TASK 4

Cyber Security Internship Report

DIGITAL EMPOWERMENT NETWORK

Presented to:

Digital Empowerment Network

Presented by:

Ali Aitazaz [LinkedIn]

+92 332 5890142

www.digitalempowermentnetwork.org

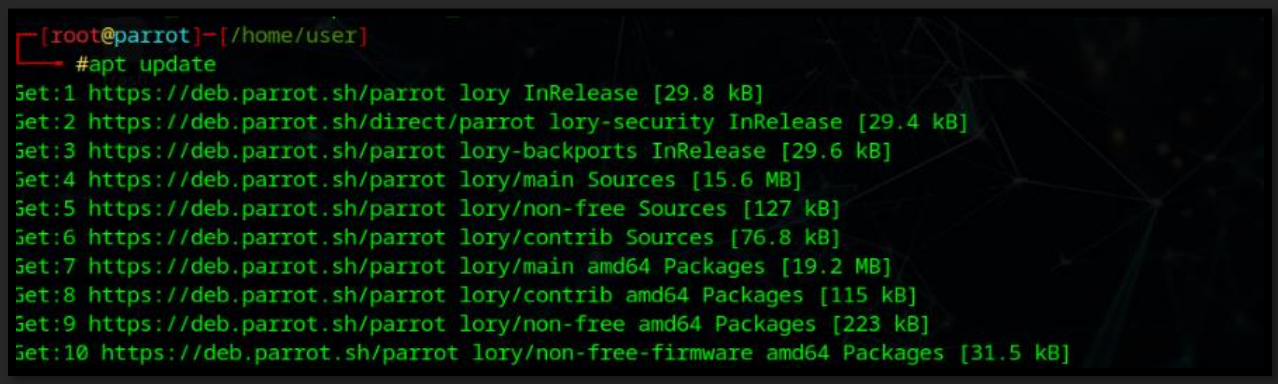
Contents

```
┌─[root@parrot]─[/home/user]
└──╼ #wget https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.8.0-1_amd64.deb && sudo WAZUH_MANAGER='192
.168.56.143' WAZUH_AGENT_GROUP='default' WAZUH_AGENT_NAME='MY-PARROT' dpkg -i ./wazuh-agent_4.8.0-1_amd64.deb
--2024-07-20 07:51:49--  https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.8.0-1_amd64.deb
Resolving packages.wazuh.com (packages.wazuh.com)... 52.84.66.43, 52.84.66.46, 52.84.66.104, ...
Connecting to packages.wazuh.com (packages.wazuh.com)|52.84.66.43|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10273502 (9.8M) [binary/octet-stream]
Saving to: 'wazuh-agent_4.8.0-1_amd64.deb'

wazuh-agent_4.8.0-1_amd64.deb   100%[=================================================>]   9.80M  1.27MB/s    in 10s

2024-07-20 07:52:00 (987 KB/s) - 'wazuh-agent_4.8.0-1_amd64.deb' saved [10273502/10273502]

Selecting previously unselected package wazuh-agent.
(Reading database ... 535376 files and directories currently installed.)
Preparing to unpack .../wazuh-agent_4.8.0-1_amd64.deb ...
Unpacking wazuh-agent (4.8.0-1) ...
Setting up wazuh-agent (4.8.0-1) ...
┌─[root@parrot]─[/home/user]
└──╼ #sudo systemctl daemon-reload
sudo systemctl enable wazuh-agent
sudo systemctl start wazuh-agent
Created symlink /etc/systemd/system/multi-user.target.wants/wazuh-agent.service → /lib/systemd/system/wazuh-agent.service.
┌─[root@parrot]─[/home/user]
└──╼ #
```

