An Analysis of Various Snort Based Techniques to Detect and Prevent Intrusions in Networks

Proposal with Code Refactoring Snort Tool in Kali Linux Environment

RaviTeja Gaddam
Research Scholar: Department of Computer Science
Pondicherry University
Puducherry, India
raviteja.csebec@gmail.com

Dr. M. Nandhini
Assistant Professor: Department of Computer Science
Pondicherry University
Puducherry, India
mnandhini2005@yahoo.com

***Abstract*—Security and reliability are the major concern of our daily life usage of any network. But with the swift advancements in network technology, attacks are becoming more sophisticated than defenses. Although firewalls and router-based packet filtering are essential elements of an overall network security topology, they are not enough on their own. So, to brace the network from unauthorized access the idea of Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) is attracting security experts. This paper briefs various trends in Intrusion Detection & Prevention. To understand various techniques in IDS, this paper analyses various approaches proposed by security researchers specifically using popular open source software Snort as their IDS tool. Being an open source IDS, Snort can be easily configured and deployed in any environment. To assess the efficiency, these research papers are analyzed in various performance aspects like Detection Accuracy, Scalability and Capability of detecting unknown attacks. To overcome various challenges like low detection rate, incapable of handling huge traffic, unsupported automated tuning, etc. that are identified during literature review, this paper proposes a level based architecture. All the levels are designed as incremental i.e. capable of providing the desired functionality and also its lower levels. To prove the efficiency of the proposed architecture, it can be integrated into Snort Tool using Code Refactoring. Also proposed an environment setup to evaluate the modified Snort Tool performance in future.**

Keywords— security; reliability; attacks; Intrusion Detection System; Intrusion Prevention System; Snort Tool; Kali Linux

I. INTRODUCTION

Digital transformation of enterprises is becoming essential as the Internet became a part of daily life. From dawn to dusk, people use Internet to access information like news, stock markets, etc. and to perform online shopping and business transactions. As the rapid advancement in technology not only giving ease of access to the common people but also sophisticated techniques to the cyber criminals. This leads to huge number of cyber-attacks on both individuals and organizations. According to Heidi Shey, Senior Analyst at Forester "Hackers are carefully picking their victim organization, learning its businesses, understanding its partner relationships, and testing for weaknesses and vulnerabilities". Intrusion Detection and Prevention Systems play a vital role against those attacks by shielding critical information.

To understand various techniques and problems in the real time scenarios, this paper analyses various approaches of IDS especially using Snort as their tool. Being an Open Source IDS, Snort can be easily configured and deployed in any environment. To overcome the challenges with Snort this paper proposes an architecture. To evaluate the improved Snort, offensive Operating System Kali Linux environment can be used.

This paper is organized as follows: Section II discusses various categories of Intrusion Detection & Prevention Systems followed by overview about Snort & Kali Linux in Section III. Section IV analyses various Snort based techniques to detect and prevent the intrusions followed by discussing major challenges of IDS in Section V. In Section VI, we propose an architecture for better intrusion detection and environment setup for evaluation. In Section VII, we make a conclusion and discuss about future work.

II. ABOUT IDS & IPS

A. Intrusion Detection System (IDS)

a. About IDS

Intrusion detection can be a collection of techniques that are used to monitor and report the suspicious activities of a system or network. Webster's dictionary describes, an intrusion as "the act of thrusting in, or of entering into a place or state without invitation or welcome".

IDS can be a software or hardware or a combination of both that detects intrusions into a system or a network [1]. Active IDS tries to block the attacks and counter measures or at least alert administrators. Passive IDS just log intrusion details or create traces for audit.

The term "Intrusion Detection" covers an extensive range of technologies that are involved in the detection and reporting of network security events. These techniques can

help to reduce following type of threats by providing attack information to security experts.

- Unauthorized Access
- Data Destruction
- Buffer Overflow attempt
- System or Network Eavesdropping
- Denial of Service (DoS)

There are two key approaches for detecting intrusions: signature-based and anomaly-based. Signature-based detection functions same as a virus scanner. It searches for a known signature for every intrusion event. Any organization need of implementing a more systematic and safer solution, must consider anomaly-based detection. It scans the incoming traffic of a network. It filters out all legitimate traffic. It is well known that to secure a network, we should use a combination of these techniques.

