# A Novel Multi-threaded Network-Based Intrusion Detection and Prevention System

[1]Raz M. Yousufi, [2]Punit Lalwani, [2]M.B.Potdar

*[1]PG Student, Gujarat Technological University, Ahmedabad, India [2]Bhaskaracharya Institute for Space Applications and Geo-informatics Gandhinagar, India*

## Abstract

*Recently, the security of an individual computer to large networks as a result of a dramatic growth of new devices connecting to the internet, has become one of the biggest challenges. Along with growing new types of security attacks, many protection mechanisms have taken to improve the privacy and security of sensitive information. Detection of abnormal behavior can help network administrators to identify intrusions but cannot prevent them from breaking into home network. Furthermore, using traditional methods which firewall and IDPS systems reside in different machines that results to low performance by filtering and checking traffic in multi points.*

*This paper is providing an efficient method of both detecting and preventing network threats efficiently. To achieve such goal, we used a combination of Suricata NIDS and Netfilter Packet filtering framework of Linux based systems.*

***Keyword:-*** *IDS, IPS, Netfilter, Intrusion Detection Systems, NIDPS, Multi-threading.*

---

## 1. INTRODUCTION

Intrusion Detection and Prevention System (IDPS) is designed to monitor individual systems or networks to find out suspicious activities and generate an in-time alarms to administrators as well as react to malicious traffics accordingly. There are different methods for detecting and preventing intrusions, but false-positives and false-negatives are two common weaknesses of all types [1]; there are techniques taken to reduce two negative points such as using a hybrid system [2]. As the network traffic is growing rapidly, the traditional IDPS systems face the lacking of expected performance. Hence this paper provides an acceptable technique to provide high level security to networks as well as high performance while using the advantages of multi-threading.

Firstly, we discussed on background and related works in section II. Then in section III we explained the IDS/IPS systems with types and techniques that are used to detect/prevent malicious traffic. Section IV defined the netfilter packet filtering framework and how it allows users to handle and manipulate the packets from kernel space. Then, we defined Suricata multi-threaded IDS and then, we have implemented our proposed architecture and flow chart. Finally, in section V we have concluded the paper and outline future work.

## 2. RELATED WORK

From security point of view, a great work has been done to both detecting and preventing malicious traffic from entering to the internanetwork such as customizing the detection and prevention part [4], using only misuse or anomaly system at the same time etc. There are many methods and techniques [5] [6] that previous papers have dealt with to increase the security in the network while not affecting the performance of the system. But they are not as efficient as the size of network traffic is dramatically increasing.

(Daniel Kavan, et al., 2014) combined the advantages of different IDS systems. He used Snort for signature-based attack detection and CAMNEP for behavior detection purposes but the core system of Snort is single-threaded. Furthermore, combining multiple systems causes a performance degradation and resource consuming.

(Nen-Fu Huang, et al., 2015) designed an IPS system using the OpenFlow communication protocol. They used Snort to detect botnet/malware and an SDN controller to centrally control the network. They used honeypot to trap attackers and an anti-scanning system to alarm against scanners. With all of these, the snort is not capable of handling high traffic as well as zero-day attacks will remain undetected.

There should be such a system that can handle gbps traffic while the security of the system should not be affected. Here, this paper provides an efficient architecture of the IDPS that can overcome both the weaknesses.

## 3. Multi-threading Technology

### 3.1. Overview

A thread represents as the smallest unit of execution to which a processor allocates time [14]. Multi-threading is a single core or multi-core processor’s ability to execute multiple threads at the same time (as shown in figure 1). A scheduler is used to switch between instructions execution from multiple threads on a regular bases called Time Slicing (i.e. 1 milli sec).

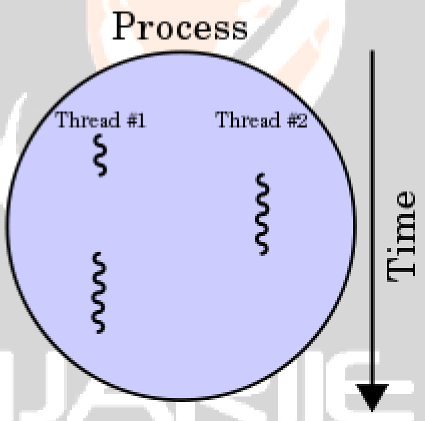

**Fig -1:** Multi-threading

The figure above depicts a process with two threads and the instructions from both threads are executed in regular time slices.

