



Faculty of Engineering and Technology
Computer Science and Cyber Security Department
Intrusion Detection and Prevention System

IDS/IPS Practical Projects.

Team 8 – DNS Security & Tunneling Detection.

Prepared by:-

Afnan Manasrah – 1221150.

Aseel Rafea – 1222375.

Instructor:-

Rasheed Yousef.

Section Number: 1.

BIRZEIT

Abstract:-

This report aims to enhance the security of the Domain Name System (DNS) protocol by detecting malicious activities related to DNS tunneling operation used by attackers to establish backdoors or route data off the network. Normal DNS traffic was analyzed using Zeek and Suricata logs to differentiate between legitimate and anomalous behavior. Rules were developed using Suricata and Zeek to detect suspicious patterns, such as highly random queries, long queries (especially TXT), and abnormal spikes in NXDOMAIN responses, as well as pattern signatures of tunneling tools like iodine and dns2tcp. The report also presents a practical demonstration of this project, outlining the attack steps, detection mechanisms, and how to prevent attacks using IPS and firewalls.

Contents

Abstract:-	2
Introduction:-	5
Architecture Design:-	5
Implementation:-	6
1. General Setup:-	6
2. Internal Setup:-	6
a) Sensor machine:-	6
b) Victim (UbuntuServer) machine:-	13
c) Attacker (Kali) machine:-	14
3 Monitoring basic, normal DNS traffic:	15
4. Run Iodine:-	17
5. Other detects:-	19
6. Analysis and Metrics:-	21
7. Block via IPS or host firewall guidance:-	22
• Suricata IPS:-	22
• Firwall:-	23
Challenges:-	24
Recommendations:-	24
Conclusion:-	24

Figures

Figure 1: architecture design 1	6
Figure 2: Machines	6
Figure 3: sensor network sittings	7
Figure 4: change Suricata interface.....	8
Figure 5: change paths for some related files	8
Figure 6: create the directories and change the ownership	8
Figure 7: Suricata rules	9
Figure 8: Suricata is running.....	10
Figure 9: Zeek install	10
Figure 10: decompress the Zeek file	10
Figure 11:configuration build	10
Figure 12: Zeek is running.....	10
Figure 13: change the interface.....	11
Figure 14:Zeek script – high entropy.....	11
Figure 15: Zeek script – long TXT	12
Figure 16:Zeek script - NXDOMAIN.....	12
Figure 17: put it in local.zeek	13
Figure 18: victim network sittings	13
Figure 19: change the nameserver	14
Figure 20: install iodine tool	14
Figure 21: normal DNS requests 1.....	15
Figure 22: We can see it from Zeek.....	15
Figure 23: normal DNS requests 2.....	16
Figure 24: We can see it from Suricata.....	16
Figure 25: start iodine on victim.....	17
Figure 26: start iodine on attacker	17
Figure 27: Zeek detection	18
Figure 28: Suricata detection 1	18
Figure 29:Suricata detection 2	19
Figure 30: NXDOMAIN requests.....	19
Figure 31: Suricata detect	19
Figure 32: long TXT and high entropy requests	20
Figure 33: Suricata detect	20
Figure 34: Zeek detect	21
Figure 35: resources usage.....	21
Figure 36: add IPS inline line	22
Figure 37: convert alert to drop	22
Figure 38: send requests.....	22
Figure 39: Suricata Prevention.....	23
Figure 40: Firewall sittings	23

Introduction:-

DNS is a fundamental system on the internet responsible for translating simple website names into computer-understood IP addresses. This system is essential for ensuring users can access websites and services quickly and easily. Despite its seemingly simple function, it is vulnerable to attacks, whether through DNS Tunneling which enables attackers to steal data or create a hidden channel that bypasses standard security systems, or any another ways.

To detect these malicious activities, the project relies on two analysis and detection mechanisms, starting with **Zeek – Network Security Monitoring**. Zeek is a powerful network monitoring and traffic analysis tool that converts data into detailed logs, allowing for an understanding of DNS traffic behavior within the network. This means it relies on behavioral analysis, which helps detect activities that lack permanent signatures. In this project, Zeek was used to collect and analyze DNS logs, build a baseline for normal DNS traffic, detect anomalous behavior, and provide data to support the development of Zeek scripts and Suricata rules.

Secondly, **Suricata – IDS/IPS Engine**. Suricata was used as a tool to detect and prevent malicious activities. It works by creating rules dedicated to detecting suspicious patterns in DNS traffic, especially those related to tunneling. Suricata was used in the project, where rules were written based on the patterns detected, detecting tunneling activity, issuing alerts, and activating IPS mode to perform prevention actions. Suricata represents the detection and protection layer that works directly on network traffic.

This means that the security design in this project relies on combining behavioral analysis (Zeek) with signature-based detection and blocking (Suricata). Zeek identifies anomalous behavior, while Suricata performs real-time detection through rules and takes appropriate blocking action.

As for the attack, we will use the iodine tool, which is a DNS tunneling tool that hides data within DNS queries so that it is not detected and is usually used by attackers for the purpose of smuggling data or obtaining sensitive data.

Architecture Design:-

The architecture design for this experiment will be as follows: the victim's device will be located on its own network, which will also have an IDS between the firewall and the DNS server to verify all requests coming out of the network or even the responses coming in to it. The victim will generate innocent requests which the attacker, whether on the same network or outside it, will exploit to smuggle data.

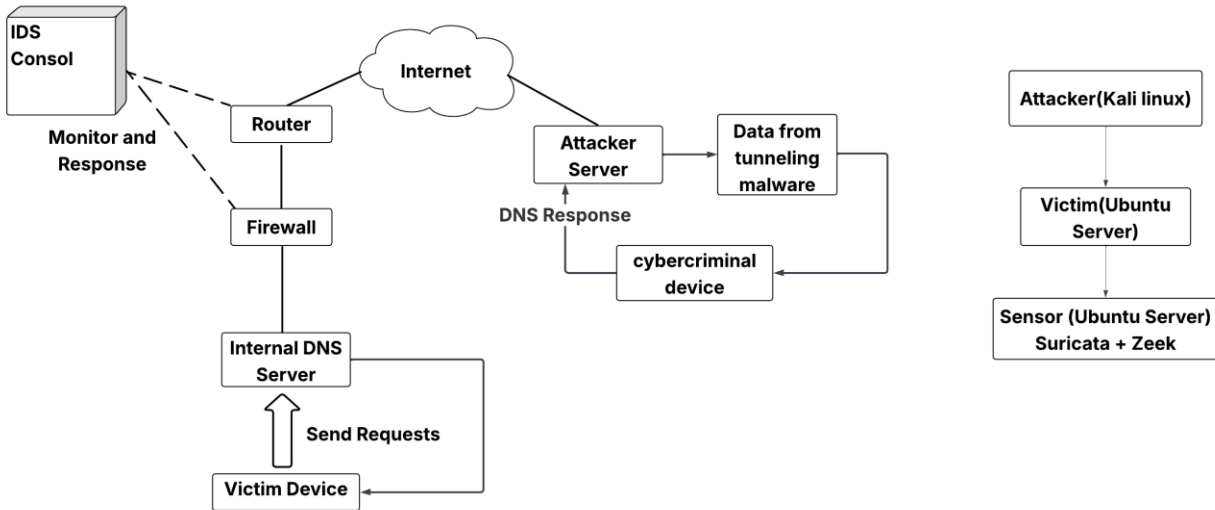


Figure 1: architecture design 1

Implementation:-

1. General Setup:-

To build the experiment environment, we will make three virtual machines: two Ubuntu servers, one for the victim and one to act as a sensor, and the third machine will be kali to act as attacker. Each machine will have two adapters: one NAT and the other is Host-only. The NAT for internet connection, and Host-only to create an isolated environment to implement the attack without affecting the network.

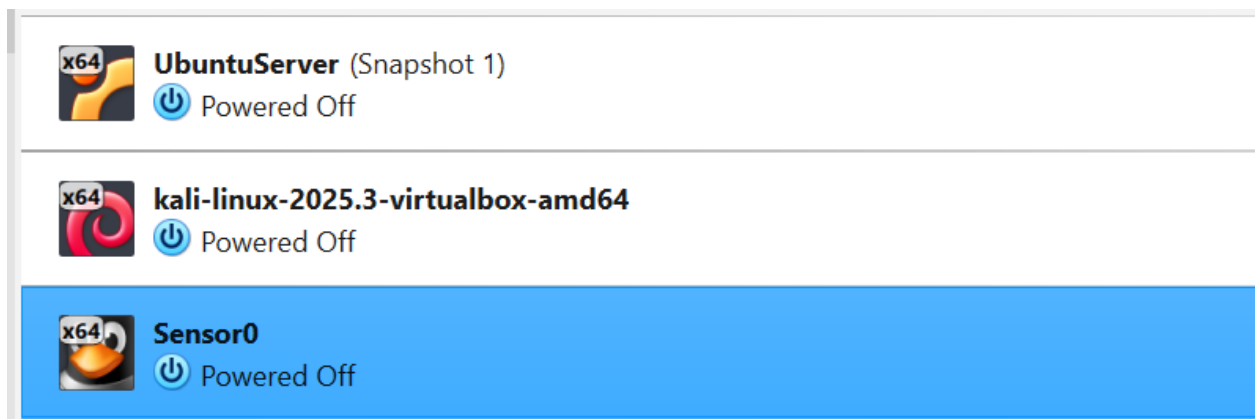


Figure 2: Machines

2. Internal Setup:-

a) Sensor machine:-

In the sensor machine the first thing we do is network settings and we make two adapter and make the setting as follow:

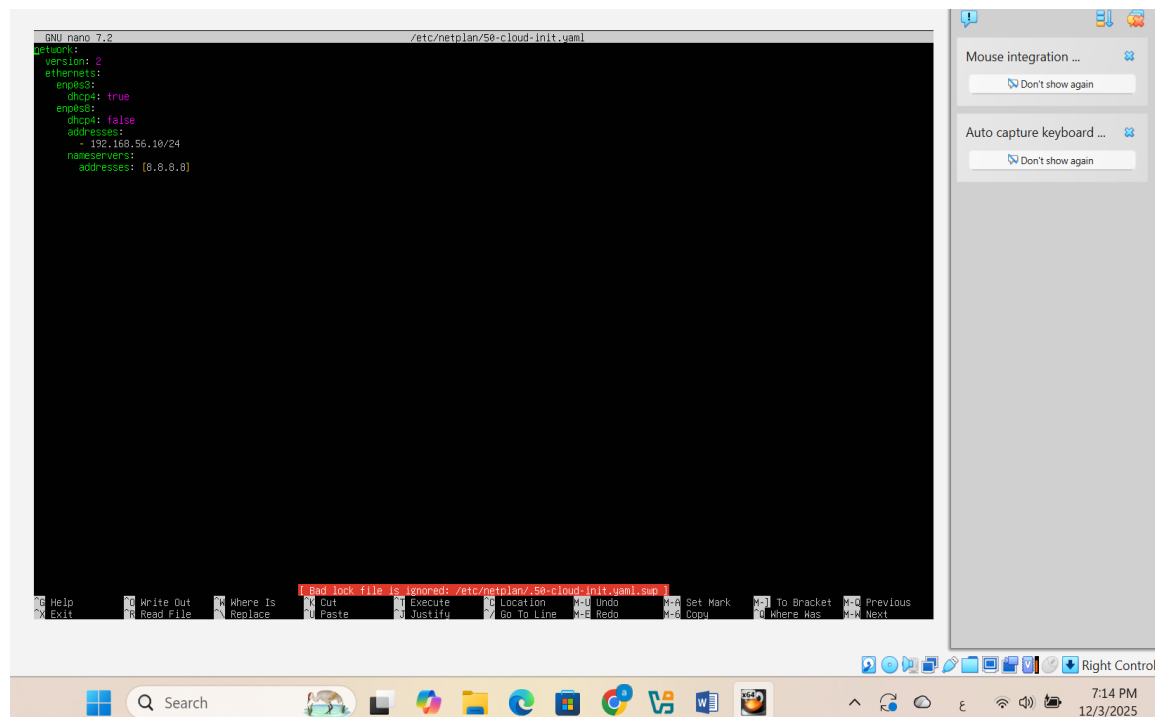


Figure 3: sensor network settings

We did it this way because the detection it will be perform on the enp0s8 interface and it's the same interface the victim will use to make DNS requests.

