SURICATA

MIS311 PROJECT

FATMA ÜNAL 16030411051

MAKBULE MERVE AKARLAR 16030411045

Information security is the protection of corporate and personal information against unauthorized access. The fact that information is stored securely in digital archives, and that it can only be accessed and changed by authorized persons, is the basic element of information security. In today's conditions, where the need for security is very important, it is a process that includes many facts in terms of information systems.

With the spread of information systems and especially Internet technologies, security incidents arising from vulnerabilities in these systems have started to increase. Malicious Activities that can be carried out using computer networks can occur as a result of unconscious use. However, the rate of attack activities that intentionally harm the system by taking advantage of the vulnerabilities of the system is higher. Especially as a result of the development and spread of web technologies, the number and types of attacks have increased. Many tools have been developed to prevent these vulnerabilities and ensure security. It has been tried to develop security solutions such as authentication, authorization, antivirus programs against many different types of attacks.

Information security threats are basically divided into two in terms of the source point of the incident. Attacks originating from people working within the organization are called "internal threats", and attacks originating from people outside the organization are called "external threats". These threats are the second biggest threat after viruses.

With the rapid development of computer and information technologies, the use of electronic storage media is increasing day by day. Despite the convenience provided by this situation, the protection of information stored in cloud environments has also become a great and very important need. The protection systems to be used may differ according to the importance of the information. The main purpose of these systems is to provide maximum information security by increasing the level of precaution against malicious people and attacks.

With the technological developments, the importance of information security is increasing day by day in the digital information age. In terms of individuals, institutions, and organizations, it is very important that the information is stored and confidential, and that it is accessible only by authorized persons when requested.

Intrusion Detection System (IDS)

Devices or software used to monitor malicious activities or policy violations against networks or systems. Any detected activity or violation is either reported to an administrator or collected centrally using a security information and incident management (SIEM) system. The SIEM system combines output from various sources and uses alarm filtering techniques to

separate malicious alarms from false alarms.

Intrusion detection system mainly includes the following actions;

• Sending a warning to the administrator in case of attack,

• Dropping malicious packages,

• Blocking traffic at the source address,

• Resetting the connection,

• Correcting CRC errors,

• Combine the packet flow,

• Sorting the incoming segments by looking at the Sequence Number at the TCP layer and reporting if there is any missing, etc.

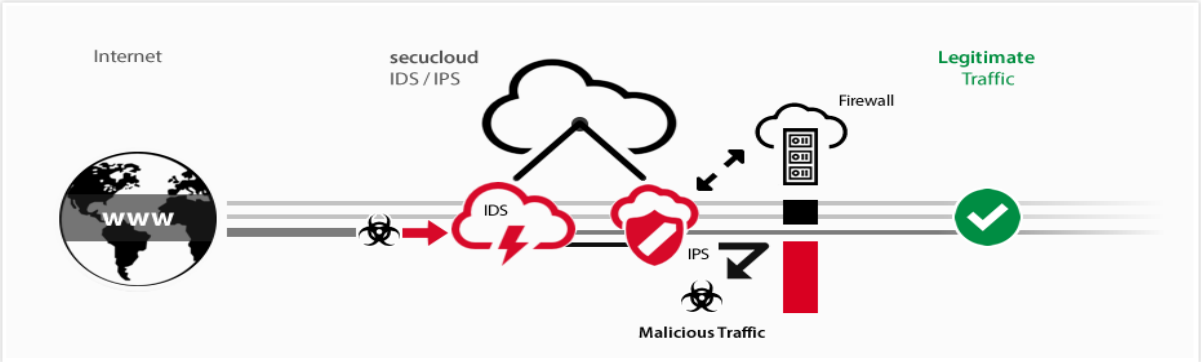

Figure 1

The main difference between Intrusion Detection Systems and Intrusion Prevention Systems; Intrusion detection systems only detect and report attacks, whereas intrusion prevention systems have the ability to prevent attacks.

