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:

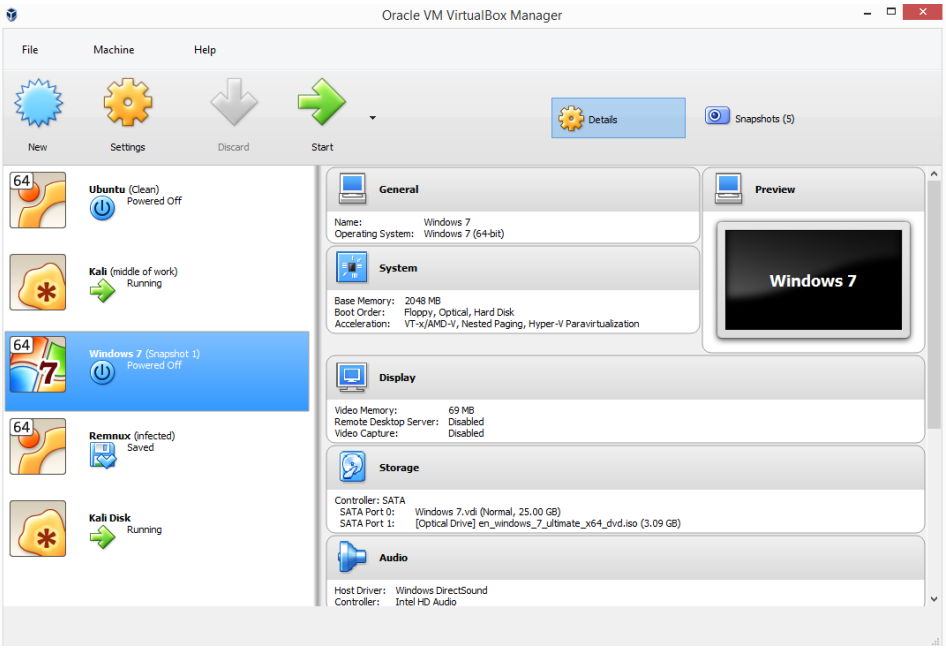
- When did the capture start and end?
- Which OS is the scanned machine running?
- How many frames are in the first TCP stream?
- What domain was the host querying?
- Originally the odd domain was not registered and was therefore not returning any results. To stop the worm from spreading Malware Tech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

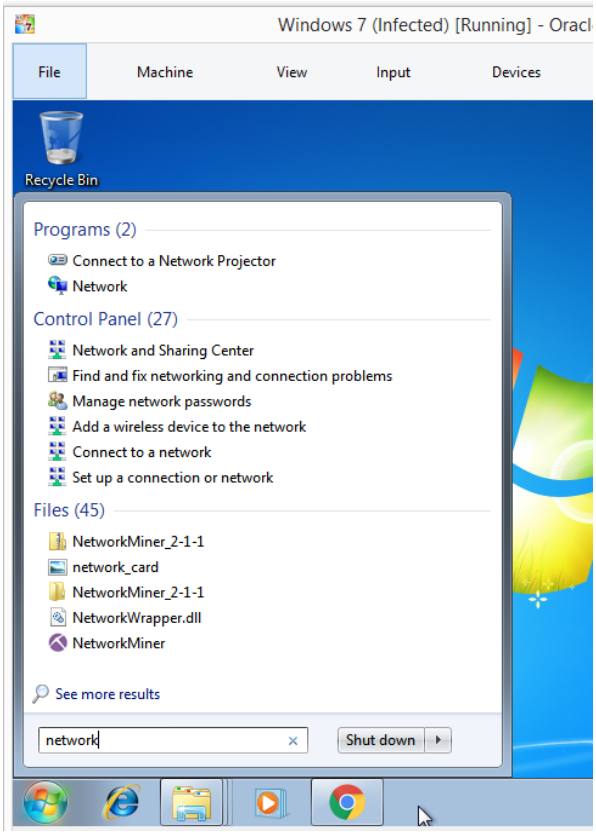
- Snort (Kali)
- Network Miner (Windows)