After that we install Suricata to perform detection, we install it as follows:

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
[Mon Nov 24 21:16:01] afnansensor@sensor:~$ sudo wget https://raw.githubusercontent.com/OISF/suricata/master/suricata.yaml.in -O /etc/suricata/suricata.yaml
--2025-11-24 21:17:31-- https://raw.githubusercontent.com/OISF/suricata/master/suricata.yaml.in
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... failed: Name or service not known.
wget: unable to resolve host address 'raw.githubusercontent.com'
[Mon Nov 24 21:17:31] afnansensor@sensor:~$ sudo wget https://raw.githubusercontent.com/OISF/suricata/master/suricata.yaml.in -O /etc/suricata/suricata.yaml
--2025-11-24 21:17:45-- https://raw.githubusercontent.com/OISF/suricata/master/suricata.yaml.in
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)[185.199.108.133]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 93287 (91K) [text/plain]
Saving to: '/etc/suricata/suricata.yaml'

/etc/suricata/suricata.yaml      100%[=====] 91.10K  476KB/s  in 0.2s
2025-11-24 21:17:46 (476 KB/s) - '/etc/suricata/suricata.yaml' saved [93287/93287]
```

Commands:

- `sudo apt install suricata -y` : to install Suricata tool.
- `sudo wget https://raw.githubusercontent.com/OISF/suricata/master/suricata.yaml.in -O /etc/suricata/suricata.yaml` : to install the configuration file that will work with it. And we make some changes in it, like the interface that will work in it and make it enp0s8, and change some file paths to save the results of detection and all related work in it.

```
##
# Linux high speed capture support
af-packet:
- interface: enp0s8
  # 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: 254
  # 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.
  # Recommended modes are cluster_flow on most boxes and cluster_cpu or cluster_qm on system
  # with capture card using RSS (requires cpu affinity tuning and system IRQ tuning)
  # cluster_rollover has been deprecated; if used, it'll be replaced with cluster_flow.
  cluster-type: cluster_flow
  # In some fragmentation cases, the hash can not be computed. If "defrag" is set
  # to yes, the kernel will do the needed defragmentation before sending the packets.
```

Figure 4: change Suricata interface

```
##
## Configure Suricata to load Suricata-Update managed rules.
##
default-rule-path: /var/lib/suricata/rules

rule-files:
- suricata.rules

##
## Auxilliary configuration files.
##
classification-file: /etc/suricata/classification.config
reference-config-file: /etc/suricata/reference.config
# threshold-file: @e_sysconfdir@threshold.config

##
## Suricata as a Firewall options (experimental)
##
firewall:
# toggle to enable firewall mode
#enabled: no

# Firewall rule file are in their own path and are not managed
# by Suricata-Update.
```

Figure 5: change paths for some related files

And then we make the directories we will need to store operating files, log files, rules, and cache, and then change ownership of directories to ensure that Suricata have a permission on it.

```
[Mon Nov 24 21:40:02] afnansensor@sensor:~$ sudo mkdir -p /var/run/suricata
[Mon Nov 24 21:40:30] afnansensor@sensor:~$ sudo mkdir -p /var/log/suricata
[Mon Nov 24 21:40:36] afnansensor@sensor:~$ sudo mkdir -p /var/lib/suricata
[Mon Nov 24 21:40:44] afnansensor@sensor:~$ sudo mkdir -p /var/lib/suricata/rules
[Mon Nov 24 21:40:53] afnansensor@sensor:~$ sudo mkdir -p /var/cache/suricata
[Mon Nov 24 21:41:08] afnansensor@sensor:~$ sudo chown -R suricata:suricata /var/run/suricata
chown: invalid user: 'suricata:suricata'
[Mon Nov 24 21:41:42] afnansensor@sensor:~$ getent passwd suricata
[Mon Nov 24 21:44:15] afnansensor@sensor:~$ sudo useradd --no-create-home --system --shell /usr/sbin/nologin suricata
[Mon Nov 24 21:45:35] afnansensor@sensor:~$ sudo groupadd --system suricata
groupadd: group 'suricata' already exists
[Mon Nov 24 21:45:53] afnansensor@sensor:~$ sudo usermod -a -G suricata suricata
[Mon Nov 24 21:46:37] afnansensor@sensor:~$ sudo useradd --no-create-home --system --shell /usr/sbin/nologin suricata
useradd: user 'suricata' already exists
[Mon Nov 24 21:46:45] afnansensor@sensor:~$ sudo chown -R suricata:suricata /var/run/suricata
[Mon Nov 24 21:46:52] afnansensor@sensor:~$ sudo chown -R suricata:suricata /var/log/suricata
[Mon Nov 24 21:47:21] afnansensor@sensor:~$ sudo chown -R suricata:suricata /var/lib/suricata
[Mon Nov 24 21:47:41] afnansensor@sensor:~$ sudo chown -R suricata:suricata /var/cache/suricata
[Mon Nov 24 21:47:47] afnansensor@sensor:~$
```

Figure 6: create the directories and change the ownership

Then, for Suricata to do its job and detect malicious DNS requests, we wrote the following rules:

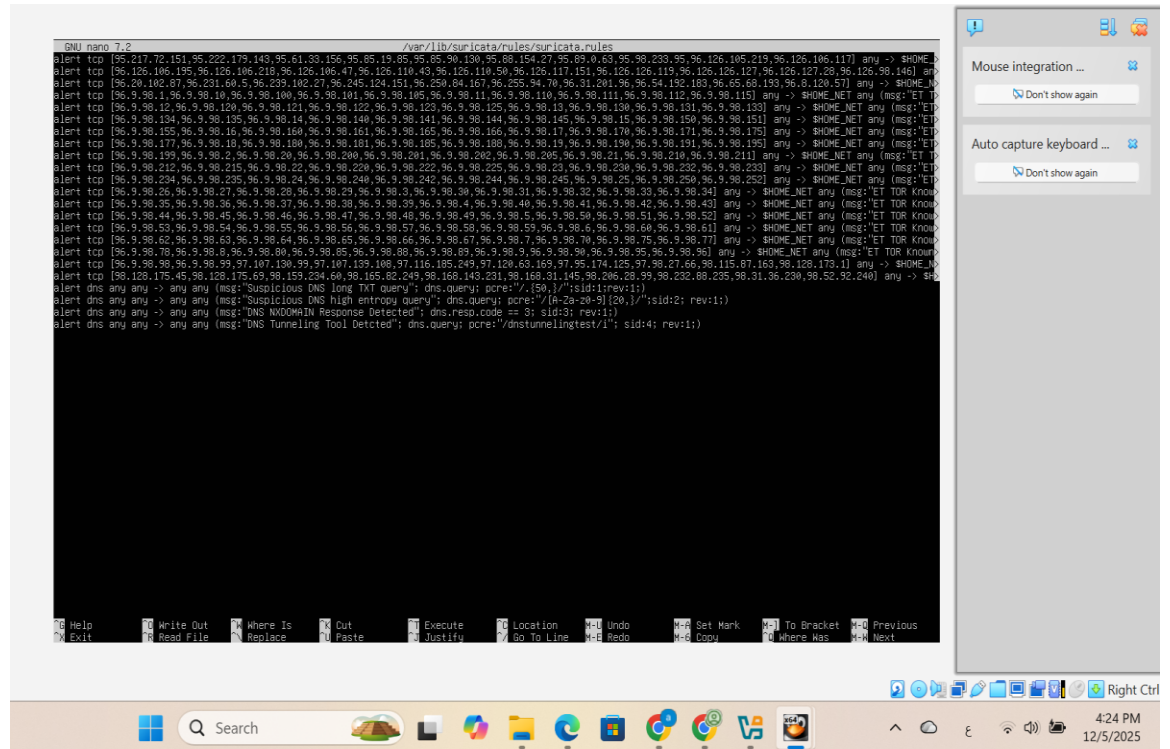


Figure 7: Suricata rules

- Alert dns any any -> any any (msg:"Suspicious DNS Long TXT Query"; dns.query; pcre:"/{50,}/"; sid:1; rev:1;
This query to detect and make alert to any DNS request from any IP and any port that go to any IP and any port and detect any long domains or subdomains with repeated characters exceeding 50 and when give alert the rule will generate the message "Suspicious DNS Long TXT Query", sid: 1 is the number of rule and rev:1 is the number of version.
- Alert dns any any -> any any (msg:"Suspicious DNS high entropy Query"; dns.query; pcre:"/[A-Za-z0-9]{20,}/"; sid:2; rev:1;
This query to detect and make alert to any DNS request from any IP and any port that go to any IP and any port and detect any Base64 or Base32 domains or subdomains and when give alert the rule will generate the message "Suspicious DNS high entropy Query", sid:2 is the number of rule and rev:1 is the number of version.
- Alert dns any any -> any any (msg:"DNS NXDOMAIN Response Detected"; dns.resp.code==3; sid:3; rev:1;
This query to detect and make alert to any DNS request from any IP and any port that go to any IP and any port and detect any response containing the phrase NXDOMAIN and the response code is 3 and its known to NXDOMAIN responses, and when give alert the rule will generate the message "DNS NXDOMAIN Response Detected", sid:3 is the number of rule and rev:1 is the number of version.

- Alert dns any any -> any any (msg:"DNS Tunneling Tool Detected";dns.query;content:"dnstunnelingtest";sid:4; rev:1;
This query to detect and make alert to any DNS request from any IP and any port that go to any IP and any port and detect any request containing the phrase dnstunnelingtest, and when give alert the rule will generate the message "DNS Tunneling Tool Detected", sid:4 is the number of rule and rev:1 is the number of version.

```
[Thu Dec 04 03:14:40] afnansensor@sensor:~$ sudo suricata -T -c /etc/suricata/suricata.yaml
[sudo] password for afnansensor:
Info: conf-yaml-loader: Configuration node 'enabled' redefined.
Info: conf-yaml-loader: Configuration node 'enabled' redefined.
i: suricata: This is Suricata version 7.0.3 RELEASE running in SYSTEM mode
i: suricata: Configuration provided was successfully loaded. Exiting.
[Thu Dec 04 03:34:54] afnansensor@sensor:~$
```

Figure 8: Suricata is running

And then install Zeek to perform detection, we install it as follows:

```
[Mon Nov 24 17:12:14] afnansensor@sensor:~$ wget https://download.zeek.org/zeek-8.0.4.tar.gz
--2025-11-24 17:12:42-- https://download.zeek.org/zeek-8.0.4.tar.gz
Resolving download.zeek.org (download.zeek.org)... 52.84.45.124, 52.84.45.68, 52.84.45.34, ...
Connecting to download.zeek.org (download.zeek.org)|52.84.45.124|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 99613526 (95M) [application/x-gzip]
Saving to: 'zeek-8.0.4.tar.gz'

zeek-8.0.4.tar.gz      100%[=====] 95.00M 3.10MB/s   in 24s
2025-11-24 17:13:07 (3.99 MB/s) - 'zeek-8.0.4.tar.gz' saved [99613526/99613526]
[Mon Nov 24 17:13:07] afnansensor@sensor:~$
```

Figure 9: Zeek install

```
2025-11-24 17:13:07 (3.99 MB/s) - 'zeek-8.0.4.tar.gz' saved [99613526/99613526]
[Mon Nov 24 17:13:07] afnansensor@sensor:~$ tar -xvzf zeek-8.0.4.tar.gz
```

Figure 10: decompress the Zeek file

```
-bash: ./configure: No such file or directory
[Mon Nov 24 17:14:37] afnansensor@sensor:~/zeek-8.0.4$ ./configure
```

Figure 11: configuration build

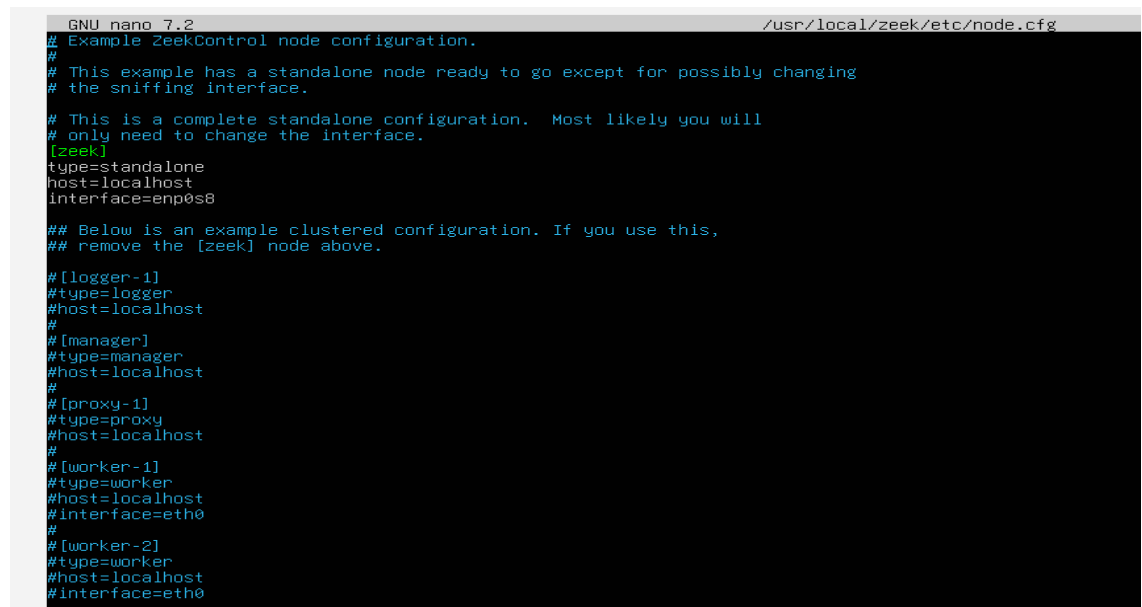
Commands:

- wget <https://download.zeek.org/zeek-8.0.4.tar.gz>: install zeek directory from the organization site.
- tar -xvzf zeek-8.0.4.tar.gz : decompress the file.
- cd zeek-8.0.4 && ./configure: To inspect the device and prepare the construction file.
- make && sudo make install: to build and compile the program and install the resulting files within the system.