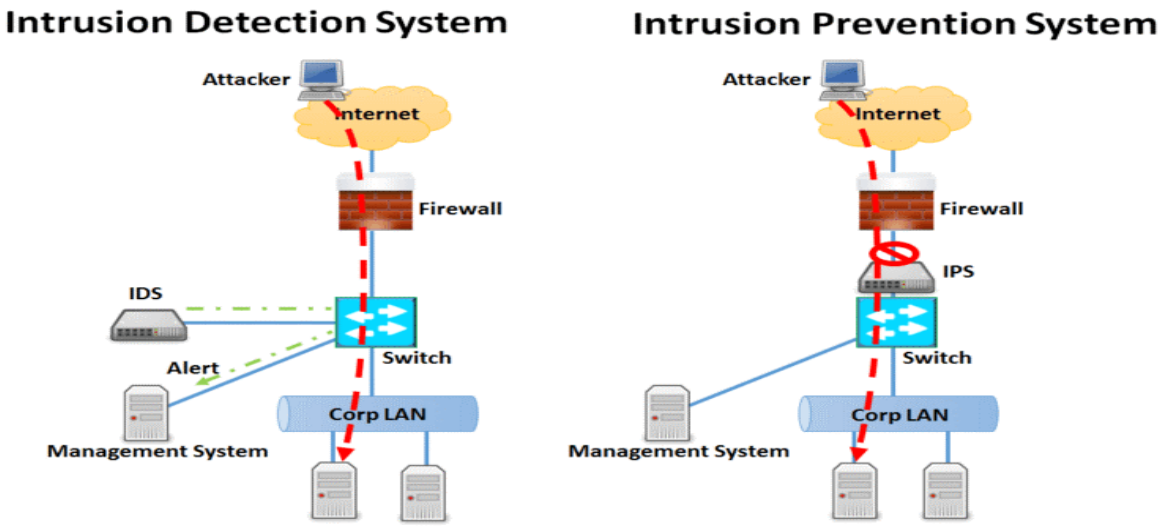

Figure 2 - Difference between IDS and IPS

Intrusion Prevention System (IPS)

The main things they do are:

• Detects attacks from program or person.

• Records attack patterns to improve detection logic.

• Gives warning and report.

• Records all attacks in a database for evidence.

• Quarantines the damaged system.

• Ensures data integrity, accessibility, and confidentiality.

Intrusion Prevention System Software

• Snort; It is an open-source intrusion detection and intrusion prevention system and was first developed by Martin Roesch in 1998.

• Suricata; is an open-source intrusion detection system and intrusion prevention system. Developed by the Open Security Foundation.

• Bro (Zeek); It is a free and open-source software network analysis framework.

• OSSEC; It is a free, open-source host-based intrusion detection system. It performs log analysis, integrity check, Windows registry monitoring, rootkit detection, time-based alert, and active response.

• Samhain Labs; The host-based intrusion detection system (HIDS) provides rootkit detection, port monitoring, detection of rogue SUID executables, and stealth operations, as well as file integrity checking and log file monitoring/analysis.

Many different studies have been made and continue to be done to ensure security in information systems. In this study, Suricata, one of the Intrusion Detection System tools, which is one of the indispensable tools of information security systems, has been examined in detail.

SURICATA

1. Development

Suricata is an open-source intrusion detection and prevention system distributed under the GPLv2 license. It is developed and supported by the OISF (Open Information Security Foundation), a non-profit community. First beta release in December 2009, June
In 2010, the first stable version was published. It works based on signature/rule just like the Snort intrusion detection system, which was announced about 10 years ago and is widely used. Supporting the ruleset used by Snort was effective in its acceptance in a short time.

Suricata, named after a carnivorous mammal (meerkat) native to Africa, has come with important innovations in the field of intrusion detection.

