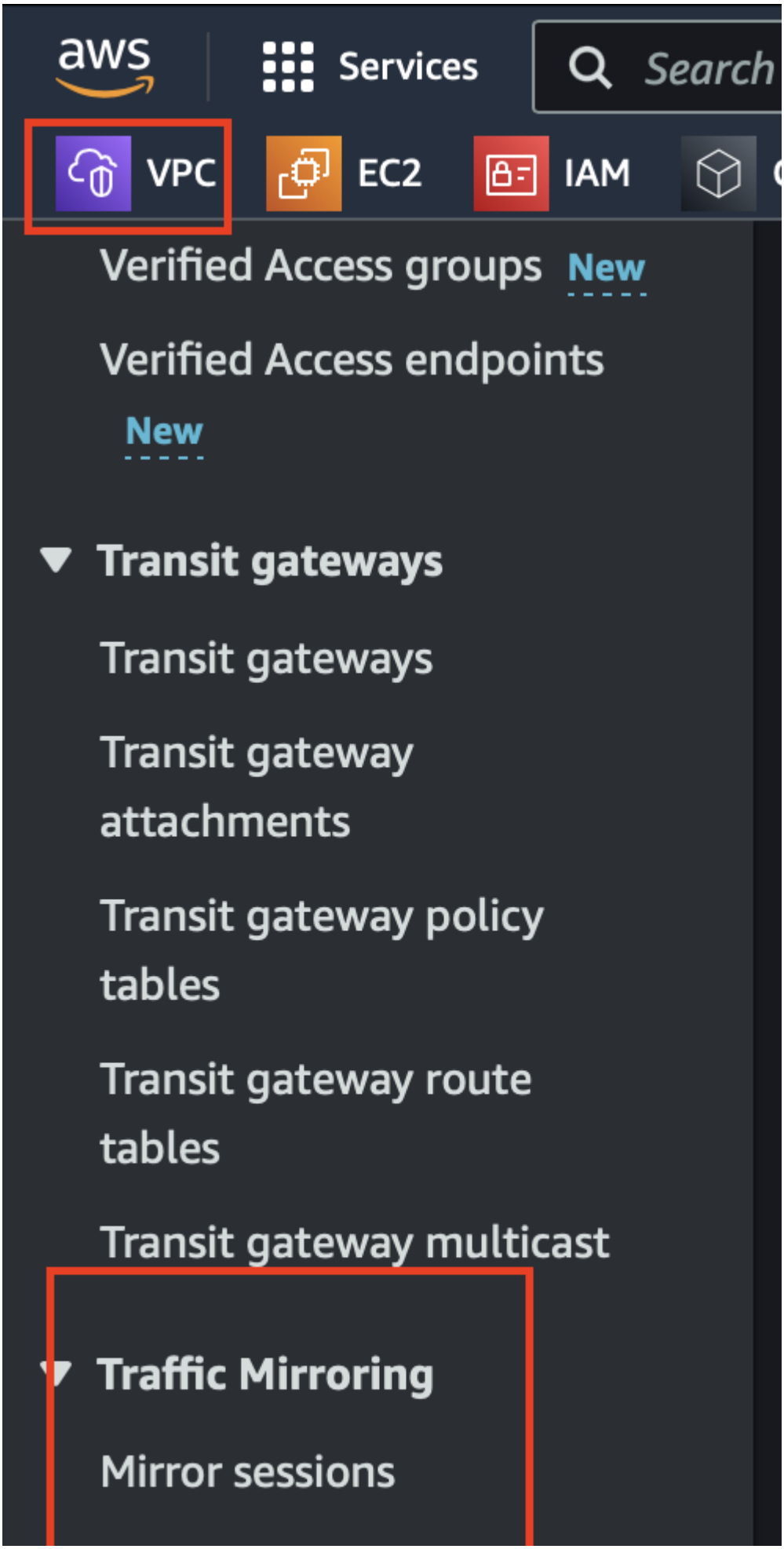
04. AWS SecOnion Lab Setup - Traffic Mirroring for SOC

