF	Central Processing Unit Random Access Memory Media Access Control Open Information Security Foundation

APPENDIX B: RELATED ISSUES

root@a	nips: ~/Pych															tų En	4)) 6	:13 PM 🔱
0	root@anip																	
							Ubuntu SMF						2018/10,	24				
							te: 2018/1											
	18:11:30	сри					oftirq %s					:pu_						
	18:11:32	all	94	0		0	Θ	0	0	0	299							
		1	0	0	0	0	0	0	0	0	100							
		2	0	0	0	0	0	0	0	0	100							
=	18:11:34	-11	1 97	0 0	0	0	0	0 0	0 0	0	99							
	18:11:54		97	0		0	0	0	0		295							
		1	1	0	1	0	0	0	0	0	99							
		2	ō	õ	ō	õ	õ	õ	õ	õ	99							
		3	1	õ	õ	õ	õ	õ	õ	2	96							
	18:11:36		97	õ	6	õ	õ	õ	õ	0	298							
I		1	1	0	0	0	0	0	0	0	99							
لثا			Θ	Θ	0	0	Θ	0	0	Θ	100							
			0				0	0		0	100							
A	18:11:38	all	95		8	0	Θ	0	0	Θ	297							
				0	0	0	Θ	0	0	0	99							
a,				Θ	Θ	Θ	Θ	0	0	Θ	99							
		3	1	0	0	0	0	0	0	0	99							
	18:11:40	all	99			0	0	0	0		290							
100		1	3	0	0	e O	0	0	0	0	97							
		2	3	0	1	0	0	0	0	0	96							
		3	1	0	1	0	0	0	0	2	96							
Contract	18:11:42		152	õ	20	0	õ	õ	0	1	226							
P- 1	1																	
		1	21	O	5	o	0	0	0	1	74							
		2	19	0	4	0	Ø	0	0	0	78							
		3	19	Θ		Θ	Θ	0	0	Θ	77							
	18:11:44	all	172		26						201							
			29	Θ		Θ	Θ	0	0		64							
			21	0	9	0	0	0	0	0	70							
		3	27	0	6	0	1	0	0	0	66							
	18:11:46	all	139	Θ	14	Θ	Θ	0	0	4	244							
					_													

Figure A1: Atop output for experimenting CPU utilization

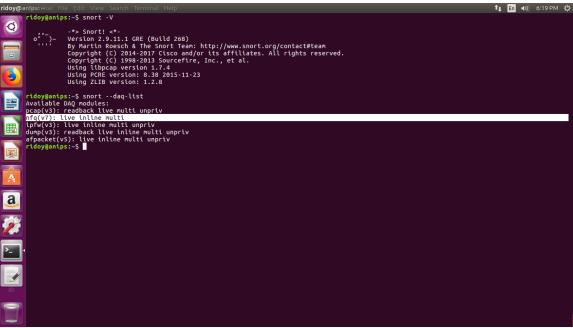


Figure A2: Checking Snort NFQ mode

ridoy@anips:~\$ sudo		list-runmodes
RunMode Type	Custom Mode	Description
PCAP_DEV	single	Single threaded pcap live mode
"pcap_live_auto" w	autofp here packets from	Multi threaded pcap live mode. Packets from each flow are assigned to a single detect thread, unli the same flow can be processed by any detect thread
	workers	Workers pcap live mode, each thread does all tasks from acquisition to logging
PCAP_FILE	single	Single threaded pcap file mode
"pcap-file-auto" w	autofp here packets from	Multi threaded pcap file mode. Packets from each flow are assigned to a single detect thread, unli the same flow can be processed by any detect thread
PFRING(DISABLED) ring_auto" where p		Multi threaded pfring mode. Packets from each flow are assigned to a single detect thread, unlike ' me flow can be processed by any detect thread
	single	Single threaded pfring mode
	workers	Workers pfring mode, each thread does all tasks from acquisition to logging
NFQ	autofp	Multi threaded NFQ IPS mode with respect to flow
	workers	Multi queue NFQ IPS mode with one thread per queue
NFLOG	autofp	Multi threaded nflog mode
	single	Single threaded nflog mode
	workers	Workers nflog mode
IPFW	autofp	Multi threaded IPFW IPS mode with respect to flow
	workers	Multi queue IPFW IPS mode with one thread per queue
ERF_FILE	single	Single threaded ERF file mode
	autofp	Multi threaded ERF file mode. Packets from each flow are assigned to a single detect thread
ERF_DAG uto" where packets	autofp from the same flo	Multi threaded DAG mode. Packets from each flow are assigned to a single detect thread, unlike "day w can be processed by any detect thread

