Table of Contents:

A.Setup:

Network address	10.140.100.0/24
Computer name	PC 3 00-0c-29-f0-27-67
Teammate computer name	PC 1 00-0c-29-9b-f4-91
Ip address	10. 140.100.103
Teammate IP address	10.140.100.104

B.Snort Overview:

Parameter	**Function**
Snort -i <if>	Listen on interface <if>
Snort -v	Be verbose
Snort -l <ld>	Log to directory <ld>
Snort -W	List available interfaces **(Win32 only!!**)
Snort -d	Dump the application layer
Snort -e	Display the second layer header info

C. Snort Sniffer Mode

Brief: When you run **'snort -W',** it will show available interfaces to sniff **[assuming Win32]**
In the image below → **interface 1** will be used to run snort commands

Section 1: Sniff interface 1 with **[snort -v -i1]** [verbose]

Snort is now running from the **Network Adapter 1** and listens to the network traffic.

```
c:\Snort\bin>snort -v -i1
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
The DAQ version does not support reload.
Acquiring network traffic from "\Device\NPF_{E3645972-9DF7-46AD-97E6-D27208887D8
1}".
Decoding Ethernet
```

While snort runs**[snort -v -i1]**, **ping partner PC**[10.140.100.104]

```
C:\Users\Student>ping 10.140.100.104 -t

Pinging 10.140.100.104 with 32 bytes of data:
Reply from 10.140.100.104: bytes=32 time=1ms TTL=128
Reply from 10.140.100.104: bytes=32 time=1ms TTL=128
Reply from 10.140.100.104: bytes=32 time=1ms TTL=128
Reply from 10.140.100.104: bytes=32 time=1ms TTL=128
Reply from 10.140.100.104: bytes=32 time=1ms TTL=128
Reply from 10.140.100.104: bytes=32 time=1ms TTL=128
```

Observe **ICMP** traffic in snort on **my PC**[10.140.100.103] **after ping**

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/27-20:35:19.497080 10.140.100.103 -> 10.140.100.104
ICMP TTL:128 TOS:0x0 ID:8373 IpLen:20 DgmLen:60
Type:8  Code:0  ID:1   Seq:612  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
04/27-20:35:19.498248 10.140.100.104 -> 10.140.100.103
ICMP TTL:128 TOS:0x0 ID:27444 IpLen:20 DgmLen:60
Type:0  Code:0  ID:1  Seq:612  ECHO REPLY
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

While snort runs**[snort -v -i1]**, **Partner Pings my PC**[10.140.100.103]

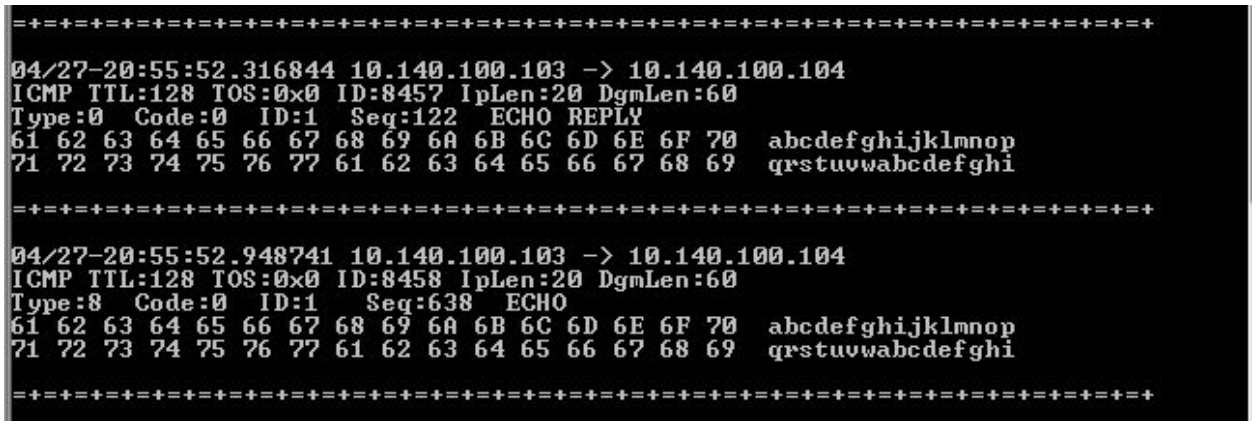
```
C:\Users\Student>ping 10.140.100.103 -t

Pinging 10.140.100.103 with 32 bytes of data:
Reply from 10.140.100.103: bytes=32 time=2ms TTL=128
Reply from 10.140.100.103: bytes=32 time=1ms TTL=128
Reply from 10.140.100.103: bytes=32 time=2ms TTL=128
Reply from 10.140.100.103: bytes=32 time=1ms TTL=128
```