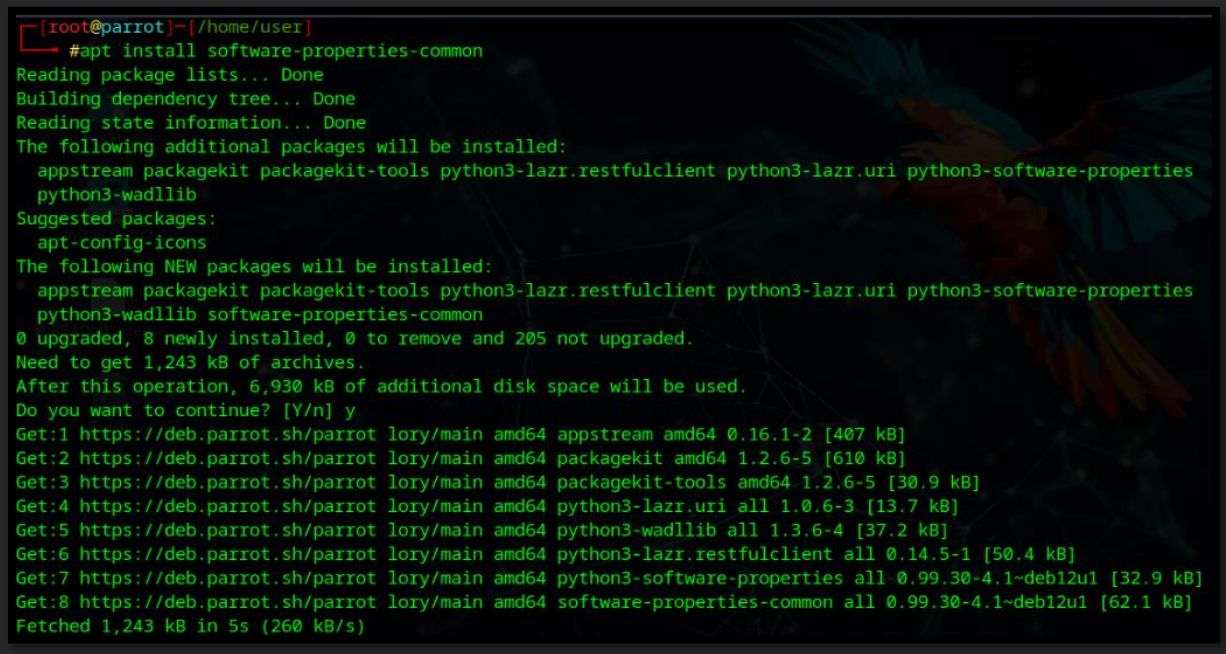
```
┌─[root@parrot]─[/home/user]
└──╼ #ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:f3:8f:c5 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.56.149/24 brd 192.168.56.255 scope global dynamic noprefixroute ens33
       valid_lft 1790sec preferred_lft 1790sec
    inet6 fe80::bebc:8028:7127:3cd1/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Implement a software based firewall and connect it with wazuh

- To start, we will open up our PARROT endpoint and run the following commands to update and upgrade our PARROT.



- Next is to import the [Open Information Security Foundation (OISF)] repository from the Suricata server. Run the following commands to do so and hit Enter to



```
┌[✗]-[root@parrot]-[/home/user]
└─► #echo "deb http://oisf.net/suricata/stable/ suricata-stable main" >> /etc/apt/sources.list
┌[root@parrot]-[/home/user]
└─► #
```

- Next is to update the repository

```
┌[root@parrot]-[/home/user]
└─► #apt update
Get:1 https://deb.parrot.sh/parrot lory InRelease [29.8 kB]
Get:2 https://deb.parrot.sh/direct/parrot lory-security InRelease [29.4 kB]
Get:3 https://deb.parrot.sh/parrot lory-backports InRelease [29.6 kB]
Get:4 https://deb.parrot.sh/parrot lory/main Sources [15.6 MB]
Get:5 https://deb.parrot.sh/parrot lory/non-free Sources [127 kB]
Get:6 https://deb.parrot.sh/parrot lory/contrib Sources [76.8 kB]
Get:7 https://deb.parrot.sh/parrot lory/main amd64 Packages [19.2 MB]
Get:8 https://deb.parrot.sh/parrot lory/contrib amd64 Packages [115 kB]
Get:9 https://deb.parrot.sh/parrot lory/non-free amd64 Packages [223 kB]
Get:10 https://deb.parrot.sh/parrot lory/non-free-firmware amd64 Packages [31.5 kB]
```

- Now we will install Suricata with the following commands.