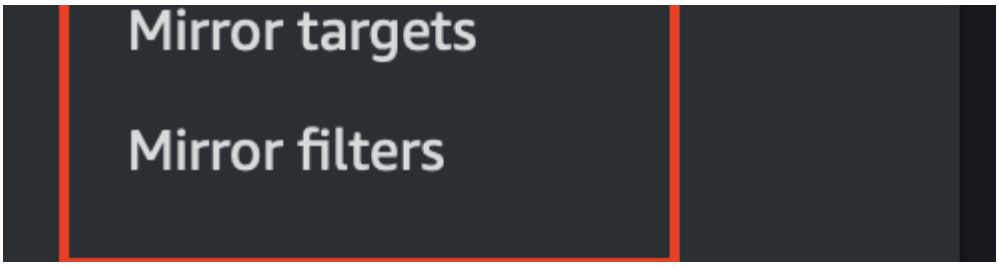
AWS SecOnion Lab Setup - Traffic Mirroring for SOC

Things to do:

Create Mirror Targets
Create Mirror Filters
Create a Mirror Session

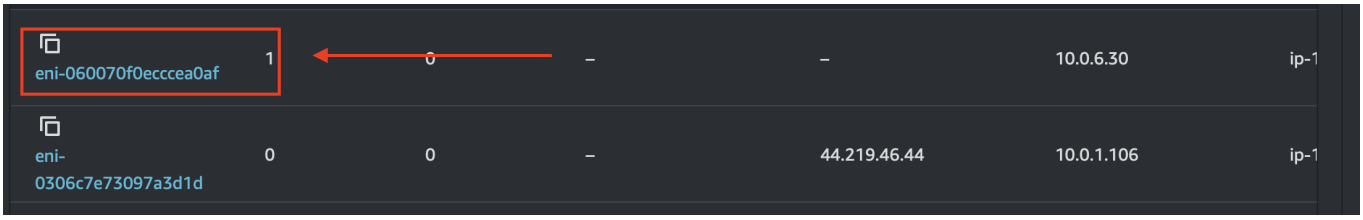
Click on VPC and scroll down on the left pane to "Traffic mirroring"


aws
Services
Search
VPC
EC2
IAM
Verified Access groups New
Verified Access endpoints
New
Transit gateways
Transit gateways
Transit gateway attachments
Transit gateway policy tables
Transit gateway route tables
Transit gateway multicast
Traffic Mirroring
Mirror sessions