Above discussed two approaches have pros and cons which means none of them is better than the other. With the database of known signatures, signature-based IDS is more reliable and works better when the system receives patterns that match, but is not able to detect new attacks. On the other hand, anomaly-based IDS is able to detect unknown attacks with the drawback of increasing the number of false alarms.

b. IDS Classifications

IDS can be classified into three types:

- Host Based IDS (HIDS)
- Network Based IDS (NIDS)
- Hybrid IDS

Host Based IDS (HIDS)

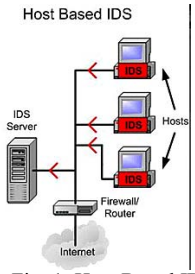

Fig. 1. Host Based IDS

HIDS resides on the host and scans activities. Normally, it scans the log files of Operating System, Application or Data Base for activity traces. Which means it fully depends on the log files. Hence, if the log data corrupted or manipulated by the attacker then HIDS will not able to detect the attack. HIDS scan result will be logged into a secure Database and compared to detect any malicious activity as shown in Fig. 1.

Advantages of HIDS

- Direct control over system entities
- Cost effective

Disadvantages of HIDS

- Harder to manage
- Can be disabled by certain DoS attacks
- Not well suited if host is a part of network
- Cost effective

Network Based IDS (NIDS)

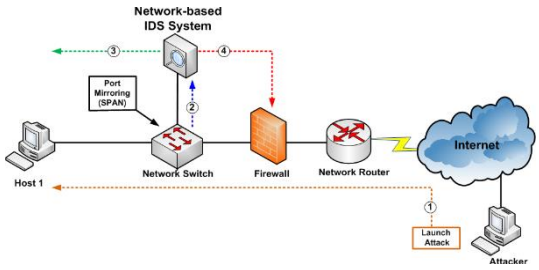

Fig. 2. Network Based IDS

NIDS is liable for detecting abnormal and unauthorized activity in a network. As shown in Fig. 2, it is designed to monitor the packets on a network segment. Most NIDS are pattern based and require identity to alert the attack. The accuracy of these methods are subject to the fine tuning of NIDS.

Advantages of NIDS

- Deployment has little impact on existing network
- If well-placed, can monitor a large network
- Can made invisible to the attackers

Disadvantages of NIDS

- Cannot analyze encrypted information
- Fails to process packets during high traffic
- Can only discern whether an attack was initiated, not if it was successful.

Hybrid IDS

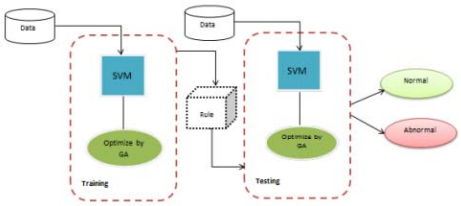

Fig. 3. Hybrid IDS

The Hybrid IDS is an IDS that combines both the features of a HIDS and a NIDS. It monitors events occurring in a system and activities of a network as shown in Fig. 3.

B. *Intrusion Prevention System (IPS)*

a. About IPS

An IPS is a network security or threat prevention technology that scans the traffic and prevents exploits like unauthorized access of an application or system [2]. With a successful exploit, the attacker can bring down the target application or can fully access the compromised application or system.

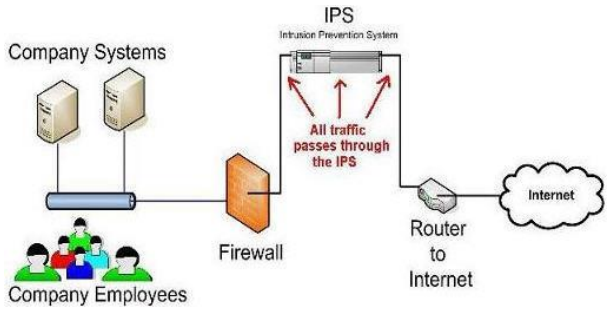

Fig. 4. Intrusion Prevention System

b. Preventing Attacks

The IPS is located behind the firewall to analyse dangerous content. Unlike its predecessor IDS, just scans and reports – the IPS can actively analyse and takes actions on network traffic. For example action like:

- Sending alert to the administrator
- Dropping the malicious packets
- Blocking illegal traffic

IPS should work in a rapid manner without degrading network performance. And accurate detection is essential to prevent threats and false positives.

c. Detecting Attacks

Mainly IPS can detect the attacks by using signature-based or statistical anomaly-based mechanisms. Signature-based detection is based on signatures in the code of each exploit. After detecting the exploit, its signature is stored in the signatures database. Signature detection falls into two types:

- **Exploit-facing** matches with an exploit-facing signature in the traffic
- **Vulnerability-facing** protect networks from different types of exploits that are unknown.