```
starting ...
starting zeek ...
[Wed Nov 26 22:37:37] afnansensor@sensor:~$ sudo /usr/local/zeek/bin/zeekctl status
Name      Type      Host      Status  Pid   Started
zeek      standalone localhost running 1491   26 Nov 22:37:35
[Wed Nov 26 22:38:28] afnansensor@sensor:~$
```

Figure 12: Zeek is running

After that we change the interface that will work in it and make it enp0s8.



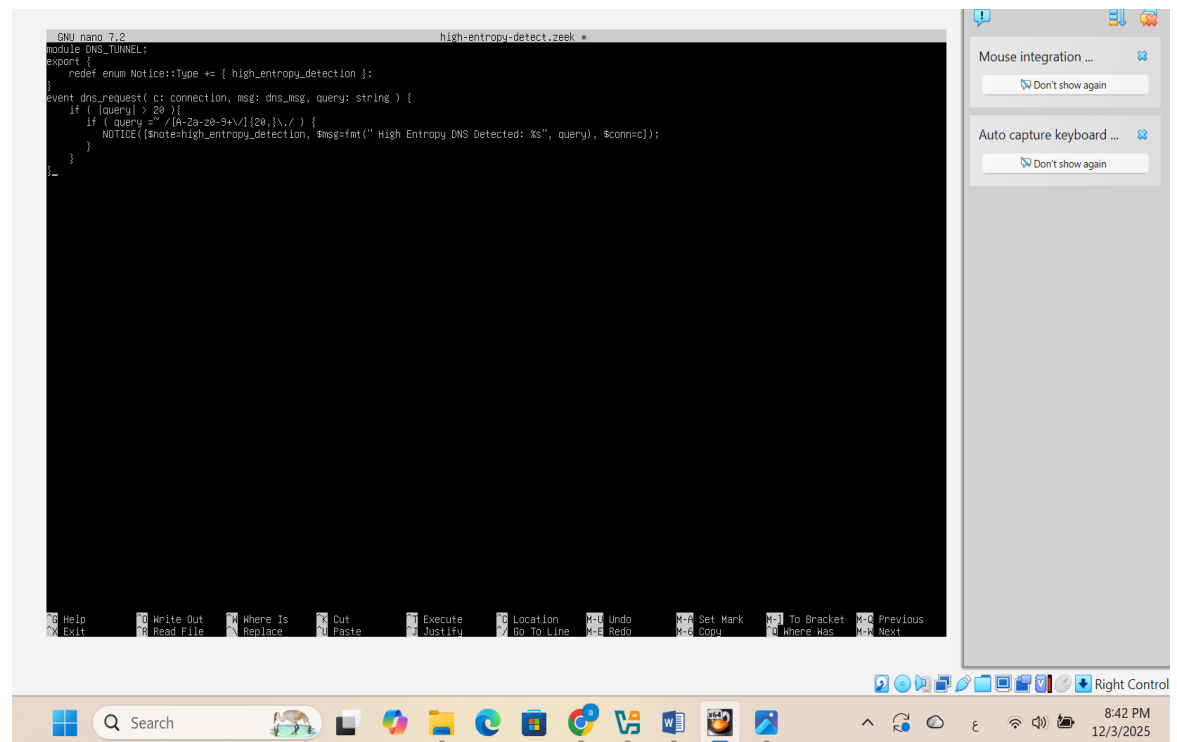
```
GNU nano 7.2 /usr/local/zeek/etc/node.cfg
# Example ZeekControl node configuration.
#
# This example has a standalone node ready to go except for possibly changing
# the sniffing interface.
#
# This is a complete standalone configuration. Most likely you will
# only need to change the interface.
[zeek]
type=standalone
host=localhost
interface=enp0s8

## Below is an example clustered configuration. If you use this,
## remove the [zeek] node above.

#[logger-1]
#type=logger
#host=localhost
#
#[manager]
#type=manager
#host=localhost
#
#[proxy-1]
#type=proxy
#host=localhost
#
#[worker-1]
#type=worker
#host=localhost
#interface=eth0
#
#[worker-2]
#type=worker
#host=localhost
#interface=eth0
```

Figure 13: change the interface

Then we convert the Suricata rules to Zeek scripts (they work in the same way and for the same purpose, but the writing method differs due to the different tools).



```
GNU nano 7.2 high-entropy-detect.zeek
module DNS_TUNNEL;
export {
  redef enum Notice::Type += { high_entropy_detection };
}
event dns_request(c: connection, msg: dns_msg, query: string) {
  if ( (query) > 20 ) {
    if ( query =~ /[A-Za-z0-9+~]{20,}/ ) {
      NOTICE([notice::high_entropy_detection, $msg=fmt(" High Entropy DNS Detected: %s", query), $conn=c]);
    }
  }
}
```

Figure 14:Zeek script – high entropy

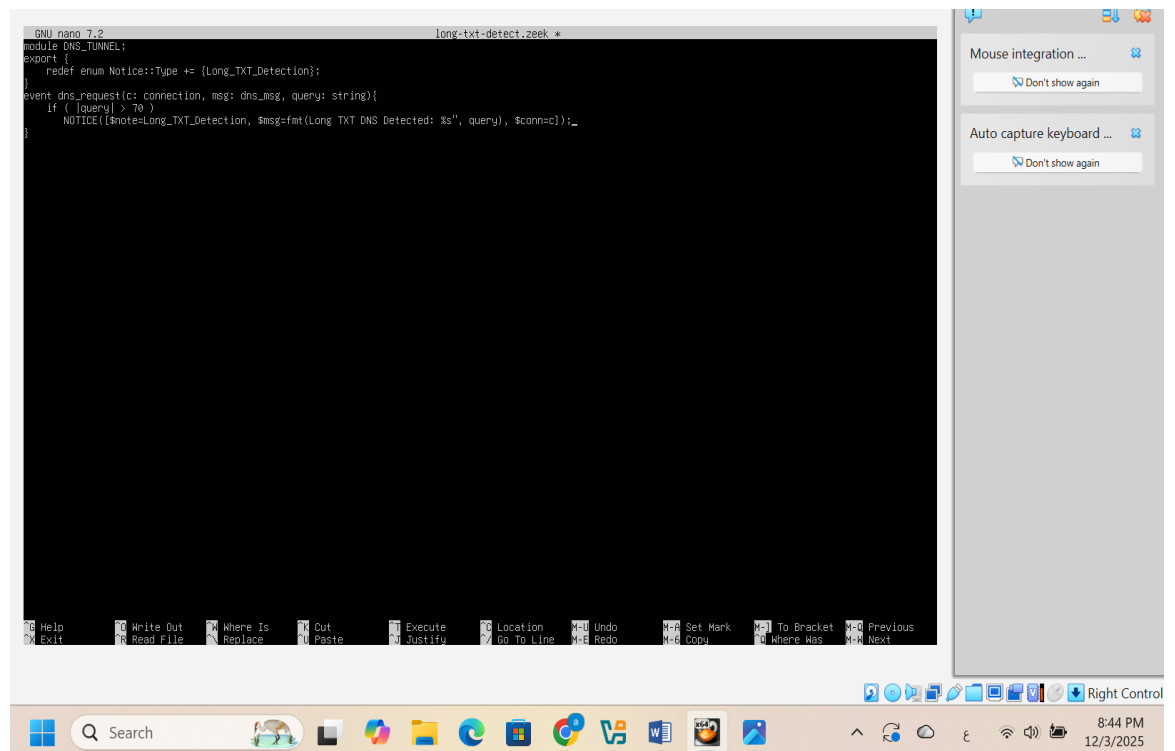


Figure 15: Zeek script – long TXT

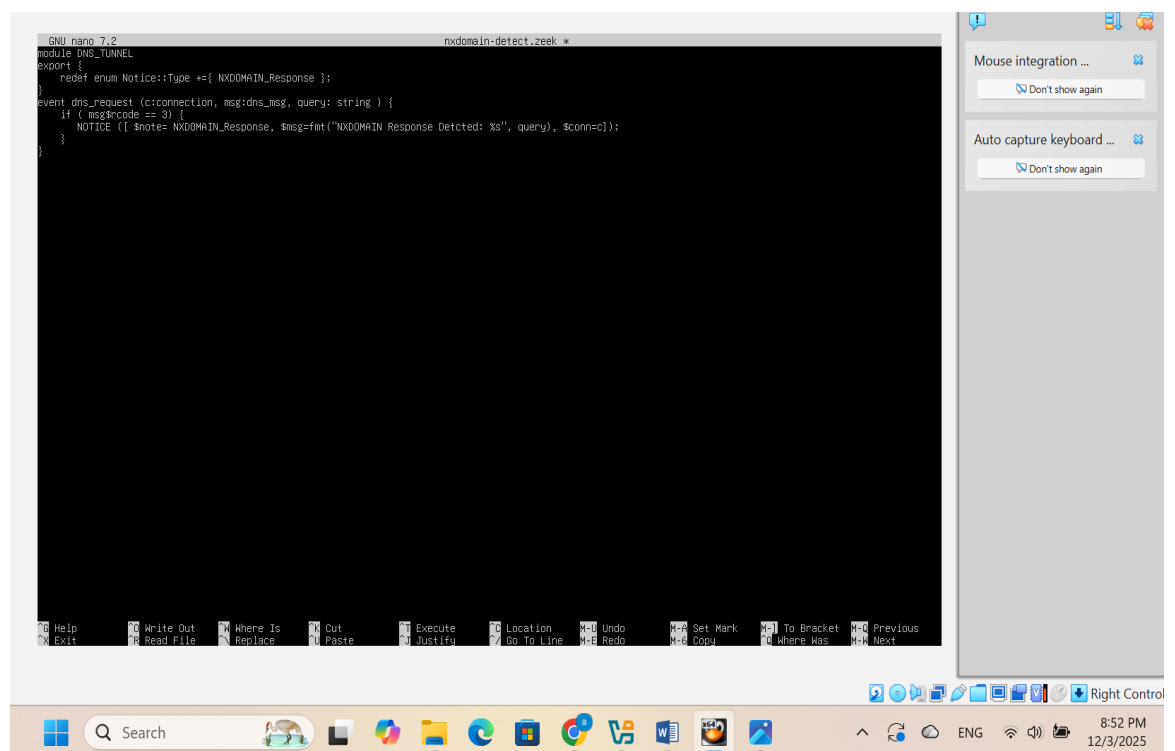


Figure 16: Zeek script - NXDOMAIN

And then we put it in local.zeek to load it and use it to detect after we changed the permissions to be enforceable.

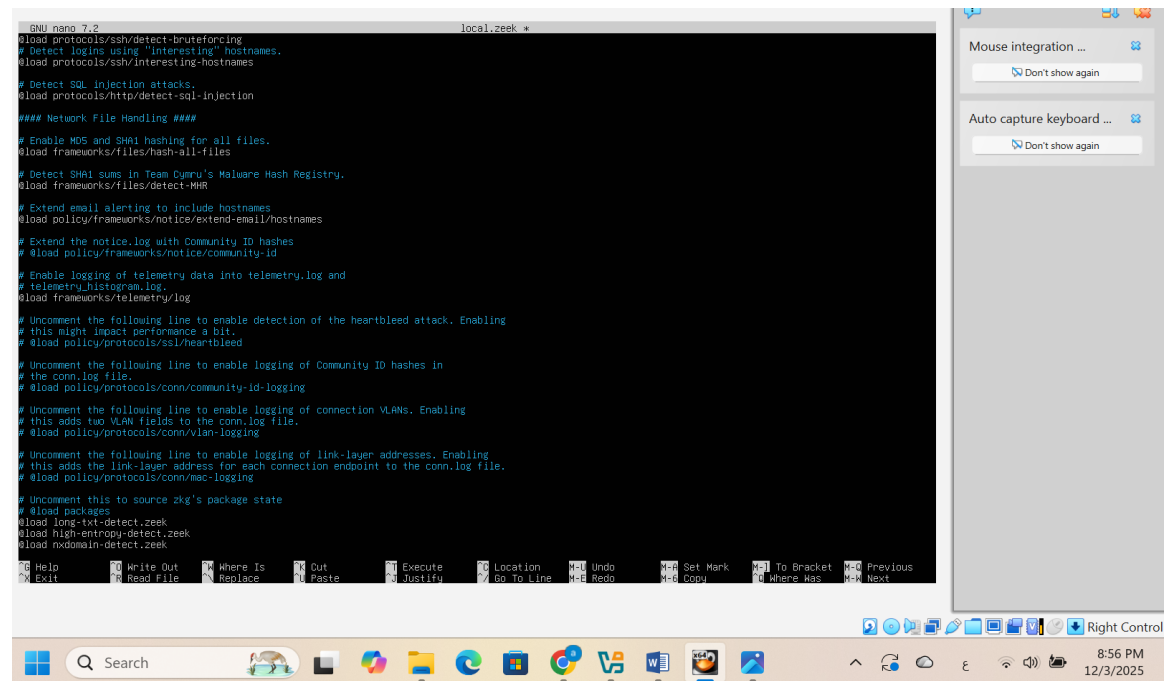


Figure 17: put it in local.zeek

b) Victim (UbuntuServer) machine:-

In the victim machine the first thing we do is network settings and we make two adapter and make the setting as follow:

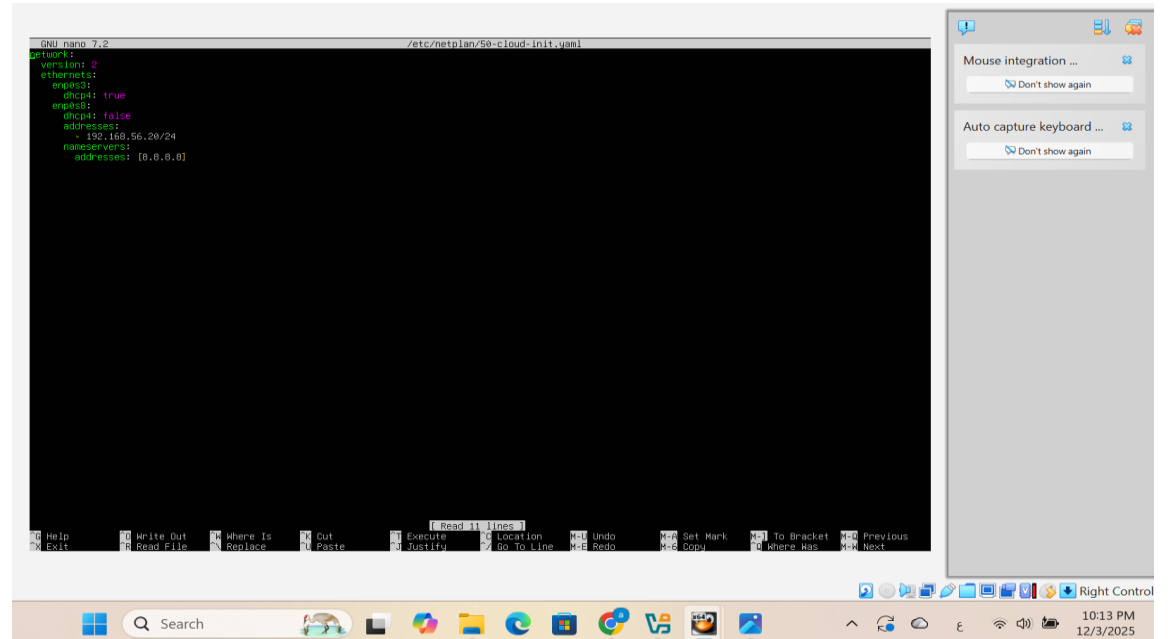


Figure 18: victim network settings

Then we changed the IP of the DNS server to be the same as the victim's IP to represented the trusted DNS system, because the IDS is located between it and the firewall and therefore to hold all the traffic of the DNS, whether innocent or malicious.