```
┌[✗]-[root@parrot]-[/home/user]
└─► #apt install suricata -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libhiredis0.14 libhtp2 libhyperscan5 libnetfilter-log1 suricata-update
Suggested packages:
  libtcmalloc-minimal4
Recommended packages:
  snort-rules-default
The following NEW packages will be installed:
  libhiredis0.14 libhtp2 libhyperscan5 libnetfilter-log1 suricata suricata-update
0 upgraded, 6 newly installed, 0 to remove and 205 not upgraded.
Need to get 4,633 kB of archives.
After this operation, 23.7 MB of additional disk space will be used.
Get:1 https://deb.parrot.sh/parrot lory/main amd64 libhyperscan5 amd64 5.4.0-2 [2,489 kB]
Get:2 https://deb.parrot.sh/parrot lory/main amd64 libhiredis0.14 amd64 0.14.1-3 [35.9 kB]
Get:3 https://deb.parrot.sh/parrot lory/main amd64 libhtp2 amd64 1:0.5.42-1 [70.4 kB]
Get:4 https://deb.parrot.sh/parrot lory/main amd64 libnetfilter-log1 amd64 1.0.2-3 [13.4 kB]
Get:5 https://deb.parrot.sh/parrot lory/main amd64 suricata amd64 1:6.0.10-1 [1,963 kB]
Get:6 https://deb.parrot.sh/parrot lory/main amd64 suricata-update amd64 1.2.7-1 [61.4 kB]
Fetched 4,633 kB in 15s (306 kB/s)
```

- When we run the following command we see that our Suricata is installed and we can see the version of the Suricata.

```
┌──[root@parrot]─[/home/user]
└─── #suricata --build-info
This is Suricata version 6.0.10 RELEASE
Features: NFQ PCAP_SET_BUFF AF_PACKET HAVE_PACKET_FANOUT LIBCAP_NG LIBNET1.1 HAVE_HTP_URI_NORMALIZE_HOOK PCRE_JIT HAVE_NSS HAV
E_LUA HAVE_LUAJIT HAVE_LIBJANSSON TLS TLS_C11 MAGIC RUST
SIMD support: none
Atomic intrinsics: 1 2 4 8 byte(s)
64-bits, Little-endian architecture
GCC version 12.2.0, C version 201112
compiled with _FORTIFY_SOURCE=2
L1 cache line size (CLS)=64
thread local storage method: _Thread_local
compiled with LibHTP v0.5.42, linked against LibHTP v0.5.42

Suricata Configuration:
  AF_PACKET support:                       yes
  eBPF support:                            yes
  XDP support:                             yes
  PF_RING support:                         no
  NFQueue support:                         yes
  NFLOG support:                           yes
  IPFW support:                            no
  Netmap support:                          no  using new api: no
  DAG enabled:                             no
  Napatech enabled:                        no
  WinDivert enabled:                       no
```

Configure and create rule on firewall and test it with home lab

- **First changes we are going to do on the YAML file is to change the HOME_NET IP address to the IP address of our PARROT endpoint.**

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

vars:
  # more specific is better for alert accuracy and performance
  address-groups:
    HOME_NET: "[192.168.56.149/24]"
    #HOME_NET: "[192.168.0.0/16]"
    #HOME_NET: "[10.0.0.0/8]"
    #HOME_NET: "[172.16.0.0/12]"
```

- **Next is to go to af-packet and change the interface to that of our PARROT endpoint.**

```
# Linux high speed capture support
af-packet:
  - interface: enp2s1
    # 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:
```

- Next is to change the pcap interface to ours.

```
 GNU nano 7.2                                  /etc/suricata/suricata.yaml
pcap:
  - interface: enp2s1
    # On Linux, pcap will try to use mmap'ed capture and will use "buffer-size"
    # as total memory used by the ring. So set this to something bigger
    # than 1% of your bandwidth.
    #buffer-size: 16777216
    #bpf-filter: "tcp and port 25"
    # Choose checksum verification mode for the interface. At the moment
    # of the capture, some packets may have an invalid checksum due to
    # the checksum computation being offloaded to the network card.
```

- **We change the community-id from False to True.**

```
    # Community Flow ID
    # Adds a 'community_id' field to EVE records. These are meant to give
    # records a predictable flow ID that can be used to match records to
    # output of other tools such as Zeek (Bro).
    #
    # Takes a 'seed' that needs to be same across sensors and tools
    # to make the id less predictable.

    # enable/disable the community id feature.
    community-id: true
    # Seed value for the ID output. Valid values are 0-65535.
    community-id-seed: 0
```

