7.1.1 Sniffer mode:

1 start server 1 and 2, and check it—

eda491@firewall:~$ sudo sh /home/eda491/netsec-lab4/prepare.sh

Done!

Verify

eda491@firewall:~$ netstat -an | grep LISTEN

tcp 0 0 0.0.0.0:8080 0.0.0.0:\* LISTEN

tcp 0 0 0.0.0.0:5555 0.0.0.0:\* LISTEN

tcp 0 0 127.0.0.53:53 0.0.0.0:\* LISTEN

tcp 0 0 0.0.0.0:3000 0.0.0.0:\* LISTEN

unix 2 [ ACC ] STREAM LISTENING 24097 /run/user/1000/systemd/private

unix 2 [ ACC ] STREAM LISTENING 24104 /run/user/1000/bus

unix 2 [ ACC ] STREAM LISTENING 24105 /run/user/1000/gnupg/S.dirmngr

unix 2 [ ACC ] STREAM LISTENING 24106 /run/user/1000/gnupg/S.gpg-agent.browser

unix 2 [ ACC ] STREAM LISTENING 24107 /run/user/1000/gnupg/S.gpg-agent.extra

unix 2 [ ACC ] STREAM LISTENING 24108 /run/user/1000/gnupg/S.gpg-agent.ssh

unix 2 [ ACC ] STREAM LISTENING 24109 /run/user/1000/gnupg/S.gpg-agent

unix 2 [ ACC ] STREAM LISTENING 24110 /run/user/1000/pk-debconf-socket

unix 2 [ ACC ] STREAM LISTENING 24111 /run/user/1000/pulse/native

unix 2 [ ACC ] STREAM LISTENING 24112 /run/user/1000/snapd-session-agent.socket

unix 2 [ ACC ] STREAM LISTENING 21870 @/tmp/.X11-unix/X0

unix 2 [ ACC ] STREAM LISTENING 17663 /run/acpid.socket

unix 2 [ ACC ] STREAM LISTENING 17664 /run/avahi-daemon/socket

unix 2 [ ACC ] STREAM LISTENING 25259 @/tmp/dbus-nYxdxeIMDJ

unix 2 [ ACC ] STREAM LISTENING 25412 /run/user/1000/keyring/control

unix 2 [ ACC ] STREAM LISTENING 17666 /run/dbus/system\_bus\_socket

unix 2 [ ACC ] STREAM LISTENING 25413 /run/user/1000/keyring/ssh

unix 2 [ ACC ] STREAM LISTENING 17668 /run/snapd.socket

unix 2 [ ACC ] STREAM LISTENING 25417 /run/user/1000/keyring/pkcs11

unix 2 [ ACC ] STREAM LISTENING 17670 /run/snapd-snap.socket

unix 2 [ ACC ] STREAM LISTENING 17672 /run/uuidd/request

unix 2 [ ACC ] STREAM LISTENING 24922 /tmp/ssh-ZggRqPKhg8g3/agent.694

unix 2 [ ACC ] STREAM LISTENING 25392 /tmp/.ICE-unix/694

unix 2 [ ACC ] STREAM LISTENING 25391 @/tmp/.ICE-unix/694

unix 2 [ ACC ] STREAM LISTENING 15580 /run/systemd/private

unix 2 [ ACC ] STREAM LISTENING 21871 /tmp/.X11-unix/X0

unix 2 [ ACC ] STREAM LISTENING 15582 /run/systemd/userdb/io.systemd.DynamicUser

unix 2 [ ACC ] STREAM LISTENING 15593 /run/systemd/fsck.progress

unix 2 [ ACC ] STREAM LISTENING 15603 /run/systemd/journal/stdout

unix 2 [ ACC ] SEQPACKET LISTENING 15607 /run/udev/control

unix 2 [ ACC ] STREAM LISTENING 16429 /run/systemd/journal/io.systemd.journal

eda491@firewall:~$

START SNORT

* Start Snort in sniffer mode, by default, Snort will capture all traffic on the specified interface. You can customize the types of traffic that Snort captures by modifying the rules in your Snort configuration file (usually snort.conf).

eda491@firewall:~$ sudo snort -i enp0s3 -c /home/eda491/netsec-lab4/snort.conf

Running in IDS mode

--== Initializing Snort ==--

Initializing Output Plugins!

