

Analyzing the Network Capture of WannaCry Ransomware

Background:

Network analysis can get very detailed. There are several different types of evidence. Full packet captures, logs and netflow information. This exercise focuses on full packet capture.

The evidence in this lab is a packet capture from the WannaCry attack in 2017. By analyzing network traffic, a malware analyst, MalwareTech, was able to successfully stop the malware from spreading.

Evidence:

Wcry-pcap.pcap

For additional practice Netresec keeps a list of publicly available pcap files:

<http://www.netresec.com/?page=PcapFiles>.

Questions:

Provide a description of the traffic with a theory of the malware propagation. Include the following details in your summary:

- When did the capture start and end?
- Which OS is the scanned machine running?
- How many frames are in the first TCP stream?
- What domain was the host querying?
- Originally the odd domain was not registered and was therefore not returning any results. To stop the worm from spreading Malware Tech registered that domain. What does the query return now?

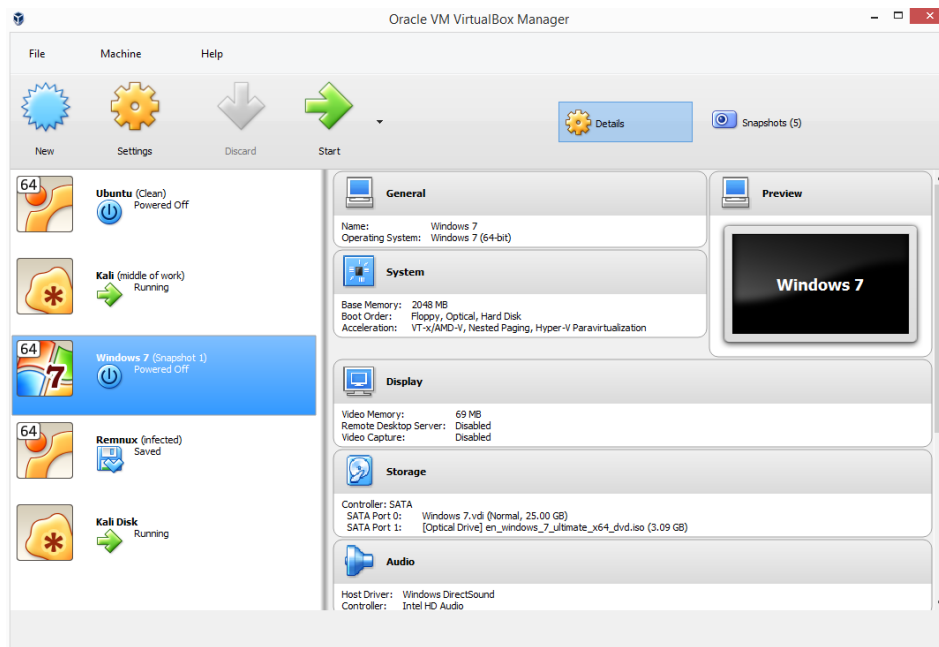
Setup:

For this exercise we will need two virtual machines, one Kali and one Windows.

Additional tools necessary:

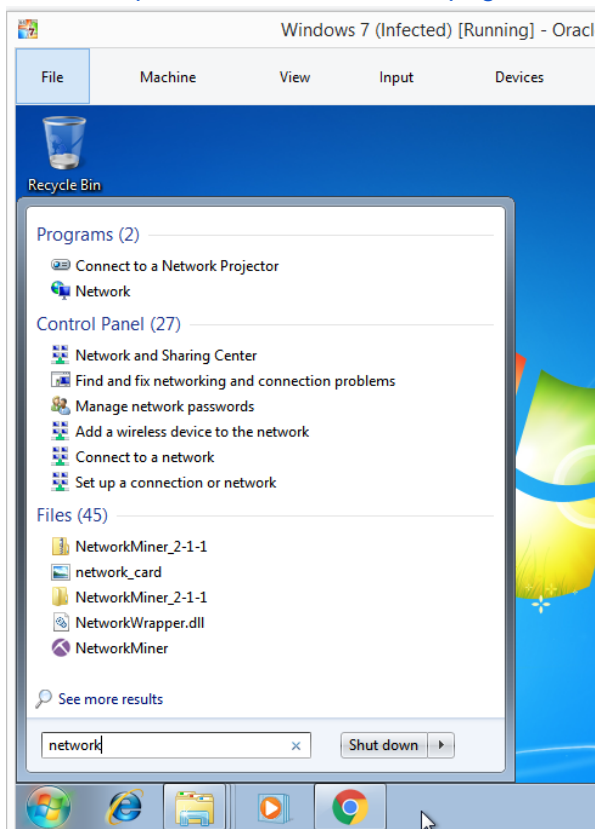
- Snort (Kali)
- Network Miner (Windows)