```
GNU nano 7.2 /etc/resolv.conf
# This is /run/systemd/resolve/stub-resolv.conf managed by man:systemd-resolved(8).
# Do not edit.
#
# This file might be symlinked as /etc/resolv.conf. If you're looking at
# /etc/resolv.conf and seeing this text, you have followed the symlink.
#
# This is a dynamic resolv.conf file for connecting local clients to the
# internal DNS stub resolver of systemd-resolved. This file lists all
# configured search domains.
#
# Run "resolvectl status" to see details about the uplink DNS servers
# currently in use.
#
# Third party programs should typically not access this file directly, but only
# through the symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a
# different way, replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.

nameserver 192.168.56.20
options edns0 trust-ad
search .
```

Figure 19: change the nameserver

c) Attacker (Kali) machine:-

In this machine, we will simulate the DNS tunneling using two tools, dns2tcp & iodine, which we will install as follows:

Commands:

- Sudo apt install iodine: to install iodine tool.

```
root@kali: ~
└─$ sudo apt update
Get:1 http://kali.download/kali kali-rolling InRelease [34.0 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [28.9 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [26.6 MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [114 kB]
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [299 kB]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Packages [188 kB]
Get:7 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [894 kB]
Get:8 http://kali.download/kali kali-rolling/non-free-firmware amd64 Packages [11.7 kB]
Get:9 http://kali.download/kali kali-rolling/non-free-firmware amd64 Contents (deb) [26.5 kB]
Fetched 75.1 MB in 38s (1,976 kB/s)
3200 packages can be upgraded. Run 'apt list --upgradable' to see them.

root@kali: ~
└─$ sudo apt install iodine
Upgrading:
  iodine
Summary:
Upgrading: 1, Installing: 0, Removing: 0, Not Upgrading: 1199
Download size: 85.7 kB
Space needed: 0 B / 63.5 GB available

Get:1 http://kali.download/kali kali-rolling/main amd64 iodine amd64 0.7.0-12 [85.7 kB]
Fetched 85.7 kB in 1s (138 kB/s)
Preparing packages ...
(Reading database ... 43718 files and directories currently installed.)
Preparing to unpack .../iodine_0.7.0-12_amd64.deb ...
Unpacking iodine (0.7.0-12) over (0.7.0-11) ...
Setting up iodine (0.7.0-12) ...
iodine.service is a disabled or a static unit not running, not starting it.
Processing triggers for man-db (2.13.1-1) ...
Processing triggers for kali-menu (2025.3.2) ...

root@kali: ~
└─$ iodine -h
iodine IP over DNS tunneling client
Usage: iodine [-v] [-h] [-r] [-u user] [-t chrootdir] [-d device] [-P password] [-m maxfragsize] [-M maxlen] [-T type] [-O enc] [-L l1] [-l sec] [-r context] [-f pidfile] [nameserver] topdomain
Options to try if connection doesn't work:
  -f force dns type: NUL, PRIVATE, TXT, SRV, MX, CNAME, A (default: autodetect)
  -O force downstream encoding for -f other than NUL: Base32, Base64, Base64u,
    Base128, or (only for TXT): Raw (default: autodetect)
  -I max interval between requests (default 4 sec) to prevent DNS timeouts
  -l l: use lazy mode for low-latency (default). 0: don't (implies -I)
  -m max size of downstream fragments (default: autodetect)
  -M max size of upstream hostnames (-100-255, default: 255)
  -r to skip raw UDP mode attempt
  -P password used for authentication (max 32 chars will be used)
```

Figure 20: install iodine tool

3. Monitoring basic, normal DNS traffic:

To monitor typical DNS traffic, we will generate requests from the victim (we tested it over several days) using dig and nslookup commands to see if they will be seen by the tools or not.

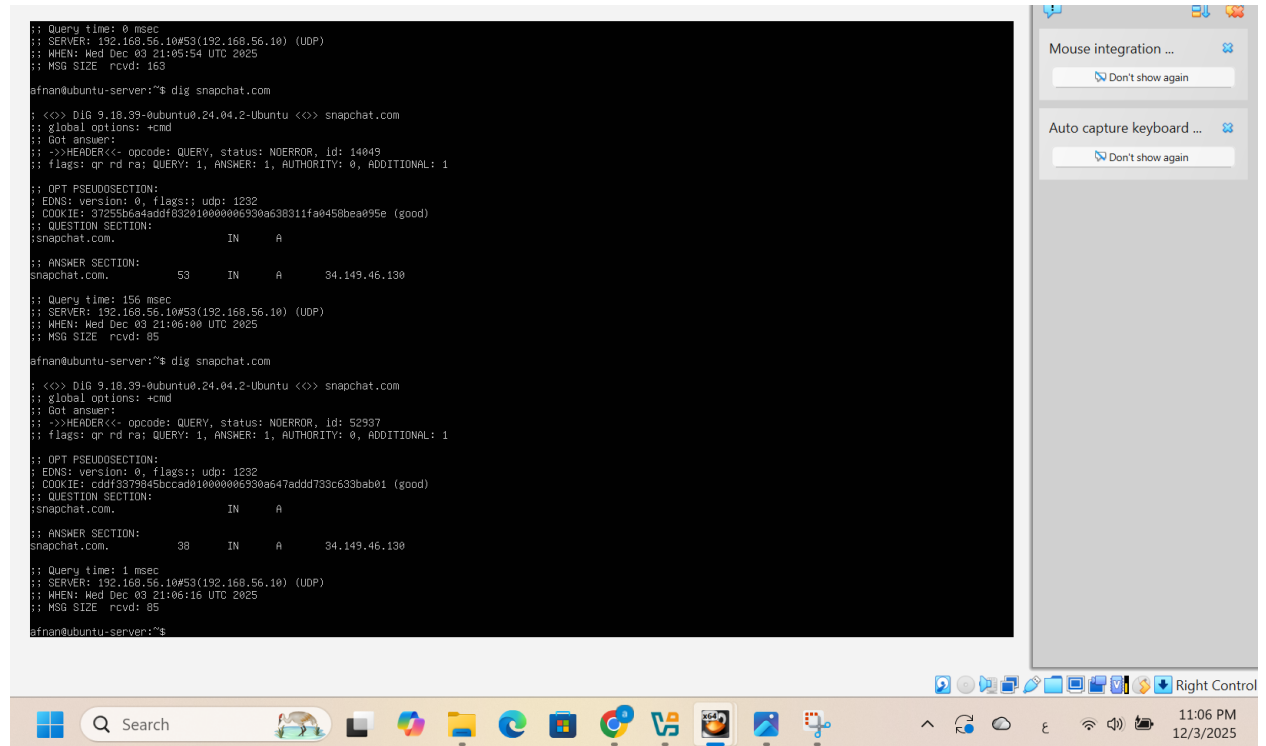


Figure 21: normal DNS requests 1

```
[Wed Dec 03 23:06:10] afnansensor@sensor:~$ sudo cat /usr/local/zeek/logs/current/dns.log
```

#separator	\x09														
#set_separator	,														
#empty_field	(empty)														
#unset_field	-														
#path	dns														
#open	2025-12-03-23-03-12														
#fields	ts uid id.orig_h id.orig_p id.resp_h id.resp_p proto trans_id rtt query qclass qclass_name qtype														
#types	time rcode addr port addr port enum count interval answers TTLs rejected count string string count string bool bool														
#bool	bool count														
1764795782.086563	-	-	-	-	-	53	udp	10125	-	google.com	1	C_INTERNET	1	-	-
1764795782.423410	-	-	-	-	-	53	udp	43092	-	google.com	1	C_INTERNET	2	-	-
1764795785.354449	-	-	-	-	-	53	udp	17311	-	google.com	1	C_INTERNET	1	-	-
1764795798.250264	-	-	-	-	-	53	udp	37788	-	google.com	1	C_INTERNET	1	-	-
1764795799.502660	-	-	-	-	-	53	udp	20219	-	google.com	1	C_INTERNET	1	-	-
1764795799.507405	-	-	-	-	-	53	udp	53297	-	google.com	1	C_INTERNET	2	-	-
1764795879.949460	-	-	-	-	-	53	udp	13729	-	microsoft.com	1	C_INTERNET	1	-	-
1764795880.094290	-	-	-	-	-	53	udp	59137	-	microsoft.com	1	C_INTERNET	2	-	-
1764795884.119108	-	-	-	-	-	53	udp	25006	-	microsoft.com	1	C_INTERNET	1	-	-
1764795884.121967	-	-	-	-	-	53	udp	33774	-	microsoft.com	1	C_INTERNET	2	-	-
1764795894.216546	-	-	-	-	-	53	udp	13655	-	google.com	1	C_INTERNET	1	-	-
1764795903.515842	-	-	-	-	-	53	udp	42040	-	facebook.com	1	C_INTERNET	1	-	-
1764795936.168911	-	-	-	-	-	53	udp	24101	-	facebook.com	1	C_INTERNET	1	-	-
1764795946.663686	-	-	-	-	-	53	udp	33323	-	instagram.com	1	C_INTERNET	1	-	-
1764795954.084839	-	-	-	-	-	53	udp	11502	-	google.com	1	C_INTERNET	1	-	-
1764795959.870163	-	-	-	-	-	53	udp	14049	-	snapchat.com	1	C_INTERNET	1	-	-

```
[Wed Dec 03 23:06:19] afnansensor@sensor:~$
```

Figure 22: We can see it from Zeek

4. Run Iodine:-

First, we start the tool on the victim to create the tunnel and to deal with the requests that arrive from the attacker and process them and forward the responses to him.