Initializing Preprocessors!

Initializing Plug-ins!

Parsing Rules file "/home/eda491/netsec-lab4/snort.conf"

Tagged Packet Limit: 256

Log directory = /var/log/snort

Max Expected Streams: 272

Stream global config:

Track TCP sessions: ACTIVE

Max TCP sessions: 8192

TCP cache pruning timeout: 30 seconds

TCP cache nominal timeout: 3600 seconds

Memcap (for reassembly packet storage): 8388608

Track UDP sessions: INACTIVE

Track ICMP sessions: INACTIVE

Track IP sessions: INACTIVE

Log info if session memory consumption exceeds 1048576

Send up to 0 active responses

Protocol Aware Flushing: ACTIVE

Maximum Flush Point: 16384

WARNING: using static flush points.

Stream TCP Policy config:

Bound Address: default

Reassembly Policy: FIRST

Timeout: 30 seconds

Maximum number of bytes to queue per session: 1048576

Maximum number of segs to queue per session: 2621

Options:

Static Flushpoint Sizes: YES

Reassembly Ports:

+++++++++++++++++++++++++++++++++++++++++++++++++++

Initializing rule chains...

3 Snort rules read

3 detection rules

0 decoder rules

0 preprocessor rules

3 Option Chains linked into 3 Chain Headers

0 Dynamic rules

+++++++++++++++++++++++++++++++++++++++++++++++++++

+-------------------[Rule Port Counts]---------------------------------------

| tcp udp icmp ip

| src 0 0 0 0

| dst 0 0 0 0

| any 1 1 1 0

| nc 1 1 1 0

| s+d 0 0 0 0

+----------------------------------------------------------------------------

+-----------------------[detection-filter-config]------------------------------

| memory-cap : 1048576 bytes

+-----------------------[detection-filter-rules]-------------------------------

| none

-------------------------------------------------------------------------------

+-----------------------[rate-filter-config]-----------------------------------

| memory-cap : 1048576 bytes

+-----------------------[rate-filter-rules]------------------------------------

| none

-------------------------------------------------------------------------------

+-----------------------[event-filter-config]----------------------------------

| memory-cap : 1048576 bytes

+-----------------------[event-filter-global]----------------------------------

+-----------------------[event-filter-local]-----------------------------------

| none

+-----------------------[suppression]------------------------------------------

| none

-------------------------------------------------------------------------------

Rule application order: activation->dynamic->pass->drop->sdrop->reject->alert->log

Verifying Preprocessor Configurations!

[ Port Based Pattern Matching Memory ]

pcap DAQ configured to passive.

Acquiring network traffic from "enp0s3".

Reload thread starting...

Reload thread started, thread 0x7fae4df0c700 (1375)

Decoding Ethernet

--== Initialization Complete ==--

,,\_ -\*> Snort! <\*-

o" )~ Version 2.9.7.0 GRE (Build 149)

'''' By Martin Roesch & The Snort Team: http://www.snort.org/contact#team

Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.

Copyright (C) 1998-2013 Sourcefire, Inc., et al.

Using libpcap version 1.9.1 (with TPACKET\_V3)

Using PCRE version: 8.39 2016-06-14

Using ZLIB version: 1.2.11

Commencing packet processing (pid=1370)

KILL SNORT

* **Start Snort in sniffer mode**, by default, Snort will capture all traffic on the specified interface. You can customize the types of traffic that Snort captures by modifying the rules in your Snort configuration file (usually snort.conf).

Firewall config**:**

* If you just want to print out the TCP/IP packet headers to the screen (i.e. sniffer mode), try this: **“. /snort -v”**
* This command will run Snort and just show the IP and TCP/UDP/ICMP headers, nothing else. If you want to see the application data in transit, try the following**:”./snort -vd”**
* This instructs Snort to display the packet data as well as the headers. If you want an even more descriptive display, showing the data link layer headers, do this: **“./snort -vde”**

*sudo snort -i enp0s3 -vde*

Test message:

echo TESTMESSAGE SNORTLAB EDA491 | nc 10.0.0.3 5555

Q1: To see the string, you had to add the payload option, but when the payload was captured, the output increased in size. Elaborate briefly whether payload data should be captured (consider log sizes, attack coverage, and amount of traffic passing the IDS).

