TryHackMe: Snort Challenge – Live Attacks

The following is a writeup for a room hosted on TryHackMe. The room involves investigating a series of traffic data to stop malicious activity under two different scenarios. I thoroughly enjoyed this room as it helped put my Snort experience to practice.

Scenario 1: Brute-Force

In the first scenario, we are alerted to a brute-force attack in progress. We are tasked with starting Snort in sniffer mode to try and figure out the source of the attack, service, and port. We then need to write an IPS rule and run Snort in IPS mode to stop the brute-force attack.

Let's start off by starting Snort in sniffer mode, we can do this by entering:

- sudo snort -dev -l .

I let this run for about 1 minute and received a total of 14860 packets. Let's now investigate the produced log file:

```
ubuntu@ip-10-10-82-152:~$ sudo snort -r snort.log.1719634459
```

If you investigate the traffic, majority is towards port 80 which could indicate brute forcing a login form or basic authentication:

```
TCP TTL:64 TOS:0x0 ID:55737 IpLen:20 DgmLen:52 DF
***A**** Seq: 0xEE15FBAE  Ack: 0xA569BD18  Win: 0x2F6F  TcpLen: 32
TCP Options (3) => NOP NOP TS: 2500366951 1334869801
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
06/29-04:15:22.999492 10.100.1.242:46104 -> 10.10.82.152:80
TCP TTL:64 TOS:0x0 ID:55738 IpLen:20 DgmLen:52 DF
***A**** Seq: 0xEE15FBAE  Ack: 0xA56A6BE1  Win: 0x2E7B  TcpLen: 32
TCP Options (3) => NOP NOP TS: 2500366951 1334869802
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

RNING: No preprocessors configured for policy 0.
RNING: No preprocessors configured for policy 0.
/29-04:15:23.000054 10.100.1.242:46104 -> 10.10.82.152:80
TCP TTL:64 TOS:0x0 ID:55739 IpLen:20 DgmLen:52 DF
***A**** Seq: 0xEE15FBAE  Ack: 0xA56AA5C4  Win: 0x2F9F  TcpLen: 32
TCP Options (3) => NOP NOP TS: 2500366952 1334869802
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
06/29-04:15:23.000101 10.100.1.242:46104 -> 10.10.82.152:80
TCP TTL:64 TOS:0x0 ID:55740 IpLen:20 DgmLen:52 DF
***A**** Seq: 0xEE15FBAE  Ack: 0xA56ACF15  Win: 0x2F8F  TcpLen: 32
TCP Options (3) => NOP NOP TS: 2500366952 1334869803
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
06/29-04:15:23.009504 10.100.1.242:46104 -> 10.10.82.152:80
TCP TTL:64 TOS:0x0 ID:55741 IpLen:20 DgmLen:100 DF
***AP*** Seq: 0xEE15FBAE  Ack: 0xA56ACF15  Win: 0x2FBF  TcpLen: 32
TCP Options (3) => NOP NOP TS: 2500366961 1334869803
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
06/29-04:15:23.056558 10.100.1.242:46104 -> 10.10.82.152:80
TCP TTL:64 TOS:0x0 ID:55742 IpLen:20 DgmLen:68 DF
***AP*** Seq: 0xEE15FBDE  Ack: 0xA56AD50D  Win: 0x2FBF  TcpLen: 32
TCP Options (3) => NOP NOP TS: 2500367008 1334869827
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

However, there is also a large amount of traffic towards port 22 (SSH). To filter for port 22 traffic, I simply enter the sudo snort -r snort.log port 22, this gave us 2418 results. This means that someone is likely brute forcing ssh, as this amount of SSH traffic is not normal:

```
06/29-04:15:21.797306 10.10.140.29:22 -> 10.10.245.36:46488
TCP TTL:64 TOS:0x0 ID:51544 IpLen:20 DgmLen:332 DF
***AP*** Seq: 0x2DDE7614  Ack: 0xF86C981B  Win: 0x1E3  TcpLen: 32
TCP Options (3) => NOP NOP TS: 4119659275 1884551835
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
06/29-04:15:21.810429 10.10.245.36:46488 -> 10.10.140.29:22
TCP TTL:64 TOS:0x0 ID:21131 IpLen:20 DgmLen:68 DF
***AP*** Seq: 0xF86C981B  Ack: 0x2DDE772C  Win: 0x1E1  TcpLen: 32
TCP Options (3) => NOP NOP TS: 1884551843 4119659275
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
06/29-04:15:21.829305 10.10.245.36:46488 -> 10.10.140.29:22
TCP TTL:64 TOS:0x0 ID:21132 IpLen:20 DgmLen:104 DF
***AP*** Seq: 0xF86C982B  Ack: 0x2DDE772C  Win: 0x1E1  TcpLen: 32
TCP Options (3) => NOP NOP TS: 1884551843 4119659275
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

You can also determine that the inbound SSH traffic is mostly coming from the same IP address with the exception of when the SSH server responds. We can now answer two of the questions:

What is the name of the service under attack?

SSH

What is the used protocol/port in the attack?

