



**INSTITUTE FOR ADVANCED COMPUTING
AND
SOFTWARE DEVELOPMENT
AKURDI, PUNE**

**DOCUMENTATION ON
IDPS using Snort/Suricata and alert notification via mail**

PG-DITISS March-2023

SUBMITTED BY:

GROUP NO: 26

SANSKAR GUPTA (233452)

KULDEEP PATEL (233416)

MR. KARTIK AWARI
PROJECT GUIDE

MR. ROHIT PURANIK
CENTRE COORDINATOR

INDEX

CHAPTER NO.	TITLE	PAGE NO.
1	Introduction	3
2	Problem Statement	4
3	Proposed Solution	5
4	Technology used	6
5	Information about IDPS tools	7
6	Advantages of IDPS tools with example	8
7	Architecture of Snort and Suricata	9,10
8	Alert system process	11
9	Implementation Screenshots	12-22
10	Future Enhancement	23
11	Conclusion	24
12	References	25

1. Introduction:

Intrusion detection and prevention systems (IDS/IPS) are security tools that monitor network traffic for malicious activity. IDS systems detect intrusions, while IPS systems can also prevent them.

Snort and Suricata are two popular open-source IDS/IPS systems. They are both capable of detecting a wide range of threats, including network scanning, port scanning, and denial-of-service attacks.

Alert notification is an important feature of IDS/IPS systems. It allows you to be notified of potential threats so that you can take action to mitigate them. Email is a common method for alert notification.

In this project, we will use Snort or Suricata to detect intrusions and send alerts via email. We will also configure the system to detect specific threats, such as network scanning or port scanning.

In this project we will demonstrate how to use Snort or Suricata to protect a network from malicious activity. It will also show how to configure alert notification to ensure that you are notified of potential threats in a timely manner.

2. Problem Statement:

Network security is a critical issue for businesses of all sizes. Intrusions and other malicious activities can cause significant damage, including data loss, financial losses, and reputational harm.

IDS/IPS systems are an important tool for protecting networks from malicious activity. However, they can be complex to configure and maintain. Additionally, they can generate a large number of alerts, which can make it difficult to identify and respond to real threats.

3. Proposed Solution:

This project will use Snort or Suricata to detect intrusions and send alerts via email. The system will be configured to detect specific threats, such as network scanning or port scanning. This will help to reduce the number of false alerts and make it easier to identify and respond to real threats.

The project will also demonstrate how to use Snort or Suricata to protect a network from malicious activity. It will also show how to configure alert notification to ensure that you are notified of potential threats in a timely manner.

Benefits:

The benefits of this project include:

- Improved network security
- Reduced number of false alerts
- Easier identification and response to real threats
- Increased visibility into network traffic
- Improved compliance with security standards

4. Technology Used:

4.1 Hardware Requirement :

- RAM: 16 GB
- HDD: 512GB

4.2 Software Requirement :

- Operating System: Windows 10
- Tool: VMWare Workstation Pro

5. Information about IDPS tools:

Snort is an open-source, free and lightweight network intrusion detection system (NIDS) software for Linux and Windows to detect emerging threats. It is one of the most popular IDS tools available. Snort can be used to detect a wide range of threats, including network scanning, port scanning, and denial-of-service attacks. It can also be used to detect more advanced threats, such as malware and zero-day attacks.

Suricata is another popular open-source IDS/IPS system. It is similar to Snort in many ways, but it offers some additional features, such as multi-threading and support for more protocols. Suricata is also more flexible than Snort, making it a good choice for advanced users.

Both Snort and Suricata are capable of detecting a wide range of threats. The choice of which tool to use depends on your specific needs and requirements. If you are looking for a lightweight and easy-to-use IDS tool, Snort is a good choice. If you need a more powerful and flexible IDS tool, Suricata is a good option.

Here are some of the key features of Snort and Suricata:

Snort:

Lightweight and easy to use

Wide range of detection capabilities

Active community of users and developers

Large library of rules and signatures

Suricata:

More powerful and flexible than Snort

Multi-threaded architecture for improved performance

Support for more protocols

Active community of users and developers

Large library of rules and signatures

6. Advantages of IDPS tools with example:

Some of the advantages of IDPS tools:

Improved network security: IDPS tools can help to protect networks from a variety of threats, including malware, denial-of-service attacks, and network intrusions.

Reduced number of false alerts: IDPS tools can be configured to filter out false alerts, which can help to reduce the workload on security teams.

Easier identification and response to real threats: IDPS tools can help to identify real threats more quickly, which can help to reduce the damage caused by these threats.

Increased visibility into network traffic: IDPS tools can provide visibility into network traffic, which can help to identify potential threats and vulnerabilities.

Improved compliance with security standards: IDPS tools can help organizations to comply with security standards, such as PCI DSS and HIPAA.

Here are some specific examples of how IDPS tools have been used to improve network security:

- In 2017, the Mirai botnet was used to launch a massive denial-of-service attack against Dyn, a major DNS provider. The attack disrupted access to a number of popular websites, including Twitter, Netflix, and PayPal. IDPS tools were used to identify and block the attack, helping to mitigate the damage.
- In 2018, the WannaCry ransomware attack infected over 200,000 computers worldwide. The attack encrypted files on infected computers and demanded a ransom payment to decrypt them. IDPS tools were used to identify and block the attack, helping to protect many organizations from being infected.
- In 2019, the NotPetya ransomware attack infected over 100,000 computers worldwide. The attack was particularly destructive, causing billions of dollars in damage. IDPS tools were used to identify and block the attack, helping to protect many organizations from being infected.

These are just a few examples of how IDPS tools can be used to improve network security. IDPS tools are an important part of a layered security approach, and they can help to protect organizations from a variety of threats.