Figure A3: Prove of Suricata's multi-threading capability

ridoy@	anips: ~			tı 🛙	in 🜒)	6:17 PM 🔱
Q	ridoy@anips:~\$ sudo iptables -L Chain INPUT (policy ACCEPT) target prot opt source	destination				
	Chain FORWARD (policy ACCEPT) target prot opt source	destination				
۷	Chain OUTPUT (policy ACCEPT) target prot opt source ridoy@anips:~\$ sudo iptables -I FORWARD ridoy@anips:~\$ sudo iptables -L	destination -j NFQUEUEqueue-num	3			
	Chain INPUT (policy ACCEPT) target prot opt source	destination				
	Chain FORWARD (policy ACCEPT) target prot opt source NFQUEUE all anywhere	destination anywhere N	FQUEUE num 3			
	Chain OUTPUT (policy ACCEPT) target prot opt source ridoy@anips:~\$	destination				
a ð						
<u>a</u> ,						
%						
<u>-</u>	•					

Figure A4: Netfilter Queue (NFQ)

Figure	A5: Packet	dropping i	n Suricata

ridoy@anips: /etc/suricata/rules	★1. En 40) 5:57 PM 43+
	[WDrop] [**] [1:10000001:1] ICMP test [**] [Classification: (null)] [Priority: 3] {ICMP} 172.217.26.78:0 -> 192.168
.50.40:0	[WDrop] [**] [1:10000001:1] ICMP test [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.50.40:8 -> 172.217
.26.78:0	
.50.40:0	
10/24/2018-17:55:47.364200	[WDrop] [**] [1:10000001:1] ICMP test [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.50.40:8 -> 172.217
10/24/2018-17:55:47.412906	[WDrop] [**] [1:10000001:1] ICMP test [**] [Classification: (null)] [Priority: 3] {ICMP} 172.217.26.78:0 -> 192.168
10/24/2018-17:55:48.366018	[WDrop] [**] [1:10000001:1] ICMP test [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.50.40:8 -> 172.217
26.78:0 10/24/2018-17:55:48.412410	[wDrop] [**] [1:100000001:1] ICMP test [**] [Classification: (null)] [Priority: 3] {ICMP} 172.217.26.78:0 -> 192.168
. 50. 40:0	
10/24/2018-17:55:49.367509	[wDrop] [**] [1:10000001:1] ICMP test [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.50.40:8 -> 172.217
10/24/2018-17:55:49.413570	[WDrop] [**] [1:10000001:1] ICMP test [**] [Classification: (null)] [Priority: 3] {ICMP} 172.217.26.78:0 -> 192.168
10/24/2018-17:55:50.368627	[wDrop] [**] [1:10000001:1] ICMP test [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.50.40:8 -> 172.217
10/24/2018-17:55:50.417922	[wDrop] [**] [1:100000001:1] ICMP test [**] [Classification: (null)] [Priority: 3] {ICMP} 172.217.26.78:0 -> 192.168
.50.40:0	
10/24/2018-17:55:51.370036	[wDrop] [**] [1:10000001:1] ICMP test [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.50.40:8 -> 172.217
50,40:0	[WDrop] [**] [1:10000001:1] ICMP test [**] [Classification: (null)] [Priority: 3] {ICMP} 172.217.26.78:0 -> 192.168
10/24/2018-17:55:52.371545	[WDrop] [**] [1:10000001:1] ICMP test [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.50.40:8 -> 172.217
.26.78:0	[wDrop] [**] [1:10000001:1] ICMP test [**] [Classification: (null)] [Priority: 3] {ICMP} 172.217.26.78:0 -> 192.168
. 50. 40:0	
10/24/2018-17:55:53.372992	[WDrop] [**] [1:100000001:1] ICMP test [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.50.40:8 -> 172.217
<pre>10/24/2018-17:55:53.418357 .50.40:0</pre>	[wDrop] [**] [1:10000001:1] ICMP test [**] [Classification: (null)] [Priority: 3] {ICMP} 172.217.26.78:0 -> 192.168
10/24/2018-17:55:54.374479	[WDrop] [**] [1:10000001:1] ICMP test [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.50.40:8 -> 172.217
.26.78:0 10/24/2018-17:55:54.421202	[wDrop] [**] [1:10000001:1] ICMP test [**] [Classification: (null)] [Priority: 3] {ICMP} 172.217.26.78:0 -> 192.168
.50.40:0	[wDrop] [**] [1:100000001:1] ICMP test [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.50.40:8 -> 172.217
.26.78:0	
10/24/2018-17:55:55.425356	[wDrop] [**] [1:10000001:1] ICMP test [**] [Classification: (null)] [Priority: 3] {ICMP} 172.217.26.78:0 -> 192.168