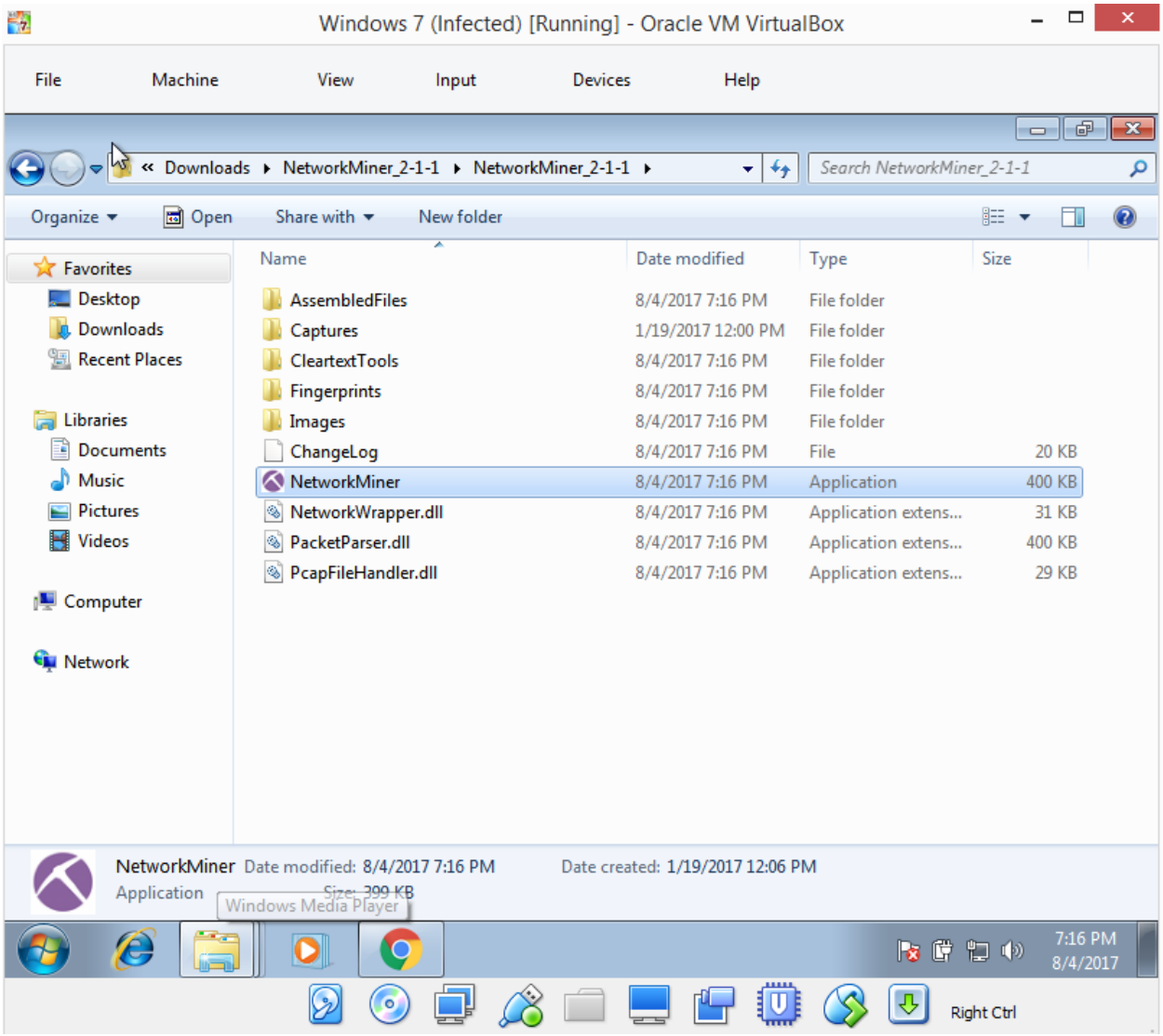
Open Windows Virtual Machine



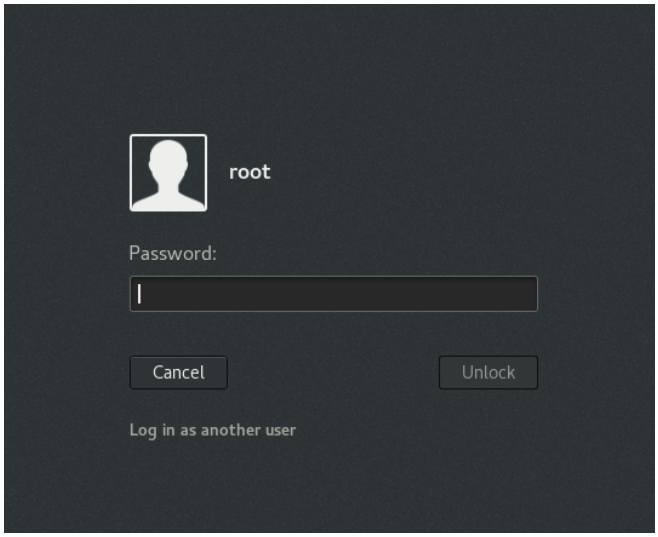
Check for Network Miner

- 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.