7. Architecture of Snort and Suricata:

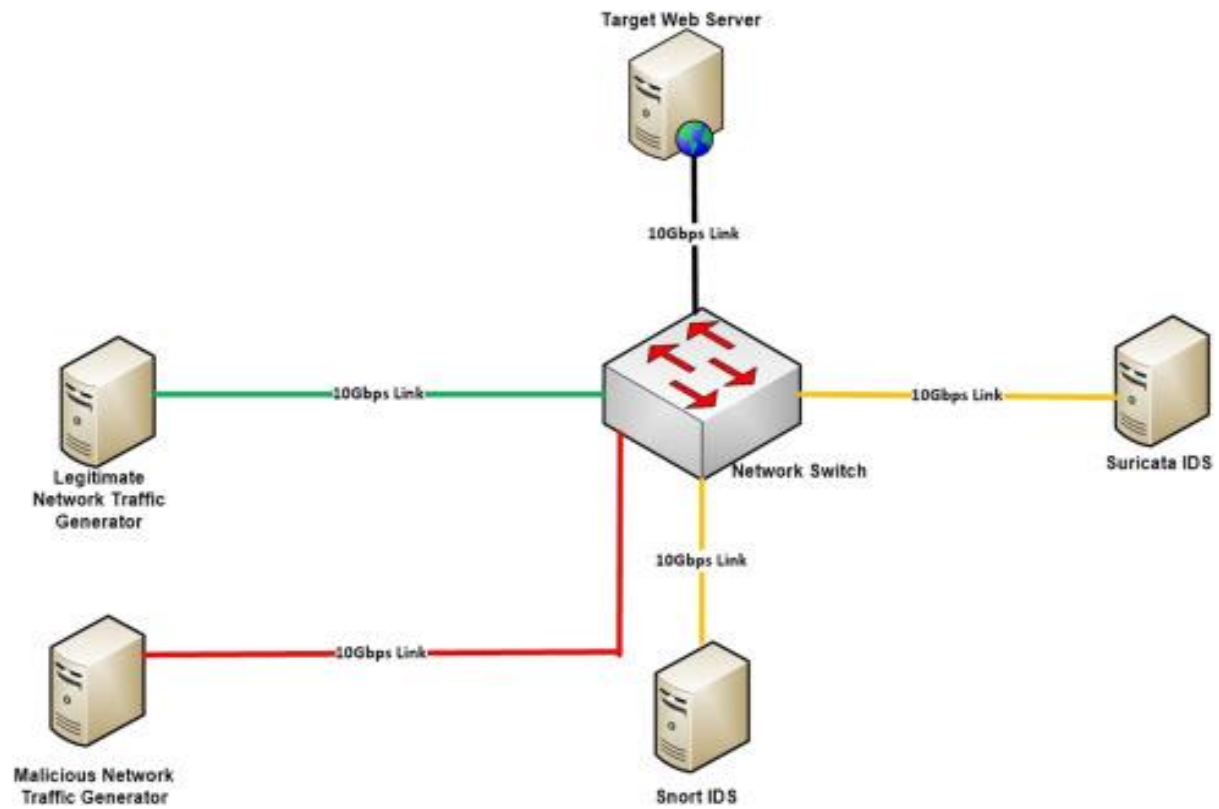


Diagram 1

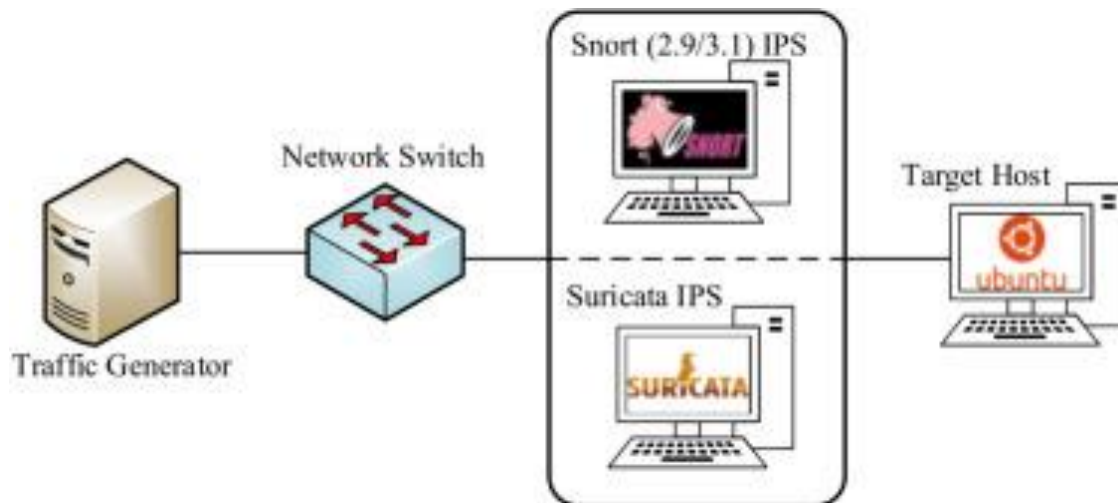


Diagram 2

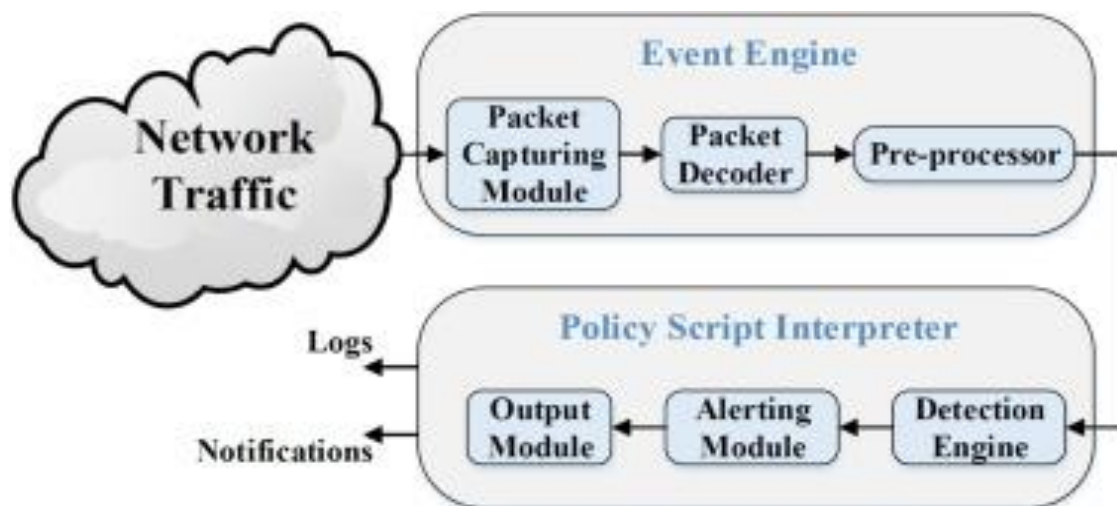


Diagram 3

8. Alert system process:

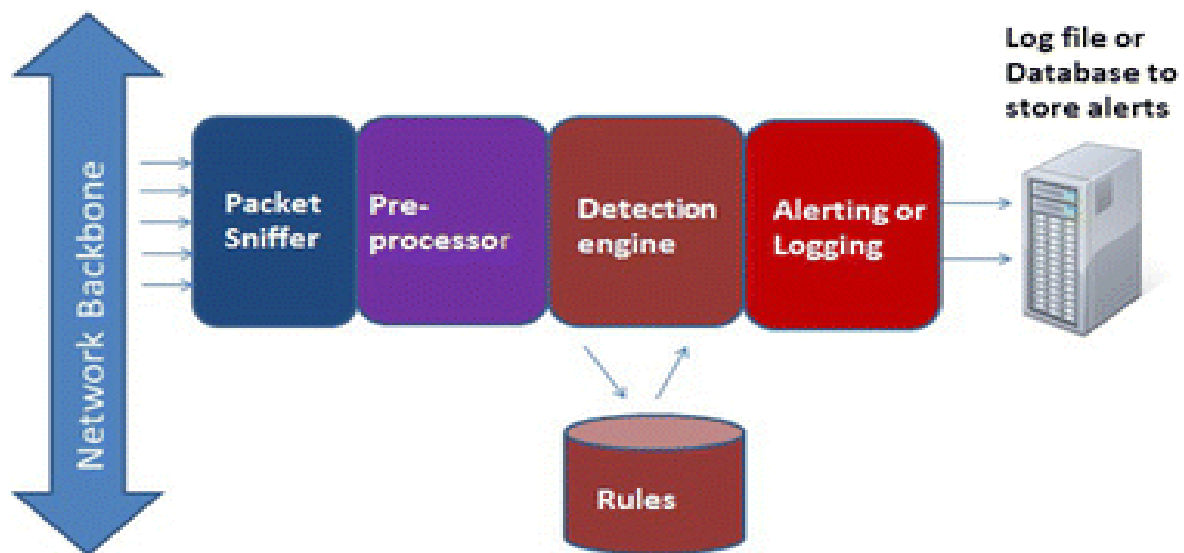


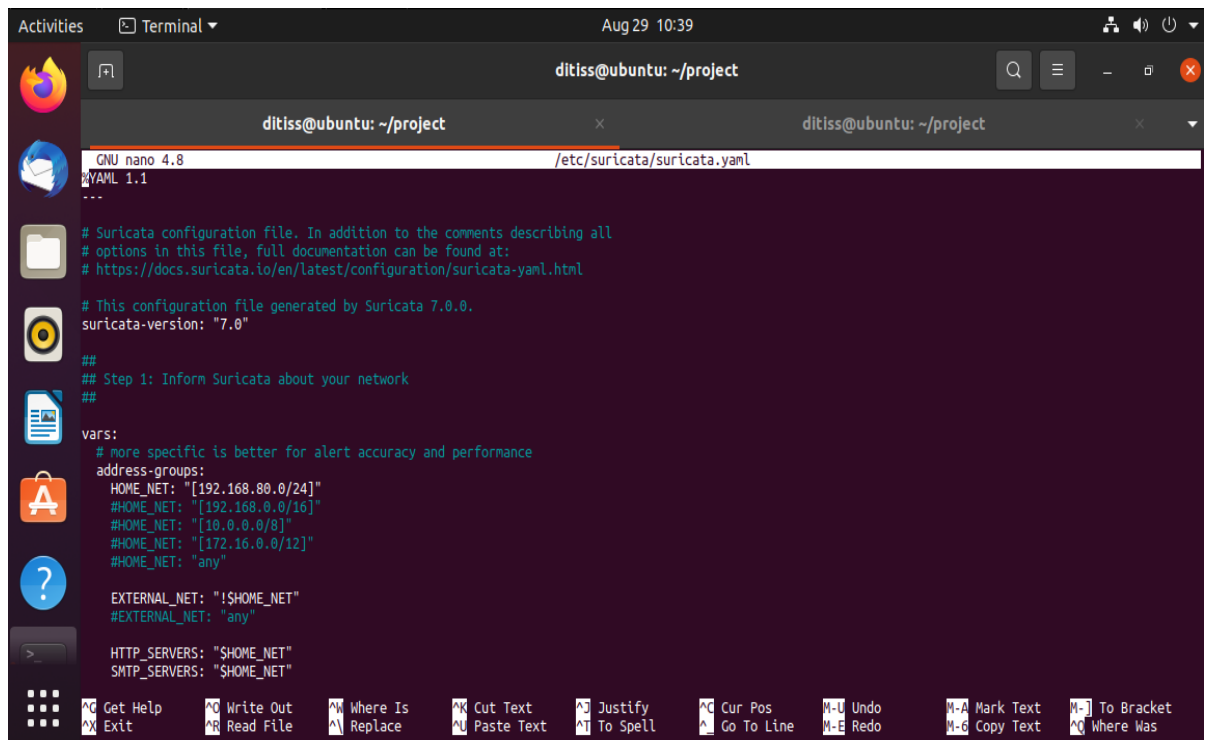
Diagram 4



Diagram 5

9. Implementation Screenshots:

- a. In Ubuntu system, first we have to install suricata and for installation we have to refer official documentation of suricata.
- b. After installation we have to enable suricata service and also we have to check the status of suricata.
- c. Also, there are some free sources available for suricata. We can enable those sources and install some extra and efficient libraries for suricata.
- d. Now, to configure suricata we have to set network ip address and interface of system where suricata is running.



The screenshot shows a terminal window titled "ditiss@ubuntu: ~/project" with a timestamp of "Aug 29 10:39". The window contains a nano text editor editing the file "/etc/suricata/suricata.yaml". The editor's status bar at the top indicates "GNU nano 4.8" and "YAML 1.1". The file content is a Suricata configuration file with the following visible text:

```
# Suricata configuration file. In addition to the comments describing all
# options in this file, full documentation can be found at:
# https://docs.suricata.io/en/latest/configuration/suricata.yaml.html

# This configuration file generated by Suricata 7.0.0.
suricata-version: "7.0"

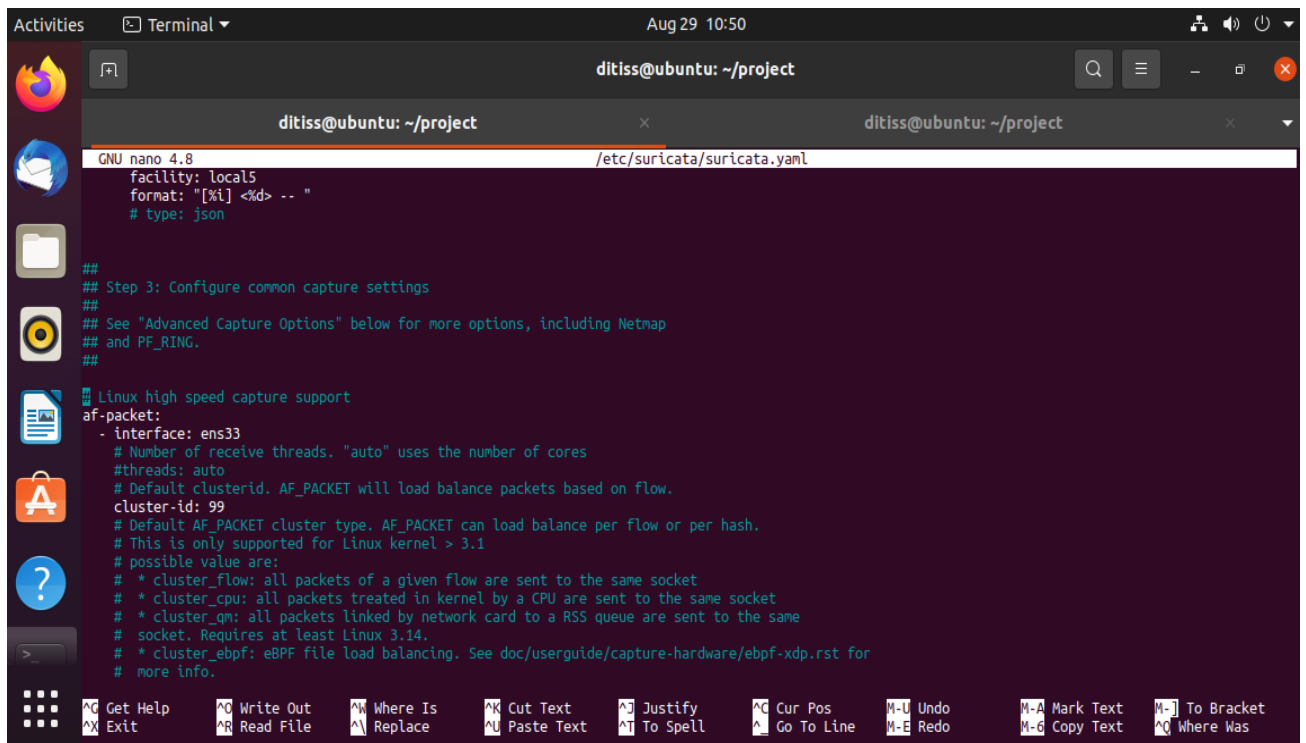
##
## Step 1: Inform Suricata about your network
##

vars:
  # more specific is better for alert accuracy and performance
  address-groups:
    HOME_NET: "[192.168.0.0/24]"
    #HOME_NET: "[192.168.0.0/16]"
    #HOME_NET: "[10.0.0.0/8]"
    #HOME_NET: "[172.16.0.0/12]"
    #HOME_NET: "any"

    EXTERNAL_NET: "!$HOME_NET"
    #EXTERNAL_NET: "any"

    HTTP_SERVERS: "$HOME_NET"
    SMTP_SERVERS: "$HOME_NET"
```

The bottom of the terminal window displays a row of keyboard shortcuts for the nano editor, including "Get Help", "Exit", "Write Out", "Read File", "Where Is", "Replace", "Cut Text", "Paste Text", "Justify", "To Spell", "Cur Pos", "Go To Line", "Undo", "Redo", "Mark Text", "Copy Text", "To Bracket", and "Where Was".