ridoy@anips:~ 11 En 40) 6:28 PM 🕸
03.48.119.119,80,94:DE:80:03:A0:1F,D4:CA:6D:81:6D:56,0xE2,***A****,0x439F2ABE,0x4CD7B749,,0x8180,55,0,48763,212,217088,,,,
10/15-21:12:22.011516 ,1,2013504,3,"ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management",TCP,192.168.50.40,54642,1
03.48.119.119.80,94:DE:80:03:A0:1F,D4:CA:6D:81:6D:56,0xE8,***4,****,0x439F2B5E,0x4CD7B749,,0x8180,55,0,48763,218,223232,,,,
10/15-21:12:22.011516, 1,2013504,3,"ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management",TCP,192.168.50.40,54642,1
03.48.119.119,80,94:DE:80:03:A0:1F,D4:CA:6D:81:6D:56,0xF9,***A****,0x439F2C04,0x4CD7B749,,0x8180,55,0,48763,235,240640,,,,,
10/15-21:12:22.011516 ,1,2013504,3,"ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management", TCP, 192.168.50.40,54642,1
1 03.48.119.119,80,94:DE:80:03:A0:1F,D4:CA:6D:81:6D:55,0xF4,****A****,0x439F2CBB,0x4CD7B749,0x8180,55,0,48763,230,235520,,,,
10/15-21:12:22.011516 ,1,2013504,3,"ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management", TCP, 192.168.50.40,54642,1 03.48.119.119,80.9410E81033A031F, D041CA15015160;556,9475,************************************
5.46.119.119.00,94.02.80.03.40.17.94.04.00.01.00.30,04.7.9.40.40.00.00.00.00.00.00.00.00.00.00.00.
10/13 / 11/12/10/13/10
10/15-21:12:22.011516 ,1,2013504,3, "ET POLICY CNU/LINUX APT User-Agent Outbound Likely related to package management", TCP, 192.168.50.40,54642,1
■ 03.48.119.119.80.94:DE:80:03:A0:1F,D4:CA:6D:81:6D:56.0xF7.***4A****.0x439F2ED4.0x4CD7B7490x8180.55.0.48763.233.238592
10/15-21:12:22.013778 ,1.2013504,3,"ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management", TCP, 192.168.50.40,54642,1
03.48.119.119,80,94:DE:80:03:A0:1F,D4:CA:6D:81:6D:56,0xFA,*****,0x439F2F89,0x4CD7B749,,0x8D00,55,0,48764,236,241664,,,,
10/15-21:12:22.150445 ,1,2013504,3,"ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management",TCP,192.168.50.40,54642,1
03.48.119.119,80,94:DE:80:03:A0:1F,D4:CA:6D:81:6D:56,0xF7,***A****,0x439F3041,0x4CD81D19,,0x9880,55,0,48783,233,238592,,,,
10/15-21:12:22.551332 ,1,2013504,3,"ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management", TCP, 192.168.50.40,54642,1
03.48.119.119,80,94:DE:80:03:A0:1F,D4:CA:6D:81:6D:56,0XF3,***A****,0X439F30F6,0X4CDA4D11,,0XA380,55,0,48883,229,234496,,,,
10/15-21:16:45.052415 ,1,10000001,1,"PING ATTACK",ICMP,172.217.194.101,,192.168.50.40,,D4:CA:6D:81:6D:56,94:DE:80:03:A6:16,0x62,,,,,,43,0,0,84,
80016,0,0,5994,1 10/15-2:116:46.596599 ,1,100000001,1,"PING ATTACK",ICMP,172.217.194.101,192.168.50.40,D4:CA:6D:81:6D:56,94:DE:80:03:A0:1F,0x62,,43,0,0,84,