Observe **ICMP** traffic in snort **on partner PC**[10.140.100.104] **after ping**

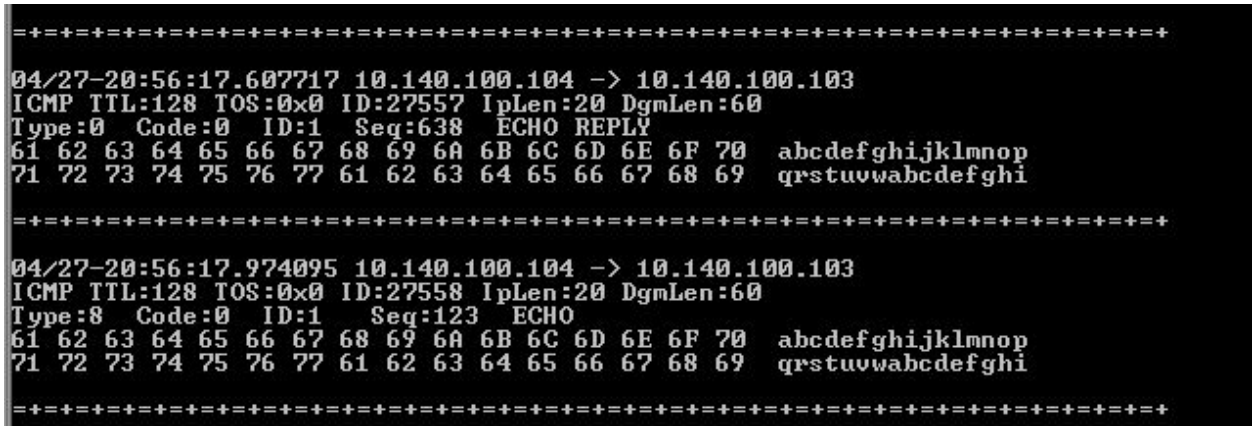
```
Commencing packet processing (pid=344)
04/27-20:35:47.069152 10.140.100.104 -> 10.140.100.103
ICMP TTL:128 TOS:0x0 ID:27449 IpLen:20 DgmLen:60
Type:8  Code:0  ID:1   Seq:92  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
04/27-20:35:47.070567 10.140.100.103 -> 10.140.100.104
ICMP TTL:128 TOS:0x0 ID:8378 IpLen:20 DgmLen:60
Type:0  Code:0  ID:1  Seq:92  ECHO REPLY
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Section 2: Sniff interface 1 with **[snort -v -d -i1]** [verbose] [dump(application layer)]
Snort is now running from the **Network Adapter 1** and listens to the network traffic.

While snort runs**[snort -v -d -i1]**, **ping partner PC**[10.140.100.104]
Observe **ICMP** traffic in snort **on partner PC**[10.140.100.104] **after ping**
NOTE: Application layer data



While snort runs**[snort -v -d -i1]**, **partner pings my PC**[10.140.100.103]
Observe **ICMP** traffic in snort **on my PC**[10.140.100.104] **after ping**
NOTE: Application layer data