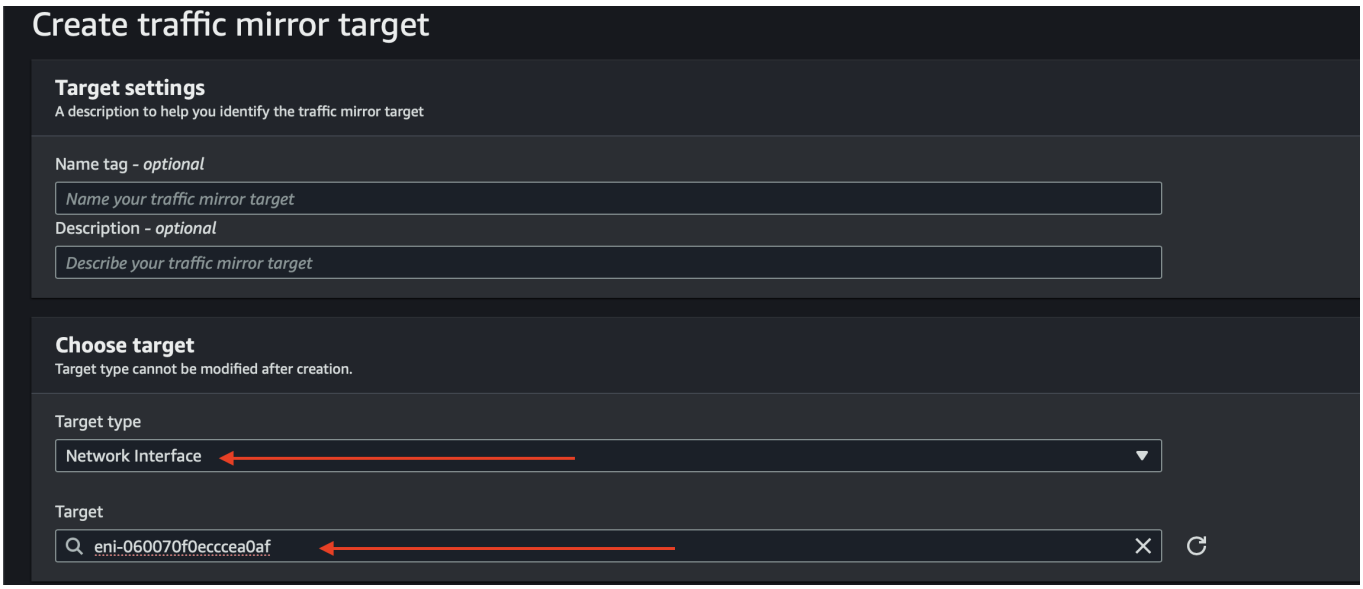


Creating a Mirror Target

Click on Mirror target > Create mirror target
The mirror target will be the Elastic Monitoring/Sniffing interface:



In the "Choose target" section ensure the following is edited:
Target type: Network Interface
Target: Your sniffing interface you created earlier:

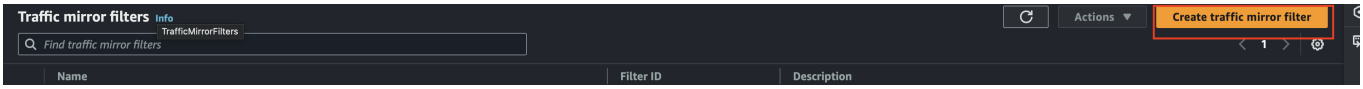


Click Save

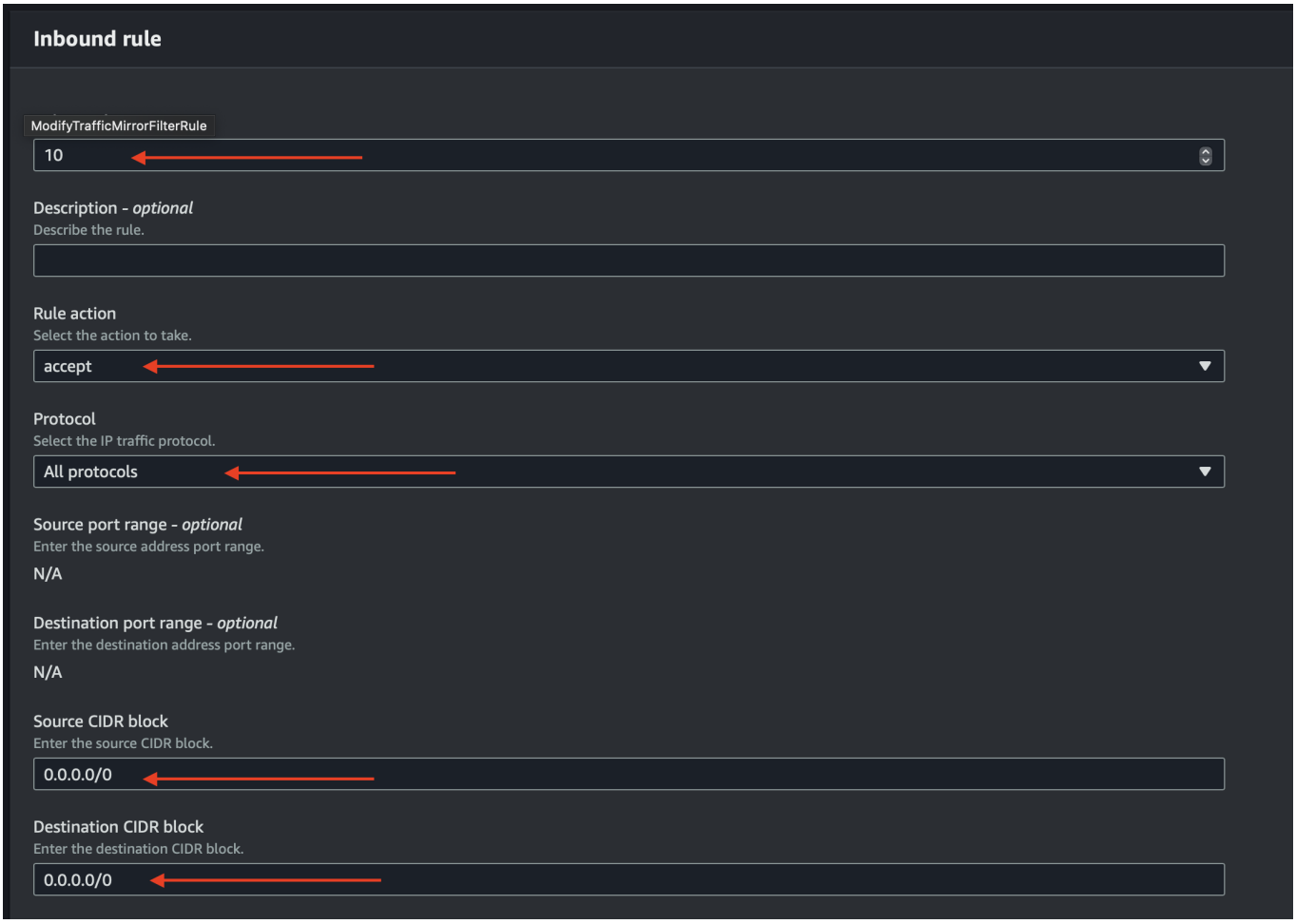
Creating Mirror Filters

Next, in the left pane, click "Mirror filters"

Click Create traffic mirror filters



Inbound Rule:



Outbound Rule:

Save it and then go to next section.

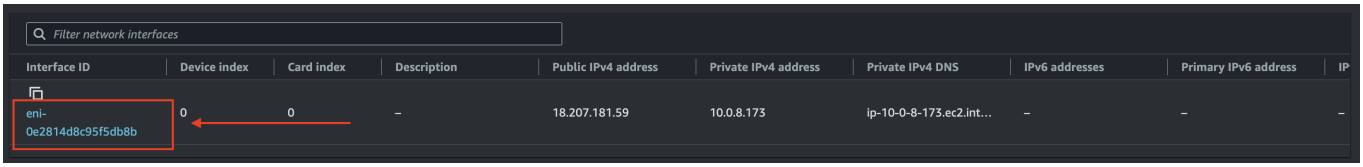
Creating a Mirror Session

Next, in the left pane, click "Mirror sessions"

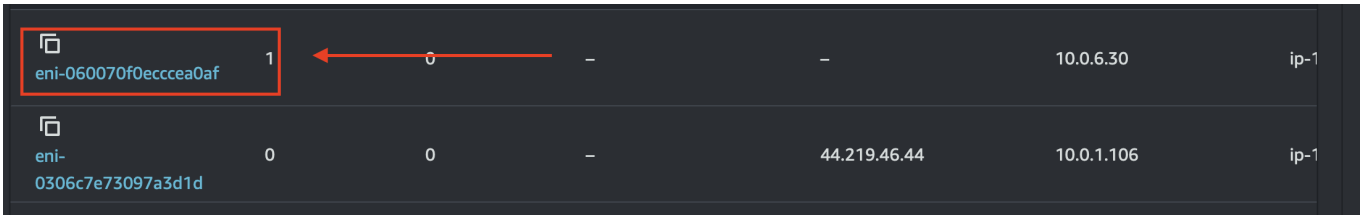
Click "Create a mirror session"