The screenshot shows a terminal window titled "Activities" with a date and time of "Aug 29 10:50". The user is logged in as "ditiss@ubuntu" and is in the directory "~/project". The terminal displays the contents of the file "/etc/suricata/suricata.yaml" using the "GNU nano 4.8" editor. The configuration file is in YAML format and includes settings for the Suricata engine, such as the interface (ens33), number of threads (auto), and cluster settings. The bottom of the terminal shows a row of keyboard shortcuts for nano editor commands.

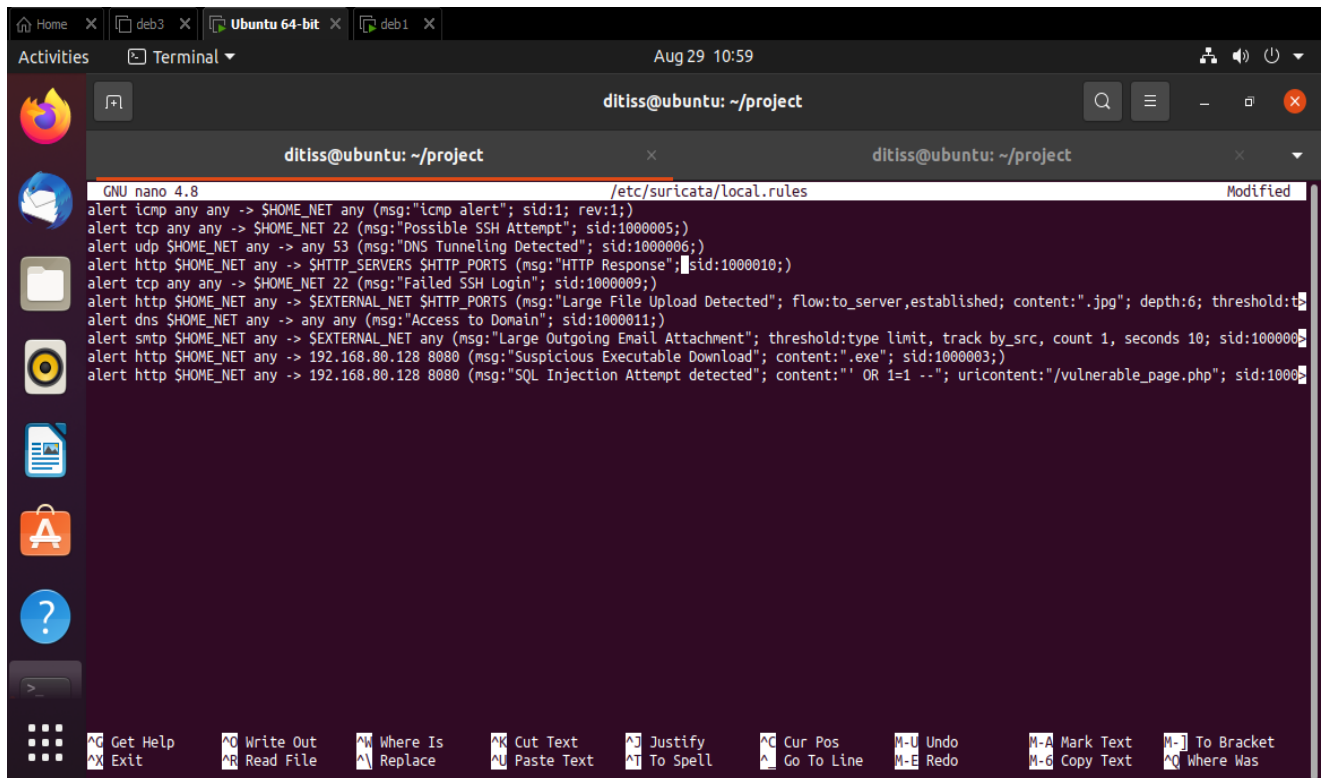
```
GNU nano 4.8 /etc/suricata/suricata.yaml
facility: local5
format: "[%i] <%d> -- "
# type: json

##
## Step 3: Configure common capture settings
##
## See "Advanced Capture Options" below for more options, including Netmap
## and PF_RING.
##

Linux high speed capture support
af-packet:
  - interface: ens33
    # Number of receive threads. "auto" uses the number of cores
    #threads: auto
    # Default clusterid. AF_PACKET will load balance packets based on flow.
    cluster-id: 99
    # Default AF_PACKET cluster type. AF_PACKET can load balance per flow or per hash.
    # This is only supported for Linux kernel > 3.1
    # possible value are:
    # * cluster_flow: all packets of a given flow are sent to the same socket
    # * cluster_cpu: all packets treated in kernel by a CPU are sent to the same socket
    # * cluster_qm: all packets linked by network card to a RSS queue are sent to the same
    # socket. Requires at least Linux 3.14.
    # * cluster_ebpf: eBPF file load balancing. See doc/userguide/capture-hardware/ebpf-xdp.rst for
    # more info.
```

- e. Now, we have to configure suricata rule files by adding some widely referred alert rules.

Note: We have to set rules in /etc/suricata/local.rules path.