Section 3: Sniff interface 1 with **[snort -v -d -e -i1]** [verbose] [dump(application layer)] & [second layer header info]
Snort is now running from the **Network Adapter 1** and listens to the network traffic.

While snort runs**[snort -v -d -e -i1]**, **ping partner PC**[10.140.100.104]
Observe **ICMP** traffic in snort **on partner PC**[10.140.100.104] **after ping**
NOTE: Application layer data & **second layer header data (Mac Address/Type/Len)**

While snort runs**[snort -v -d e -i1]**, **partner pings my PC**[10.140.100.103]
Observe **ICMP** traffic in snort **on my PC**[10.140.100.104] **after ping**
NOTE: Application layer data **& second layer header data(Mac Address/Type/Len)**

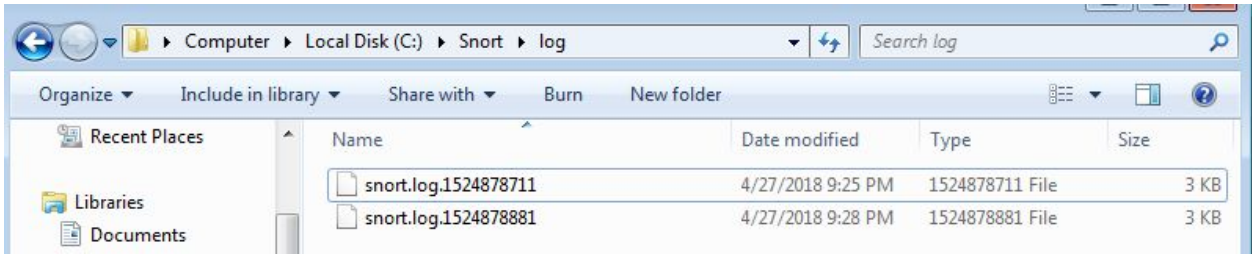
D. Packet Logger Mode

---

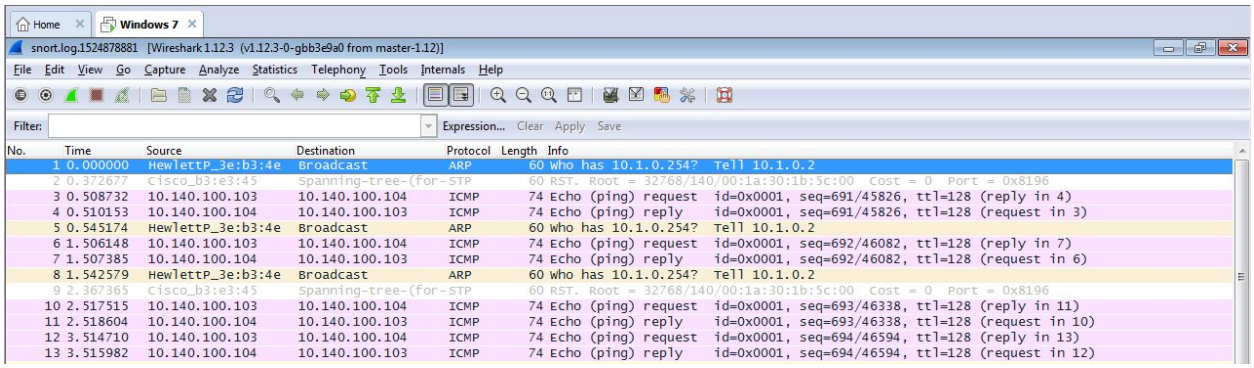
snort -dev -i1 -l c:\snort\log → creates a local snort.log file

snort -dev -i1 -l c:\snort\log -b → creates a local snort.log binary file [for high speed network]

After pings, both **standard & binary snort.log files** are present inside **c:\snort\log dir**