A. BLOCK specific countries (example China, Russia etc.) traffic

- **Install Populate IP sets for China and Russia database:**

```
┌─[✗]─[root@parrot]─[/home/user]
└──╼ #sudo apt-get install geoip-bin geoip-database
sudo apt-get install xtables-addons-common
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
geoip-bin is already the newest version (1.6.12-10).
geoip-database is already the newest version (20240708-1~bpo12+1).
0 upgraded, 0 newly installed, 0 to remove and 204 not upgraded.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libnet-cidr-lite-perl xtables-addons-dkms
Recommended packages:
  linux-headers
The following NEW packages will be installed:
```

```
┌─[root@parrot]─[/home/user]
└──╼ #wget -O cn.zone http://www.ipdeny.com/ipblocks/data/countries/cn.zone
wget -O ru.zone http://www.ipdeny.com/ipblocks/data/countries/ru.zone
--2024-07-20 12:02:01--  http://www.ipdeny.com/ipblocks/data/countries/cn.zone
Resolving www.ipdeny.com (www.ipdeny.com)... 51.15.12.186
Connecting to www.ipdeny.com (www.ipdeny.com)|51.15.12.186|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 135039 (132K) [text/plain]
Saving to: 'cn.zone'

cn.zone                    100%[=========================================>] 131.87K   168KB/s    in 0.8s

2024-07-20 12:02:02 (168 KB/s) - 'cn.zone' saved [135039/135039]

--2024-07-20 12:02:02--  http://www.ipdeny.com/ipblocks/data/countries/ru.zone
Resolving www.ipdeny.com (www.ipdeny.com)... 51.15.12.186
Connecting to www.ipdeny.com (www.ipdeny.com)|51.15.12.186|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 171374 (167K) [text/plain]
Saving to: 'ru.zone'

ru.zone                    100%[=========================================>] 167.36K   264KB/s    in 0.6s

2024-07-20 12:02:04 (264 KB/s) - 'ru.zone' saved [171374/171374]
```

```
┌─[✗]─[root@parrot]─[/home/user]
└──╼ #sudo ipset create china hash:net
sudo ipset create russia hash:net
ipset v7.17: Set cannot be created: set with the same name already exists
ipset v7.17: Set cannot be created: set with the same name already exists
┌─[✗]─[root@parrot]─[/home/user]
└──╼ ## Add China IP ranges to the IP set
while read ip; do
  sudo ipset add china $ip
done < cn.zone

# Add Russia IP ranges to the IP set
while read ip; do
  sudo ipset add russia $ip
done < ru.zone
ipset v7.17: Element cannot be added to the set: it's already added
ipset v7.17: Element cannot be added to the set: it's already added
ipset v7.17: Element cannot be added to the set: it's already added
ipset v7.17: Element cannot be added to the set: it's already added
ipset v7.17: Element cannot be added to the set: it's already added
ipset v7.17: Element cannot be added to the set: it's already added
```

```
┌─[root@parrot]─[/home/user]
└──╼ #sudo iptables -I INPUT -m set --match-set china src -j DROP
sudo iptables -I INPUT -m set --match-set russia src -j DROP
┌─[root@parrot]─[/home/user]
└──╼ #sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  anywhere             anywhere             match-set russia src
DROP       all  --  anywhere             anywhere             match-set china src

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

- **Create a rule file (/etc/suricata/rules/ipset.rules):**

```
  GNU nano 7.2                                /etc/suricata/rules/ipset.rules
# /etc/suricata/rules/ipset.rules
alert ip any any -> any any (msg:"ALERT: Traffic from China IP"; content:"CN"; sid:1000001; rev:1;)
alert ip any any -> any any (msg:"ALERT: Traffic from Russia IP"; content:"RU"; sid:1000002; rev:1;)
```

B. **Create rules whuch restrict the user from specified websites**
C. **Administrator privileges rule**

- **Create a rule file /etc/suricata/rules/local.rules):**

```
  GNU nano 7.2                                          local.rules
alert icmp any any -> $HOME_NET any (msg:"THIS IS AN ICMP Ping"; sid:1; rev:1;)
# /etc/suricata/rules/local.rules
drop http any any -> any any (msg:"DROP Access to Facebook"; content:"facebook.com"; http_host; sid:1000003; rev:1;)
# /etc/suricata/rules/local.rules
alert tcp any any -> any any (msg:"Admin Action Detected"; content:"admin"; http_uri; sid:1000004; rev:1;)
alert tcp any any -> any 22 (msg:"SSH connection attempt"; flow:to_server; sid:100004; rev:1;)
```

- **Include the rule file in suricata.yaml:**

```
  GNU nano 7.2                                          /etc/suricata/suricata.yaml
##
## Configure Suricata to load Suricata-Update managed rules.
##

default-rule-path: /etc/suricata/rules

rule-files:
  - /etc/suricata/rules/ipset.rules
  - /etc/suricata/rules/local.rules

##
## Auxiliary configuration files.
##

classification-file: /etc/suricata/classification.config
reference-config-file: /etc/suricata/reference.config
# threshold-file: /etc/suricata/threshold.config
```

- **Now we will save and close the YAML file then update the Suricata configuration with the following command.**