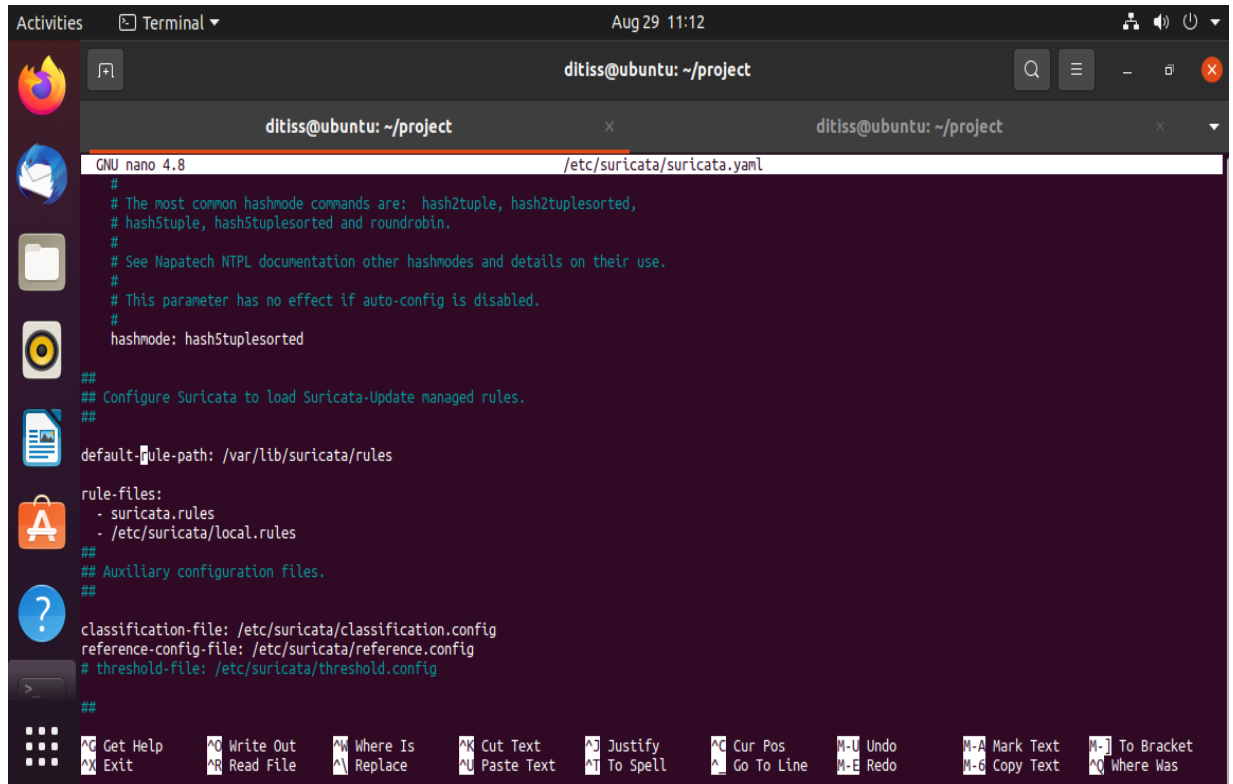
```
GNU nano 4.8 /etc/suricata/local.rules Modified
alert icmp any any -> $HOME_NET any (msg:"icmp alert"; sid:1; rev:1;)
alert tcp any any -> $HOME_NET 22 (msg:"Possible SSH Attempt"; sid:1000005;)
alert udp $HOME_NET any -> any 53 (msg:"DNS Tunneling Detected"; sid:1000006;)
alert http $HOME_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"HTTP Response"; sid:1000010;)
alert tcp any any -> $HOME_NET 22 (msg:"Failed SSH Login"; sid:1000009;)
alert http $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS (msg:"Large File Upload Detected"; flow:to_server,established; content:".jpg"; depth:6; threshold:t
alert dns $HOME_NET any -> any any (msg:"Access to Domain"; sid:1000011;)
alert smtp $HOME_NET any -> $EXTERNAL_NET any (msg:"Large Outgoing Email Attachment"; threshold:type limit, track by_src, count 1, seconds 10; sid:1000008;)
alert http $HOME_NET any -> 192.168.80.128 8080 (msg:"Suspicious Executable Download"; content:".exe"; sid:1000003;)
alert http $HOME_NET any -> 192.168.80.128 8080 (msg:"SQL Injection Attempt detected"; content:"' OR 1=1 --"; uricontent:"/vulnerable_page.php"; sid:1000004;)
```

In this rule set, there are alert commands based on:

- . Icmp packets
- . tcp (ssh login and fail)
- . udp (DNS tunneling)
- . http (http server and ports)
- . smtp (For email)
- . dns (Access to any existing domain)

- f. After confirming all the rules, we have to paste the rule file in the main yaml file of suricata.

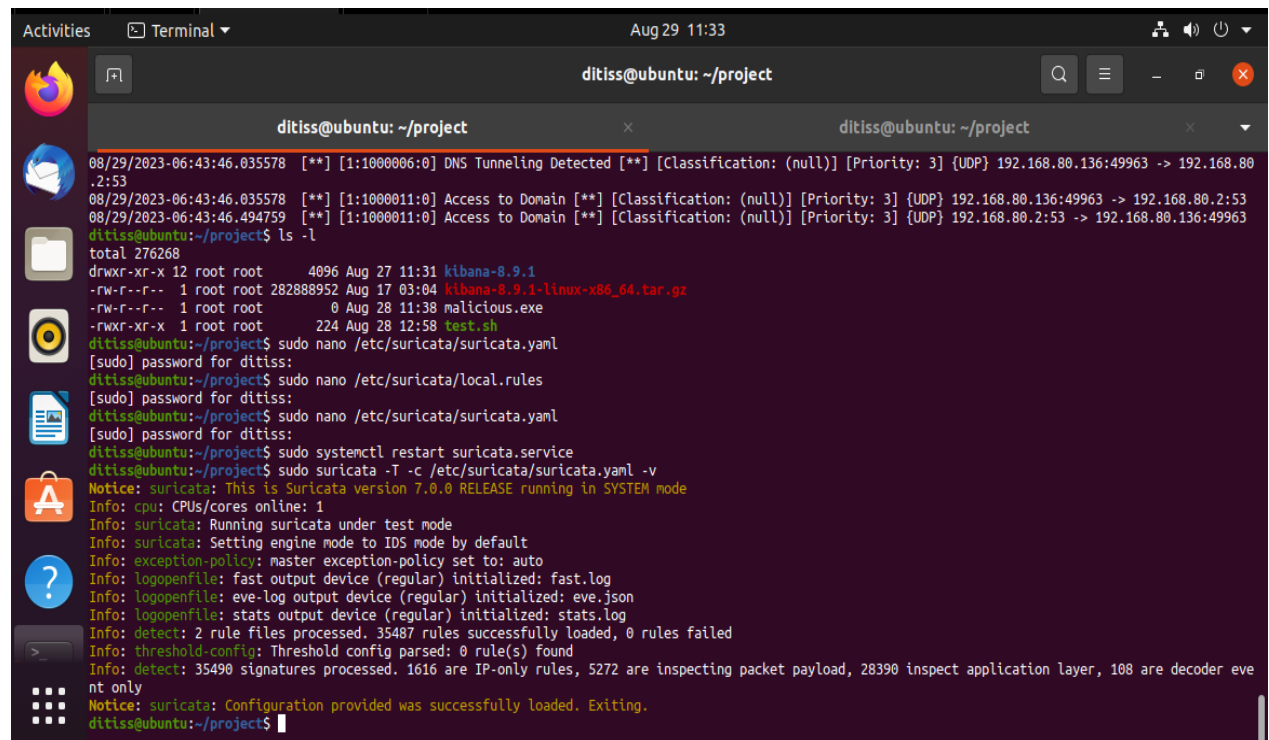
Note: Yaml file is in /etc/suricata/suricata.yaml



```
GNU nano 4.8 /etc/suricata/suricata.yaml
#
# The most common hashmode commands are: hash2tuple, hash2tuplesorted,
# hash5tuple, hash5tuplesorted and roundrobin.
#
# See Napatech NTPPL documentation other hashmodes and details on their use.
#
# This parameter has no effect if auto-config is disabled.
#
hashmode: hash5tuplesorted

##
## Configure Suricata to load Suricata-Update managed rules.
##
default-rule-path: /var/lib/suricata/rules
rule-files:
- suricata.rules
- /etc/suricata/local.rules
##
## Auxiliary configuration files.
##
classification-file: /etc/suricata/classification.config
reference-config-file: /etc/suricata/reference.config
# threshold-file: /etc/suricata/threshold.config
##
```


- g. Now, we have to test the configuration file of suricata to check whether all the rules are mentioned accurately.