On the other hand, statistical anomaly detection compares the random network traffic sample to a pre-calculated baseline performance level. IPS takes action when the sample activity is beyond baseline performance.

After discussing various types of IDS & IPS, next section gives a brief introduction about Snort and Kali Linux, which are the main focus of this paper.

III. About Snort & Kali Linux

A. *About Snort*

Snort [3] is an open source NIDS created by Martin Roesch in 1998. It can do real-time traffic analysis and packet logging on IP Networks. It can analyze the protocols and can search for matching content.

Snort can also be used to detect various attacks like Operating System fingerprinting attempts, Buffer Overflows, Stealth port scans and so on.

Snort mainly consists of four components: Data sniffers, pre-processor, detection engine as well as log and alarm system. A packet read from the network card is first processed by the pre-processor, and then through rule detection packet in detection engine, if the packet matches the rule, it will be processed in accordance with the rules. The overall architecture is shown in Fig. 5.

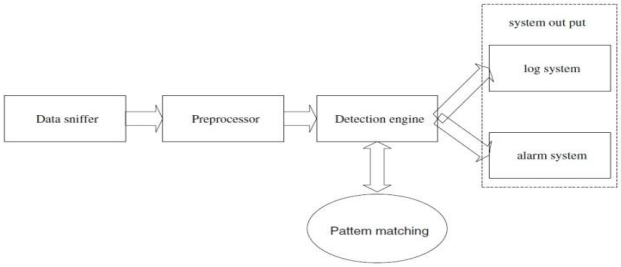

Fig. 5. Snort Architecture

B. *About Kali Linux*

Kali Linux [4] is an Open Source Operating System which can be used to exploit the vulnerabilities of a system/network

Kali Linux is designed for digital forensics and penetration testing and supports the platform for developing and executing security exploits.

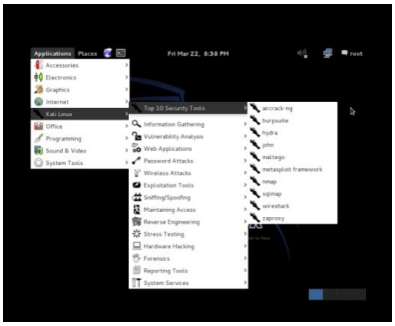

Fig. 6. Kali Linux sample tools list

IV. Literature Review

In this section various Snort based IDS techniques that are proposed in various research papers are discussed. Every approach improvises certain features but with some

drawbacks. Papers from several domains like data mining, Cloud Computing, Neural Networks, etc. are discussed for better understanding of the Snort usage.

Authors of [5] proposed a rule-base IDS along with Efficient Port Scan Detection Rules (EPSDR) for port attack scanning. Snort and BASE tools were used to view the scanning results. After discussing several research proposal methods for port scanning, authors proposed EPSDR. Here the results were evaluated without EPSDR and with EPSDR and it shown a 10% improvement. Due to the nature of Snort detection, i.e. Snort depends on signature match, for new signatures, the port attack detection rate decreases. One more drawback is, it applied only for TCP protocol and left the rest of protocols like UDP.

Authors of [6] used Honeypot to generate Snort IDS rules. According to this, Honeypot collects the attack data, supply this data to IDS and then IDS will evaluate the results and generate the rules automatically. To reduce the resource consumption it used Honeyd, a low-interaction honeypot. By using Backtrack (which is a predecessor of Kali Linux) it configures Honeypot, IDS, Server and Attacker in a virtual environment. During the evaluation of this method rules were generated and detected attacks successfully. After a great appraisal of this innovative work, a major drawback identified that it may not handle all attacks during high traffic.