TCP/22

We now need to create a rule to block this brute-force traffic and start snort in IPS mode. We can do this by first modifying the local.rules files located at /etc/snort/rules/local.rules:

```
Open  +                    local.rules
                           /etc/snort/rules
1 # $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
2 # ----------------
3 # LOCAL RULES
4 # ----------------
5 # This file intentionally does not come with signatures.  Put your local
6 # additions here.
```

Let's now add the following rule:

```
drop tcp any 22 <> any any (msg:"SSH Brute-Force Detected"; sid:1001; rev:1;)
```

All this rule does is drop all inbound port 22 TCP traffic to any IP address and port. Simply save the file, and now all we need to do is start Snort in IPS mode by entering:

```
sudo snort -c /etc/snort/snort.conf -q -Q --daq afpacket -i eth0:eth1 -A full
```

Let this run for a bit, eventually the flag will pop up on the desktop:

Scenario 2: Reverse-Shell

The second scenario involves a potential reverse-shell, we are tasked with starting Snort in sniffer mode like done previously to try and figure out the attack source, service, and port. We then also need to create an IPS rule to block the reverse-shell.

Let's start off by starting Snort in sniffer mode, we can do this by entering:

- sudo snort -dev -l .

I let this run for about 1 minute and received a total of 9271packets. Let's now investigate the produced log file:

```
sudo snort -r snort.log.1719636226
```

Once again there is a lot of traffic to port 80, however, after doing some scrolling you can find inbound and outbound traffic to port 4444:

```
WARNING: No preprocessors configured for policy 0.
06/29-04:44:29.780877 10.10.196.55:54120 -> 10.10.144.156:4444
TCP TTL:64 TOS:0x0 ID:65207 IpLen:20 DgmLen:54 DF
***AP*** Seq: 0xE99CA1AB  Ack: 0x2D7325FD  Win: 0x1EB  TcpLen: 32
TCP Options (3) => NOP NOP TS: 2358173208 1980340906
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
06/29-04:44:29.800929 10.10.144.156:4444 -> 10.10.196.55:54120
TCP TTL:64 TOS:0x0 ID:56063 IpLen:20 DgmLen:52 DF
***A**** Seq: 0x2D7325FD  Ack: 0xE99CA1AD  Win: 0x1E9  TcpLen: 32
TCP Options (3) => NOP NOP TS: 1980340907 2358173208
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

This is obviously the reverse shell as 4444 is a common listening port. However, even if it was a random port, the large amount of inbound and outbound traffic to and from that port is suspicious. Therefore, we can answer two questions with this information:

What is the used protocol/port in the attack?

TCP/4444

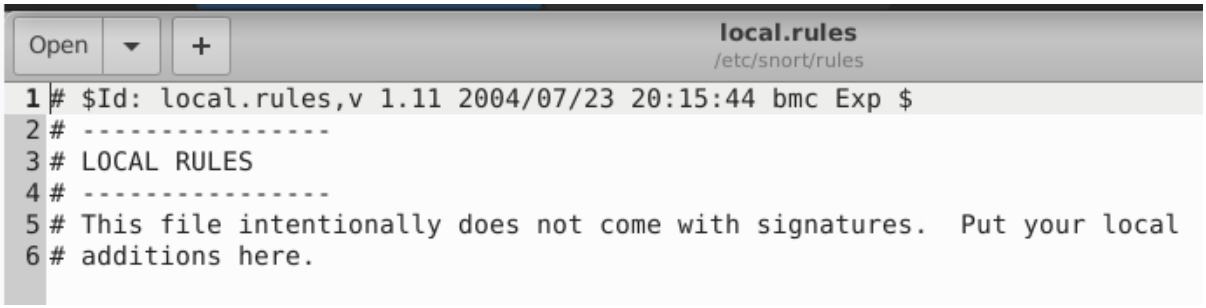
What tool is highly associated with this specific port number?

port 4444

port 4444 **metasploit**

Metasploit is the answer.

We now need to create a rule to block this brute-force traffic and start snort in IPS mode. We can do this by first modifying the local.rules files located at /etc/snort/rules/local.rules:



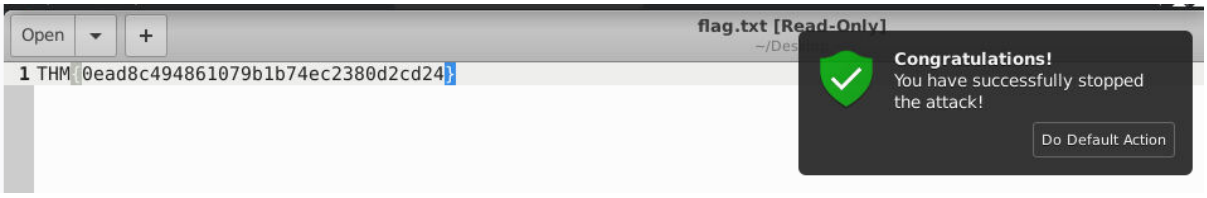
Let's now add the following rule:

```
drop tcp any 4444 -> any any (msg:"Reverse Shell Detected"; sid:1002; rev:1;)
```

This rule drops any TCP traffic originating from port 4444. Simply save the file, and now all we need to do is start Snort in IPS mode by entering:

```
sudo snort -c /etc/snort/snort.conf -q -Q --daq afpacket -i eth0:eth1 -A full
```

Let this run for a bit, eventually the flag will pop up on the desktop:



Completing this room was highly educational and enjoyable for any beginner like myself. It was a great exercise in applying basic Snort knowledge, especially in using its IP functionality for the first time.