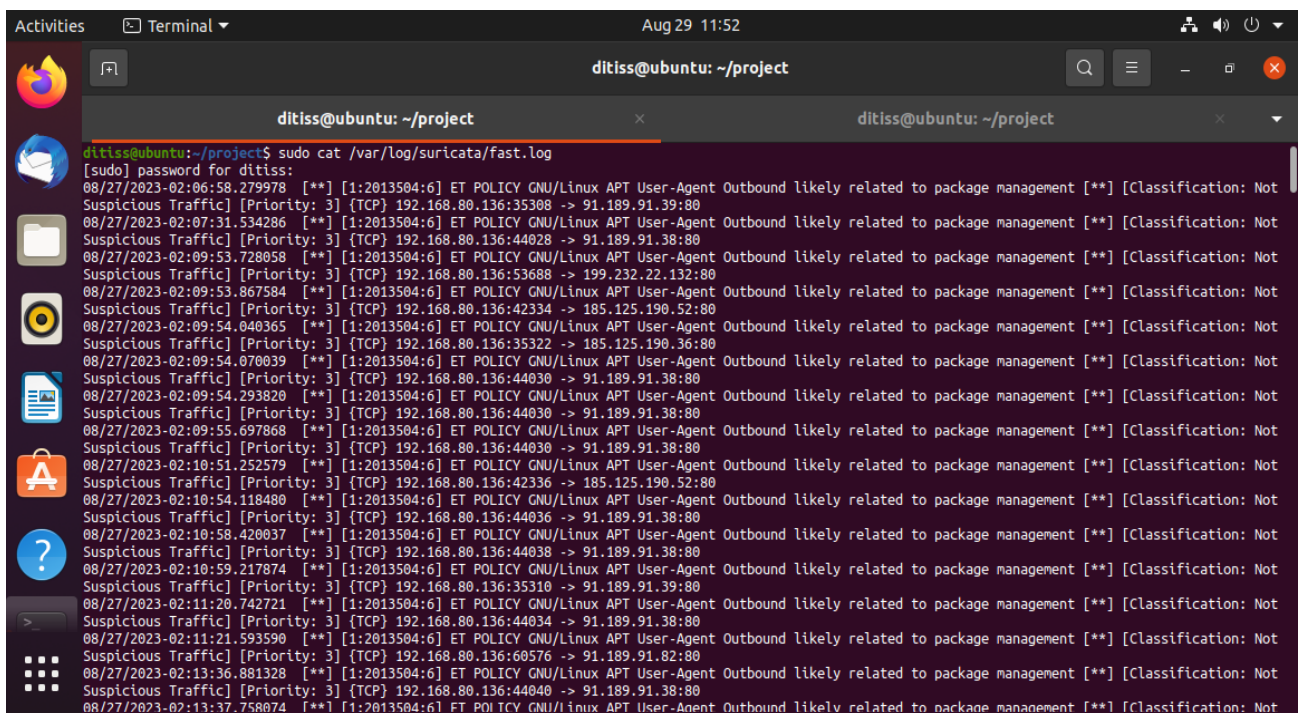
```
Activities  Terminal  Aug 29 11:33
ditiss@ubuntu: ~/project

ditiss@ubuntu: ~/project
ditiss@ubuntu: ~/project

08/29/2023-06:43:46.035578  [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:49963 -> 192.168.80.2:53
08/29/2023-06:43:46.035578  [**] [1:1000011:0] Access to Domain [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:49963 -> 192.168.80.2:53
08/29/2023-06:43:46.494759  [**] [1:1000011:0] Access to Domain [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.2:53 -> 192.168.80.136:49963
ditiss@ubuntu:~/project$ ls -l
total 276268
drwxr-xr-x 12 root root    4096 Aug 27 11:31 kibana-8.9.1
-rw-r--r--  1 root root 282888952 Aug 17 03:04 kibana-8.9.1-linux-x86_64.tar.gz
-rw-r--r--  1 root root      0 Aug 28 11:38 malicious.exe
-rwxr-xr-x  1 root root    224 Aug 28 12:58 test.sh
ditiss@ubuntu:~/project$ sudo nano /etc/suricata/suricata.yaml
[sudo] password for ditiss:
ditiss@ubuntu:~/project$ sudo nano /etc/suricata/local.rules
[sudo] password for ditiss:
ditiss@ubuntu:~/project$ sudo nano /etc/suricata/suricata.yaml
[sudo] password for ditiss:
ditiss@ubuntu:~/project$ sudo systemctl restart suricata.service
ditiss@ubuntu:~/project$ sudo suricata -T -c /etc/suricata/suricata.yaml -v
Notice: suricata: This is Suricata version 7.0.0 RELEASE running in SYSTEM mode
Info: cpu: CPUs/cores online: 1
Info: suricata: Running suricata under test mode
Info: suricata: Setting engine mode to IDS mode by default
Info: exception-policy: master exception-policy set to: auto
Info: logopenfile: fast output device (regular) initialized: fast.log
Info: logopenfile: eve-log output device (regular) initialized: eve.json
Info: logopenfile: stats output device (regular) initialized: stats.log
Info: detect: 2 rule files processed. 35487 rules successfully loaded, 0 rules failed
Info: threshold-config: Threshold config parsed: 0 rule(s) found
Info: detect: 35490 signatures processed. 1616 are IP-only rules, 5272 are inspecting packet payload, 28390 inspect application layer, 108 are decoder eve
nt only
Notice: suricata: Configuration provided was successfully loaded. Exiting.
ditiss@ubuntu:~/project$
```

h. Now, to check the output or final alert logs we have to run log command.

Note: Logs are stored in /var/log/suricata/fast.log.



```
ditiss@ubuntu: ~/project
[ditiss@ubuntu: ~/project]$ sudo cat /var/log/suricata/fast.log
[sudo] password for ditiss:
08/27/2023-02:06:58.279978  *** [1:2013504:6] ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management *** [Classification: Not Suspicious Traffic] [Priority: 3] {TCP} 192.168.80.136:35308 -> 91.189.91.39:80
08/27/2023-02:07:31.534286  *** [1:2013504:6] ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management *** [Classification: Not Suspicious Traffic] [Priority: 3] {TCP} 192.168.80.136:44028 -> 91.189.91.38:80
08/27/2023-02:09:53.728058  *** [1:2013504:6] ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management *** [Classification: Not Suspicious Traffic] [Priority: 3] {TCP} 192.168.80.136:53688 -> 199.232.22.132:80
08/27/2023-02:09:53.867584  *** [1:2013504:6] ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management *** [Classification: Not Suspicious Traffic] [Priority: 3] {TCP} 192.168.80.136:42334 -> 185.125.190.52:80
08/27/2023-02:09:54.040365  *** [1:2013504:6] ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management *** [Classification: Not Suspicious Traffic] [Priority: 3] {TCP} 192.168.80.136:35322 -> 185.125.190.36:80
08/27/2023-02:09:54.070039  *** [1:2013504:6] ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management *** [Classification: Not Suspicious Traffic] [Priority: 3] {TCP} 192.168.80.136:44030 -> 91.189.91.38:80
08/27/2023-02:09:54.293820  *** [1:2013504:6] ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management *** [Classification: Not Suspicious Traffic] [Priority: 3] {TCP} 192.168.80.136:44030 -> 91.189.91.38:80
08/27/2023-02:09:55.697868  *** [1:2013504:6] ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management *** [Classification: Not Suspicious Traffic] [Priority: 3] {TCP} 192.168.80.136:44030 -> 91.189.91.38:80
08/27/2023-02:10:51.252579  *** [1:2013504:6] ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management *** [Classification: Not Suspicious Traffic] [Priority: 3] {TCP} 192.168.80.136:42336 -> 185.125.190.52:80
08/27/2023-02:10:54.118480  *** [1:2013504:6] ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management *** [Classification: Not Suspicious Traffic] [Priority: 3] {TCP} 192.168.80.136:44036 -> 91.189.91.38:80
08/27/2023-02:10:58.420037  *** [1:2013504:6] ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management *** [Classification: Not Suspicious Traffic] [Priority: 3] {TCP} 192.168.80.136:44038 -> 91.189.91.38:80
08/27/2023-02:10:59.217874  *** [1:2013504:6] ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management *** [Classification: Not Suspicious Traffic] [Priority: 3] {TCP} 192.168.80.136:35310 -> 91.189.91.39:80
08/27/2023-02:11:20.742721  *** [1:2013504:6] ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management *** [Classification: Not Suspicious Traffic] [Priority: 3] {TCP} 192.168.80.136:44034 -> 91.189.91.38:80
08/27/2023-02:11:21.593590  *** [1:2013504:6] ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management *** [Classification: Not Suspicious Traffic] [Priority: 3] {TCP} 192.168.80.136:60576 -> 91.189.91.82:80
08/27/2023-02:13:36.881328  *** [1:2013504:6] ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management *** [Classification: Not Suspicious Traffic] [Priority: 3] {TCP} 192.168.80.136:44040 -> 91.189.91.38:80
08/27/2023-02:13:37.758074  *** [1:2013504:6] ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management *** [Classification: Not Suspicious Traffic] [Priority: 3] {TCP} 192.168.80.136:44040 -> 91.189.91.38:80
```