There are several types of multi-threading but among all the Simultaneous Multi-Threading (SMT) is the most advanced one which is able to execute multiple instructions from multiple threads concurrently [14].

### 3.2. Advantages of Multi-threading

In a thread, instructions highly depend on each other’s result so CPUs experience idle cycles and cache miss. So, multi-threading allows the CPU to execute instructions from other threads. This way, threads are executed concurrently and the overall performance is increasing.

Although in a single-core CPU one cannot experience the real power of multi-threading but in today’s multi-core CPUs or Multi-processor systems, multi-threading plays a crucial role in system performance.

## 4. WHY NIDPS IS NEEDED?

Before discussing the importance of NIDPS, it is better to know the role of IPS in a network and how it can react to attacks.

### 4.1. IDPS

IPS or IDPS1 is basically an IDS which can detect malicious traffic but at the same time it will be able to block them as it resides in-line on the network [16] and acts as a gateway, so that attacks such as DoS should not be succeeded to reach their targets. IPS systems plays a crucial role in detecting and blocking both intrusions such as malwares and DoS activities. Without an IPS system, an attacker can easily compromise the systems by propagating malwares or attacks like DoS which will result in deletion, modification and stealth of confidential data or unavailability of crucial services.

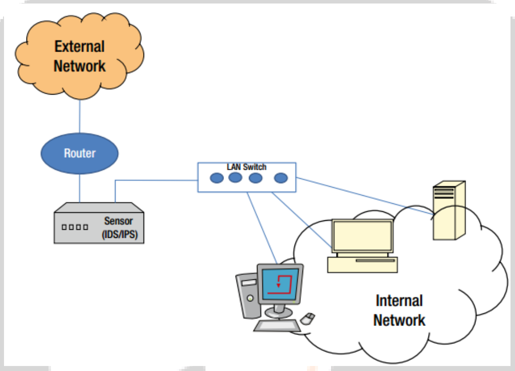

**Fig 2:** IDPS inline mode

IPSs are classified into four main types [12] [16]

a. Host Based IPS
b. Network Based IPS
c. Network Behavior Analysis
d. Wireless IPS

### 4.2. NEED FOR NETWORK IDPS

There are many advantages for selecting Network-based IDPS rather than other systems and they are:

- ✓ Only a single system is needed to monitor and protect a network of hundreds or thousands computers.
- ✓ A single system can be monitored and managed easily rather than multiple systems.
- ✓ NIDPS can protect a network against any type of attacks DoS, DDoS, Ping Flood etc.)
- ✓ This system can protect network devices such as Firewalls, routers, Printers etc. against attacks as well.

So, NIDPS is a mandatory and crucial system for any type of network ranging from small sized to medium and large sized networks.

### 4.3. Intrusion Detection Methods

### 4.3.1. Misuse/Signature based Detection

Signature based detection system uses from a database of known signatures (known attacks) for detection purposes. It is simple to configure and employ. It generates less false alarms compared to anomaly based system. But, the worst drawback of such systems is that they are unable to detect any attacks which does not match the signatures [7] (obfuscations). See example below:
script/./././././admin.pwd=script/admin.pwd

In this example, for a normal system both commands are the same while for signature-based IDS they are different commands. See Figure 3.