**Answer:**

Log Sizes: If we store payload data, the size of logs will increase, and it will need more storage. Might affect performance.

Attack Coverage: If we store payload data, we can quickly identify some attacks out of experience – Signature based detection

Amount of Traffic: Processing and storing payload data for all traffic passing through the IDS can be resource-intensive, especially in high-traffic environments.

Capture link and network layer

We have a picture on phone

Capture transport layer and payload

Picture onphone

STORE PACKETS IN ASCII FORMAT – see picture on phone

GNU nano 4.8 ICMP\_ECHO

05/13-09:24:57.585887 10.0.0.4 -> 10.0.0.3

ICMP TTL:64 TOS:0x0 ID:26067 IpLen:20 DgmLen:84 DF

Type:8 Code:0 ID:2 Seq:1 ECHO

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/13-09:24:58.608977 10.0.0.4 -> 10.0.0.3

ICMP TTL:64 TOS:0x0 ID:26103 IpLen:20 DgmLen:84 DF

Type:8 Code:0 ID:2 Seq:2 ECHO

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/13-09:24:59.612096 10.0.0.4 -> 10.0.0.3

ICMP TTL:64 TOS:0x0 ID:26193 IpLen:20 DgmLen:84 DF

Type:8 Code:0 ID:2 Seq:3 ECHO

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/13-09:25:00.613811 10.0.0.4 -> 10.0.0.3

ICMP TTL:64 TOS:0x0 ID:26310 IpLen:20 DgmLen:84 DF

Type:8 Code:0 ID:2 Seq:4 ECHO

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/13-09:25:01.615072 10.0.0.4 -> 10.0.0.3

ICMP TTL:64 TOS:0x0 ID:26373 IpLen:20 DgmLen:84 DF

Type:8 Code:0 ID:2 Seq:5 ECHO

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/13-09:25:02.642407 10.0.0.4 -> 10.0.0.3

ICMP TTL:64 TOS:0x0 ID:26428 IpLen:20 DgmLen:84 DF

Type:8 Code:0 ID:2 Seq:6 ECHO

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/13-09:25:03.643628 10.0.0.4 -> 10.0.0.3

ICMP TTL:64 TOS:0x0 ID:26623 IpLen:20 DgmLen:84 DF

Type:8 Code:0 ID:2 Seq:7 ECHO

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/13-09:25:04.644339 10.0.0.4 -> 10.0.0.3

ICMP TTL:64 TOS:0x0 ID:26805 IpLen:20 DgmLen:84 DF

Type:8 Code:0 ID:2 Seq:8 ECHO

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/13-09:25:05.645805 10.0.0.4 -> 10.0.0.3

ICMP TTL:64 TOS:0x0 ID:26828 IpLen:20 DgmLen:84 DF

Type:8 Code:0 ID:2 Seq:9 ECHO

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/13-09:25:06.647565 10.0.0.4 -> 10.0.0.3

ICMP TTL:64 TOS:0x0 ID:26976 IpLen:20 DgmLen:84 DF

Type:8 Code:0 ID:2 Seq:10 ECHO

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/13-09:25:07.664474 10.0.0.4 -> 10.0.0.3

ICMP TTL:64 TOS:0x0 ID:27211 IpLen:20 DgmLen:84 DF

Type:8 Code:0 ID:2 Seq:11 ECHO

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/13-09:25:08.666755 10.0.0.4 -> 10.0.0.3

ICMP TTL:64 TOS:0x0 ID:27365 IpLen:20 DgmLen:84 DF

Type:8 Code:0 ID:2 Seq:12 ECHO

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/13-09:25:09.668147 10.0.0.4 -> 10.0.0.3

ICMP TTL:64 TOS:0x0 ID:27480 IpLen:20 DgmLen:84 DF

Type:8 Code:0 ID:2 Seq:13 ECHO

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/13-09:25:10.668946 10.0.0.4 -> 10.0.0.3

ICMP TTL:64 TOS:0x0 ID:27516 IpLen:20 DgmLen:84 DF

Type:8 Code:0 ID:2 Seq:14 ECHO

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/13-09:25:11.671403 10.0.0.4 -> 10.0.0.3