Activities Terminal Aug 29 11:54

ditiss@ubuntu: ~/project

ditiss@ubuntu: ~/project

```
.2:53
08/28/2023-07:20:50.040023  [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:37791 -> 192.168.80.128:53
08/28/2023-07:23:24.868425  [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:52883 -> 192.168.80.128:53
08/28/2023-07:25:40.085752  [**] [1:2:2] Possible SSH Brute Force Attempt [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.80.1:50479 -> 192.168.80.128:22
08/28/2023-07:25:49.995947  [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:43314 -> 192.168.80.128:53
08/28/2023-07:28:24.840024  [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:35110 -> 192.168.80.128:53
08/28/2023-07:29:11.396679  [**] [1:2:2] Possible SSH Brute Force Attempt [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.80.1:50502 -> 192.168.80.128:22
08/28/2023-07:29:24.683534  [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.128:56732 -> 192.168.80.128:53
08/28/2023-07:29:25.196541  [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.128:56521 -> 192.168.80.128:53
08/28/2023-07:29:25.469864  [**] [1:2013504:6] ET POLICY GNU/Linux APT User-Agent Outbound likely related to package management [**] [Classification: Not Suspicious Traffic] [Priority: 3] {TCP} 192.168.80.128:51734 -> 199.232.22.132:80
08/28/2023-07:30:49.981821  [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:39395 -> 192.168.80.128:53
08/28/2023-07:32:11.491503  [**] [1:2034636:2] ET INFO Python SimpleHTTP ServerBanner [**] [Classification: Misc activity] [Priority: 3] {TCP} 192.168.80.128:8000 -> 192.168.80.136:41682
08/28/2023-07:33:24.868247  [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:41265 -> 192.168.80.128:53
08/28/2023-07:35:50.039790  [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:51013 -> 192.168.80.128:53
08/28/2023-07:36:13.364664  [**] [1:2034636:2] ET INFO Python SimpleHTTP ServerBanner [**] [Classification: Misc activity] [Priority: 3] {TCP} 192.168.80.128:8000 -> 192.168.80.136:41684
08/28/2023-07:38:24.868359  [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:40949 -> 192.168.80.128:53
08/28/2023-07:40:50.039812  [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:38645 -> 192.168.80.128:53
```

Activities Terminal ▾ Aug 29 11:59

ditliss@ubuntu: ~/project

ditliss@ubuntu: ~/project

```
08/28/2023-10:00:31.207555 [**] [1:2034636:2] ET POLICY AND/OR LINUX API USER AGENT OUTGOING likely related to package management [**] [Classification: Misc activity] [Priority: 3] {TCP} 192.168.80.136:47278 -> 91.189.91.38:80
Suspicious Traffic] [Priority: 3] {TCP} 192.168.80.136:47278 -> 91.189.91.38:80
08/28/2023-10:00:50.178790 [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:51678 -> 192.168.80.2:53
08/28/2023-10:03:06.156736 [**] [1:1000005:0] Possible SSH Attempt [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.80.1:51702 -> 192.168.80.136:22
08/28/2023-10:03:24.964323 [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:51020 -> 192.168.80.2:53
08/28/2023-10:03:44.774170 [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:40001 -> 192.168.80.2:53
08/28/2023-10:03:44.789547 [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:59205 -> 192.168.80.2:53
08/28/2023-10:05:48.084111 [**] [1:1000005:0] Possible SSH Attempt [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.80.1:50502 -> 192.168.80.128:22
08/28/2023-10:05:50.182760 [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:35789 -> 192.168.80.2:53
08/28/2023-10:06:30.982793 [**] [1:2034636:2] ET INFO Python SimpleHTTP ServerBanner [**] [Classification: Misc activity] [Priority: 3] {TCP} 192.168.80.128:8000 -> 192.168.80.136:41696
08/28/2023-10:07:06.829873 [**] [1:2034636:2] ET INFO Python SimpleHTTP ServerBanner [**] [Classification: Misc activity] [Priority: 3] {TCP} 192.168.80.128:8000 -> 192.168.80.136:41698
08/28/2023-10:08:24.971880 [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:40230 -> 192.168.80.2:53
08/28/2023-10:09:38.274693 [**] [1:1000005:0] Possible SSH Attempt [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.80.1:50502 -> 192.168.80.128:22
08/28/2023-10:10:50.184910 [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:36760 -> 192.168.80.2:53
08/28/2023-10:12:39.976560 [**] [1:1000005:0] Possible SSH Attempt [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.80.1:50502 -> 192.168.80.128:22
08/28/2023-10:13:24.968999 [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:52182 -> 192.168.80.2:53
08/28/2023-10:15:39.678940 [**] [1:1000005:0] Possible SSH Attempt [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.80.1:50502 -> 192.168.80.128:22
08/28/2023-10:15:50.183788 [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:40766 -> 192.168.80.2:53
```