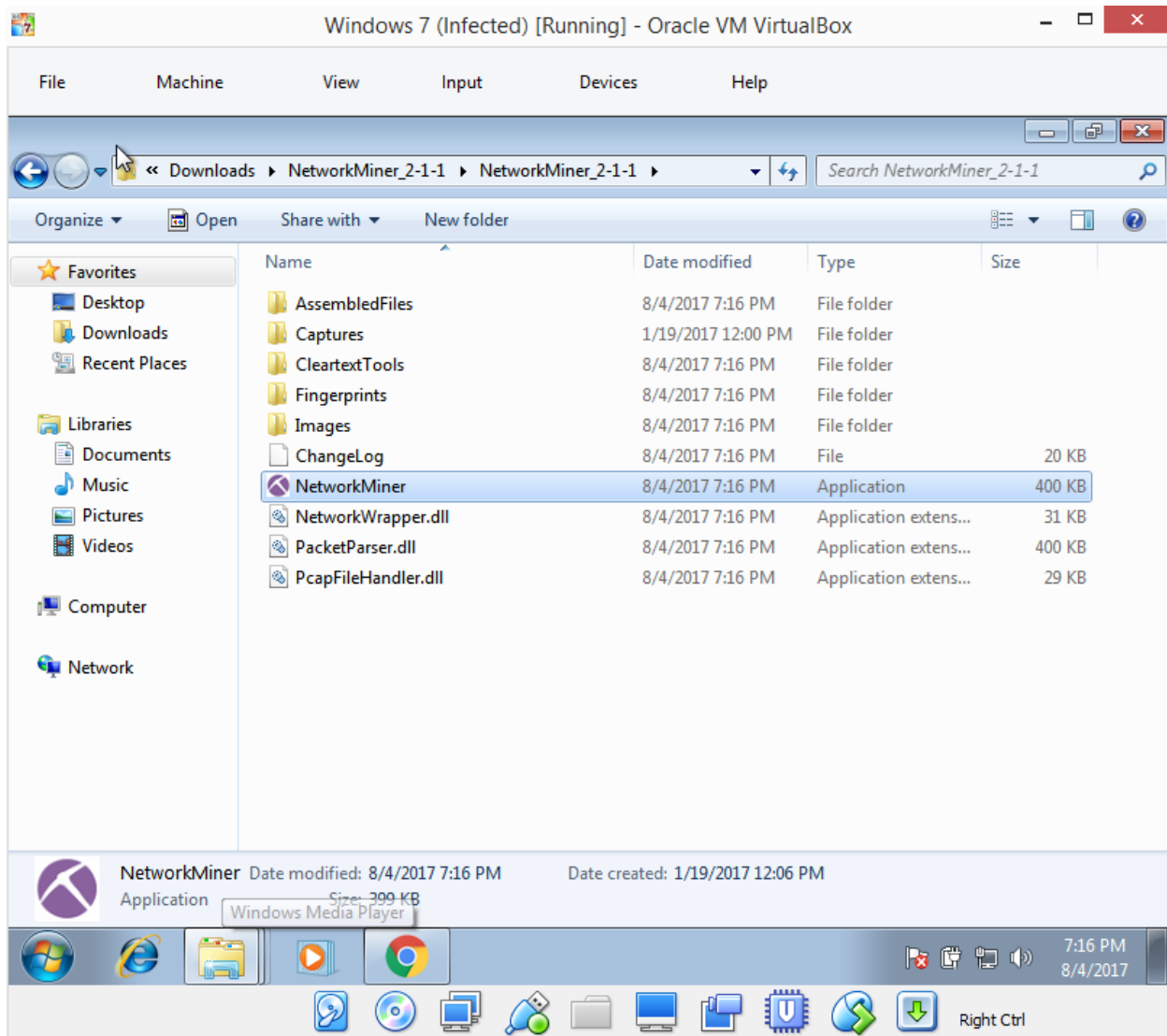
Open Windows Virtual Machine



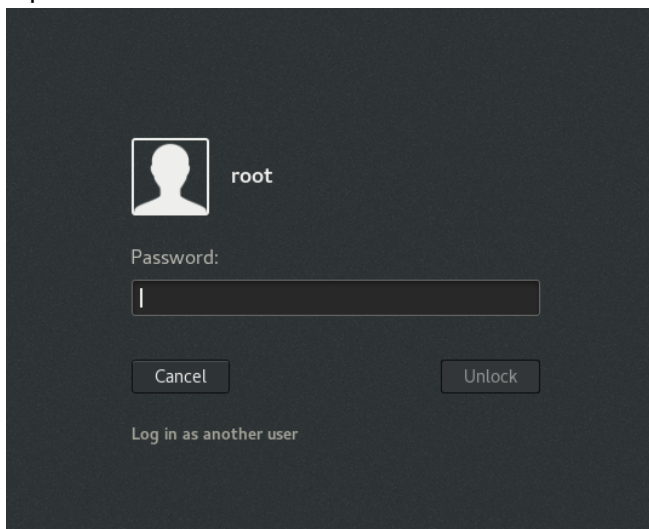
Check for Network Miner

- Click on the windows button in the left-hand corner and start typing the word “network”. If you see the Network Miner program with the purple icon, great.
- If it is not listed, download the free version from: <http://www.netresec.com/?page=NetworkMiner> and install.