Edits to make:
Tag - Enter a session name if you'll have more than one session

Mirror Source: Linux interface

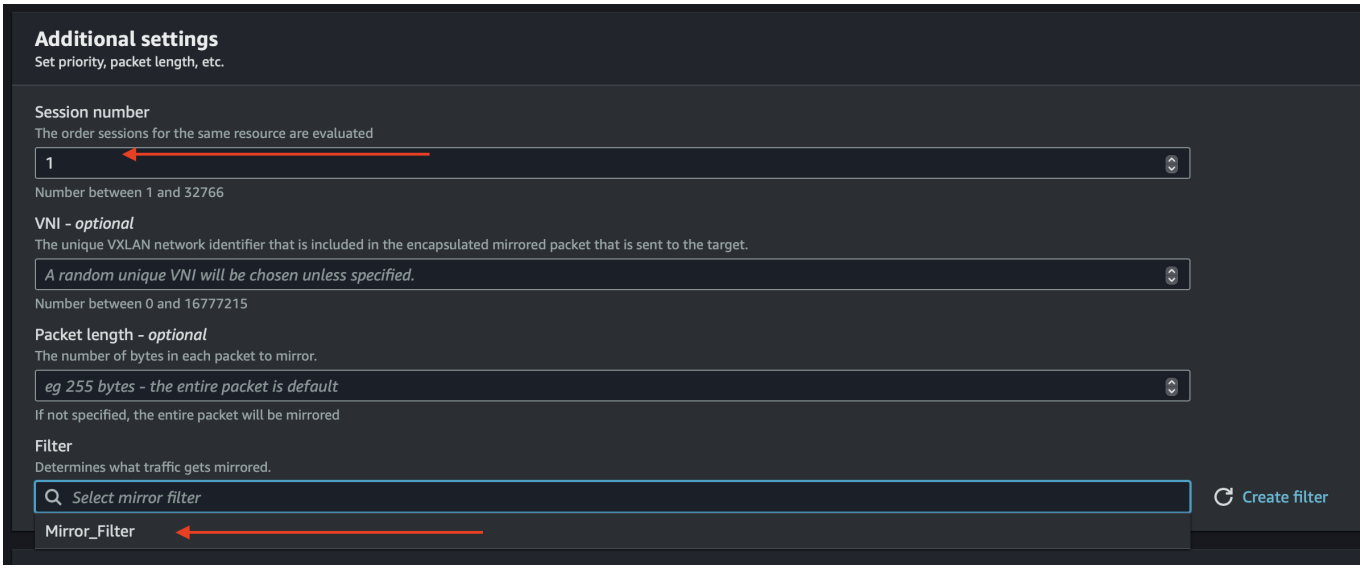


Mirror Target: Sniffing Interface



Session Settings

Additional Settings



Final Checks

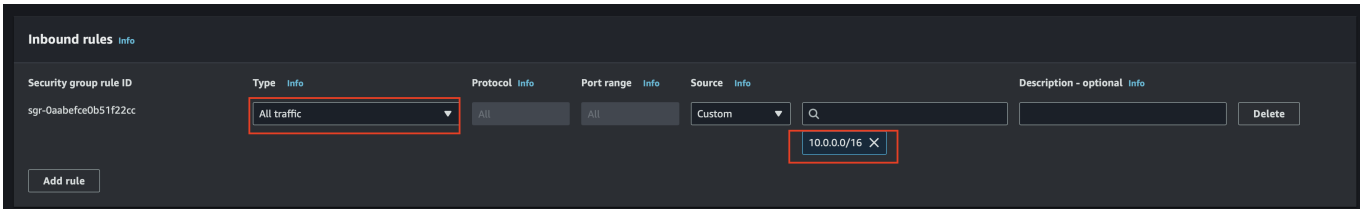
Ensuring that all security groups have an inbound rule that allows all internal traffic through the VPC:

Linux_SG

Manager_SG

Sniffing_SG

Example:

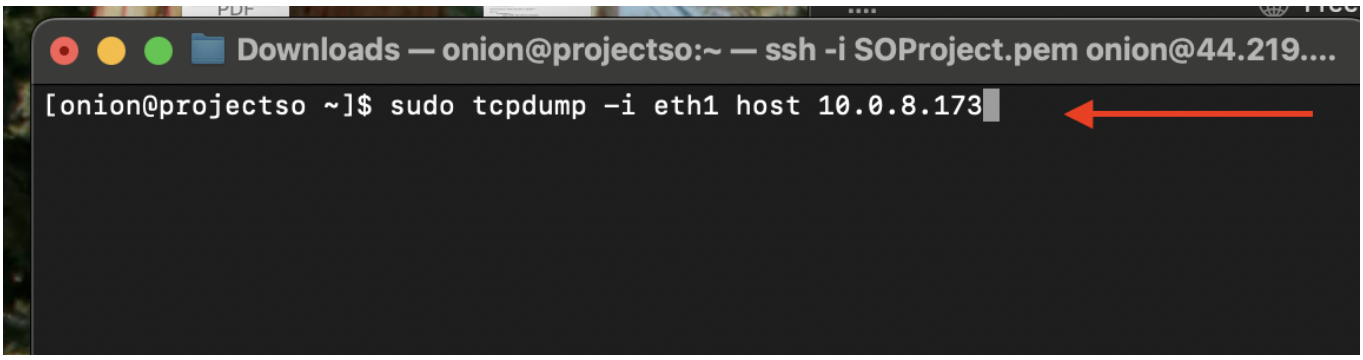


Test the connection by sending traffic to trigger alerts

In the SO cli run the tcpdump command pointing to the "private ip of the Unix/Linux host"

```
sudo tcpdump -i eth1 host <your-linux-host-ip>
```

```
sudo tcpdump -i eth1 host 10.0.8.173
```