G. Ahmed, M. Khan and M. Bashir in [7] discussed various methods and drawbacks of IDPS and proposed a six component framework for Linux-based IDS. Along with scanning incoming traffic, it scanned outgoing traffic for better detection and prevention of attacks. Proposed framework first uses a firewall to scan the traffic. Then deep inspection is carried out by decoding and pre-processing components. Later based on signature database, intrusion is detected and this will be logged for analysis purpose. For experimentation it used Ubuntu Linux and Snort running with new set of rules. Explicitly attacked the network with TCP SYN flooding DoS and the system successfully detected it. Main drawback identified is, it uses pcap to capture the packets and framework rearranges the packets to meet the rules of database. This can lead to a bottleneck situation if number of attacks increased. Performance degrades if different protocols are used for DoS attack.

Wonhyung Park and Seongjin Ahn in [8] analyzed the performance of two prominent open source Intrusion Detection Systems: Snort and Suricata. Various features of both the systems like their capability, running modes, processing of packets, alerting, etc. were discussed. According to the authors, Snort processing unit is single threaded whereas Suricata's is multi-threaded. Which clearly states that Suricata has higher detection rate. However with great stability and good detection, Snort has the bigger market share. To test both IDSs, Security Onion Operating System was used and applied Host based IDS mode. Both Snort and Suricata run in two modes i.e. single and multi-threaded. Results were depicting that Suricata performance was better than that of Snort. From this, it was clear that Snort needs enhancements in order to deploy in multi core environment to improve the detection rate.

Authors of [9] discussed about accelerating Snort NIDS using NetFPGA-based Bloom filter. It [9] mainly concentrated on the limitations of Snort like single threaded and medium performance in high speed networks. To overcome these drawbacks Rami Al-Dalky, Khaled Salah, Hadi Otrok and Mahmoud Al-Qutayri proposed a two-line defense mechanism. NetFPGA-based hardware layer acts as the first line of defense and Snort is only used as a second line of defense where deep packet inspection required. To perform pattern matching at NetFPGA, Bloom filter is used. Proposed framework emphasis on combining both hardware-based and software-based components and will result in better detection rate with minimum packet loss in high speed networks. To analyze the performance, proposed mechanism was tested on a network which is connected with a switch network TAP. Resulting graph shown the improvement in detection rate and reduction in packet loss. But this hardware assisted Snort NIDS may not handle DDoS attacks as the hardware layer needs to send the packets to Snort layer for deep inspection.

In [10] authors discussed about detecting attacks in Cloud environment by combining Snort and Back-Propagation Neural Network (BPN). To detect known attacks it [10] used Snort and BPN for unknown attacks. By considering the weaknesses of BPN like slow detection speed, low detection accuracy, etc. authors proposed an optimization algorithm to improvise BPN detection rate. Proposed framework was based on both signature based and anomaly based. It emphasizes on detecting Dos attacks but not concluded how the framework can efficiently prevent DoS and DDoS attacks and share this information among other IDS that are in the Cloud.

Khamkone Sengaphay, Saiyan Saiyod and Nunnapus Benjamas in [11] discussed about Intrusion Detection in private Cloud system. They proposed to improvise the Snort rules and multi-sensors for better behavior detection in private cloud. Authors proposed major rules for Snort like port scanning, behavior checking operating system, etc. For experimentation purpose authors used virtual machines with roles like sensors, attackers, database and monitoring and running on Ubuntu and Windows OS. To evaluate the performance, it used MIT-DARPA 1999 data set and Nmap. During testing, different results identified at different sensors. Major drawback identified is preparing rules for every sensor and coordinating all sensors for better detection rate.

After discussing various techniques, Table I gives a consolidated view of the research papers in various performance aspects like detection accuracy, scalability and capability of detecting unknown attacks.

Various challenges like low detection rate, incapable of handling huge traffic, unsupported automated tuning, etc. are identified during literature review. After discussing some more real time challenges of Intrusion Detection in next section, this paper proposes a level based architecture in subsequent section.

TABLE I. ASSESSMENT OF VARIOUS APPROACHES OF LITERATURE REVIEW

Reference	Approach	Detection Accuracy	Scalability	Capability to detect unknown attacks
[5]	Efficient Port Scan Detection Rules	30%	Yes	No
[6]	Honeypot	100%	No	No
[7]	Firewall based IDPS	100%	Yes	Yes
[8]	Snort Vs Suricata	Full - partial 8 - 19 19 - 58	Yes	Yes
[9]	NetFPGA-based Bloom filter	100%	Yes	Yes
[10]	Back-Propagation Neural Network	------	Yes	Yes
[11]	Multiple Sensors in Cloud Computing	51%	Yes	Yes

V. MAIN IDS CHALLENGES

As discussed in the previous section, many of today's IDS are focused on Signature Detection and designed for Mbps speed network environments. IDS have failed to keep up with the increased sophistication of attacks. Current IDS acts like "sniffers" - which can detect attacks but cannot block before the damage is done [12]. To deploy IDS in any network there are challenges like:

Partial attack coverage: Majority of IDS focus on Signature or Anomaly or DoS detection. Administrators have to procure and integrate separate solutions from different vendors in order to make networks not vulnerable to attacks.