Activities Terminal Aug 29 12:07

ditiss@ubuntu: ~/project

ditiss@ubuntu: ~/project

```
08/28/2023-10:54:42.328199 :22 [**] [1:1000005:0] Possible SSH Attempt [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.80.1:51917 -> 192.168.80.136
08/28/2023-10:55:40.935900 :22 [**] [1:1000005:0] Possible SSH Attempt [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.80.1:51702 -> 192.168.80.136
08/28/2023-10:55:45.171022 :22 [**] [1:1000005:0] Possible SSH Attempt [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.80.1:50502 -> 192.168.80.128
08/28/2023-10:55:45.178209 :22 [**] [1:2034636:2] ET INFO Python SimpleHTTP ServerBanner [**] [Classification: Misc activity] [Priority: 3] {TCP} 192.168.80.128:8080 -> 192.168.80.136:37904
08/28/2023-10:55:50.184169 :2:53 [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:43185 -> 192.168.80.128
08/28/2023-11:05:20.584674 :22 [**] [1:1000005:0] Possible SSH Attempt [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.80.1:51702 -> 192.168.80.136
08/28/2023-11:05:20.584674 :2:53 [**] [1:1000009:0] Failed SSH Login [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.80.1:51702 -> 192.168.80.136:22
08/28/2023-11:05:50.271020 :2:53 [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:44685 -> 192.168.80.128
08/28/2023-11:08:25.006085 :2:53 [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:39623 -> 192.168.80.128
08/28/2023-11:10:50.179618 :2:53 [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:51339 -> 192.168.80.128
08/28/2023-11:13:24.978014 :2:53 [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:57205 -> 192.168.80.128
08/28/2023-11:15:50.178335 :2:53 [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:45367 -> 192.168.80.128
08/28/2023-11:18:24.979962 :2:53 [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:51150 -> 192.168.80.128
08/28/2023-11:19:01.279280 :22 [**] [1:1000005:0] Possible SSH Attempt [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.80.1:50502 -> 192.168.80.128
08/28/2023-11:19:01.279280 :2:53 [**] [1:1000009:0] Failed SSH Login [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.80.1:50502 -> 192.168.80.128:22
08/28/2023-11:20:50.177465 :2:53 [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:46696 -> 192.168.80.128
08/28/2023-11:21:21.611942 :2:53 [**] [1:2034636:2] ET INFO Python SimpleHTTP ServerBanner [**] [Classification: Misc activity] [Priority: 3] {TCP} 192.168.80.128:8080 -> 192.168.80.1:52345
08/28/2023-11:21:21.611942 :2:53 [**] [1:2210038:2] SURICATA STRFAM ETN out of window [**] [Classification: Generic Protocol Command Decode] [Priority: 3] {TCP} 192.168.80.1:52345 -> 192.168.80.1:52345
```

Activities Terminal Aug 29 12:17

ditiss@ubuntu: ~/project

ditiss@ubuntu: ~/project

```
08/28/2023-23:57:10.648535 [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:42628 -> 192.168.80.2:53
08/28/2023-23:57:10.648535 [**] [1:1000011:0] Access to Domain [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:42628 -> 192.168.80.2:53
08/28/2023-23:57:10.896813 [**] [1:1000011:0] Access to Domain [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.2:53 -> 192.168.80.136:42628
08/28/2023-23:58:13.852088 [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:46076 -> 192.168.80.2:53
08/28/2023-23:58:13.852088 [**] [1:1000011:0] Access to Domain [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:46076 -> 192.168.80.2:53
08/28/2023-23:58:13.888906 [**] [1:1000011:0] Access to Domain [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.2:53 -> 192.168.80.136:46076
08/29/2023-00:00:21.874291 [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:47121 -> 192.168.80.2:53
08/29/2023-00:00:21.874291 [**] [1:1000011:0] Access to Domain [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:47121 -> 192.168.80.2:53
08/29/2023-00:00:22.096705 [**] [1:1000011:0] Access to Domain [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.2:53 -> 192.168.80.136:47121
08/29/2023-00:01:09.802250 [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:40805 -> 192.168.80.2:53
08/29/2023-00:01:09.802250 [**] [1:1000011:0] Access to Domain [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:40805 -> 192.168.80.2:53
08/29/2023-00:01:10.048067 [**] [1:1000011:0] Access to Domain [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.2:53 -> 192.168.80.136:40805
08/29/2023-00:01:22.213095 [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:47061 -> 192.168.80.2:53
08/29/2023-00:01:22.213095 [**] [1:1000011:0] Access to Domain [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:47061 -> 192.168.80.2:53
08/29/2023-00:01:22.221409 [**] [1:1000011:0] Access to Domain [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.2:53 -> 192.168.80.136:47061
08/29/2023-00:01:22.225843 [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:36837 -> 192.168.80.2:53
08/29/2023-00:01:22.225843 [**] [1:1000011:0] Access to Domain [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:36837 -> 192.168.80.2:53
08/29/2023-00:01:22.235000 [**] [1:1000011:0] Access to Domain [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.2:53 -> 192.168.80.136:36837
08/29/2023-00:04:37.902128 [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:53731 -> 192.168.80.2:53
08/29/2023-00:04:37.902128 [**] [1:1000011:0] Access to Domain [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:53731 -> 192.168.80.2:53
08/29/2023-00:04:37.921547 [**] [1:1000011:0] Access to Domain [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.2:53 -> 192.168.80.136:53731
08/29/2023-00:06:09.774479 [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:43365 -> 192.168.80.2:53
08/29/2023-00:06:09.774479 [**] [1:1000011:0] Access to Domain [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:43365 -> 192.168.80.2:53
08/29/2023-00:06:09.817698 [**] [1:1000011:0] Access to Domain [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.2:53 -> 192.168.80.136:43365
08/29/2023-00:09:37.903140 [**] [1:1000006:0] DNS Tunneling Detected [**] [Classification: (null)] [Priority: 3] {UDP} 192.168.80.136:42065 -> 192.168.80.2:53
```

10. Future Enhancement:

Here are some ideas for future enhancements for your project:

- Use a more powerful IDPS tool: If you are using Snort, you could consider using Suricata instead. Suricata is more powerful and flexible than Snort, and it offers some additional features, such as multi-threading and support for more protocols.
- Use a cloud-based IDPS tool: Cloud-based IDPS tools are becoming increasingly popular. These tools offer a number of advantages, such as scalability, ease of deployment, and cost-effectiveness.
- Integrate your IDPS with other security tools: You could integrate your IDPS with other security tools, such as firewalls and intrusion prevention systems. This would allow you to get a more comprehensive view of your network security and to take a more proactive approach to threat detection and mitigation.
- Use machine learning to improve your IDPS's detection capabilities: Machine learning can be used to improve the detection capabilities of IDPS tools. This is because machine learning can be used to identify patterns in network traffic that are indicative of malicious activity.
- Automate the response to IDPS alerts: You could automate the response to IDPS alerts. This would allow you to take action on alerts more quickly and efficiently.
- Use a SIEM tool to correlate IDPS alerts with other security data: A SIEM tool can be used to correlate IDPS alerts with other security data, such as firewall logs and network traffic data. This can help you to identify and investigate potential threats more effectively.

11. Conclusion:

In conclusion, this project has demonstrated how to use Snort or Suricata to detect intrusions and send alerts via email. The system was configured to detect specific threats, such as network scanning or port scanning. This helped to reduce the number of false alerts and made it easier to identify and respond to real threats.

The project also showed how to configure alert notification to ensure that you are notified of potential threats in a timely manner. This is an important step in protecting your network from malicious activity.

The project has several limitations. First, it only uses a single IDPS tool. In a real-world environment, you would likely use a combination of IDPS tools to get a more comprehensive view of your network security. Second, the project does not address the issue of false positives. False positives are alerts that are generated by the IDPS tool but are not caused by malicious activity. You will need to configure the IDPS tool to filter out false positives so that you can focus on real threats.

Despite these limitations, the project has demonstrated the basic concepts of IDPS and alert notification. This is a good starting point for further research and development.

12. References:

- . <https://suricata.io/>
- . <https://www.snort.org/>
- . <https://bard.google.com/>
- . <https://www.youtube.com/>
- . Some extra notes and suggestions from mentors.