Open Kali Virtual Machine



If you end up on the root sign-in page and do not know the password, try the default password: toor.

Once you are logged in, open a terminal and check if the snort tool is installed.

```
root@kali> snort -V
```

If not, install it

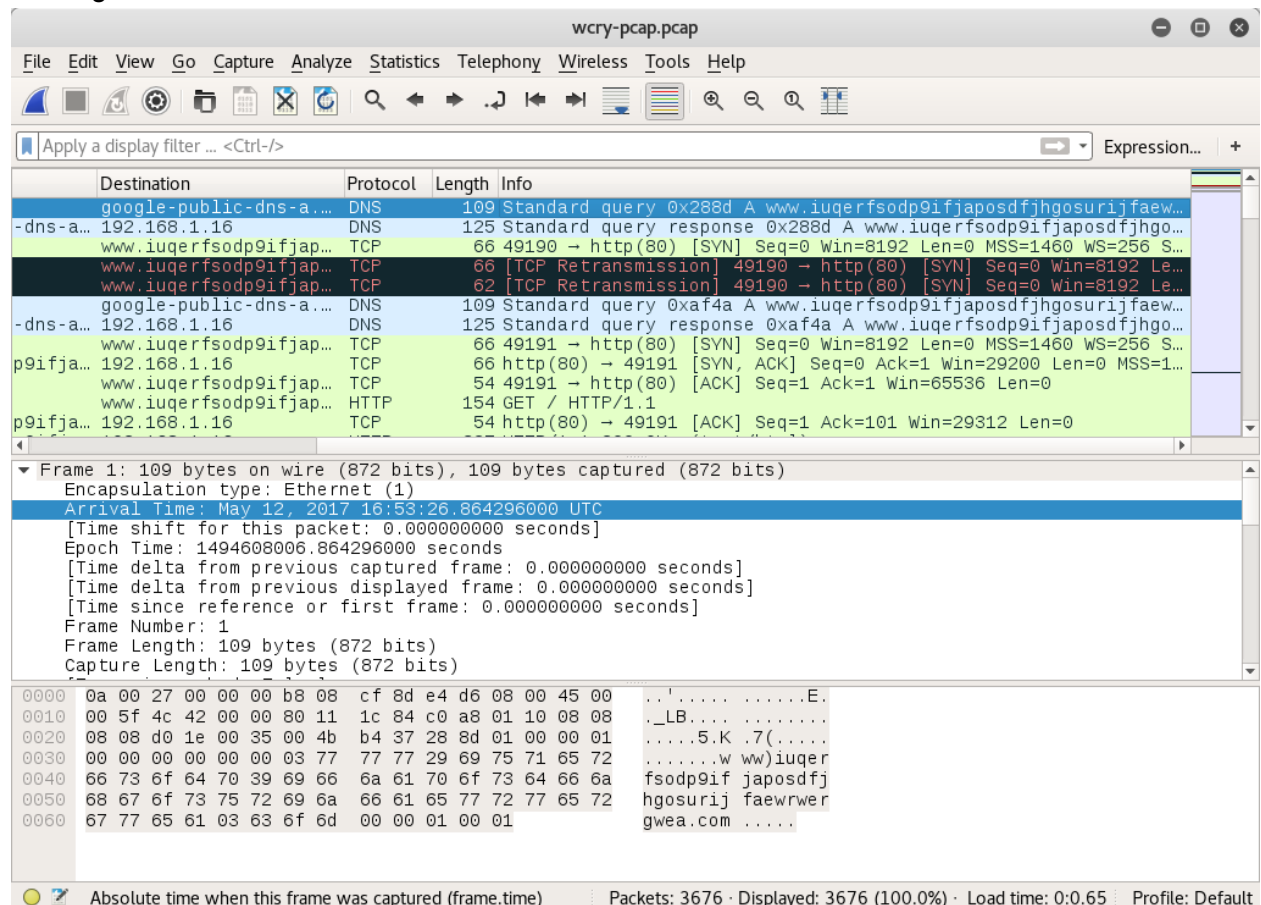
```
root@kali > apt-get install snort
```