Download or curl the traffic generator:

Navigate to tmNIDS on GitHub
https://github.com/3CORESec/testmynids.org

Scroll down on the README section and use the "Interactive Mode" cURL command:

testmynids.org

A website and framework for testing NIDS detection

What is it?

A simple project that aims to centralize testing for detection of malicious events by network intrusion detection systems (NIDS). The tests in this project are built against rulesets, not software. Therefore, if you're using ET Open, coverage for these tests will work.

There are two parts to it:

- **A website** - that is used to hold some tests/files. It will also hold sub-domains and DNS records for when testing via DNS is required.
- **A script** - That runs/simulates interaction with the website or with 3rd party websites, meant to be executed on the client for which you want to test coverage of your NIDS sensor.

Usage

Only requirements are `curl` and `nc`, which should be included in your distribution.

One-liner to download and execute in interactive mode:

```
curl -sSL https://raw.githubusercontent.com/3CORESec/testmynids.org/master/tmNIDS -o /tmp/tn
```

One-liner to download and execute in script mode:

```
curl -sSL https://raw.githubusercontent.com/3CORESec/testmynids.org/master/tmNIDS -o /tmp/tn
```

On the unix/linux host, run the tmNIDS traffic check to create alerts:
(Copy the command below and run it)

```
curl -sSL
https://raw.githubusercontent.com/3CORESec/testmynids.org/master/tmNIDS -o
/tmp/tmNIDS && chmod +x /tmp/tmNIDS && /tmp/tmNIDS
```

```
Downloads — ec2-user@ip-10-0-8-173:~ — ssh -i projects_linux.pem ec2-user@18...
[ec2-user@ip-10-0-8-173 ~]$ curl -sSL https://raw.githubusercontent.com/3CORESec/testmyni
ds.org/master/tmNIDS -o /tmp/tmNIDS && chmod +x /tmp/tmNIDS && /tmp/tmNIDS
```