```
aseel@aseel-VMware-Virtual-Platform:~$ sudo iodined -f -P 1234 -l 1 192.168.17.133 -p 53 10.0.0.1/24 tunnel.test
Opened dns1
Setting IP of dns1 to 10.0.0.1
Setting MTU of dns1 to 1130
Opened IPv4 UDP socket
Listening to dns for domain tunnel.test
```

Figure 25: start iodine on victim

Command: `sudo iodined -f -P 1234 -l 192.168.17.133 -p 10.0.0.1/24 tunnel.test`: through this command, we create a tunnel and set the password to it (1234), specifying the server's IP (192.168.17.133), the port will listen to it (53), the determine the IP range within which the tunnel will work (10.0.0.1/24), and determine the domain that will be used within the queries (tunnel.test).

```
(kali㉿kali)-[~]
$ sudo iodine -P 1234 192.168.17.133 tunnel.test

Opened dns2
Opened IPv4 UDP socket
Sending DNS queries for tunnel.test to 192.168.17.133
Autodetecting DNS query type (use -T to override).
Using DNS type NULL queries
Version ok, both using protocol v 0x00000502. You are user #0
Setting IP of dns2 to 10.0.0.2
Setting MTU of dns2 to 1130
Server tunnel IP is 10.0.0.1
Testing raw UDP data to the server (skip with -r)
Server is at 192.168.17.133, trying raw login: OK
Sending raw traffic directly to 192.168.17.133
Connection setup complete, transmitting data.
Detaching from terminal...

(kali㉿kali)-[~]
$
```

Figure 26: start iodine on attacker

Command: `sudo iodined -P 1234 -l 192.168.17.133 tunnel.test`: through this command, we run the iodine tool on the attacker's machine and put the password for it (1234), specifying the server's IP (192.168.17.133), and determining the domain that will be used within the queries (tunnel.test). Through it, encrypted data is sent within queries for the same domain.

Meanwhile, the Suricata and Zeek tools captured the DNS queries and gave an alerts to them, and the type of the queries that were detected is: Null queries and long TXT queries.

```
aseel@aseel-VMware-Virtual-Platform:~$ sudo tail -f /opt/zeek/logs/current/dns.log
[sudo] password for aseel:
1764842530.544966 CDJtuyIuYUlnmrca 192.168.17.1 63643 224.0.0.252 5355 udp 40622 - desktop-m424s4v 1 C_INTERNET
255 * - - F F F F 0 - - -F
1764842590.065162 Cg1Ax01F5Xz0U86FLf 192.168.17.133 52645 8.8.8.853 udp 35290 - - - - - 3
NXDOMAIN F F F F 0 - - T
1764842615.081117 CK6Dm740c8F2D4uq9d 192.168.17.133 55916 8.8.8.853 udp 22261 - - - - - 3
NXDOMAIN F F F F 0 - - T
1764842615.132591 CpVCGw1wjyWctAaLHL 192.168.17.133 33768 8.8.8.853 udp 22587 - - - - - 3
NXDOMAIN F F F F 0 - - T
1764842644.539846 CL9VuGMLPeuCqCnK7 192.168.17.129 45338 192.168.17.133 53 udp 21620 - yrbgbe.tunnel.test 1 C_I
VNET 10 NULL F F T F 0 - - F
1764842644.541665 CL9VuGMLPeuCqCnK7 192.168.17.129 45338 192.168.17.133 53 udp 29347 - vaaaakayeu.tunnel.test 1 C_I
VNET 10 NULL F F T F 0 - - F
1764842644.542716 CL9VuGMLPeuCqCnK7 192.168.17.129 45338 192.168.17.133 53 udp 37074 - ladxta41mxnen4g5sad4v2zs2sydjajq.tu
rnel.test 1 C_INTERNET 10 NULL - - F F TF 0 - - F
1764842644.550825 CL9VuGMLPeuCqCnK7 192.168.17.129 45338 192.168.17.133 53 udp 44801 - lagbh.tunnel.test 1 C_I
VNET 10 NULL - - F F T F 0 - - F
1764842799.100714 C395xf22yaaBkx8GwL 192.168.17.133 48335 8.8.8.853 udp 6668 - connectivity-check.ubuntu.com - -
-- 0 NOERROR F F T 0 185.125.190.98,185.125.190.48,91.189.91.97,91.189.91.49,185.125.190.18,91.189.91.48,
31.189.91.96,185.125.190.49,185.125.190.97,185.125.190.96,91.189.91.98,185.125.190.17 13.000000,13.000000,13.000000,13.000000,13.000000,13.0
30000,13.000000,13.000000,13.000000,13.000000,13.000000 F
1764843099.110507 CcHZVX22dK6E0puot7 192.168.17.133 56306 8.8.8.853 udp 6785 - connectivity-check.ubuntu.com - -
-- 0 NOERROR F F T 0 185.125.190.98,185.125.190.48,91.189.91.97,91.189.91.49,185.125.190.18,91.189.91.97,
185.125.190.96,91.189.91.96,185.125.190.18,185.125.190.49,91.189.91.48,185.125.190.17 13.000000,13.000000,13.000000,13.000000,13.000000,13.0
30000,13.000000,13.000000,13.000000,13.000000,13.000000 F
1764843399.079061 C1wLPX2aodNvd0f8of 192.168.17.133 42193 8.8.8.853 udp 16908 - connectivity-check.ubuntu.com - -
-- 0 NOERROR F F T 0 91.189.91.98,185.125.190.98,185.125.190.97,91.189.91.49,185.125.190.18,91.189.91.98,185.125.190.9
7,185.125.190.48,185.125.190.17,91.189.91.97,185.125.190.96,91.189.91.48,91.189.91.96 57.000000,57.000000,57.000000,57.000000,57.000000,57.0
30000,57.000000,57.000000,57.000000,57.000000,57.000000 F
1764843440.482923 CnnF3mtCUpLuv3Nee fe80::366c:2dc5:16fe:785b 56019 ff02::1:3 5355 udp 6091 - desktop-m424s4v 1
C_INTERNET 255 * - - F F F 0
```

Figure 27: Zeek detection

```
12:29
aseel@aseel-VMware-Virtual-Platform:~$
9 -> 192.168.17.133:53
/04/2025-12:24:20.583999 [**] [1:400001:1] DNS Tunneling Suspicion - Long TXT
Jery [**] [Classification: A Network Trojan was detected] [Priority: 1] {UDP}
2.168.17.129:49658 -> 192.168.17.133:53
/04/2025-12:24:50.690982 [**] [1:400001:1] DNS Tunneling Suspicion - Long TXT
Jery [**] [Classification: A Network Trojan was detected] [Priority: 1] {UDP}
2.168.17.129:54809 -> 192.168.17.133:53
/04/2025-12:25:20.772557 [**] [1:400001:1] DNS Tunneling Suspicion - Long TXT
Jery [**] [Classification: A Network Trojan was detected] [Priority: 1] {UDP}
2.168.17.129:34812 -> 192.168.17.133:53
/04/2025-12:25:50.856310 [**] [1:400001:1] DNS Tunneling Suspicion - Long TXT
Jery [**] [Classification: A Network Trojan was detected] [Priority: 1] {UDP}
2.168.17.129:58713 -> 192.168.17.133:53
/04/2025-12:26:20.919566 [**] [1:400001:1] DNS Tunneling Suspicion - Long TXT
Jery [**] [Classification: A Network Trojan was detected] [Priority: 1] {UDP}
2.168.17.129:55625 -> 192.168.17.133:53
/04/2025-12:26:50.990800 [**] [1:400001:1] DNS Tunneling Suspicion - Long TXT
Jery [**] [Classification: A Network Trojan was detected] [Priority: 1] {UDP}
2.168.17.129:41543 -> 192.168.17.133:53
/04/2025-12:27:21.058724 [**] [1:400001:1] DNS Tunneling Suspicion - Long TXT
Jery [**] [Classification: A Network Trojan was detected] [Priority: 1] {UDP}
2.168.17.129:39725 -> 192.168.17.133:53
/04/2025-12:27:51.115409 [**] [1:400001:1] DNS Tunneling Suspicion - Long TXT
Jery [**] [Classification: A Network Trojan was detected] [Priority: 1] {UDP}
2.168.17.129:52158 -> 192.168.17.133:53
/04/2025-12:28:21.207313 [**] [1:400001:1] DNS Tunneling Suspicion - Long TXT
Jery [**] [Classification: A Network Trojan was detected] [Priority: 1] {UDP}
2.168.17.129:37549 -> 192.168.17.133:53
/04/2025-12:28:51.284551 [**] [1:400001:1] DNS Tunneling Suspicion - Long TXT
Jery [**] [Classification: A Network Trojan was detected] [Priority: 1] {UDP}
2.168.17.129:48680 -> 192.168.17.133:53
```

Figure 28: Suricata detection 1


```
aseel@aseel-VMware-Virtual-Platform:~$ sudo tail -f /var/log/suricata/fast.log
[sudo] password for aseel:
12/04/2025-11:46:43.896377  [**] [1:2029994:1] ET HUNTING Suspicious NULL DNS Request [**] [Classification: Misc activity] [Priority: 3] {UDP} 192.168.17.129:59139 -> 192.168.17.133:53
12/04/2025-12:01:19.189207  [**] [1:2029994:1] ET HUNTING Suspicious NULL DNS Request [**] [Classification: Misc activity] [Priority: 3] {UDP} 192.168.17.129:47407 -> 192.168.17.133:53
12/04/2025-12:01:26.174894  [**] [1:2029994:1] ET HUNTING Suspicious NULL DNS Request [**] [Classification: Misc activity] [Priority: 3] {UDP} 192.168.17.129:47407 -> 192.168.17.133:53
12/04/2025-12:01:29.883258  [**] [1:2022973:1] ET INFO Possible Kali Linux hostname in DHCP Request Packet [**] [Classification: Potential Corporate Privacy Violation] [Priority: 1] {UDP} 192.168.17.129:68 -> 192.168.17.254:67
12/04/2025-12:01:40.295974  [**] [1:2029994:1] ET HUNTING Suspicious NULL DNS Request [**] [Classification: Misc activity] [Priority: 3] {UDP} 192.168.17.129:47407 -> 192.168.17.133:53
12/04/2025-12:04:04.539846  [**] [1:2029994:1] ET HUNTING Suspicious NULL DNS Request [**] [Classification: Misc activity] [Priority: 3] {UDP} 192.168.17.129:45338 -> 192.168.17.133:53
12/04/2025-12:04:04.541665  [**] [1:2029994:1] ET HUNTING Suspicious NULL DNS Request [**] [Classification: Misc activity] [Priority: 3] {UDP} 192.168.17.129:45338 -> 192.168.17.133:53
12/04/2025-12:04:04.542716  [**] [1:400001:1] DNS Tunneling Suspicion - Long TXT Query [**] [Classification: A Network Trojan was detected] [Priority: 1] {UDP} 192.168.17.129:45338 -> 192.168.17.133:53
12/04/2025-12:04:04.542716  [**] [1:2029994:1] ET HUNTING Suspicious NULL DNS Request [**] [Classification: Misc activity] [Priority: 3] {UDP} 192.168.17.129:45338 -> 192.168.17.133:53
12/04/2025-12:04:04.550825  [**] [1:2029994:1] ET HUNTING Suspicious NULL DNS Request [**] [Classification: Misc activity] [Priority: 3] {UDP} 192.168.17.129:45338 -> 192.168.17.133:53
12/04/2025-12:16:29.908678  [**] [1:2022973:1] ET INFO Possible Kali Linux hostname in DHCP Request Packet [**] [Classification: Potential Corporate Privacy Violation] [Priority: 1] {UDP} 192.168.17.129:68 -> 192.168.17.254:67
```

Figure 29: Suricata detection 2

5. Other detects:-

- Sent requests for a domain that doesn't exist:

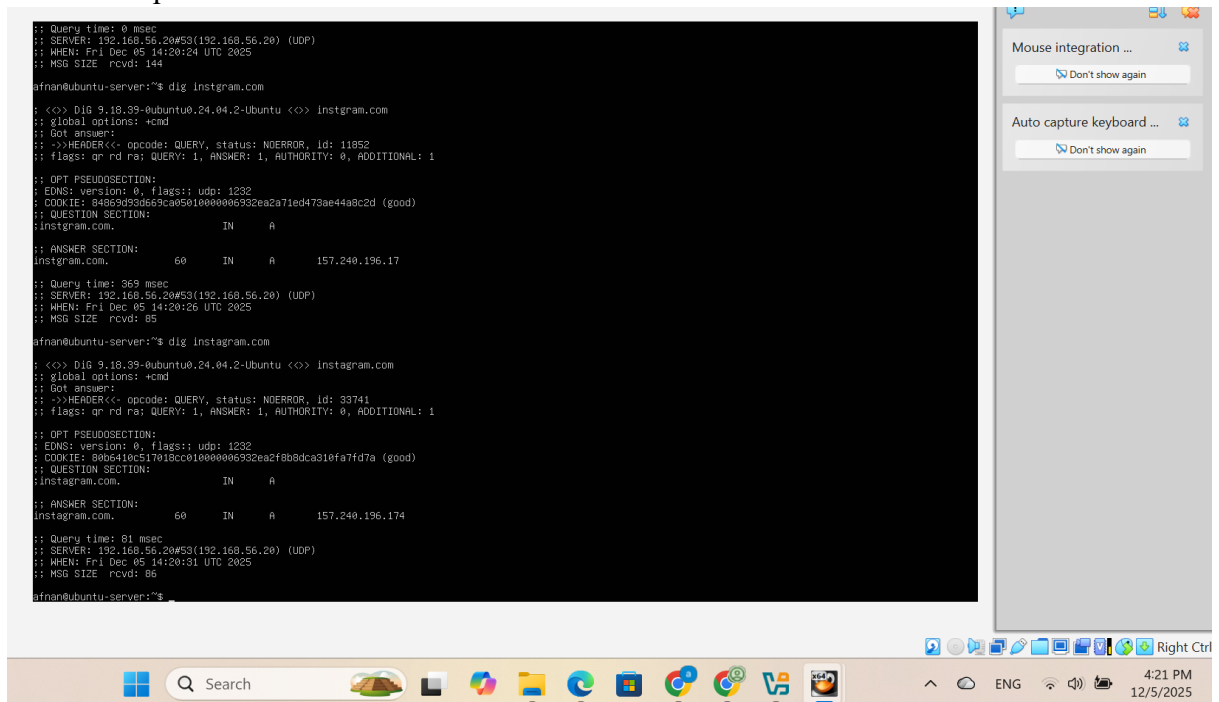


Figure 30: NXDOMAIN requests