Check for successful installation by getting the version

```
root@kali > snort -V
```

Analysis

Finding basic information:



The screenshot shows the Wireshark interface with a packet capture of a DNS query and response. The packet list shows a DNS query from 192.168.1.16 to google-public-dns-a... and a response from google-public-dns-a... to 192.168.1.16. The packet details pane shows the structure of the DNS packet, including the query type and the response data.

Destination	Protocol	Length	Info
google-public-dns-a...	DNS	109	Standard query 0x288d A www.iuqerfsodp9ifjaposdfjhgosurijfaew...
-dns-a... 192.168.1.16	DNS	125	Standard query response 0x288d A www.iuqerfsodp9ifjaposdfjhgo...
www.iuqerfsodp9ifjap...	TCP	66	49190 → http(80) [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 S...
www.iuqerfsodp9ifjap...	TCP	66	[TCP Retransmission] 49190 → http(80) [SYN] Seq=0 Win=8192 Le...
www.iuqerfsodp9ifjap...	TCP	62	[TCP Retransmission] 49190 → http(80) [SYN] Seq=0 Win=8192 Le...
google-public-dns-a...	DNS	109	Standard query 0xaf4a A www.iuqerfsodp9ifjaposdfjhgosurijfaew...
-dns-a... 192.168.1.16	DNS	125	Standard query response 0xaf4a A www.iuqerfsodp9ifjaposdfjhgo...
www.iuqerfsodp9ifjap...	TCP	66	49191 → http(80) [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 S...
p9ifja... 192.168.1.16	TCP	66	http(80) → 49191 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1...
www.iuqerfsodp9ifjap...	TCP	54	49191 → http(80) [ACK] Seq=1 Ack=1 Win=65536 Len=0
www.iuqerfsodp9ifjap...	HTTP	154	GET / HTTP/1.1
p9ifja... 192.168.1.16	TCP	54	http(80) → 49191 [ACK] Seq=1 Ack=101 Win=29312 Len=0

Frame 1: 109 bytes on wire (872 bits), 109 bytes captured (872 bits)
Encapsulation type: Ethernet (1)
Arrival Time: May 12, 2017 16:53:26.864296000 UTC
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1494608006.864296000 seconds
[Time delta from previous captured frame: 0.000000000 seconds]
[Time delta from previous displayed frame: 0.000000000 seconds]
[Time since reference or first frame: 0.000000000 seconds]
Frame Number: 1
Frame Length: 109 bytes (872 bits)
Capture Length: 109 bytes (872 bits)