```
┌─[root@parrot]─[/home/user]
└──╼ #suricata-update
20/7/2024 -- 11:18:41 - <Info> -- Using data-directory /var/lib/suricata.
20/7/2024 -- 11:18:41 - <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml
20/7/2024 -- 11:18:41 - <Info> -- Using /etc/suricata/rules for Suricata provided rules.
20/7/2024 -- 11:18:41 - <Info> -- Found Suricata version 6.0.10 at /usr/bin/suricata.
20/7/2024 -- 11:18:41 - <Info> -- Loading /etc/suricata/suricata.yaml
20/7/2024 -- 11:18:41 - <Info> -- Disabling rules for protocol http2
20/7/2024 -- 11:18:41 - <Info> -- Disabling rules for protocol modbus
20/7/2024 -- 11:18:41 - <Info> -- Disabling rules for protocol dnp3
20/7/2024 -- 11:18:41 - <Info> -- Disabling rules for protocol enip
20/7/2024 -- 11:18:41 - <Info> -- No sources configured, will use Emerging Threats Open
20/7/2024 -- 11:18:41 - <Info> -- Fetching https://rules.emergingthreats.net/open/suricata-6.0.10/emerging.rules.tar.gz.
   2% - 131072/4386085
```

- We will verify the Suricata configuration file by using the built-in test command:

```
┌─[root@parrot]─[/home/user]
└──╼ #sudo suricata -T -c /etc/suricata/suricata.yaml
20/7/2024 -- 12:48:52 - <Info> - Running suricata under test mode
20/7/2024 -- 12:48:52 - <Notice> - This is Suricata version 6.0.10 RELEASE running in SYSTEM mode
20/7/2024 -- 12:48:52 - <Notice> - Configuration provided was successfully loaded. Exiting.
┌─[root@parrot]─[/home/user]
└──╼ #
```

```
┌─[✗]─[root@parrot]─[/home/user]
└──╼ #sudo suricata -T -c /etc/suricata/suricata.yaml -v
20/7/2024 -- 13:01:37 - <Info> - Running suricata under test mode
20/7/2024 -- 13:01:37 - <Notice> - This is Suricata version 6.0.10 RELEASE running in SYSTEM mode
20/7/2024 -- 13:01:37 - <Info> - CPUs/cores online: 2
20/7/2024 -- 13:01:37 - <Info> - fast output device (regular) initialized: fast.log
20/7/2024 -- 13:01:37 - <Info> - eve-log output device (regular) initialized: eve.json
20/7/2024 -- 13:01:37 - <Info> - stats output device (regular) initialized: stats.log
20/7/2024 -- 13:01:37 - <Info> - 2 rule files processed. 6 rules successfully loaded, 0 rules failed
20/7/2024 -- 13:01:37 - <Info> - Threshold config parsed: 0 rule(s) found
20/7/2024 -- 13:01:37 - <Info> - 6 signatures processed. 1 are IP-only rules, 2 are inspecting packet payload, 2 inspect appli
cation layer, 0 are decoder event only
20/7/2024 -- 13:01:38 - <Notice> - Configuration provided was successfully loaded. Exiting.
20/7/2024 -- 13:01:38 - <Info> - cleaning up signature grouping structure... complete
┌─[root@parrot]─[/home/user]
└──╼ #
```

Enable and start Suricata and also check its status:

```
┌─[✗]─[root@parrot]─[/home/user]
└──╼ #sudo systemctl enable suricata
Synchronizing state of suricata.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable suricata
Use of uninitialized value $service in hash element at /usr/sbin/update-rc.d line 26, <DATA> line 44.
Use of uninitialized value $service in hash element at /usr/sbin/update-rc.d line 26, <DATA> line 44.
┌─[root@parrot]─[/home/user]
└──╼ #sudo systemctl start suricata
┌─[root@parrot]─[/home/user]
└──╼ #systemctl status suricata
● suricata.service - Suricata IDS/IDP daemon
     Loaded: loaded (/lib/systemd/system/suricata.service; enabled; preset: enabled)
     Active: active (running) since Sat 2024-07-20 12:57:43 UTC; 8min ago
       Docs: man:suricata(8)
             man:suricatasc(8)
             https://suricata-ids.org/docs/
   Main PID: 148234 (Suricata-Main)
      Tasks: 8 (limit: 3409)
     Memory: 46.9M
        CPU: 7.168s
     CGroup: /system.slice/suricata.service
             └─148234 /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml --pidfile /run/suricata.pid