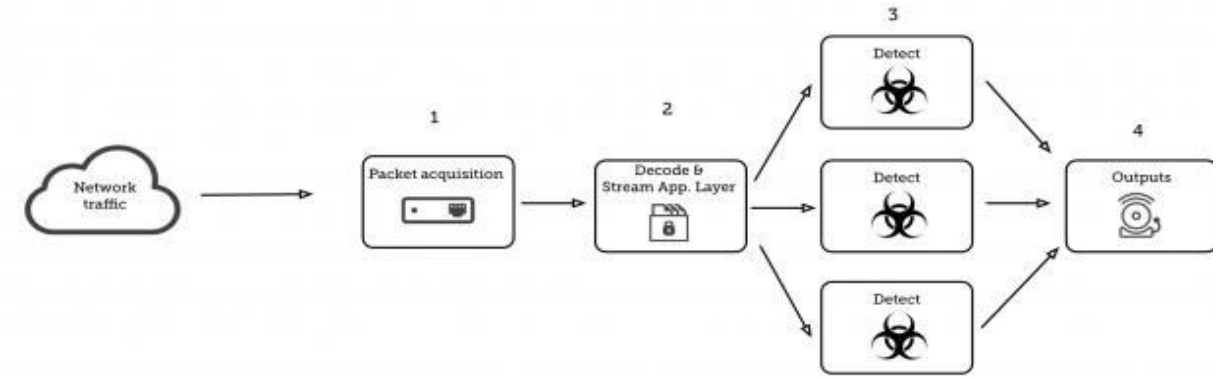

Figure 3 Suricata threading example

The first of these is the new HTTP normalization tool called the HTP library and developed by Ivan Ristic from the Suricata project team. The most important feature of this new tool, which enables the parsing of HTP traffic, is that it is designed as "security-aware". That is, it has the capacity to capture various techniques that attackers can use to circumvent intrusion detection systems. However, the library has different parsers for HTP protocol request line, request header, URI, user agent, response line, server response line, cookie, "basic" and "digest" authentication. Another important feature of Suricata is multi-threaded, to support the work. In other words, in architectures with more than one processor unit, packet processing is distributed in different units with different threads. Each CPU unit acts as a separate single-threaded machine. Thus, load balance is achieved and performance is increased.

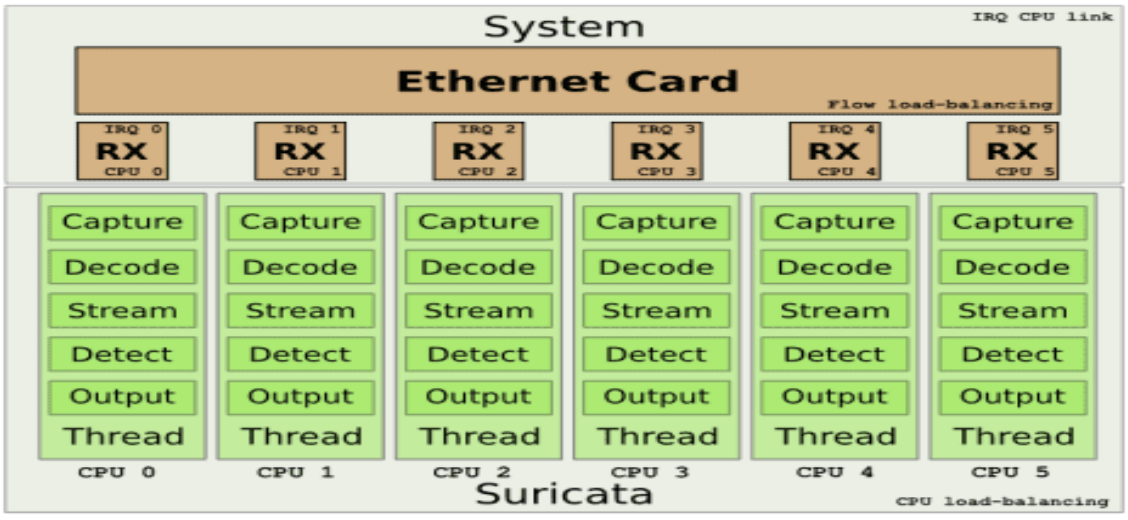

Figure 4 – Multi-thread Operation of Suricata

Working with a single thread, Snort can handle a maximum of 100-200 megabits of traffic, while Suricata can handle 10 gigabits of real traffic.

2. Features

The characteristics of Suricata can be listed as follows:

• It can be used in operating modes such as intrusion detection systems (IDS) and intrusion prevention systems (IPS).

• It monitors the network traffic, records the traffic in pcap format, and then analyzes these saved files offline. It also works in Unix socket mode for the analysis of pcap files.