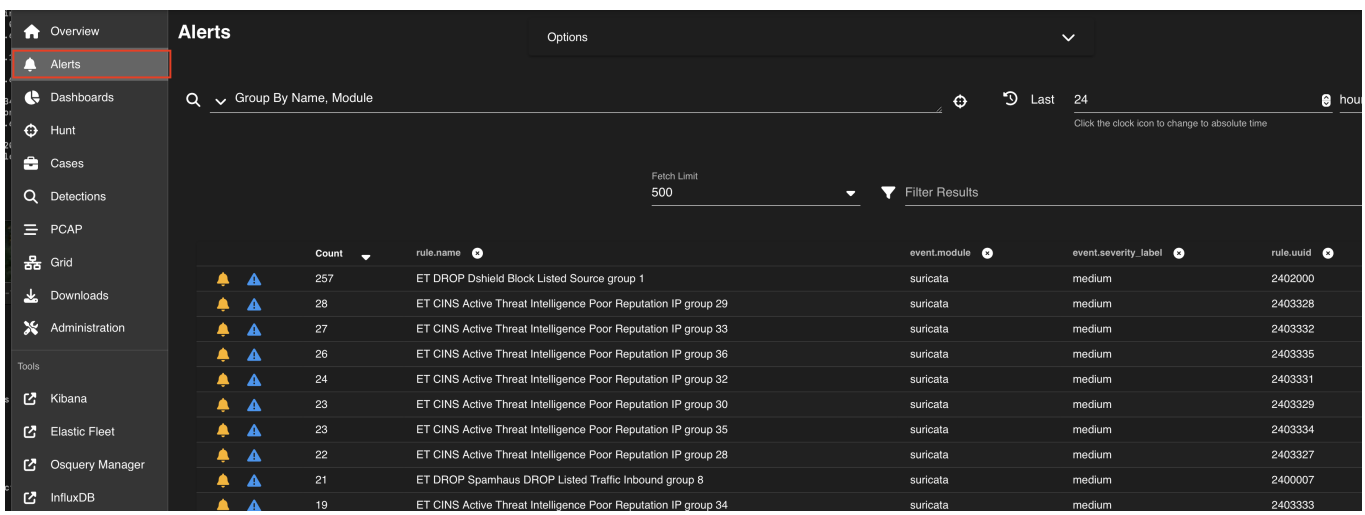
Hit Enter and type "11"

```
Downloads — ec2-user@ip-10-0-8-173:~ — ssh -i projects_linux.pem ec2-user@18.207.181.59 —
 _             _   _ _____ _____   _____
| |           | \ | |_   _|  __ \ / ____|
| |_ _ __ ___ |  \| | | | | |  | | (___
| __|  _   _ \| .   | | | | |  | |\___ \
| |_| | | | | | |\  |_| |_| |__| |____) |
 \__|_| |_| |_|_| \_|_____|_____/|_____/

tmNIDS - NIDS detection tester - @3CORESec
Project: https://github.com/3CORESec/testmynids.org

Choose which test you'd like to run:

 1) Linux UID
 2) HTTP Basic Authentication
 3) HTTP Malware User-Agent
 4) Bad Certificates & CAs
 5) Tor .onion DNS response and known IPs connection
 6) EXE or DLL download over HTTP
 7) PDF download with Embedded File
 8) Simulate SSH Outbound Scan
 9) Miscellaneous domains (TLD's, Sinkhole, DDNS, etc)
10) Anonymous filesharing website
11) External IP Address Lookup
12) URL Shortener
13) Policy Violation - Gaming
14) Adware PUP
15) Malware - Command & Control - Beacon
16) CHAOS! RUN ALL!
17) Quit!
#? 11
```

Hit enter again and check the security onion GUI for traffic and then the cli for alerts:

```
Downloads — onion@projectso:~ — ssh -i SOProject.pem onion@44.219....
vxlan: VXLAN, flags [I] (0x08), vni 10368543
IP 233.211.203.35.bc.googleusercontent.com.54646 > ip-10-0-8-173.ec2.internal.46
372: Flags [S], seq 787794939, win 65535, options [mss 1460], length 0
19:44:52.820500 IP ip-10-0-8-173.ec2.internal.65446 > ip-10-0-6-30.ec2.internal.
vxlan: VXLAN, flags [I] (0x08), vni 10368543
IP ip-10-0-8-173.ec2.internal.46372 > 233.211.203.35.bc.googleusercontent.com.54
646: Flags [R.], seq 0, ack 787794940, win 0, length 0
19:44:56.849334 IP ip-10-0-8-173.ec2.internal.65531 > ip-10-0-6-30.ec2.internal.
vxlan: VXLAN, flags [I] (0x08), vni 10368543
IP 147.185.133.182.50360 > ip-10-0-8-173.ec2.internal.13978: Flags [S], seq 4141
537818, win 65535, options [mss 1460], length 0
19:44:56.849513 IP ip-10-0-8-173.ec2.internal.65531 > ip-10-0-6-30.ec2.internal.
vxlan: VXLAN, flags [I] (0x08), vni 10368543
IP ip-10-0-8-173.ec2.internal.13978 > 147.185.133.182.50360: Flags [R.], seq 0,
ack 4141537819, win 0, length 0
19:45:03.176718 IP ip-10-0-8-173.ec2.internal.65446 > ip-10-0-6-30.ec2.internal.
vxlan: VXLAN, flags [I] (0x08), vni 10368543
IP 198.235.24.149.54356 > ip-10-0-8-173.ec2.internal.commplex-link: Flags [S], s
eq 3497612851, win 65535, options [mss 1460], length 0
19:45:03.176893 IP ip-10-0-8-173.ec2.internal.65446 > ip-10-0-6-30.ec2.internal.
vxlan: VXLAN, flags [I] (0x08), vni 10368543
IP ip-10-0-8-173.ec2.internal.commplex-link > 198.235.24.149.54356: Flags [R.],
seq 0, ack 3497612852, win 0, length 0
```

GUI



Ensure the Unix/Linx client IP is visible in the logs