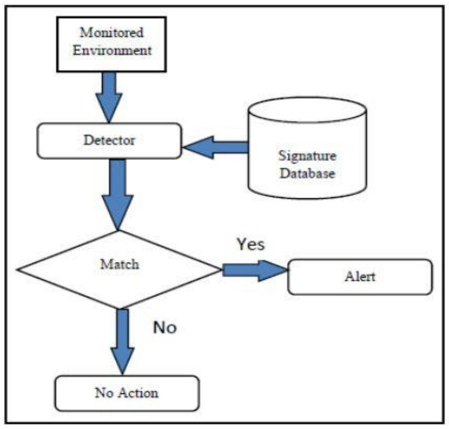

**Fig-3:** Signature based detection technique

### 4.3.2. Anomaly based Detection

In anomaly detection method, the system first creates a baseline for normal behavior of the network and compares any change to the baseline. If there is any abnormal behavior, then it mark it as intrusion and generates alarm. Anomaly detection technique is excellent in detecting unknown or Zero-day attacks[2] but they are suffering from large number of false alarms.
Examples of abnormal behaviors are: highly consuming system resources, initiating many connections, consuming more network bandwidth etc. See Figure 3.

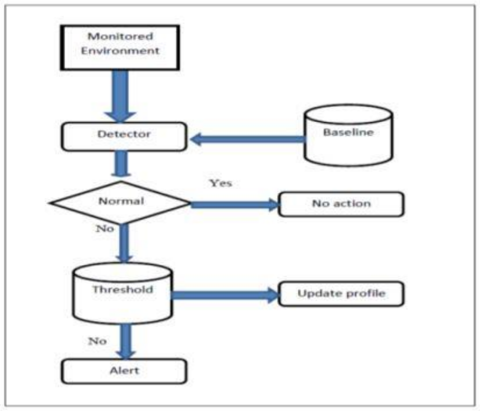

**Fig-4:** Anomaly based detection technique

### 4.3.3. Stateful Protocol Analysis

SPA known as deep packet inspection, is a system which adds the stateful characteristic to normal protocol analysis. Protocol analysis helps to analyze TCP or UDP payloads that carries other protocols such as HTTP, FTP and DNS. IPSs knows how these protocols work based on their RFCs and any suspicious behavior can be easily detected but the protocol analysis examines only single request or response. An attack would not becarried on a single request but will succeeded on a series of request and response. To detect such attacks, stateful analysis will be the best one. Because it keeps and monitors the whole events within the whole session[13]. The main drawback of such systems is the resource requirements and the system overhead because of complex analysis of traffic[8].

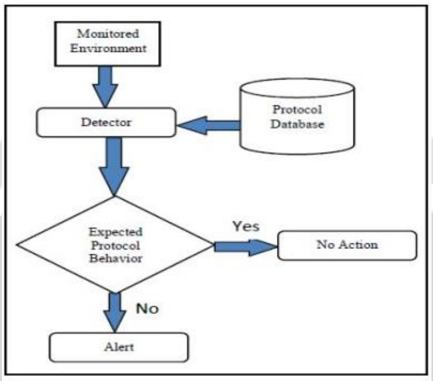

**Fig-5.** Stateful Protocol Analysis

### 4.3.4. Hybrid Systems

Such systems are created as a result of combining any two or more previous system to increase the level of security. Such systems are monitored through a single console-line, to achieve high level of efficiency and accuracy while reducing the false positive/negatives remarkably [2].Figure 6 shows the hybrid system.

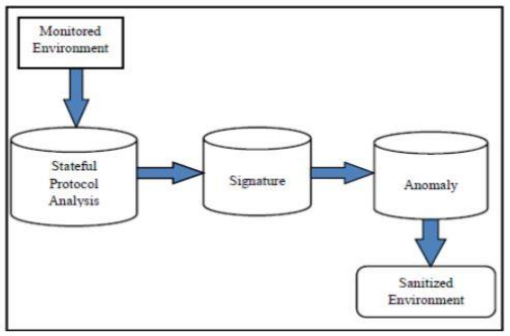

**Fig-6:** Hybrid detection technique

## 5. SYSTEM ARCHITECTURE

The proposed architecture is depicted in figure 9. As mentioned before, we use Suricata for detection purposes and netfilter firewall for protection purpose. In the followings we will discuss the netfilter framework, its different hooks and how it allows a user application to manipulate packets and after, we will briefly explain suricata multithreaded IDS system.