Jul 20 12:57:43 parrot systemd[1]: Starting suricata.service - Suricata IDS/IDP daemon...
Jul 20 12:57:43 parrot suricata[148232]: 20/7/2024 -- 12:57:43 - <Notice> - This is Suricata version 6.0.10 RELEASE running i>
Jul 20 12:57:43 parrot systemd[1]: Started suricata.service - Suricata IDS/IDP daemon.
lines 1-16/16 (END)
```

- Now we are going to verify that our Suricata is working and is able to track and log any threat on our PARROT endpoint.To do that we will first disable Suricata with the following command.

```
┌─[root@parrot]─[/home/user]
└──╼ #sudo ethtool -K ens33 gro off lro off
┌─[root@parrot]─[/home/user]
└──╼ #
```

- Stop the Suricata with the following command.
  sudo systemctl stop suricata
- Remove the Suricata PID file so we can have a clean restart.

```
┌─[root@parrot]─[/home/user]
└──╼ #sudo rm -rf /var/run/suricata.pid
┌─[root@parrot]─[/home/user]
└──╼ #
```

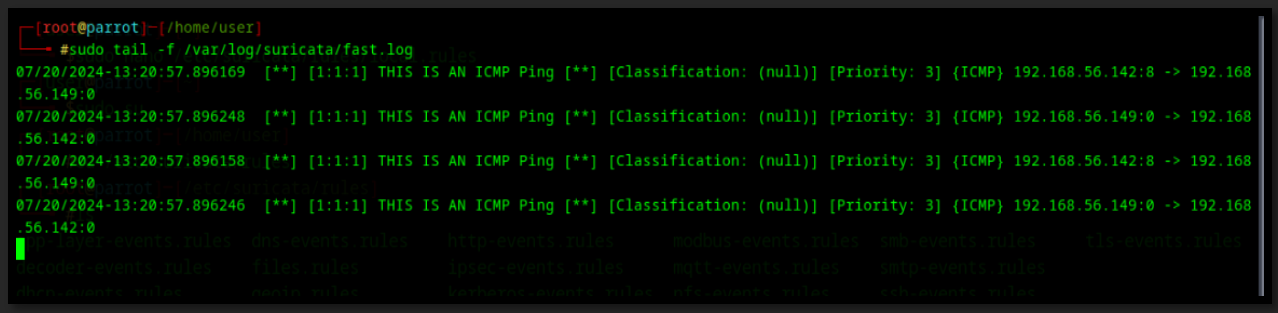
- Now we will run Suricata manually

```
┌─[root@parrot]─[/home/user]
└──╼ #sudo suricata -D -c /etc/suricata/suricata.yaml -i ens33
20/7/2024 -- 13:13:40 - <Notice> - This is Suricata version 6.0.10 RELEASE running in SYSTEM mode
┌─[root@parrot]─[/home/user]
└──╼ #
```

- Now Verifying The Implimentation with The Ping Log Rules:
  Now we will Ping our PARROT endpoint and see how Suricata tracks and logs the pings.I will log into my Kali Linux and preform the Ping on our PARROT endpoint.

```
┌──(kali㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:6a:d7:0b brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.142/24 brd 192.168.56.255 scope global dynamic noprefixroute eth0
       valid_lft 1615sec preferred_lft 1615sec
    inet6 fe80::8167:1e6:cc8e:8c6f/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:ff:cd:ba:58 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever

┌──(kali㉿kali)-[~]
└─$ ping 192.168.56.149
PING 192.168.56.149 (192.168.56.149) 56(84) bytes of data.
64 bytes from 192.168.56.149: icmp_seq=1 ttl=64 time=3.67 ms
64 bytes from 192.168.56.149: icmp_seq=2 ttl=64 time=0.788 ms
64 bytes from 192.168.56.149: icmp_seq=3 ttl=64 time=0.698 ms
64 bytes from 192.168.56.149: icmp_seq=4 ttl=64 time=0.690 ms
64 bytes from 192.168.56.149: icmp_seq=5 ttl=64 time=0.627 ms
64 bytes from 192.168.56.149: icmp_seq=6 ttl=64 time=0.576 ms
^C
── 192.168.56.149 ping statistics ──
6 packets transmitted, 6 received, 0% packet loss, time 5050ms
rtt min/avg/max/mdev = 0.576/1.174/3.667/1.116 ms