ICMP TTL:64 TOS:0x0 ID:27700 IpLen:20 DgmLen:84 DF

Type:8 Code:0 ID:2 Seq:15 ECHO

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

REPLY ----- See picture on phone

GNU nano 4.8 ICMP\_ECHO\_REPLY

05/13-09:24:57.585902 10.0.0.3 -> 10.0.0.4

ICMP TTL:64 TOS:0x0 ID:59814 IpLen:20 DgmLen:84

Type:0 Code:0 ID:2 Seq:1 ECHO REPLY

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/13-09:24:58.609073 10.0.0.3 -> 10.0.0.4

ICMP TTL:64 TOS:0x0 ID:59934 IpLen:20 DgmLen:84

Type:0 Code:0 ID:2 Seq:2 ECHO REPLY

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/13-09:24:59.612130 10.0.0.3 -> 10.0.0.4

ICMP TTL:64 TOS:0x0 ID:60108 IpLen:20 DgmLen:84

Type:0 Code:0 ID:2 Seq:3 ECHO REPLY

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/13-09:25:00.613845 10.0.0.3 -> 10.0.0.4

ICMP TTL:64 TOS:0x0 ID:60116 IpLen:20 DgmLen:84

Type:0 Code:0 ID:2 Seq:4 ECHO REPLY

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/13-09:25:01.615106 10.0.0.3 -> 10.0.0.4

ICMP TTL:64 TOS:0x0 ID:60274 IpLen:20 DgmLen:84

Type:0 Code:0 ID:2 Seq:5 ECHO REPLY

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/13-09:25:02.642441 10.0.0.3 -> 10.0.0.4

ICMP TTL:64 TOS:0x0 ID:60398 IpLen:20 DgmLen:84

Type:0 Code:0 ID:2 Seq:6 ECHO REPLY

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/13-09:25:03.643661 10.0.0.3 -> 10.0.0.4

ICMP TTL:64 TOS:0x0 ID:60574 IpLen:20 DgmLen:84

Type:0 Code:0 ID:2 Seq:7 ECHO REPLY

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/13-09:25:04.644372 10.0.0.3 -> 10.0.0.4

ICMP TTL:64 TOS:0x0 ID:60677 IpLen:20 DgmLen:84

Type:0 Code:0 ID:2 Seq:8 ECHO REPLY

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/13-09:25:05.645888 10.0.0.3 -> 10.0.0.4

ICMP TTL:64 TOS:0x0 ID:60761 IpLen:20 DgmLen:84

Type:0 Code:0 ID:2 Seq:9 ECHO REPLY

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/13-09:25:06.647591 10.0.0.3 -> 10.0.0.4

ICMP TTL:64 TOS:0x0 ID:60802 IpLen:20 DgmLen:84

Type:0 Code:0 ID:2 Seq:10 ECHO REPLY

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/13-09:25:07.664507 10.0.0.3 -> 10.0.0.4

ICMP TTL:64 TOS:0x0 ID:60937 IpLen:20 DgmLen:84

Type:0 Code:0 ID:2 Seq:11 ECHO REPLY

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/13-09:25:08.666789 10.0.0.3 -> 10.0.0.4

ICMP TTL:64 TOS:0x0 ID:61160 IpLen:20 DgmLen:84

Type:0 Code:0 ID:2 Seq:12 ECHO REPLY

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/13-09:25:09.668181 10.0.0.3 -> 10.0.0.4

ICMP TTL:64 TOS:0x0 ID:61204 IpLen:20 DgmLen:84

Type:0 Code:0 ID:2 Seq:13 ECHO REPLY

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/13-09:25:10.668981 10.0.0.3 -> 10.0.0.4

ICMP TTL:64 TOS:0x0 ID:61217 IpLen:20 DgmLen:84

Type:0 Code:0 ID:2 Seq:14 ECHO REPLY

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/13-09:25:11.671436 10.0.0.3 -> 10.0.0.4

ICMP TTL:64 TOS:0x0 ID:61370 IpLen:20 DgmLen:84

Type:0 Code:0 ID:2 Seq:15 ECHO REPLY

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

STORE PACKETS IN PCAP FORMAT

Note: we have pictures of raw and readable format

Q2: Elaborate briefly on storing log files in ASCII and pcap format. When would it be more suitable to store the files in ASCII, and when would it be more suitable to store the files in pcap format? Is there a difference in the way the log files are named?

ASCII: Its easier to read and can be read using text editors like vim. We can also do all the search and find operations.