```
[1907 - Suricata-Main] 2025-12-05 16:20:22 Error: detect-parse: An invalid action 'alert' was given
[1907 - Suricata-Main] 2025-12-05 16:20:22 Error: detect: error parsing signature "alert ip any any -> any any (msg:"SURICATA Applayer Mismatch protocol both d
rections"; flow:established; app-layer-event:applayer_mismatch_protocol_both_directions; flowint:applayer.anomaly.count,+,1; classtype:protocol-command-decode:
sid:2260000; rev:1);" from file /var/lib/suricata/rules/suricata.rules at line 1
[1907 - Suricata-Main] 2025-12-05 16:20:35 Error: detect-parse: unknown rule keyword 'dns.resp.code == 3'.
[1907 - Suricata-Main] 2025-12-05 16:20:35 Error: detect: error parsing signature "alert dns any any -> any any (msg:"DNS NXDOMAIN Response Detected"; dns.resp.
code == 3; sid:9; rev:1);" from file /var/lib/suricata/rules/suricata.rules at line 62346
[1907 - Suricata-Main] 2025-12-05 16:20:35 Info: detect: 1 rule files processed, 46530 rules successfully loaded, 2 rules failed, 0
[1907 - Suricata-Main] 2025-12-05 16:20:35 Info: threshold-config: Threshold config parsed: 0 rule(s) found
[1907 - Suricata-Main] 2025-12-05 16:20:35 Info: detect: 46533 signatures processed, 1059 are IP-only rules, 4422 are inspecting packet payload, 40823 inspect a
pplication layer, 108 are decoder event only
[1907 - Suricata-Main] 2025-12-05 16:20:36 Info: afnansensor@sensor:~$
```

Figure 31: Suricata detect

- Sent long TXT and high entropy requests:-

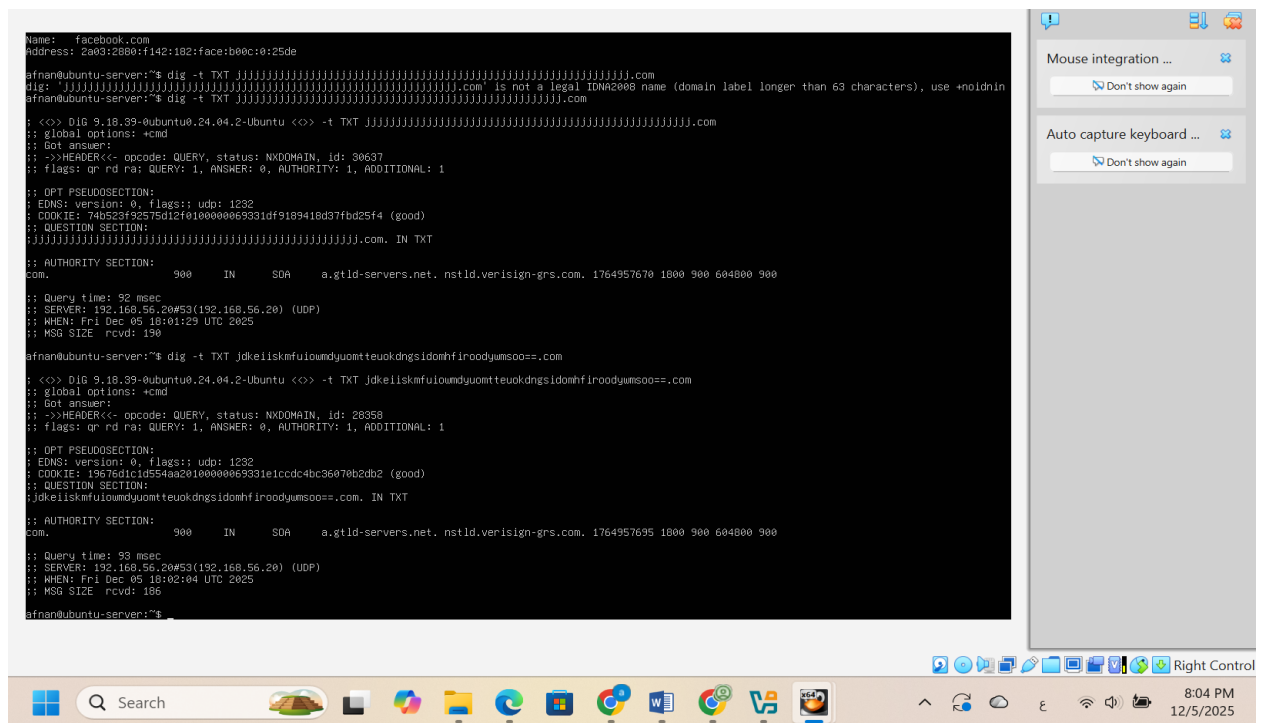


Figure 32: long TXT and high entropy requests

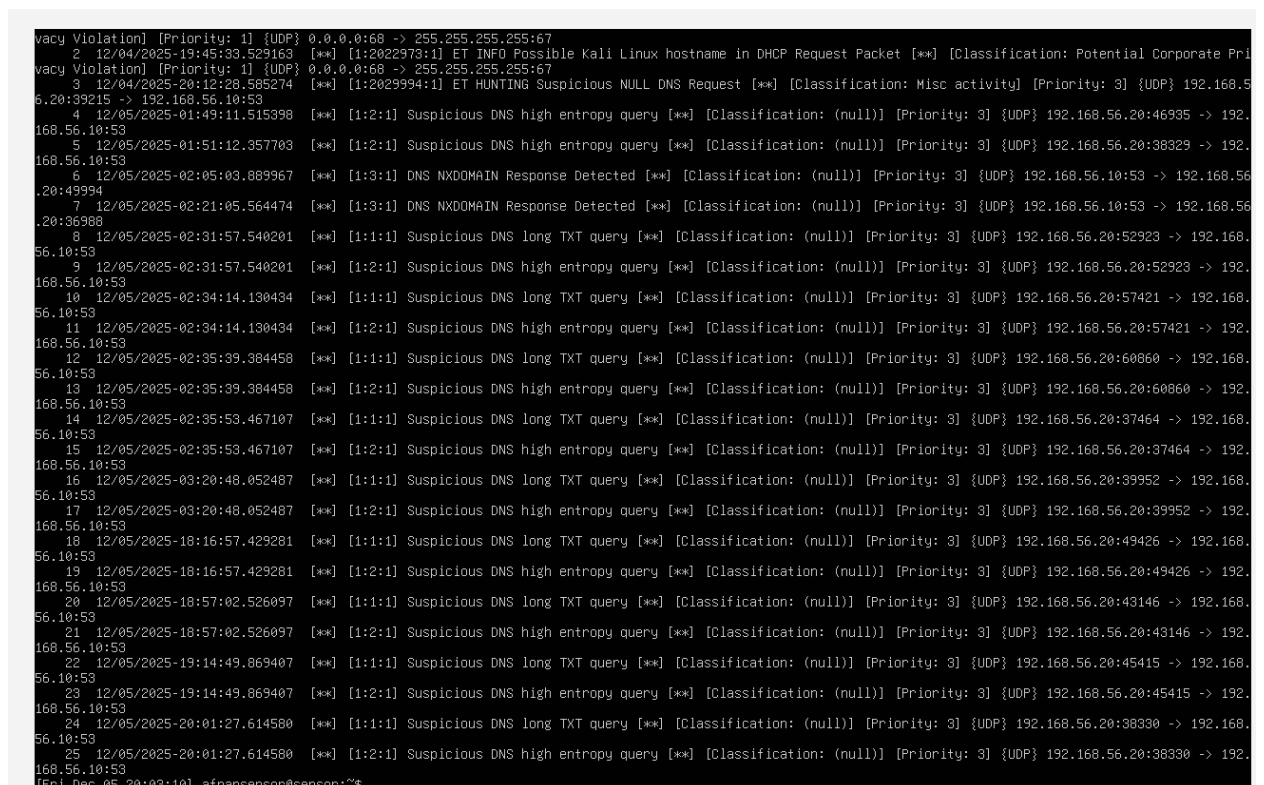


Figure 33: Suricata detect