10/13/21.10.40.390339 11,100000001,1, FINU ATTACK ,1CHF,1/2.21/134.101,,152.106.30.40,,04-CK.00.61.00.30,54-DE-60.05.K0.1F,0X02,,,,,,45,0,0,84,
1/15-21:1647.604560 ,1,10000001,1,"PING ATTACK",ICMP,172.217.194.101,,192.168.50.40,,D4:CA:6D:81:6D:56,94:DE:80:03:A0:1F,0x62,,,43,0,0,84,
86016.0.0.5994.3
[3] 10/16-12:02:53.469676 ,1,2013504,3,"ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management",TCP,192.168.50.40,46478,1
03.48.119.119.80,94:DE:80:03:A0:1F,D4:CA:6D:81:6D:56,0x145,***A****,0x7BBD69BD,0x98328E90,,0x15F,55,0,43953,311,56324,,,,
10/16-12:02:56.613059 ,1,2013504,3,"ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management",TCP,192.168.50.40,46478,1
03.48.119.119.80,94:DE:80:03:A0:1F,D4:CA:6D:81:6D:56,0x145,***A****,0x7BBD6AC0,0x983E2DC0,,0x175,55,0,44481,311,56324,,,,
10/16-12:02:58.062901 ,1,2013504,3,"ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management", TCP, 192.168.50.40,46478,1
83.48.119.119,80,94:DE:80:03:A0:1F,D4:CA:6D:81:60:55,0x141,***A****,0x7BBD6BC3,0x984457D8,0x18C,55,0,44760,307,52228,,,,
10/16-12:02:59.400688 ,1,2013504,3,"ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management", TCP,192.168.50.40,46478,1 0.348.119.119.80.9410E180:03:A0:1F,D41CA:050:181:051:56,04:155.***AP***,0A:780B60C2,0A:984850H8,0A:043,35,56,44943,311,55324,
36363.119.119.00.97.02.00.103.04.1F.j04.04.00.631.00.30.07.04.04.00.631.00.20.00.760000CC2,0X3045030-63,0X145,35,0,044763,311,303241,,,,,
0/10/12/03/04/2533 (201304), EFFOLCE UND/ELING AFT OSET AGE OUTDOUD ELINES FEED OF DEALEST EDING STREAM TO A STATUS AND A
10/16-12:03:03.334035, 1,2013504,3, "ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management, TCP, 192.168.50.40,46478,1
03.48.119.119.80.94:DE:80:03:A0:1F,D4:CA:6D:81:6D:56.0x146.***A****.0x7BBD6ECA.0x98569D900x1D0.55.0.45591.312.57348
10/16-12:03:05.960850 ,1,2013504,3,"ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management", TCP, 192.168.50.40,46478,1
03.48.119.119.80,94:DE:80:03:A0:1F,D4:CA:6D:81:6D:56,0x13F,***A****,0x7BBD6FCE,0x985FF080,,0x1E6,55,0,46016,305,50180,,,,
10/16-12:03:07.140689 ,1,2013504,3,"ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management", TCP, 192.168.50.40,46478,1
03.48.119.119,80,94:DE:80:03:A0:1F,D4:CA:6D:81:6D:56,0x140,***A****,0x7BBD70CB,0x98644AC8,,0x1FD,55,0,46213,306,51204,,,,
10/16-12:03:08.175970 ,1,2013504,3,"ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management", TCP, 192.168.50.40,46478,1
03.48.119.119,80,94:DE:80:03:A0:1F,D4:CA:6D:81:6D:56,0X140,***A****,0X7BBD71C9,0X986817A8,0X214,55,0,46386,306,51204,,,,