0000 0a 00 27 00 00 00 b8 08 cf 8d e4 d6 08 00 45 00 ..'.....E.
0010 00 5f 4c 42 00 00 80 11 1c 84 c0 a8 01 10 08 08 _LB.....
0020 08 08 d0 1e 00 35 00 4b b4 37 28 8d 01 00 00 015.K.7(.....
0030 00 00 00 00 00 00 03 77 77 77 29 69 75 71 65 72w ww)iuqer
0040 66 73 6f 64 70 39 69 66 6a 61 70 6f 73 64 66 6a fsodp9if japosdfj
0050 68 67 6f 73 75 72 69 6a 66 61 65 77 72 77 65 72 hgoserij faewrwer
0060 67 77 65 61 03 63 6f 6d 00 00 01 00 01 gwea.com

- The packet capture started on May 12, 2017 16:53 and ended on [INSERT END TIME HERE]

Pcap files are usually large and have a large amount of data to sort through. Because of that it helps to break the information into chunks for analysis.

Identifying alerts with snort

- The snort tool is an intrusion detection system that can identify suspicious activity by inspecting network traffic. We can use snort to identify areas of the .pcap file that might be useful for further analysis.

Run time for packet processing was 1.10721 seconds

Snort processed 3676 packets.

Snort ran for 0 days 0 hours 0 minutes 1 seconds

Pkts/sec: 3676

=====

Memory usage summary:

Total non-mmapped bytes (arena):	2244608
Bytes in mapped regions (hblkhd):	12906496
Total allocated space (uordblks):	1976880
Total free space (fordblks):	267728
Topmost releasable block (keepcost):	56416

=====

Packet I/O Totals:

Received:	3676
Analyzed:	3676 (100.000%)
Dropped:	0 (0.000%)
Filtered:	0 (0.000%)
Outstanding:	0 (0.000%)
Injected:	0

=====

Breakdown by protocol (includes rebuilt packets):

Eth:	3676 (100.000%)
VLAN:	0 (0.000%)
IP4:	3676 (100.000%)
Frag:	0 (0.000%)
ICMP:	0 (0.000%)
UDP:	4 (0.109%)
TCP:	3672 (99.891%)
IP6:	0 (0.000%)
IP6 Ext:	0 (0.000%)
IP6 Opts:	0 (0.000%)
Frag6:	0 (0.000%)
ICMP6:	0 (0.000%)
UDP6:	0 (0.000%)
TCP6:	0 (0.000%)
Teredo:	0 (0.000%)
ICMP-IP:	0 (0.000%)
IP4/IP4:	0 (0.000%)
IP4/IP6:	0 (0.000%)
IP6/IP4:	0 (0.000%)
IP6/IP6:	0 (0.000%)
GRE:	0 (0.000%)
GRE Eth:	0 (0.000%)
GRE VLAN:	0 (0.000%)
GRE IP4:	0 (0.000%)
GRE IP6:	0 (0.000%)
GRE IP6 Ext:	0 (0.000%)
GRE PPTP:	0 (0.000%)
GRE ARP:	0 (0.000%)
GRE IPX:	0 (0.000%)
GRE Loop:	0 (0.000%)

MPLS:	0 (0.000%)
ARP:	0 (0.000%)
IPX:	0 (0.000%)
Eth Loop:	0 (0.000%)
Eth Disc:	0 (0.000%)
IP4 Disc:	0 (0.000%)
IP6 Disc:	0 (0.000%)
TCP Disc:	0 (0.000%)
UDP Disc:	0 (0.000%)
ICMP Disc:	0 (0.000%)
All Discard:	0 (0.000%)
Other:	0 (0.000%)
Bad Chk Sum:	115 (3.128%)
Bad TTL:	0 (0.000%)
S5 G 1:	0 (0.000%)
S5 G 2:	0 (0.000%)
Total:	3676
=====	
Action Stats:	
Alerts:	0 (0.000%)
Logged:	0 (0.000%)
Passed:	0 (0.000%)
Limits:	
Match:	0
Queue:	0
Log:	0
Event:	0
Alert:	0
Verdicts:	
Allow:	3676 (100.000%)
Block:	0 (0.000%)
Replace:	0 (0.000%)
Whitelist:	0 (0.000%)
Blacklist:	0 (0.000%)
Ignore:	0 (0.000%)
Retry:	0 (0.000%)