Pcap: It stores data in binary. It is used by analysis tools like wireshark. Here, info about all IP addresses are stored in the same file.

In ASCII, we have different files for different IP addresses.

ASCII can be read by us, so when that is a priority, and when we don’t have a constraint of having limited files, we can use this. We can handle data and do operations on text.

PCAP is raw data. So when we don’t need to read it ourselves and need to analyse it via 3rd party apps like wireshark, we can use this for network packet analysis.

PCAP stores files as : snort.log. 29485798

ASCII stores files as ICMP\_ECHO

ATTACK 1:

We have picture of wireshark

**/bin/sh** command is used in scripts or other programs to execute system commands or run shell scripts.

ALERT:

[\*\*] [1:100007:0] /bin/sh message detected [\*\*]

[Priority: 0]

05/13-10:31:53.686657 10.0.0.4:44490 -> 10.0.0.3:5555

TCP TTL:64 TOS:0x0 ID:31799 IpLen:20 DgmLen:148 DF

\*\*\*AP\*\*\* Seq: 0x246EC8F4 Ack: 0xD698D098 Win: 0x1F6 TcpLen: 32

TCP Options (3) => NOP NOP TS: 2274566609 983112328

ATTACK 2: XMAS Attack

We have picture of wireshark

In the suspect2 attack, the attacker seems to **inject the XMAS attack.** The XMAS attack has a TCP packet with the URG, PUSH, and FIN flags set. These flags are also used in normal TCP traffic but they are not usually used together in a single packet.

ALERT:

[\*\*] [1:100004:0] XMAS scan detected [\*\*]

[Priority: 0]

05/13-10:26:44.475851 10.0.0.4:40445 -> 10.0.0.3:25

TCP TTL:46 TOS:0x0 ID:23951 IpLen:20 DgmLen:40

\*\*U\*P\*\*F Seq: 0x53467CD3 Ack: 0x0 Win: 0x400 TcpLen: 20 UrgPtr: 0x0

[\*\*] [1:100004:0] XMAS scan detected [\*\*]

[Priority: 0]

05/13-10:26:44.475851 10.0.0.4:40445 -> 10.0.0.3:21

TCP TTL:53 TOS:0x0 ID:33681 IpLen:20 DgmLen:40

\*\*U\*P\*\*F Seq: 0x53467CD3 Ack: 0x0 Win: 0x400 TcpLen: 20 UrgPtr: 0x0

[\*\*] [1:100004:0] XMAS scan detected [\*\*]

[Priority: 0]

05/13-10:26:44.475851 10.0.0.4:40445 -> 10.0.0.3:23

TCP TTL:46 TOS:0x0 ID:46134 IpLen:20 DgmLen:40

\*\*U\*P\*\*F Seq: 0x53467CD3 Ack: 0x0 Win: 0x400 TcpLen: 20 UrgPtr: 0x0

[\*\*] [1:100004:0] XMAS scan detected [\*\*]

[Priority: 0]

05/13-10:26:44.475851 10.0.0.4:40445 -> 10.0.0.3:111

TCP TTL:42 TOS:0x0 ID:45152 IpLen:20 DgmLen:40

\*\*U\*P\*\*F Seq: 0x53467CD3 Ack: 0x0 Win: 0x400 TcpLen: 20 UrgPtr: 0x0

[\*\*] [1:100004:0] XMAS scan detected [\*\*]

[Priority: 0]

05/13-10:26:44.475851 10.0.0.4:40445 -> 10.0.0.3:445

TCP TTL:50 TOS:0x0 ID:51857 IpLen:20 DgmLen:40

\*\*U\*P\*\*F Seq: 0x53467CD3 Ack: 0x0 Win: 0x400 TcpLen: 20 UrgPtr: 0x0

[\*\*] [1:100004:0] XMAS scan detected [\*\*]

[Priority: 0]

05/13-10:26:44.475851 10.0.0.4:40445 -> 10.0.0.3:80

TCP TTL:58 TOS:0x0 ID:45288 IpLen:20 DgmLen:40

\*\*U\*P\*\*F Seq: 0x53467CD3 Ack: 0x0 Win: 0x400 TcpLen: 20 UrgPtr: 0x0

[\*\*] [1:100004:0] XMAS scan detected [\*\*]

[Priority: 0]