Figure A6: Packet dropping in Snort

REFERENCES

- Specarfone Karen, and Peter Mell. "Guide to Intrusion Detection and Prevention Systems (IDPS)." *Guide to Intrusion Detection and Prevention Systems (IDPS)*, Feb. 2007, csrc.nist.gov/publications/detail/sp/800-94/final.
- [2] "PF_RING[™] High-Speed Packet Capture, Filtering and Analysis." *Ntop*, www.ntop.org/products/packet-capture/pf_ring/.
- [3] Sergey Bezborodov. "Intrusion Detection System and Intrusion Prevention System with Snort Provided by Security Onion." *Mikkeli University of Applied Sciences*, 2016.
- [4] Ahmad, Iftikhar, et al. "Comparative Analysis of Intrusion Detection Approaches." *International Conference on Computer Modelling and Simulation*, 2010, doi:10.1109/UKSIM.2010.112.
- [5] Kumar, B.Santos, et al. "Intrusion Detection System- Types and Prevention." International Journal of Computer Science and Information Technologies, vol. 4, no. 1, 2013, pp. 77–82.
- [6] Albin, Eugene. "A Comparative Analysis of the Snort and Suricata Intrusion-Detection Systems." *Naval Postgraduate School, September*, 2011.
- [7] Loganathan, Gobinath. "Real-Time Intrusion Detection Using Multidimensional Sequence-to-Sequence Machine Learning and Adaptive Stream Processing." *The University of Western Ontario*, 2018.
- [8] Lavrova, Darya, et al. "Wavelet-Analysis of Network Traffic Time-Series for Detection of Attacks on Digital Production Infrastructure." SHS Web of Conferences, 2018, doi:https://doi.org/10.1051/shsconf/20184400052.
- [9] Intrusion Detection Evaluation Dataset (CICIDS2017). Canadian Institute for Cybersecurity (CIC), 5 Feb. 2018, www.unb.ca/cic/datasets/ids-2017.html.
- [10] bono, Joan, and Md. Nazrul Islam. "Joanbono/Snorter." *GitHub*, 22 Aug. 2018, github.com/joanbono/Snorter.

- [11] Suricata User Guide. Open Information Security Foundation (OISF), 2018, media.readthedocs.org/pdf/suricata/latest/suricata.pdf.
- [12] Wikepedia. "Wireshark." Wireshark-Wikipedia, https://en.wikipedia.org/wiki/Wireshark.
- [13] Wikipedia. *Receiver Operating Characteristic. Confusion Matrix* https://en.wikipedia.org/wiki/Receiver_operating_characteristic.
- [14] "Threading." *Suricata*, suricata.readthedocs.io/en/suricata-4.0.5/configuration/suricata-yaml.html.