## 5.1. NETFILTER FRAMEWORK

Netfilter is a packet filtering utility which introduced in linux kernel 2.4 and later versions [8]. Iptables is connected to netfilter kernel mode and only provides an interface to the netfilter framework [11]. Netfilter is in conjunction with the TCP/IP stack [9] so allows userspace applications to perform Network Address Translition, manipulate or can remove the data grams [15]. In figure 7 the architecture of Netfilter/Iptables is depicted.

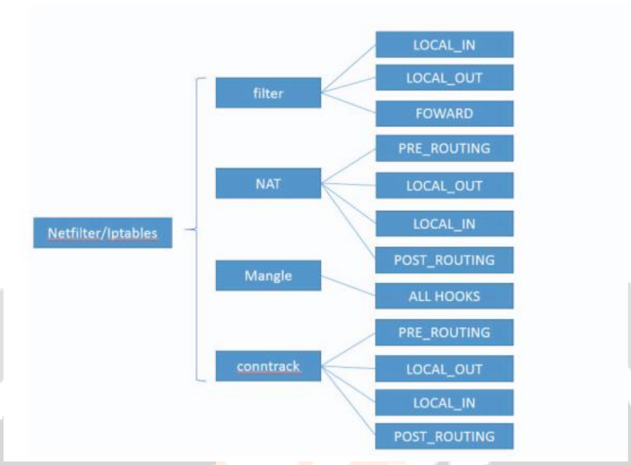

**Fig-7:** Architecture of netfilter firewall

In fact netfilter is comprised of five hooks as defined bellow [3][15]:

1) NF_INET_PRE_ROUTING: this hook is the first filtering point for incoming packets, it filters packets before they are routed for local process or other interfaces.

2) NF_INET_LOCAL_IN: after routing decision has taken, this hook is triggered and it is filtering packets before they reach its destination port.

3) NF_INET_LOCAL_OUT: this hook is used to filter packets generated from local system itself. Before routing decision, this hook is triggered.

4) NF_INET_FORWARD: if a packet does not belong to the local ports, it passes through this hook to be out for another interface.

5) NF_INET_POST_ROUTING: this hook is the last point of filtering. It filters the packets which are about to be put on the wire. This hook is a proper point for performing Source-NAT as of PRE_ROUTING which is acceptable for Destination-NAT.

Packet filtering functions can be inserted/registered with this hooks. After verdict is issued, the return values are one of the followings [15]:

- NF_DROP: the process stops and packet is dropped.
- NF_ACCEP: Packet is accepted and passed to next hook.
- NF_STOLEN: packet is not dropped but kept for further processing.
- NF_QUEUE: the packet is put in queue and can be processed or removed from user space like iptables etc.
- NF_REPEAT: not pass to the next hook but invoke it again to current packet filter-function.

Among all, NF_QUEUE is very important for our purpose, because it allow us to handle and process packets as of our wishes. Suricata can grab packets from queue to analyze them for detecting intrusions.

## 5.2. Suricata

Suricata is an open source and powerful IDS system which is recently introduced, but it is rapidly growing to take the market for its unique characteristics such as multi-threading, GPU support, IP repudiation etc.

The most problem faced with snort is being its core single-thread [10] which causes it to a low performance and it will not be a good solution for multi-cores especially GPUs. Suricata is a good replacement with its high performance at the time. Figure bellow depicts a simple architecture of Suricata.

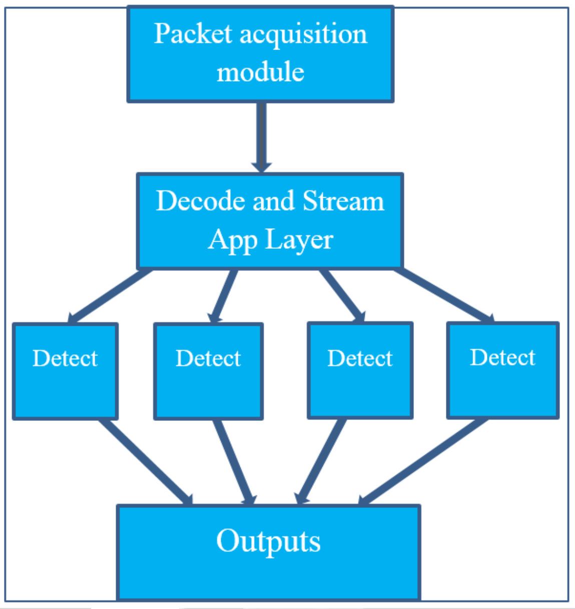

**Figure 8**: Architecture of Suricata IDS

Packet acquisition unit captures the incoming packets or stored PCAP files and then pass it to decode module, when it is decoded as per protocol, the streams is resembled to construct original stream. The resembled traffic is distributed to detection modules which basically the packets are distributed by IP streams. Finally the packets send to the output module to alert and can be logged.