Inaccurate detection: Accuracy and specificity are the two important characteristics of any IDS. Today IDS is lack of specificity and accuracy and generate too many "false positives" – informs security engineers of attacks, when nothing malicious is taking place.

Detection, not prevention: Most of the IDS concentrate on detecting the attacks. Prevention is a reactive activity, sometimes it is too late to put a stop to the intrusion.

Primary design for below 1Gbps networks: Most of the IDS**s** are not keep up with the speed and sophistication of network infrastructure; and cannot work accurately in observing high speed networks.

Poor scalability: Many IDS are designed for low-end deployments and not suitable for medium and large enterprise or government networks.

VI. OUR PROPOSAL

To overcome the above discussed limitations, we propose a new architecture as shown in Fig. 7 that can detect and prevent both known and unknown attacks. The system needs to perform this task with high accuracy.

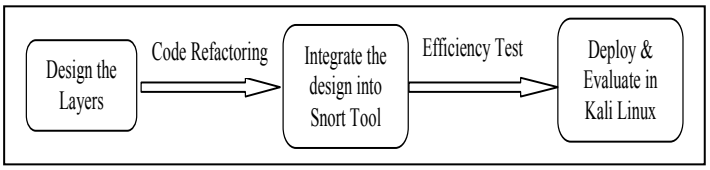

Fig. 7. Proposed Architecture

In general, attacks target at Confidentiality, Integrity and Availability of a network. By considering this, our architecture tries to detect these attacks individually by selecting different attributes for each of the three.

A. Design the Layers

To reduce system complexity and to make it incremental, i.e. making the system react to instantaneous information, and to avoid decision making at later stage, we propose a layer based design as shown in the Fig. 8. Every layer has specific features and more accurate detection rate than that of its below layer. For example, to detect a Denial of Service attack, i.e. to ensure Availability, we don't look into which file was accessed, while this becomes considerable when we would like to ensure data Integrity and Confidentiality.

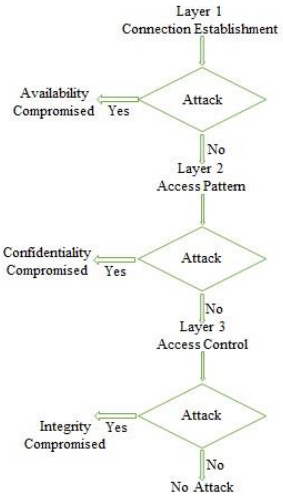

Fig. 8. Layer based design

- **Layer 1**: Connection establishment layer corresponds to user ID, IP addresses, port numbers, number of connections etc. Used to detect attacks exploiting the availability aspect such as Denial of Service attacks, etc.
- **Layer 2**: To ensure data confidentiality this layer detects attacks like port scanning, packet capturing, etc. This layer prevents the unauthorized access to device and data.
- **Layer 3**: To ensure data integrity this layer detects attacks like session hijacking i.e.; unauthorized access to services and is more concerned with the file modifications; user privileges etc.

B. *Code Refactoring the Snort Tool to integrate the proposed design*

Code refactoring is used to improve the functionality and reliability of a software. It is a process of restructuring the code without changing its behavior.

By using Code Refactoring, integrate the Layer Based design into Snort tool so that the efficiency of the Snort IDS can be enhanced.

C. *Deploy & Evaluate the modified Snort tool in Kali Linux Environment*

After getting ready with the Snort Tool, deploy the Snort IDS in a Kali Linux based system in an organization.