• It can run on almost all operating systems such as Linux, FreeBSD, OpenBSD, Mac OS X, Windows.

• The configuration file is in YAML format for easy understanding. It is supported by many programming languages. With the stable version of Suricata 2.0, the YAML file is divided into the desired parts and called from the main file.

• IPv6 protocol is fully supported.

• Teredo, GRE, IP4-IP6 tunnel protocols can be resolved.

• It performs operations such as following the session from beginning to end, queuing the flow for TCP sessions. It also has a separate module for reassembling the fragmented packages.

• Ethernet, PPP; VLAN, QINQ, etc. It supports many second layer protocols such as In addition, HTTP, SSL, TLS, SMB, SMB2, DCERPC, SMTP, FTP, SSH, DNS can be resolved from application layer protocols.

• PCRE (Perl Compatible Regular Expressions) can be used in the written rules, file type, size, MD5 summary value can be matched.

• Rule update operations such as adding or deleting new rules can be performed during operation. The application does not need to be restarted.

• Supports CUDA (Compute Unified Device Architecture) technology developed by NVIDIA and used by GPUs. Therefore, high performance will be achieved in working with such hardware and multiple threads.

• It can record HTTP requests, TLS handshakes, SSH connections. With Suricata 2.0, DNS requests/responses started to be recorded and all records were saved in JSON format that can be easily understood by many programming languages.

• Alarms generated according to the rules can be recorded in text format or sent to Syslog. It uses Unified2 binary format, which enables faster recording of alarms. Files in this format can be converted into text using the Barnyard2 open-source tool or saved in the desired database.

After Suricata 2.0, XFF (X-Forwarded-For) support came for Unified2 records as well as HTTP requests.

• Information about all files passing through HTTP traffic can be saved in JSON format together with MD5 summary values.

• In case of use in IPS mode, information about dropped packets and statistics about the operation of the application can also be recorded.

• It has IP reputation support. In rule writing, matching with the desired data can be made by using the keyword "iprep". IP reputation support can be updated at runtime, no reboot required.

• Applications such as AF_PACKET, PF_RING can be used to increase packet processing performance. It can also work with high performance on specialized hardware such as Endace, Napatech, Tilera. On the Tilera platform, which consists of 8 nodes, 80 Gbps traffic can be processed with Suricata.

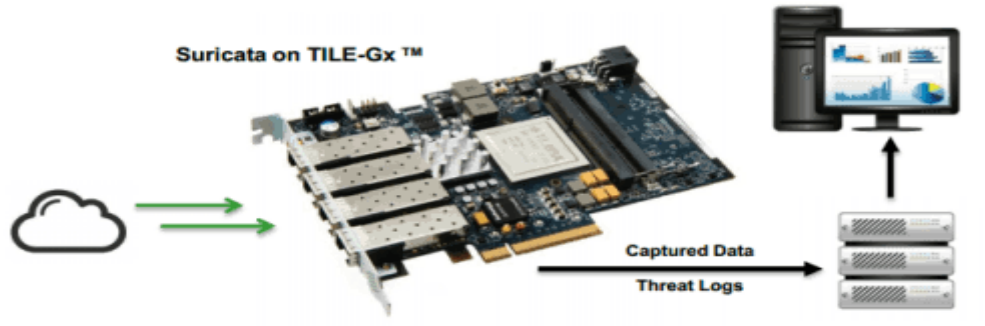

Figure 5 – Customized TILE-GxHardware for Suricata

• Suricata works in accordance with the "Sourcefire Vulnerability Research Team™ (VRT) Rules" and "Emerging Threats Rules" rulesets. In addition, the capabilities of signatures can be developed with the rules to be written in the Lua scripting language. The Lua scripting language supported sample signature is:

```
alert tcp any any -> any any (msg:"Lua rule"; luajit:test.lua; sid:1;)
```

```
function init (args)
    local needs = {}
    needs["http.request_line"] = tostring(true)
    return needs
end
-- match if packet and payload both contain HTTP
function match(args)
    a = tostring(args["http.request_line"])
    if #a > 0 then
        if a:find("^POST%s+/.+%.php%s+HTTP/1.0$") then
            return 1
        end
    end
    return 0
end
```

3. Structure

Suricata, which has many operating modes, is determined by the parameters given at the beginning in which mode it will work. Queues created for the processing of packets are made suitable for operation by organizing the packet handler threads according to the mode after the working mode is determined. In the most preferred "pcap device", that is, intrusion detection system mode, a packet goes through packet capture, packet decoding, flow processing and detection modules, respectively. According to the results of these operations, the packet is passed or an alarm is generated. Dropping and rejecting packets are also available for IPS mode.