- Snort returns no alerts for suspicious activity so we move on

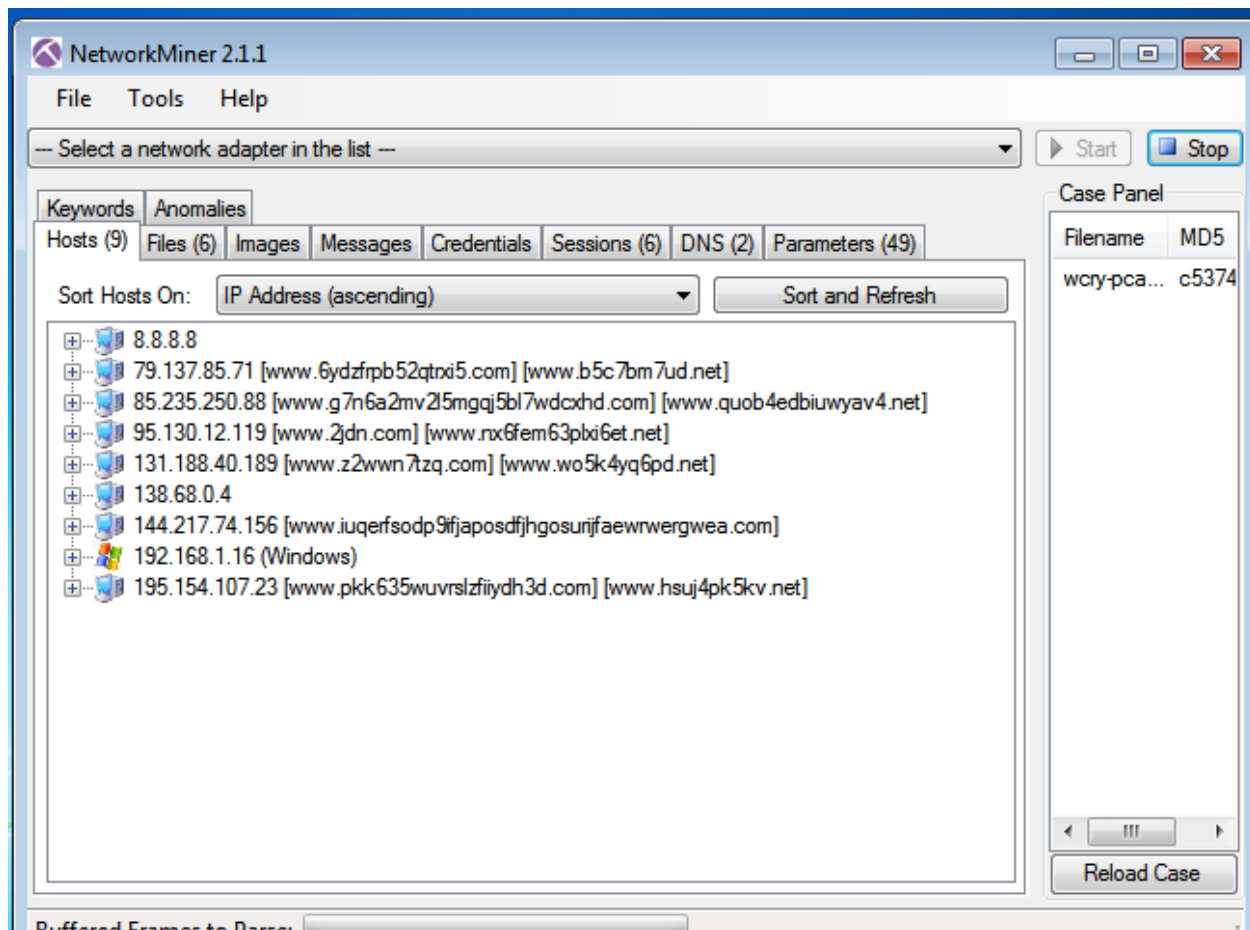
File carving using foremost

- Use file carving to extract any files that may have been transferred during the packet capture

```
root@kali:/# foremost ~/Downloads/wcry-pcap.pcap
Processing: /root/Downloads/wcry-pcap.pcap
|*|
```