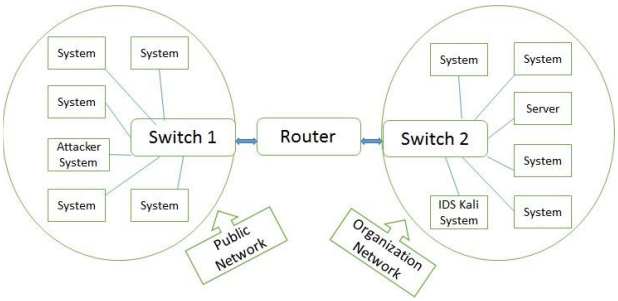

Fig. 9. Evaluation environment set up

Environment set up as shown in Fig. 9 can be used to evaluate the performance of modified Snort Tool. Try to attack the network systems including Server, in different ways (i.e. Port Scanning attack, DoS attack, etc.) with Attacker System from public network. In this setup, organization network and public network are connected through a router to simulate the usage of Internet. Gather the test results from the Snort system, analyze them and compare against earlier discussed systems.

VII. Conclusion and Future Work

After familiarizing with IDS and its classifications, different Snort based Intrusion Detection techniques are discussed in this paper to upkeep the security of an organization against attacks. Snort based IDPS using efficient rules, Bayseian Network, Honeypot, Hardware-assisted technique, Neural Networks and Multi-Sensors like techniques can protect from simple intrusions to dangerous DoS and DDoS type attacks in high speed and Cloud environments with considerable drawbacks. Various challenges are identified and discussed which are to be considered while designing efficient IDS for any network. This paper proposes an architecture to enhance the efficiency of Snort IDS. There are still many ways to enhance the efficiency of Snort based Intrusion Detection and Prevention System. In future we will integrate the proposed design into Snort tool and evaluate it to achieve better detection rate with less false alarms.

Acknowledgment

We would like to thank the anonymous reviewers for their valuable feedback. We would like to acknowledge all the authors of respective research papers that are taken for our analysis purpose. We would like to thank our Computer Science Department for providing necessary resources for our work. This paper reflects the views only of the authors, and others cannot be held responsible for any use which may be made of the information contained therein.

References

[1] "Basics of Intrusion Detection Systems - HackThis!!", *HackThis!!*, 2016. [Online]. Available: https://www.hackthis.co.uk/articles/basics-of-intrusion-detection-systems.

[2] "What is an intrusion prevention system?", *Paloaltonetworks.com*, 2016. [Online]. Available: https://www.paloaltonetworks.com/documentation/glossary/what-is-an-intrusion-prevention-system-ips.

[3] "Snort - Network Intrusion Detection & Prevention System", *Snort.org*, 2016. [Online]. Available: https://www.snort.org/.

[4] "Kali Linux", *En.wikipedia.org*, 2016. [Online]. Available: https://en.wikipedia.org/wiki/Kali_Linux.

[5] S. Patel and A. Sonker, "Rule-Based Network Intrusion Detection System for Port Scanning with Efficient Port Scan Detection Rules Using Snort", *International Journal of Future Generation Communication and Networking*, vol. 9, no. 6, pp. 339-350, 2016.

[6] A. Sagala, "Automatic SNORT IDS rule generation based on honeypot log", *2015 7th International Conference on Information Technology and Electrical Engineering (ICITEE)*, pp. 576-580, 2015.

[7] G. Ahmed, M. Khan and M. Bashir, "A Linux-based IDPS using Snort", *Computer Fraud & Security*, vol. 2015, no. 8, pp. 13-18, 2015.

[8] W. Park and S. Ahn, "Performance Comparison and Detection Analysis in Snort and Suricata Environment", *Wireless Pers Commun*, 2016.

[9] R. Al-Dalky, K. Salah, H. Otrok and M. Al-Qutayri, "Accelerating snort NIDS using NetFPGA-based Bloom filter", *2014 International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 869-874, 2014.

[10] Z. Chiba, N. Abghour, K. Moussaid, A. omri and M. Rida, "A Cooperative and Hybrid Network Intrusion Detection Framework in Cloud Computing Based on Snort and Optimized Back Propagation Neural Network", *Procedia Computer Science*, vol. 83, pp. 1200-1206, 2016.

[11] K. Sengaphay, S. Saiyod and N. Benjamas, "Creating Snort-IDS Rules for Detection Behavior Using Multi-sensors in Private Cloud", *Lecture Notes in Electrical Engineering*, pp. 589-601, 2016.

[12] F. Gong. Next generation intrusion detection systems (IDS). McAfee Network Security Technologies Group, 2002.