```
#unset_field -
#path dns
#open 2025-12-05-20-00-00
#fields ts uid id.orig_h id.orig_p id.resp_h id.resp_p proto trans_id rtt query qclass qclass_name qtype
#type_name rcode name AA TC RD RA Z answers TTLs rejected count string count string count string count string bool bool
#types time string addr port addr port enum count interval string count string count string count string bool bool
bool count vector[string] vector[interval] bool
1764957598.306046 S - - F 1 - - 192.168.56.20 50464 - 192.168.56.10 53 udp 49125 - com 1 C_INTERNET 43 D
1764957598.371197 3 - - F 1 - - 192.168.56.20 51457 - 192.168.56.10 53 udp 435 - ubuntu.com 1 C_INTERNET 4
1764957598.438921 3 DS - - F 1 - - 192.168.56.20 54801 - 192.168.56.10 53 udp 893 - com 1 C_INTERNET 48 D
1764957605.768784 NSKEY - - F 1 - - 192.168.56.20 45717 - 192.168.56.10 53 udp 29145 - microsoft 1 C_INTERNET 4
1764957615.716188 3 F DNSKEY - - F 1 - - 192.168.56.20 42726 - 192.168.56.10 53 udp 52582 - google.com 1 C_INTERNET 4
1764957634.273598 3 DS - - F 1 - - 192.168.56.1 5353 224.0.0.251 5353 udp 0 - desktop-ujdp1j3._dosvc._tcp.local 1
C_INTERNET 255 * 0 NOERROR T F F 0 desktop-ujdp1j3._dosvc._tcp.local 0.000000 F
1764957634.275731 C_XGUBP1jPVG16XN4e fe80::30c9:e660:e09d:db30 5353 udp 0 - desktop-ujdp1j3._dosvc._
tcp.local 1 C_INTERNET 255 * 0 NOERROR T F F 0 desktop-ujdp1j3._dosvc._tcp.local 0.000000
1764957634.530366 CsrP11iM2ssm0IuSGc 192.168.56.1 5353 224.0.0.251 5353 udp 0 0.508475 desktop-ujdp1j3._dosvc._tcp.local
1 1 C_INTERNET 255 * NOERROR T F F 0 desktop-ujdp1j3._dosvc._tcp.local 4500.000000 F
1764957634.531957 C_XGUBP1jPVG16XN4e fe80::30c9:e660:e09d:db30 5353 ff02::fb 5353 udp 0 0.508406 desktop-ujdp1j3._
dosvc._tcp.local 1 C_INTERNET 255 * 0 NOERROR T F F 0 desktop-ujdp1j3._dosvc._tcp.local 120.000000 F
1764957634.531957 C_XGUBP1jPVG16XN4e fe80::30c9:e660:e09d:db30 5353 ff02::fb 5353 udp 0 0.508406 desktop-ujdp1j3._
dosvc._tcp.local 1 C_INTERNET 255 * 0 NOERROR T F F 0 desktop-ujdp1j3._dosvc._tcp.local 120.000000 F
1764957634.784541 CsrP11iM2ssm0IuSGc 192.168.56.1 5353 224.0.0.251 5353 udp 0 0.256879 desktop-ujdp1j3._dosvc._tcp.local
1 1 C_INTERNET 255 * NOERROR T F F 0 desktop-ujdp1j3.local 120.000000 F
1764957634.786017 C_XGUBP1jPVG16XN4e fe80::30c9:e660:e09d:db30 5353 ff02::fb 5353 udp 0 0.256319 desktop-ujdp1j3._
dosvc._tcp.local 1 C_INTERNET 255 * 0 NOERROR T F F 0 desktop-ujdp1j3.local 120.000000 F
1764957634.159118 CEBM6gkrQ5oRQR2S4 192.168.56.20 35073 192.168.56.10 53 udp 21740 - facebook.com 1 C_INTERNET 1
1764957687.614580 3 Chp7fo4dndr6j3lnhyd 192.168.56.20 38330 192.168.56.10 53 udp 40560 - jfffffffffffffffffffffffffffffffff
1 jfffffffffffffffffffffffffffffffff 16 TXT T F 0
1764957755.105853 C1d2dg20IP8IHP1Jd 192.168.56.1 5353 224.0.0.251 5353 udp 0 - desktop-ujdp1j3._dosvc._tcp.local 1
C_INTERNET 255 * 0 NOERROR T F F 0 desktop-ujdp1j3._dosvc._tcp.local 0.000000 F
1764957755.106615 C_JqbGh4myITPF7mGue fe80::30c9:e660:e09d:db30 5353 ff02::fb 5353 udp 0 - desktop-ujdp1j3._dosvc._
tcp.local 1 C_INTERNET 255 * 0 NOERROR T F F 0 desktop-ujdp1j3._dosvc._tcp.local 0.000000
1764957755.361724 C1d2dg20IP8IHP1Jd 192.168.56.1 5353 224.0.0.251 5353 udp 0 0.508097 desktop-ujdp1j3._dosvc._tcp.local
1 1 C_INTERNET 255 * 0 NOERROR T F F 0 desktop-ujdp1j3._dosvc._tcp.local 4500.000000 F
1764957755.362947 C_JqbGh4myITPF7mGue fe80::30c9:e660:e09d:db30 5353 ff02::fb 5353 udp 0 0.508198 desktop-ujdp1j3._
dosvc._tcp.local 1 C_INTERNET 255 * 0 NOERROR T F F 0 desktop-ujdp1j3._dosvc._tcp.local 120.000000 F
1764957755.614795 C1d2dg20IP8IHP1Jd 192.168.56.1 5353 224.0.0.251 5353 udp 0 0.257457 desktop-ujdp1j3._dosvc._tcp.local
1 1 C_INTERNET 255 * 0 NOERROR T F F 0 desktop-ujdp1j3.local 120.000000 F
1764957755.616456 C_JqbGh4myITPF7mGue fe80::30c9:e660:e09d:db30 5353 ff02::fb 5353 udp 0 0.256833 desktop-ujdp1j3._
dosvc._tcp.local 1 C_INTERNET 255 * 0 NOERROR T F F 0 desktop-ujdp1j3.local 120.000000 F
[Fri Dec 05 20:03:57] afnansensor@sensor:~$
```

Figure 34: Zeek detect

6. Analysis and Metrics:-

As we saw before, Zeek and Suricata succeeded in analyzing the DNS queries and detect the DNS tunneling.

The detection rate of the two tools was high, but some malicious requests were not captured or detected, and Zeek was giving a lot of false positive alerts, and there was a delay in capturing some requests and missing some of them. Out of every five requests, two were sometimes lost, and therefore the detection rate is more than 80% because we use more than one tool to detect.

As for the exploitation of resources, we had a little because of the small size of the network and traffic in it, but the consumption of space in Suricata was higher because of keep all files, even if they were restarted, but Zeek deletes them with every stop of the system, and Zeek consumes less CPU & Memory than Suricata.

top - 20:06:40 up 12 min, 1 user, load average: 0.10, 0.11, 0.09									
Tasks: 129 total, 1 running, 128 sleeping, 0 stopped, 0 zombie									
Mem(s): 0.1 us, 0.1 su, 0.0 ni, 99.7 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st									
MiB Mem : 5886.6 total, 4512.9 free, 1164.1 used, 363.5 buff/cache									
MiB Swap: 3036.0 total, 4036.0 free, 0.0 used, 4642.6 avail Mem									
PID	USER	PPID	UID	VSZ	RES	SHR	S	WQ	TIME COMMAND
1054	root	20	0	1116656	507908	15232	S	0.7	0.5 0123.30 Suricata-Main
370	root	20	0	0	0	0	I	0.3	0.0 0102.35 kworker/1:3-events
1425	root	20	0	1724708	261908	171000	S	0.3	4.4 0103.21 zeek
2303	afnansensor	20	0	11944	5760	3584	R	0.3	0.1 0100.10 top
1	root	20	0	21153	12340	3564	S	0.0	0.2 0100.71 systemd
2	root	20	0	0	0	0	S	0.0	0.0 0100.00 kthread
3	root	20	0	0	0	0	S	0.0	0.0 0100.00 rcu1_workqueue_release
4	root	20	0	0	0	0	I	0.0	0.0 0100.00 kworker/R-rcu_g
5	root	20	0	0	0	0	I	0.0	0.0 0100.00 kworker/R-rcu_p
6	root	20	0	0	0	0	I	0.0	0.0 0100.00 kworker/R-slab
7	root	20	0	0	0	0	I	0.0	0.0 0100.00 kworker/R-netns
8	root	20	0	0	0	0	I	0.0	0.0 0100.04 kworker/0:0-mm_percpu_wq
10	root	20	0	0	0	0	I	0.0	0.0 0100.00 kworker/0:0-events_highpri
12	root	20	0	0	0	0	I	0.0	0.0 0100.00 kworker/R-mm_pg
13	root	20	0	0	0	0	I	0.0	0.0 0100.00 rcu_tasks_kthread
14	root	20	0	0	0	0	I	0.0	0.0 0100.00 rcu_tasks_rude_kthread
15	root	20	0	0	0	0	I	0.0	0.0 0100.00 rcu_tasks_trace_kthread
16	root	20	0	0	0	0	S	0.0	0.0 0100.01 softirqd
17	root	20	0	0	0	0	I	0.0	0.0 0100.09 rcu_preempt

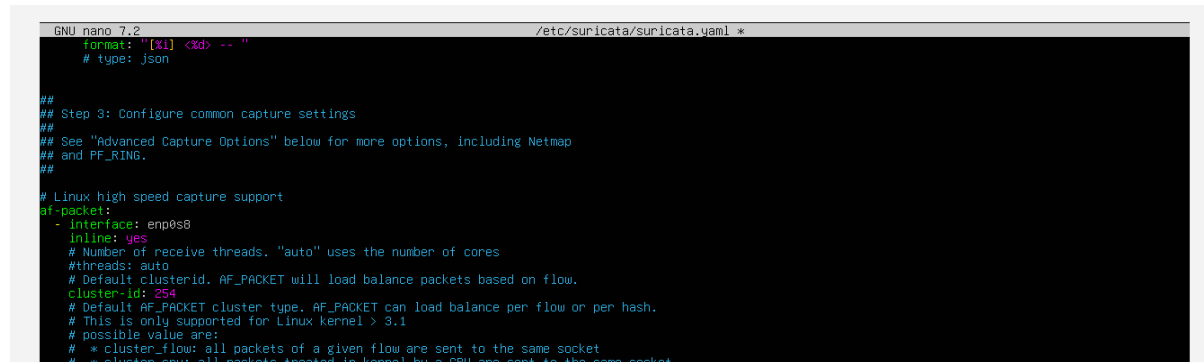
Figure 35: resources usage

7. Block via IPS or host firewall guidance:-

As for the prevention, we did it by:-

- Suricata IPS:-

We add the inline:yes line to the configuration file of Suricata to monitor the traffic in real time.



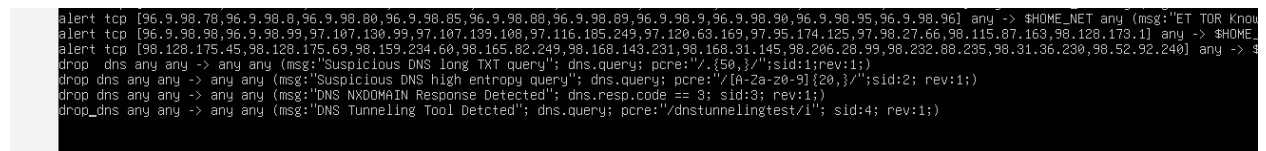
```
GNU nano 7.2 /etc/suricata/suricata.yaml *
format: "[x1] <xd> -- "
# type: json

##
## 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: enp0s8
  inline: yes
  # 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: 254
  # 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
```

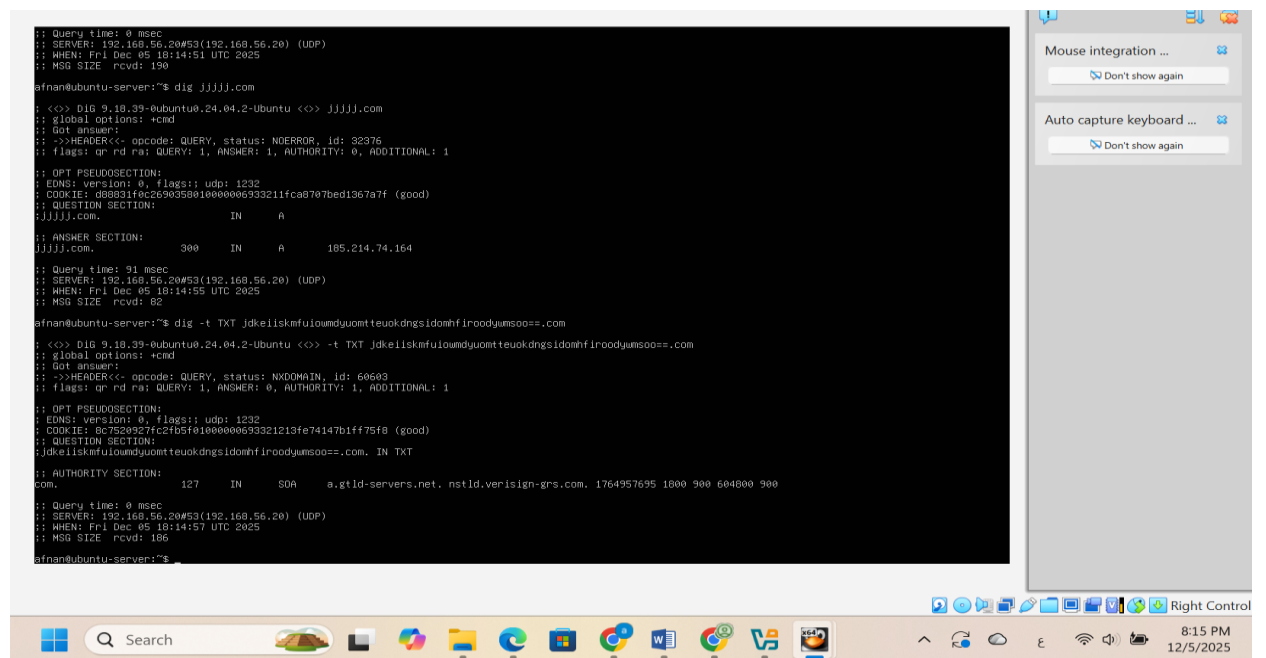
Figure 36: add IPS inline line

And we convert the alert in the rules to drop.