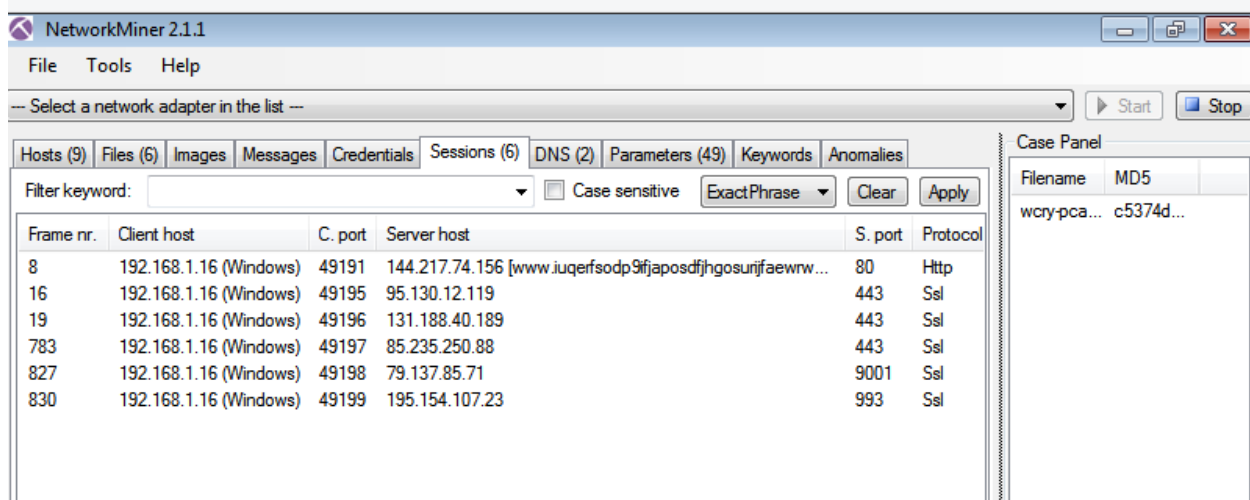
Easy analysis with Network Miner (Windows)

- In addition to the foremost tool we can use Network Miner to extract files and other information about the network capture. Open Network Miner. Open wcry-pcap.pcap. The first thing we notice is that the domain names look strange.



- Foremost carved [X number of FILES], Network Miner shows six. Check the MD5 Checksum of each file and check Virus Total for any matches to known malware.
- [INSERT FILE PAGE HERE]

Identify TCP Conversations with Network Miner



- Taking this information from Network Miner we can use Wireshark for deeper analysis. Start with frame 8.

Traffic Analysis with Wireshark

Wireshark interface showing a packet capture of a TCP stream. The packet list displays frames 8 through 13. Frame 10 is selected, showing its details in the packet details pane. The packet bytes pane shows the raw data of frame 10.

No.	Time	Source	Destination	Protocol	Length	Info
8	22.015763	192.168.1.16	www.iuqerfsodp9ifjapos...	TCP	66	49191 → http(80) [SYN] Seq=0 Win=819
10	22.015873	192.168.1.16	www.iuqerfsodp9ifjapos...	TCP	54	49191 → http(80) [ACK] Seq=1 Ack=1 W
11	22.016394	192.168.1.16	www.iuqerfsodp9ifjapos...	HTTP	154	GET / HTTP/1.1
14	22.281267	192.168.1.16	www.iuqerfsodp9ifjapos...	TCP	54	49191 → http(80) [ACK] Seq=101 Ack=2
15	22.518799	192.168.1.16	www.iuqerfsodp9ifjapos...	TCP	54	49191 → http(80) [RST, ACK] Seq=101
9	22.015796	www.iuqerfsodp9i...	192.168.1.16	TCP	66	http(80) → 49191 [SYN, ACK] Seq=0 Ac
12	22.016420	www.iuqerfsodp9i...	192.168.1.16	TCP	54	http(80) → 49191 [ACK] Seq=1 Ack=101
13	22.281210	www.iuqerfsodp9i...	192.168.1.16	HTTP	327	HTTP/1.1 200 OK (text/html)