## 5.3. IDPS System architecture

In the proposed architecture, the incoming packets pass through netfilter hooks and then the respected hooks are triggered. There are five hooks in the netfilter and each hook is a point for filtering packets. Once packet enters to the hook, it checks whether it should be queued or not, if its target is set to queue (in the filter table) then it is buffered and the user-space program here (Suricata) can handle it through Libnetfilter_queue [16]. Libnetfilter_queue is a userspace Library which provides an API to packets which queued by netfilter. Figure 9 depicts the proposed NIDPS system.

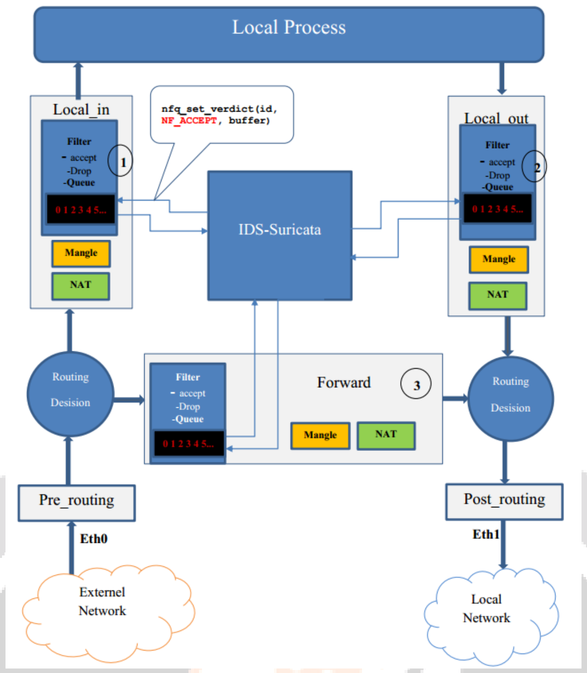

**Fig-9:** Proposed IDPS architecture

1, 2 and 3 shows three hooks that contains a filtering table. The PRE_ROUTIGN and POST_ROUTING hooks are performing source/ destination NATing and connection tracking.

- Filter table
  Filter table is used to filter packets. You can set a target for a packet (i.e. accept, drop, queue). If the target is accept, the packet is allowed to pass to next hook. If it is set to drop, packet will be dropped and process stops. But, if target is set to queue, then it is labeled (integer i.e. 1, 2, 3, 4 etc.) and queued by netfilter.
- Suricata IDS
  After the packet is queued, Suricata can capture it and after analyzing, a verdict is issued (i.e. NF_ACCEPT) to this labeled packet and then it releases the queue.