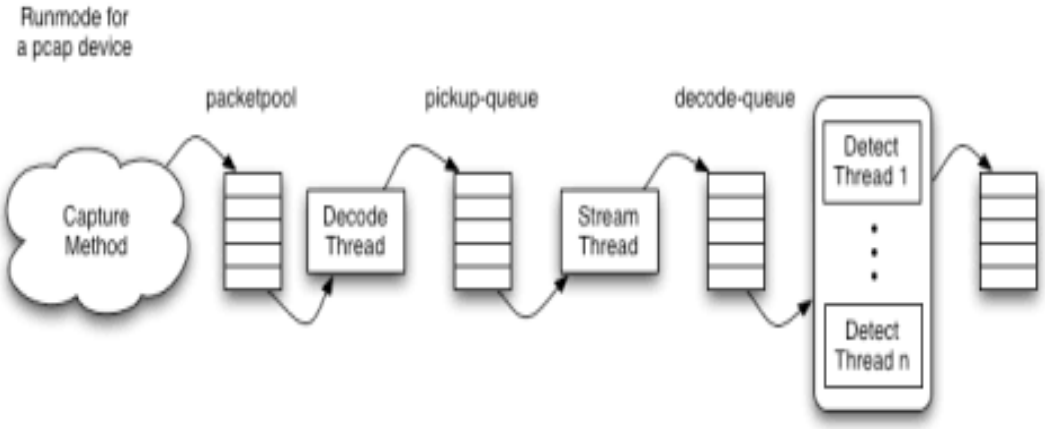

Figure 7 – Suricata Working Structure

• Packet Decoding Module: Packet decoding is responsible for buffering the packets and converting their contents to the data structure supported by Suricata. Packets are classified here according to data links (ethernet, ppp etc.) and processed in the appropriate resolvers.

• Stream Operations Module (Stream Module): It basically has 3 tasks

1. It follows the streams to have an accurate, understandable network connection.

2. It performs the process of queuing the packets so that the main stream can be rebuilt for TCP connections.

3. Performs application layer control. HTTP and DCERPC are analyzed.

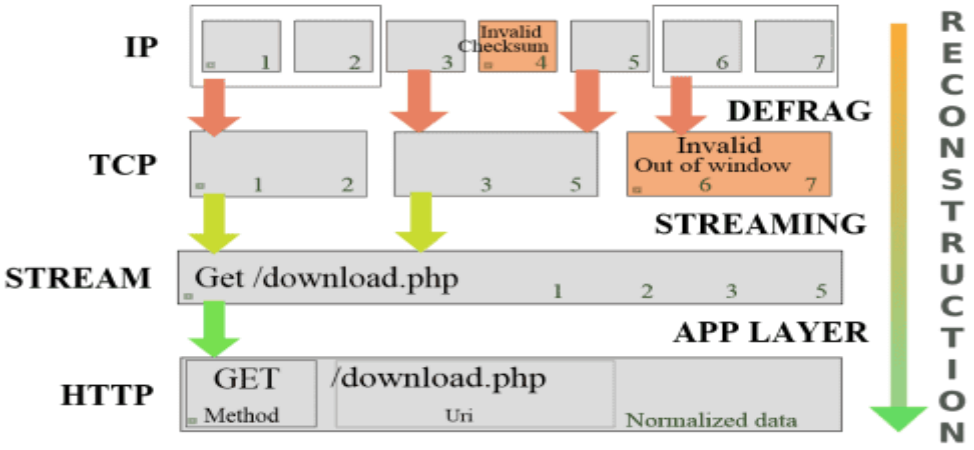

Figure 8 – Stream Module Working Mechanism

• Detect Module: It is responsible for important tasks such as loading all the rules specified in the configuration, launching the detection plugins and matching the packages with the rules by grouping them. It groups the rules within itself. For example, the TCP packet does not need to be compared with the rules written for the UDP protocol. That's why rules written for TCP can be thought of as a group.