05/13-10:26:44.475851 10.0.0.4:40445 -> 10.0.0.3:79

TCP TTL:50 TOS:0x0 ID:31780 IpLen:20 DgmLen:40

\*\*U\*P\*\*F Seq: 0x53467CD3 Ack: 0x0 Win: 0x400 TcpLen: 20 UrgPtr: 0x0

[\*\*] [1:100004:0] XMAS scan detected [\*\*]

[Priority: 0]

05/13-10:26:44.475851 10.0.0.4:40445 -> 10.0.0.3:513

TCP TTL:41 TOS:0x0 ID:22190 IpLen:20 DgmLen:40

\*\*U\*P\*\*F Seq: 0x53467CD3 Ack: 0x0 Win: 0x400 TcpLen: 20 UrgPtr: 0x0

[\*\*] [1:100004:0] XMAS scan detected [\*\*]

[Priority: 0]

05/13-10:26:44.475910 10.0.0.4:40445 -> 10.0.0.3:514

TCP TTL:52 TOS:0x0 ID:62389 IpLen:20 DgmLen:40

\*\*U\*P\*\*F Seq: 0x53467CD3 Ack: 0x0 Win: 0x400 TcpLen: 20 UrgPtr: 0x0

[\*\*] [1:100004:0] XMAS scan detected [\*\*]

[Priority: 0]

05/13-10:26:44.475910 10.0.0.4:40445 -> 10.0.0.3:512

TCP TTL:57 TOS:0x0 ID:20382 IpLen:20 DgmLen:40

\*\*U\*P\*\*F Seq: 0x53467CD3 Ack: 0x0 Win: 0x400 TcpLen: 20 UrgPtr: 0x0

ATTACK 3:

We have picture of wireshark

ALERT:

PSH + ACK flags together 🡪 Attacker is sending these files along with GET. This means it is trying to run some malicious files. Dir command will list the contents of the directory. So the attacker can upload or download some malicious files.

* flags: tcp : This specifies that the rule will only match TCP packets with the PSH and ACK flags set. This is because the rule is looking for a specific sequence of bytes that might not appear until a TCP segment is complete and pushed.
* content:"/c";: This looks for the presence of the string "/c" in the payload of the TCP segment. This is the command line switch for the Windows Command Prompt to execute a command.
* content:"/system32";: This looks for the presence of the string "/system32" in the payload of the TCP segment. This is the default location of many important system files and binaries on a Windows system.
* within:15: This specifies that the two content matches must be within 15 bytes of each other. This is to ensure that they are part of the same command being executed.
* msg:"Possible Command Execution";: This is the message that will be logged when the rule is triggered.
* sid:100005: This is the unique identifier for the rule.

Attack 4: Activation of bots for DDoS

Have picture

Attack 5:

alert tcp [172.16.0.1/12,192.168.0.1/16,10.0.0.1/8,!10.0.0.1/24] any -> $HOME\_NET any (msg:"illegal source ip address"; sid: 100009)

[\*\*] [1:100009:0] illegal source ip address [\*\*]

[Priority: 0]

05/13-11:25:32.736370 10.1.0.1:1088 -> 10.0.0.3:80

TCP TTL:30 TOS:0x0 ID:23499 IpLen:20 DgmLen:40

\*2\*\*\*\*S\* Seq: 0x28376839 Ack: 0x0 Win: 0xFFFF TcpLen: 20

[\*\*] [1:100009:0] illegal source ip address [\*\*]

[Priority: 0]

05/13-11:25:32.744596 10.1.0.64:1088 -> 10.0.0.3:80

TCP TTL:30 TOS:0x0 ID:23499 IpLen:20 DgmLen:40

\*2\*\*\*\*S\* Seq: 0x28376839 Ack: 0x0 Win: 0xFFFF TcpLen: 20

[\*\*] [1:100009:0] illegal source ip address [\*\*]

[Priority: 0]

05/13-11:25:32.751114 10.1.0.128:1088 -> 10.0.0.3:80

TCP TTL:30 TOS:0x0 ID:23499 IpLen:20 DgmLen:40

\*2\*\*\*\*S\* Seq: 0x28376839 Ack: 0x0 Win: 0xFFFF TcpLen: 20

[\*\*] [1:100009:0] illegal source ip address [\*\*]

[Priority: 0]