The flow chart of our architecture is as bellow

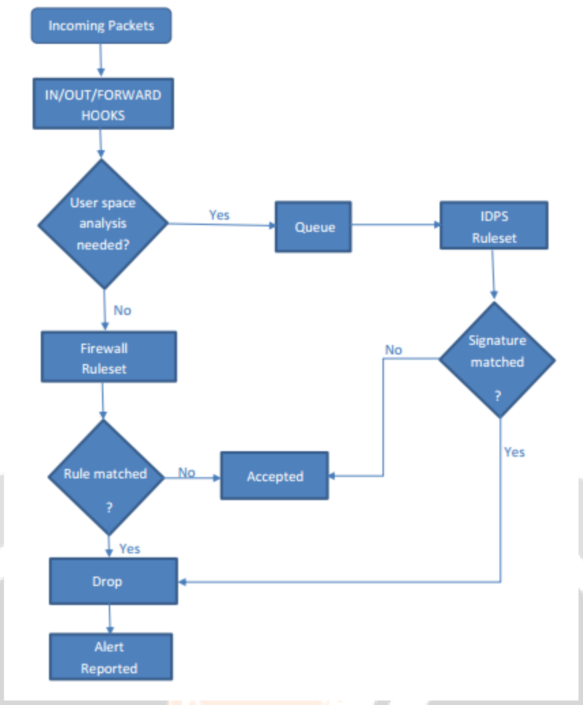

**Fig-10:** Flow chart of proposed system

## 6. CONCLUSION AND FUTURE WORK

A huge research has done to apply high level of security as the number of new and complex attacks are growing. One very effective way to protect networks and personal data from being stolen or alteration, in time, is using IDS/IPS system. IDS and IPS both are similar but the main difference is that IDS systems can reside anywhere in the network while IPS systems have to sit in-line, means that IPS should act as a bridge and all traffic should pass through it. In this case, it can allow or deny the traffic which is incoming or outgoing the network.

One of the main challenges almost every IDPS systems faces, is the bandwidth growth which results the low performance and attacks can succeed if the system is unable to handle all traffic. In this paper a multi-threaded solution has proposed to overcome this drawback.

As part of future work, we will try to make the system OS independent so that it work with any type of system without performance degradation.

## 7. REFERENCES

[1] Shivaji P. Mirashe, N. V. Kalyankar, “3Why We Need the Intrusion Detection Prevention Systems –IDPS) In IT Company”.IEEE, 2010.

[2] Rajeev Agrawal, David Mudzingwa, “A study of Methodologies used in Intrusion Detection and Prevention Systems (IDPS)”. IEEE, 2012.

[3] Justin Ellingwood. “A Deep Dive into Iptables and Netfilter Architecture,” 20 Aug 2015.

[4] Minoo Sadat Mirpuryan, Tina Tavizi, HosseinGharaee“A Comprehensive Network Intrusion Detection and Prevention System Architecture,”6'th International Symposium on Telecommunications (IST'2012), IEEE, 2012.

[5] K. Alsubhi, I. Aib, J. Franc ̧ois, and R. Boutaba, “Policy-based security configuration management application to intrusion detection and prevention,” in IEEE conference on Communications (ICC), IEEE, 2009.

[6] Wattanapongsakorn, S. Srakaew , C. Charnsripinyo, “A Practical Network-based Intrusion Detection and Prevention System”, IEEE, 2012.

[7] Corbin Del Carlo.“*Intrusion detection evasion: How Attackers get past the burglar alarm”.* SANS Institute, SANS Great Lakes, Chicago Illinois. Vol 1.4b. Sep 25, 2003.

[8] Nagesh Vaidya, Parikshit Godbole. *“Hardware Implementation of Key Functionalities of NIPS for High Speed Network,”* IEEE, 2015.

[9] Baoliang Wang, Kaining Lu, Peng Chang, *“Design and Implementation of Linux Firewall Based on the Frame of Netfilter/IPtable,”* ***IEEE**, 2016.*

[10] Eugene Albin and Neil C. Rowe,”*A Realistic Experimental Comparison of the Suricata and Snort Intrusion-Detection Systems,”***IEEE,** 2012.

[11] Michael Rash. *“Linux Firewalls”. No Starch Press. 2007,* pp 09.

[12] K. Scarfone, P. Mell,“Guide to Intrusion Detection and Prevention Systems”, NIST Special Publications, Feb 2007.

[13] *Umesh Hodeghatta Rao, Umesh Nayak. “The Infosec Handbook”. Apress. Sep 2014, pp 125-243.*

[14] Lin Gao. *“Multithreading”, cs9244 report,* 2006.

[15] Klaus Wehrle, Frank Pählke, Hartmut Ritter, Daniel Müller, Marc Bechler. *“The Linux Networking Architecture: Design and Implementation of Network Protocols in the Linux Kernel”.Prentice Hall. Aug 2004, pp 323-325.*

[16] Libnetfilter Queue Project. http://www.netfilter.org/projects/libnetfilter_queue/index.html