Open **binary snort.log 1524878881** in wireshark to see **ICMP** ping traffic

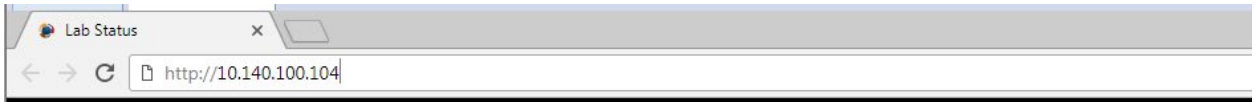


E. Intrusion Detection Mode

Create a snort configuration file c:\snort\etc\temp.conf
The file defines the **home network, external network, RULE_PATH , and 1 basic rule**

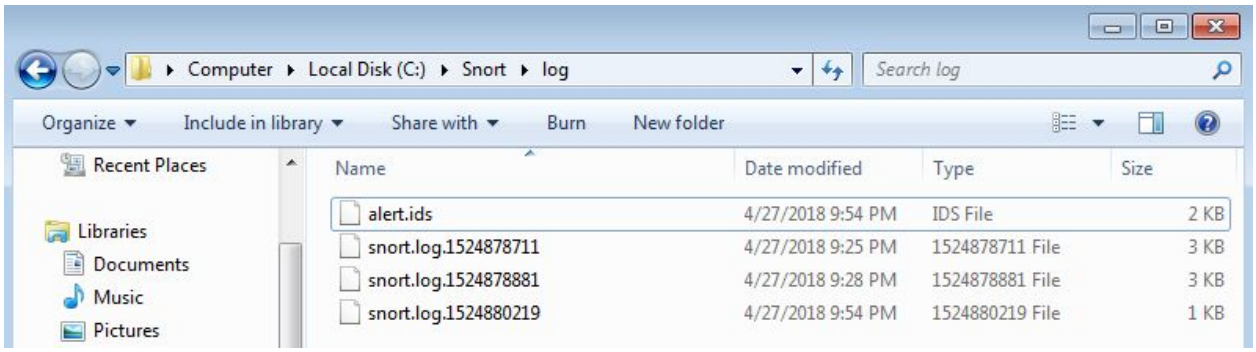
Start snort with the **temp configuration file** and create a **log file**

Instead of pinging, **create TCP traffic** by writing **teammate IP address** in browser

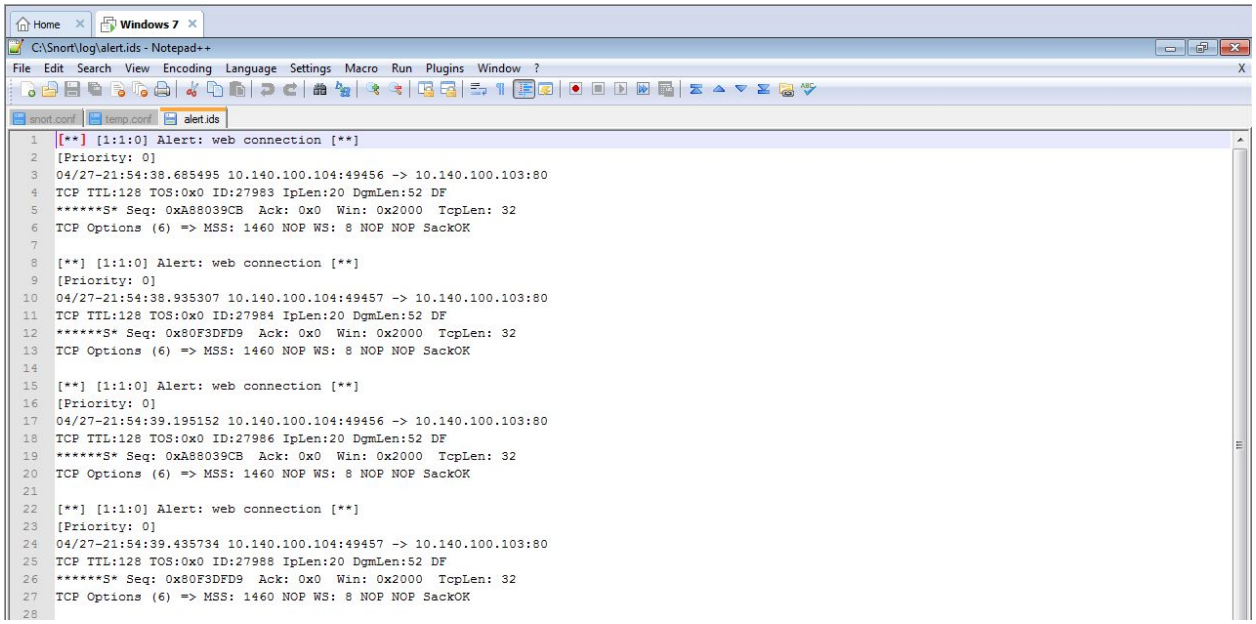


Partner creates **TCP traffic by writing** my IP address **in their browser**

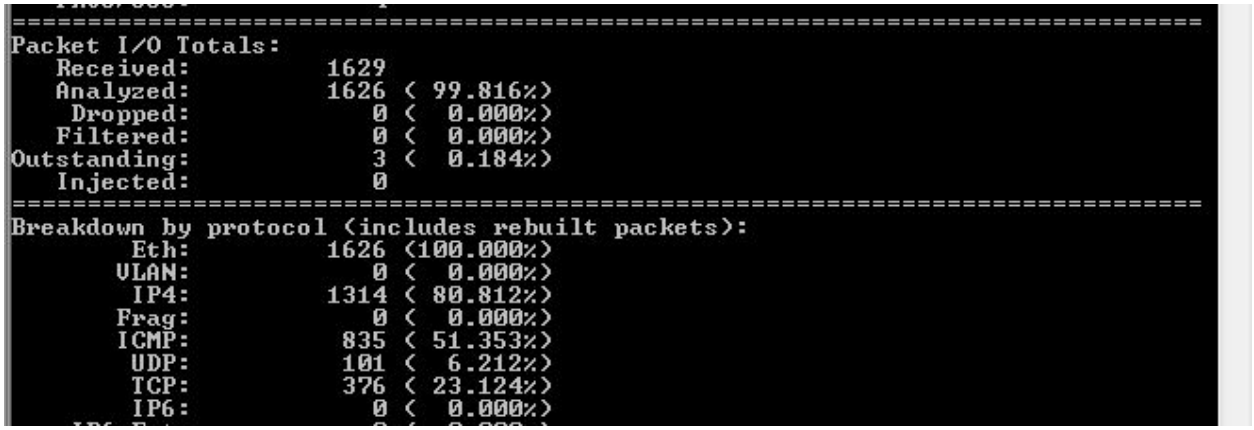
Inside the **C:\snort\log directory** on both my PC & partner PC → **alert.ids** created



Inside **alert.ids** file → See all alerts generated by **temp.conf** rule
alert tcp any any → any [80] (msg: "alert: web connection"; ssid: 1;)
NOTE: All alerts in alert.ids file are detecting TCP traffic as per the above rule..



snort summary → Throughout the lab experiment. **TCP/ UDP/ICMP** was all detected



Conclusion:

My main interests in this lab were learning how to specify different snort parameters [-v -d -e], saving binary files, analyzing the snort traffic in wireshark, and then learning how to build alerts and log them to a local file. Because of the preprocessor errors I received, I decided to download the registered rules from snort.org and write a snort.conf file that is adjusted for a windows machine. Those changes can be found in the table I created in the bonus section. Running a new snort.conf file removed any errors, at which point, I decided to create more alerts. In the local.rules file, I captured TCP, UDP, ICMP with new rules. I was able to see the corresponding alerts after I re-ran snort and viewed alert.ids.

BONUS:

"snort -i1 –l c:\snort\log –c c:\snort\etc\temp.conf" →
generates a preprocessor warning.

Dynamicpreprocessor directory c:\Snort\lib\snort_dynamicpreprocessor
Dynamicengine c:\snort\lib\snort_dynamicengine\sf_engine.dll

After running a similar command with a different conf file containing these changes:
In addition to the snort Registered rules [snortrules-snapshot-29110.tar.gz] download
The following rules were adapted for windows …