Frame 10: 54 bytes on wire (432 bits), 54 bytes captured (432 bits)
Encapsulation type: Ethernet (1)
Arrival Time: May 12, 2017 16:53:48.880169000 UTC
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1494608028.880169000 seconds
[Time delta from previous captured frame: 0.000077000 seconds]
[Time delta from previous displayed frame: 0.000077000 seconds]
[Time since reference or first frame: 22.015873000 seconds]
Frame Number: 10
Frame Length: 54 bytes (432 bits)
Captured Length: 54 bytes (432 bits)

0000 0a 00 27 00 00 00 b8 08 cf 8d e4 d6 08 00 45 00 ...E.
0010 00 28 41 6a 40 00 80 06 1c 38 c0 a8 01 10 90 d9 ... (Aj@... .8.....
0020 4a 9c c0 27 00 50 59 99 05 85 0c ac d8 d3 50 10 J...PY.P.
0030 01 00 0c 91 00 00

Epoch time when this frame was captured (frame.time_epoch) Packets: 3676 · Displayed: 8 (0.2%) · Load time: 0:0.85 Profile: Default

The TCP stream that begins with frame 8 shows the communication between the host and a strange website.

The next TCP stream we are interested in begins in frame 16.

wcry-pcap.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.stream eq 2 Expression...

No.	Time	Source	Destination	Protocol	Length	Info
16	59.968537	192.168.1.16	digi02080.digicube.fr	TCP	66	49195 → https(443) [SYN] Seq
18	59.968641	192.168.1.16	digi02080.digicube.fr	TCP	54	49195 → https(443) [ACK] Seq
22	60.188361	192.168.1.16	digi02080.digicube.fr	TLSv1.2	258	Client Hello
30	60.320411	192.168.1.16	digi02080.digicube.fr	TLSv1.2	180	Client Key Exchange, Change
36	60.402479	192.168.1.16	digi02080.digicube.fr	TLSv1.2	92	Application Data
45	60.512067	192.168.1.16	digi02080.digicube.fr	TCP	54	49195 → https(443) [ACK] Seq
46	60.513510	192.168.1.16	digi02080.digicube.fr	TLSv1.2	1111	Application Data
169	60.808377	192.168.1.16	digi02080.digicube.fr	TCP	66	49195 → https(443) [ACK] Seq
3589	127.956797	192.168.1.16	digi02080.digicube.fr	TCP	54	49195 → https(443) [ACK] Seq
17	59.968573	digi02080.digicube.fr	192.168.1.16	TCP	66	https(443) → 49195 [SYN, AC
23	60.188410	digi02080.digicube.fr	192.168.1.16	TCP	54	https(443) → 49195 [ACK] Seq
29	60.315996	digi02080.digicube.fr	192.168.1.16	TLSv1.2	811	Server Hello, Certificate,

Frame 16: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)

Encapsulation type: Ethernet (1)

Arrival Time: May 12, 2017 16:54:26.832833000 UTC

[Time shift for this packet: 0.000000000 seconds]

Epoch Time: 1494608066.832833000 seconds

[Time delta from previous captured frame: 37.449738000 seconds]

[Time delta from previous displayed frame: 0.000000000 seconds]

[Time since reference or first frame: 59.968537000 seconds]

Frame Number: 16

Frame Length: 66 bytes (528 bits)

Capture length: 66 bytes (528 bits)

0000	0a 00 27 00 00 00 b8 08	cf 8d e4 d6 08 00 45 00	..4...@...4...E.
0010	00 34 1d de 40 00 80 06	af 34 c0 a8 01 10 5f 82	..4...@...4...E.
0020	0c 77 c0 2b 01 bb 17 aa	ab ce 00 00 00 00 80 02	..w.+... ..
0030	20 00 9b ff 00 00 02 04	05 b4 01 03 03 08 01 01
0040	04 02		..

Frame (frame), 66 bytes

Packets: 3676 · Displayed: 21 (0.6%) · Load time: 0:0.90 · Profile: Default