05/13-11:25:32.758220 10.1.0.192:1088 -> 10.0.0.3:80

TCP TTL:30 TOS:0x0 ID:23499 IpLen:20 DgmLen:40

\*2\*\*\*\*S\* Seq: 0x28376839 Ack: 0x0 Win: 0xFFFF TcpLen: 20

[\*\*] [1:100009:0] illegal source ip address [\*\*]

[Priority: 0]

05/13-11:25:32.764108 10.1.0.255:1088 -> 10.0.0.3:80

TCP TTL:30 TOS:0x0 ID:23499 IpLen:20 DgmLen:40

\*2\*\*\*\*S\* Seq: 0x28376839 Ack: 0x0 Win: 0xFFFF TcpLen: 20

[\*\*] [1:100009:0] illegal source ip address [\*\*]

[Priority: 0]

05/13-11:25:32.770778 10.0.1.1:1088 -> 10.0.0.3:80

TCP TTL:30 TOS:0x0 ID:23499 IpLen:20 DgmLen:40

\*\*\*\*\*\*S\* Seq: 0x28376839 Ack: 0x0 Win: 0xFFFF TcpLen: 20

[\*\*] [1:100009:0] illegal source ip address [\*\*]

[Priority: 0]

05/13-11:25:32.776951 10.0.64.1:1088 -> 10.0.0.3:80

TCP TTL:30 TOS:0x0 ID:23499 IpLen:20 DgmLen:40

\*2\*\*\*\*S\* Seq: 0x28376839 Ack: 0x0 Win: 0xFFFF TcpLen: 20

Command:

# set your IP subnet/address here

var HOME\_NET 10.0.0.3/24

# stream5 can be useful for certain rule options

preprocessor stream5\_global: max\_tcp 8192, track\_tcp yes, track\_udp no

preprocessor stream5\_tcp: policy first, use\_static\_footprint\_sizes

#This will log everything that is sent over TCP and UDP use them as reference

#And to see what the packets are but remove them to pass the lab

log tcp any any -> $HOME\_NET any (sid:100001)

log udp any any -> $HOME\_NET any (sid:100002)

log icmp any any -> $HOME\_NET any (sid:100003)

Q3: In this assignment, you have written specific rules to discover attacks. This paradigm is known as signature-based detection. Another paradigm for intrusion detection is called anomaly-based detection. Elaborate on the advantages and disadvantages of each.

**Signature-based detection** is based on predefined signatures or patterns of known malicious activities to identify threats. It compares network traffic or system events against a signature database and triggers alerts when a match is found. It is effective in detecting known threats but may struggle with new or unknown attacks.

**Anomaly-based detection** however focuses on detecting deviations from normal behavior or baselines. Anomaly-based detection establishes a baseline of normal activity and generates alerts (Intrusion Detection System) or takes action (Intrusion Detection and Handling System) when abnormal behavior or suspicious activity is detected. It is effective in detecting previously unseen attacks but may generate more false positives and also require more resources.

Signature-based Detection:

Advantages:

* 1. Can protect against known types of attacks
  2. Fast
  3. Low false positive rate

Disadvantages:

1. Doesn’t protect against abnormal behaviour
2. We need to always keep signatures up to date

Anomaly-based Detection:

Advantages:

1. Can detect anomalies in threats – protects against 0 day attacks
2. We don’t need to update anything like we do in signature based.

Disadvantages:

1. High false positive
2. Needs to keep learning and so, computationally complex

CONF FILE

# ATTACK 1

alert tcp any any -> $HOME\_NET any (content:"/bin/sh"; msg: "/bin/sh message detected"; sid:100007)

# ATTACK 2

alert tcp any any -> $HOME\_NET any (flags: FPU; msg: "XMAS scan detected"; sid:100004)

# ATTACK 3

var HTTP\_PORTS 8080,80,443

alert tcp any any -> $HOME\_NET $HTTP\_PORTS (content:"/system32/cmd.exe"; msg:"Possible Command Execution"; nocase; sid:100005)

# ATTACK 4

alert tcp any any -> $HOME\_NET any (content:"activating botz lol"; msg:"Bot detected"; nocase; sid:100006)

# ATTACK 5

alert tcp [!10.0.0.1/24] any -> $HOME\_NET any (msg:"illegal source ip address"; sid:100009)

#10.0.0.1/8, 172.16.0.1/12, 192.168.0.1/16,