```
#snort rule path
var RULE_PATH c:Snort\rules
```

```
#snort preprocessor rule path
var  PREPROC_RULE_PATH c:\Snort\preproc_rules
```

Section 2:

```
#configure default log directory
config logdir: c:\Snort\log
```

Section 4:

```
#path to dynamic preprocessor libraries
dynamicpreprocessor directory c:\Snort\lib\snort_dynamicpreprocessor
```

```
#path to base processor engine
dynamicengine  c:\snort\lib\snort_dynamicengine\sfengine.dll
```

Section 6:

```
#syslog
```

```
output alert_syslog: LOG_AUTH LOG_ALERT
```

Section 7:

NOTE: replace all forward slashes(/) to back slashes(\) for all include statements

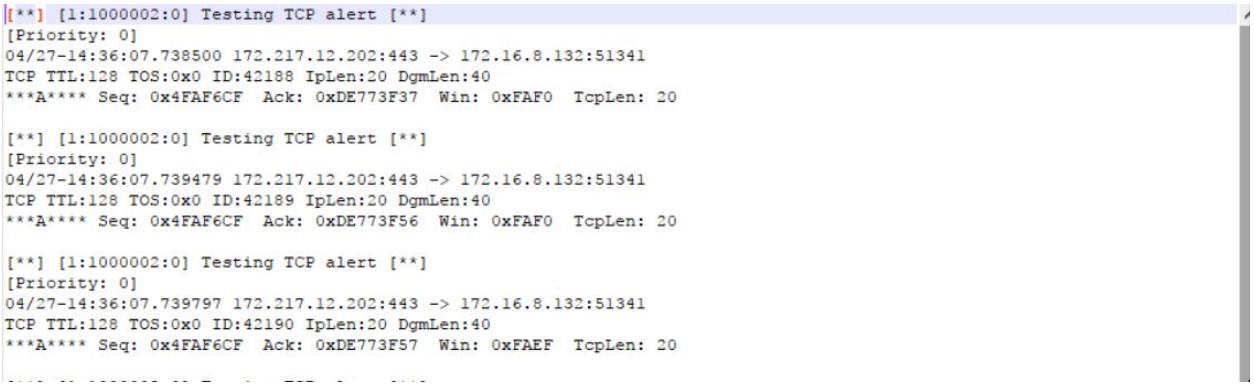
Section 8:

```
#decoder and preprocessor event rules
Include $PREPROC_RULE_PATH\preprocessor.rules
Include $PREPROC_RULE_PATH\decoder.rules
Include $PREPROC_RULE_PATH\sensitive-data.rules
```

The following rules file/local.rules was created to build alerts

```
#-------------
# LOCAL RULES
#-------------
alert ICMP any any -> any any (msg:"Testing ICMP alert"; sid: 1000001;)
alert TCP any any -> any any (msg:"Testing TCP alert"; sid: 1000002;)
alert UDp any any -> any any (msg:"Testing UDP alert"; sid: 1000003;)
```

Alerts were logged to snort\log\alert.ids



```
[**] [1:1000002:0] Testing TCP alert [**]
[Priority: 0]
04/27-14:36:07.738500 172.217.12.202:443 -> 172.16.8.132:51341
TCP TTL:128 TOS:0x0 ID:42188 IpLen:20 DgmLen:40
***A**** Seq: 0x4FAF6CF  Ack: 0xDE773F37  Win: 0xFAF0  TcpLen: 20

[**] [1:1000002:0] Testing TCP alert [**]
[Priority: 0]
04/27-14:36:07.739479 172.217.12.202:443 -> 172.16.8.132:51341
TCP TTL:128 TOS:0x0 ID:42189 IpLen:20 DgmLen:40
***A**** Seq: 0x4FAF6CF  Ack: 0xDE773F56  Win: 0xFAF0  TcpLen: 20

[**] [1:1000002:0] Testing TCP alert [**]
[Priority: 0]
04/27-14:36:07.739797 172.217.12.202:443 -> 172.16.8.132:51341
TCP TTL:128 TOS:0x0 ID:42190 IpLen:20 DgmLen:40
***A**** Seq: 0x4FAF6CF  Ack: 0xDE773F57  Win: 0xFAEF  TcpLen: 20
```