┌──(kali㉿kali)-[~]
└─$
```

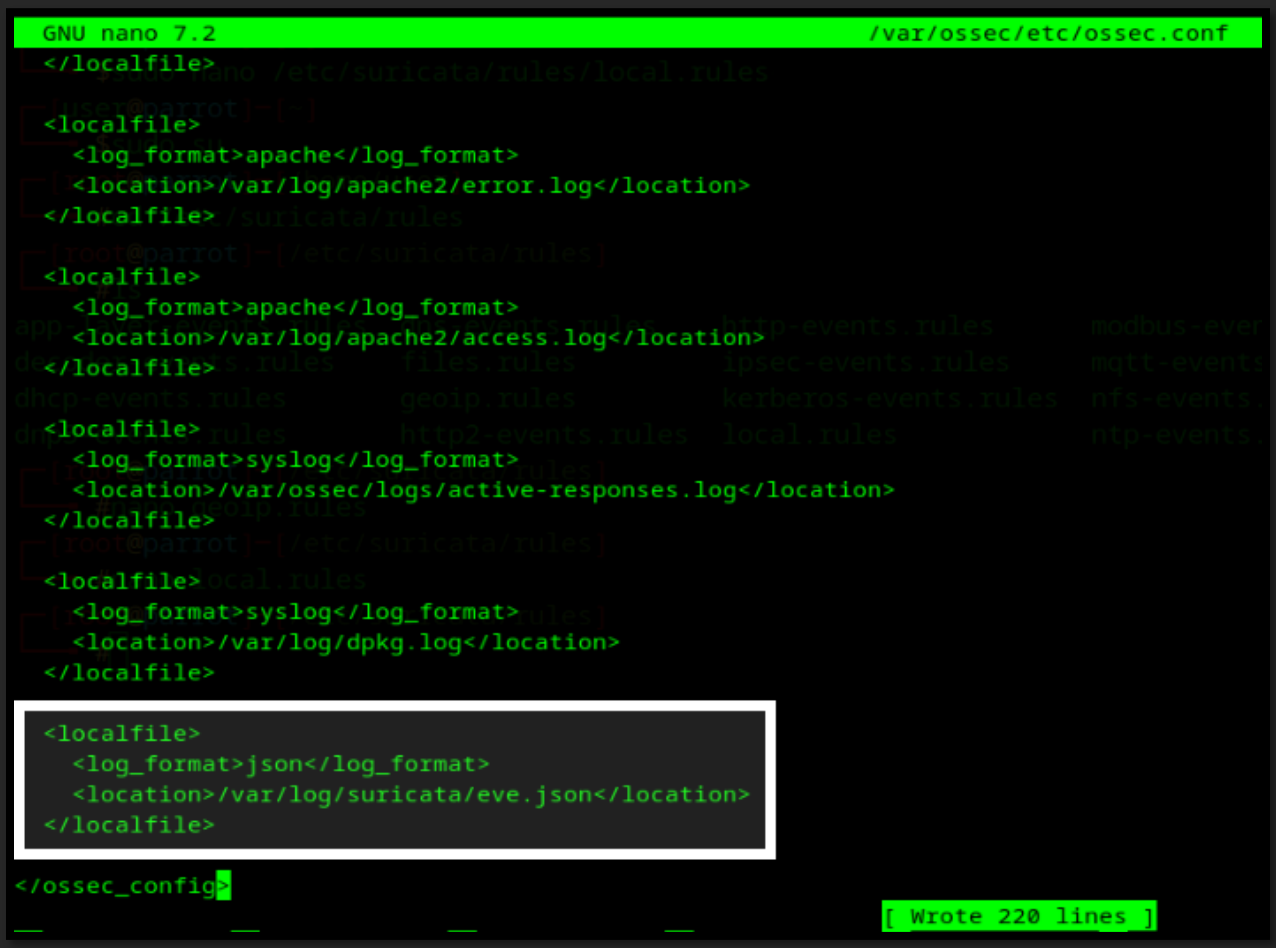
After the ping we will run the following command to view if Suricata logged the ping.



We can see the last two lines from the fast.log files shows a ping that was logged in. The source IP address is from the Kali linux with IP address 192.168.43.136 and the ping is targeted at our PARROT endpoint with IP address 192.168.43.23

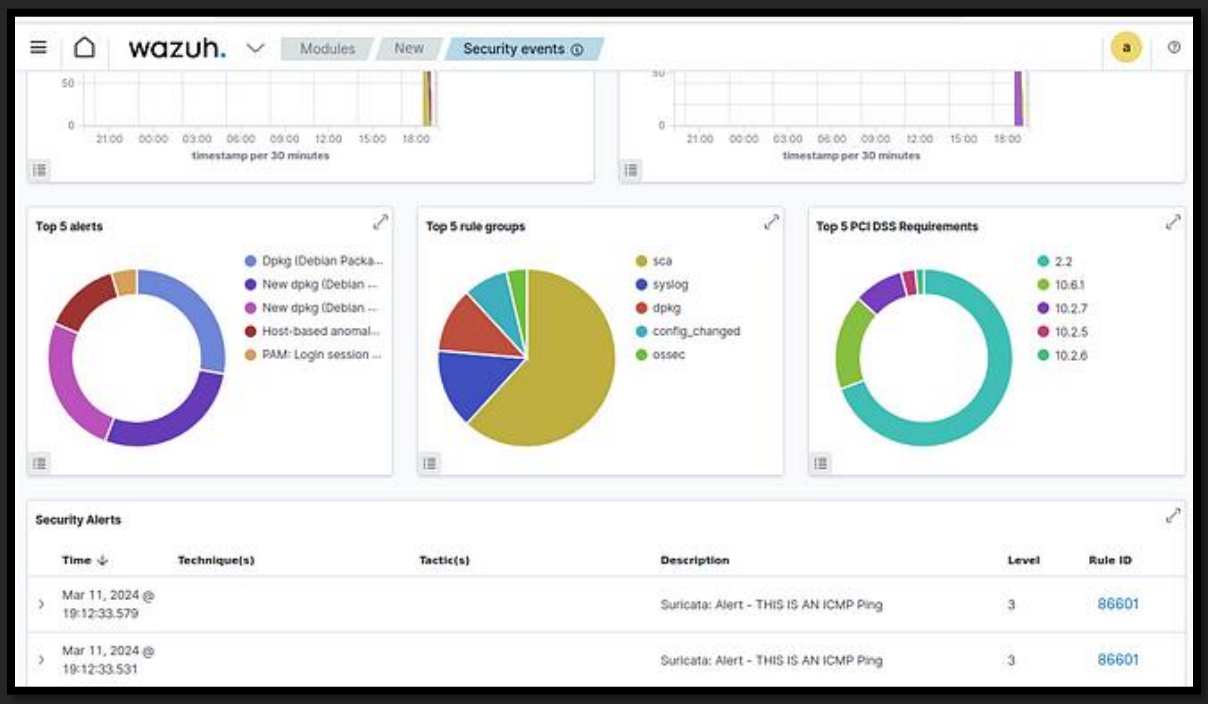
Try to Monitor the files and logs on wazuh

- Now we will integrate our Suricata with the Wazuh we already installed on our endpoint. To do that we will run the following command to open and edit our Wazuh configurations file. Go to the bottom of the file and add the following configuration to the ***/var/ossec/etc/ossec.conf*** file of the Wazuh agent. This allows the Wazuh agent to read the Suricata logs file.



2. create reports put analysis on the report regarding the traffic you have observed

Wazuh Dashboard



About the Company

At Digital Empowerment, we envision a future where youth are equipped with the skills, knowledge, and mindset needed to thrive in a rapidly evolving world. Our mission is to bridge the digital divide, foster leadership development, and enhance academic growth, empowering young minds to realize their full potential.

To achieve this, we offer comprehensive virtual internships across various domains, providing students with invaluable hands-on experience and practical skills essential for success. Furthermore, we are committed to assisting exceptional students in securing positions at prestigious companies, helping to launch their careers and build a brighter future for themselves and future generations.

Digital Empowerment Pakistan

Author’s Note

This document serves as a comprehensive guide on implementing a software-based firewall and integrating it with Wazuh, a security monitoring platform. The procedures outlined are designed to enhance the security of a network by leveraging Suricata, an open-source threat detection engine, to create and manage firewall rules. Additionally, this guide provides detailed steps on how to monitor and analyze network traffic using Wazuh, ensuring that the implementation is effective and secure.

The instructions presented in this document are tailored for cybersecurity practitioners who have a foundational understanding of network security and are seeking to apply advanced configurations to protect their systems. This guide also emphasizes practical implementation, making it a valuable resource for anyone aiming to fortify their network defenses.

The contents are based on practical experience and tested configurations, ensuring that readers can replicate the results in their environments. Readers are encouraged to adapt the provided configurations to their specific network requirements and remain vigilant for updates in security best practices.