```
alert tcp [96.9.98.78,96.9.98.8,96.9.98.80,96.9.98.85,96.9.98.88,96.9.98.89,96.9.98.9,96.9.98.90,96.9.98.95,96.9.98.96] any -> $HOME_NET any (msg:"ET TOR Know
alert tcp [96.9.98.98,96.9.98.99,97.107.130.99,97.107.139.108,97.116.185.249,97.120.63.169,97.95.174.125,97.98.27.66,98.115.87.163,98.128.173.1] any -> $HOME
alert tcp [98.128.175.45,98.128.175.69,98.159.234.60,98.165.82.249,98.168.143.231,98.168.31.145,98.206.28.99,98.232.88.235,98.31.36.230,98.52.92.240] any -> $
drop dns any any -> any any (msg:"Suspicious DNS long TXT query"; dns.query; pcre:"/.{50,}/";sid:1;rev:1)
drop dns any any -> any any (msg:"Suspicious DNS high entropy query"; dns.query; pcre:"/[A-Za-z0-9]{20,}/";sid:2; rev:1;)
drop dns any any -> any any (msg:"DNS NXDOMAIN Response Detected"; dns.resp.code == 3; sid:3; rev:1;)
drop_dns any any -> any any (msg:"DNS Tunneling Tool Dctcted"; dns.query; pcre:"/dnstunnelingtest/i"; sid:4; rev:1;)
```

Figure 37: convert alert to drop



```
Query time: 0 msec
SERVER: 192.168.56.2053(192.168.56.20) (UDP)
WHEN: Fri Dec 05 18:14:51 UTC 2025
MSG SIZE rcvd: 190

afnan@ubuntu-server:~$ dig jijiji.com

; <> DIG 9.10.39-ubuntu0.24.04.2-Ubuntu <> jijiji.com
; global options: +cdm
; Got answer:
; ->HEADER<<< opcode: QUERY, status: NOERROR, id: 32376
; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: d88831f0c269035801000006933211fca8707bed1367a7f (good)
; QUESTION SECTION:
; jijiji.com.                IN      A
; ANSWER SECTION:
;          300      IN      A      185.214.74.164
; jijiji.com.

; Query time: 91 msec
SERVER: 192.168.56.2053(192.168.56.20) (UDP)
WHEN: Fri Dec 05 18:14:55 UTC 2025
MSG SIZE rcvd: 82

afnan@ubuntu-server:~$ dig -t TXT jdkellskmfuowndyuoimt teukdngsidomhfiroodyumsoo=.com

; <> DIG 9.10.39-ubuntu0.24.04.2-Ubuntu <> -t TXT jdkellskmfuowndyuoimt teukdngsidomhfiroodyumsoo=.com
; global options: +cdm
; Got answer:
; ->HEADER<<< opcode: QUERY, status: NXDOMAIN, id: 60603
; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 8c7520a27fc2fb5f0100000693321213fe74147b1ff75f8 (good)
; QUESTION SECTION:
; jdkellskmfuowndyuoimt teukdngsidomhfiroodyumsoo=.com. IN TXT
; AUTHORITY SECTION:
;          127      IN      SOA     a.gtld-servers.net. nstld.verisign-grs.com. 1764957695 1800 900 604800 900
; com.

; Query time: 0 msec
SERVER: 192.168.56.2053(192.168.56.20) (UDP)
WHEN: Fri Dec 05 18:14:57 UTC 2025
MSG SIZE rcvd: 106

afnan@ubuntu-server:~$
```

Figure 38: send requests

```

712 [1060 - Suricata-Main] 2025-12-05 18:40:38 Info: runmodes: enp0s8: creating 3 threads
713 [1060 - Suricata-Main] 2025-12-05 18:40:38 Info: unix-manager: unix socket '/var/suricata-command.socket'
714 [1060 - Suricata-Main] 2025-12-05 18:40:38 Notice: threads: Threads created -> W: 3 FM: 1 FR: 1 Engine started.
715 [1044 - Suricata-Main] 2025-12-05 19:53:58 Notice: suricata: This is Suricata version 7.0.3 RELEASE running in SYSTEM mode
716 [1044 - Suricata-Main] 2025-12-05 19:53:58 Info: cpu: CPUs/cores online: 3
717 [1044 - Suricata-Main] 2025-12-05 19:53:58 Info: suricata: Setting engine mode to IDS mode by default
718 [1044 - Suricata-Main] 2025-12-05 19:53:58 Info: exception-policy: master exception-policy set to: auto
719 [1044 - Suricata-Main] 2025-12-05 19:53:58 Info: ioctl: enp0s8: MTU 1500
720 [1054 - Suricata-Main] 2025-12-05 19:53:58 Info: conf: Running in live mode, activating unix socket
721 [1054 - Suricata-Main] 2025-12-05 19:53:58 Info: logopenfile: fast output device (regular) initialized: fast.log
722 [1054 - Suricata-Main] 2025-12-05 19:53:58 Info: logopenfile: eve-log output device (regular) initialized: /var/log/suricata/eve.json
723 [1054 - Suricata-Main] 2025-12-05 19:53:58 Info: logopenfile: stats output device (regular) initialized: stats.log
724 [1054 - Suricata-Main] 2025-12-05 19:53:58 Error: detect-parse: An invalid action "0alert" was given
725 [1054 - Suricata-Main] 2025-12-05 19:53:58 Error: detect: error parsing signature "0alert ip any any -> any any (msg:"SURICATA Applayer Mismatch protocol both directions"; flow:established; app-layer-event:applayer_mismatch_protocol_both_directions; flowint:applayer.anomaly.count,+,1; classtype:protocol-command-decode; sid:2260000; rev:1)" from file /var/lib/suricata/rules/suricata.rules at line 1
726 [1054 - Suricata-Main] 2025-12-05 19:54:03 Error: detect-parse: unknown rule keyword 'dns.resp.code == 3'.
727 [1054 - Suricata-Main] 2025-12-05 19:54:03 Error: detect: error parsing signature "alert dns any any -> any any (msg:"DNS NXDOMAIN Response Detected"; dns.resp.code == 3; sid:3; rev:1)" from file /var/lib/suricata/rules/suricata.rules at line 62346
728 [1054 - Suricata-Main] 2025-12-05 19:54:03 Info: detect: 1 rule files processed. 46530 rules successfully loaded, 2 rules failed, 0
729 [1054 - Suricata-Main] 2025-12-05 19:54:03 Info: threshold-config: Threshold config parsed: 0 rule(s) found
730 [1054 - Suricata-Main] 2025-12-05 19:54:03 Info: detect: 46533 signatures processed. 1059 are IP-only rules, 4422 are inspecting packet payload, 40823 inspect application layer, 108 are decoder event only
731 [1054 - Suricata-Main] 2025-12-05 19:54:13 Info: runmodes: enp0s8: creating 3 threads
732 [1054 - Suricata-Main] 2025-12-05 19:54:13 Info: unix-manager: unix socket '/var/suricata-command.socket'
733 [1054 - Suricata-Main] 2025-12-05 19:54:14 Notice: threads: Threads created -> W: 3 FM: 1 FR: 1 Engine started.
734 [1054 - Suricata-Main] 2025-12-05 20:13:12 Notice: suricata: Signal Received. Stopping engine.
735 [1054 - Suricata-Main] 2025-12-05 20:13:12 Info: suricata: time elapsed 1139.075s
736 [1054 - Suricata-Main] 2025-12-05 20:13:13 Info: counters: Alerts: 2
737 [1054 - Suricata-Main] 2025-12-05 20:13:13 Notice: device: enp0s8: packets: 186, drops: 0 (0.00%), invalid chksum: 0
738 [2328 - Suricata-Main] 2025-12-05 20:13:14 Notice: suricata: This is Suricata version 7.0.3 RELEASE running in SYSTEM mode
739 [2328 - Suricata-Main] 2025-12-05 20:13:14 Info: cpu: CPUs/cores online: 3
740 [2328 - Suricata-Main] 2025-12-05 20:13:14 Info: suricata: Setting engine mode to IDS mode by default
741 [2328 - Suricata-Main] 2025-12-05 20:13:14 Info: exception-policy: master exception-policy set to: auto
742 [2328 - Suricata-Main] 2025-12-05 20:13:14 Info: ioctl: enp0s8: MTU 1500
743 [2330 - Suricata-Main] 2025-12-05 20:13:14 Info: conf: Running in live mode, activating unix socket
744 [2330 - Suricata-Main] 2025-12-05 20:13:14 Info: logopenfile: fast output device (regular) initialized: fast.log
745 [2330 - Suricata-Main] 2025-12-05 20:13:14 Info: logopenfile: eve-log output device (regular) initialized: /var/log/suricata/eve.json
746 [2330 - Suricata-Main] 2025-12-05 20:13:14 Info: logopenfile: stats output device (regular) initialized: stats.log
747 [2330 - Suricata-Main] 2025-12-05 20:13:18 Error: detect-parse: unknown rule keyword 'dns.resp.code == 3'.
748 [2330 - Suricata-Main] 2025-12-05 20:13:18 Error: detect: error parsing signature "drop dns any any -> any any (msg:"DNS NXDOMAIN Response Detected"; dns.resp.code == 3; sid:3; rev:1)" from file /var/lib/suricata/rules/suricata.rules at line 62346
749 [2330 - Suricata-Main] 2025-12-05 20:13:18 Info: detect: 1 rule files processed. 46531 rules successfully loaded, 1 rules failed, 0
750 [2330 - Suricata-Main] 2025-12-05 20:13:18 Info: threshold-config: Threshold config parsed: 0 rule(s) found
751 [2330 - Suricata-Main] 2025-12-05 20:13:18 Info: detect: 46534 signatures processed. 1059 are IP-only rules, 4422 are inspecting packet payload, 40823 inspect application layer, 108 are decoder event only
752 [2330 - Suricata-Main] 2025-12-05 20:13:28 Info: runmodes: enp0s8: creating 3 threads
753 [2330 - Suricata-Main] 2025-12-05 20:13:28 Info: unix-manager: unix socket '/var/suricata-command.socket'
754 [2330 - Suricata-Main] 2025-12-05 20:13:28 Notice: threads: Threads created -> W: 3 FM: 1 FR: 1 Engine started.
[Fri Dec 05 20:15:00] afnansensor@sensor:~$

```

Figure 39: Suricata Prevention

- Firewall:-

Limited requests, only 10 requests per minute and block requests from a specific source (192.168.56.20).

```

No VM guests are running outdated hypervisor (qemu) binaries on this host.
[Fri Dec 05 20:28:02] afnansensor@sensor:~$ sudo systemctl enable --now firewallld
[Fri Dec 05 20:28:28] afnansensor@sensor:~$ sudo firewall-cmd --state
running
[Fri Dec 05 20:28:42] afnansensor@sensor:~$ sudo firewall-cmd --zone=public --add-port=53/udp --permanent
success
[Fri Dec 05 20:28:57] afnansensor@sensor:~$ sudo firewall-cmd --zone=public --add-port=53/tcp --permanent
Error: INVALID_PORT: bad port (most likely missing protocol), correct syntax is portid[-portid]/protocol
[Fri Dec 05 20:29:04] afnansensor@sensor:~$ sudo firewall-cmd --zone=public --add-port=53/tcp --permanent
success
[Fri Dec 05 20:29:11] afnansensor@sensor:~$ sudo firewall-cmd --reload
success
[Fri Dec 05 20:29:29] afnansensor@sensor:~$ sudo firewall-cmd --zone=public --add-rich-rule='rule family="ipv4" service name="dns" limit value="10/m" accept' --permanent
success
[Fri Dec 05 20:31:30] afnansensor@sensor:~$ sudo firewall-cmd --reload
success
[Fri Dec 05 20:31:39] afnansensor@sensor:~$ sudo firewall-cmd --zone=public --add-rich-rule='rule family="ipv4" source address="192.168.56.20" service name="dns" limit value="10/m" accept' --permanent
Error: INVALID_RULE: unknown element source
[Fri Dec 05 20:33:01] afnansensor@sensor:~$ sudo firewall-cmd --zone=public --add-rich-rule='rule family="ipv4" source address="192.168.56.20" service name="dns" limit value="10/m" accept' --permanent
success
[Fri Dec 05 20:33:24] afnansensor@sensor:~$

```

Figure 40: Firewall settings

Challenges:-

- The difficulty of writing rules and scripts manually and the difficulty of reconciling them completely.
- The Zeek tool deletes files when the system is shutdown.
- Difficulty reading the eve.json file in Suricata tool to monitoring the network traffic.

Recommendations:-

- We observed that adding a simple dashboard to display results and alerts is preferable. This would provide a faster way to monitor traffic and alerts, rather than relying solely on log files. It would facilitate monitoring during any attack, making the system easier to use and understand.
- Suricata's rules and Zeek's scripts must be continuously developed to improve their accuracy, avoid false positive alerts, and easily distinguish between legitimate and malicious traffic.
- Running Zeek for a longer period provides a clearer picture of legitimate network traffic, making the detection of any unusual activity easier and more accurate.
- After testing, we observed that recording observations at each stage of the process helped us track errors and improve results. Therefore, it is recommended to document steps and modifications regularly to facilitate future development and review.

Conclusion:-

This report highlights the importance of combining behavioral analysis and signature detection in a network security environment. Zeek provides the ability to monitor and analyze DNS traffic and extract anomalous behavior, while Suricata acts as a real-time detection layer, creating custom rules and implementing prevention measures. Attack testing was conducted using DNS Tunneling tools on a Kali machine, while Sensor VM analyzed the traffic and generated appropriate alerts.

An integrated environment was provided for proof of concept and documentation of results. The experiment confirms that combining the two mechanisms provides a higher level of detection accuracy and reduces the chances of successful DNS attacks.

References:-

[Suricata rules](#)

[Zeek Scripts](#)

[Git Repo](#)