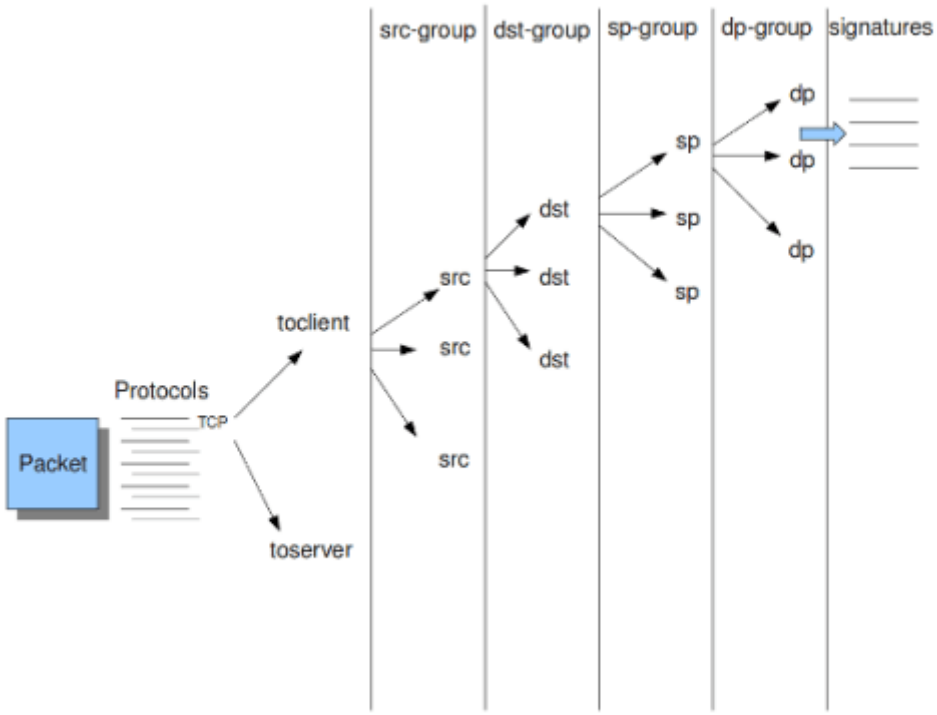

Figure 9 – Grouping of Packets with the Detection Module

The number of groups to be created can be determined by the user. Determining the number of groups is a memory/performance problem. A small number of groups causes poor performance and less memory usage, while an increase in the number of groups causes an increase in performance and memory usage. There are 3 profiles defined as “high, medium and low” in Suricata, the default profile is “medium” which creates a balance between memory usage and performance.

4. Installing Suricata

First we need to load a few necessary links.

```
$ sudo apt-get install wget build-essential libpcre3-dev libpcre3-dbg automake autoconf libtool libpcap-dev
libnet1-dev libyaml-dev zlib1g-dev libcap-ng-dev libjansson-dev
```

```
$ sudo yum install wget libpcap-devel libnet-devel pcre-devel gcc-c++ automake autoconf libtool make libyaml-devel
zlib-devel file-devel jansson-devel nss-devel
```

Then download and build the Suricata source code. Enter all codes in order.

```
$ wget http://www.openinfosecfoundation.org/download/suricata-2.0.8.tar.gz
$ tar -xvf suricata-2.0.8.tar.gz
$ cd suricata-2.0.8
$ ./configure --sysconfdir=/etc --localstatedir=/var
```

```
$ make
$ sudo make install
```

```
$ sudo make install-conf
```

```
$ sudo make install-rules
```

To configure;

```
$ sudo vi /etc/suricata/suricata.yaml
```

```
$ sudo /usr/local/bin/suricata --list-runmodes
```

```
$ sudo /usr/local/bin/suricata -c
/etc/suricata/suricata.yaml -i eth0 --init-errors-fatal
```