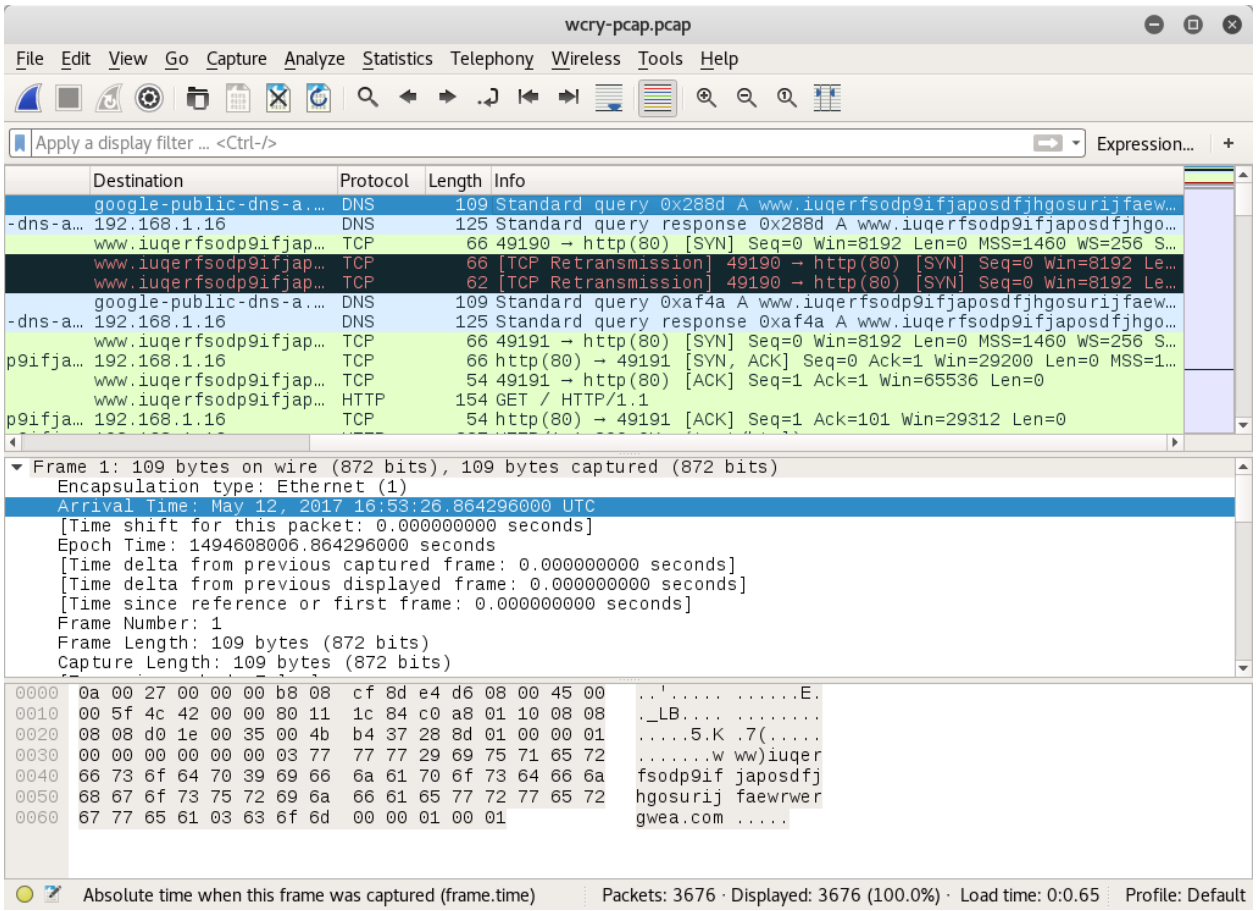
Open Kali Virtual Machine



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

<u>Analysis</u>

Finding basic information:



- 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

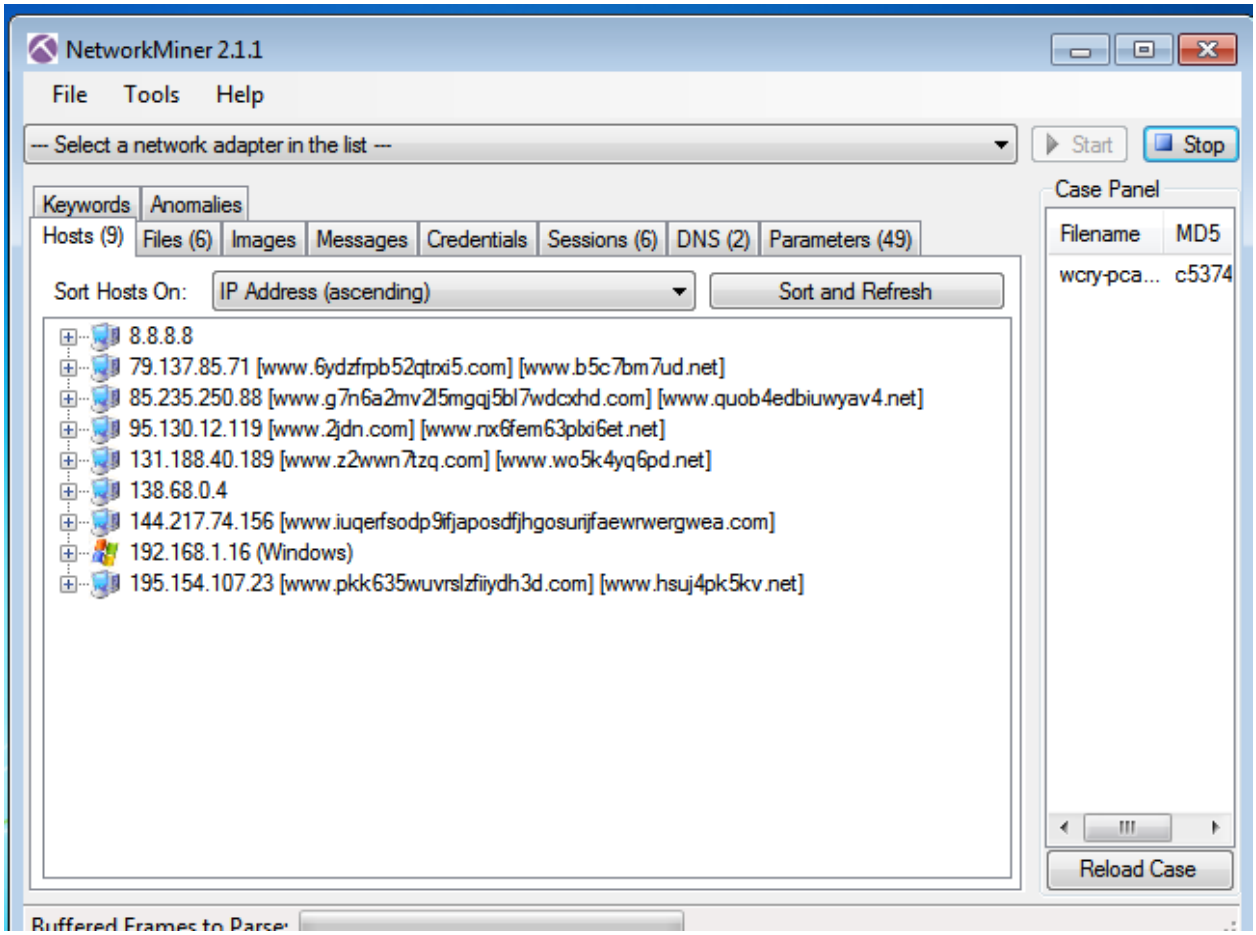
File carving using foremost

- 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
|*|
```

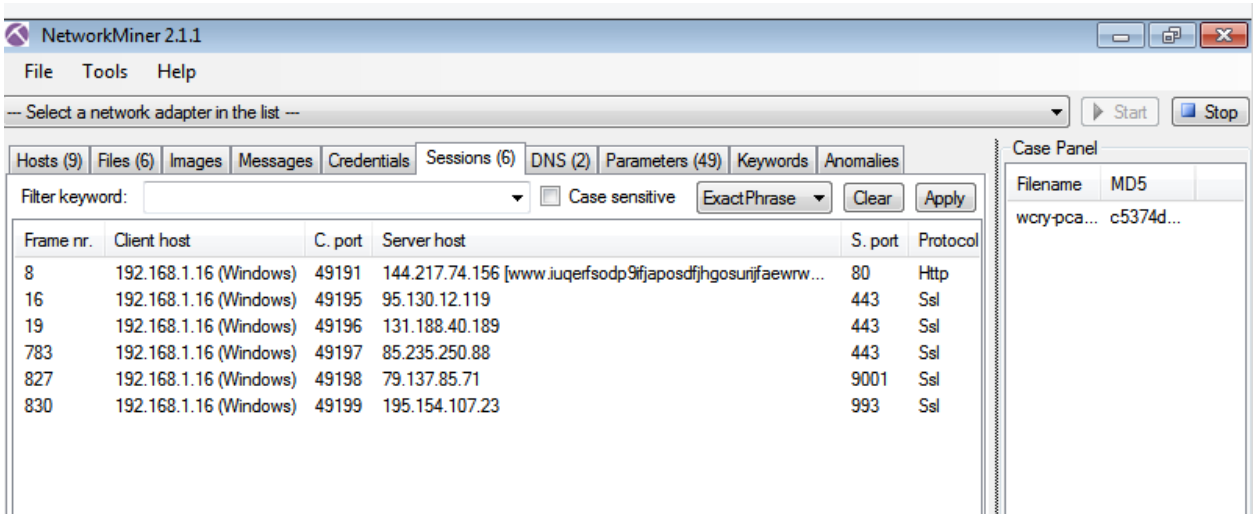
Easy analysis with Network Miner (Windows)

- 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.



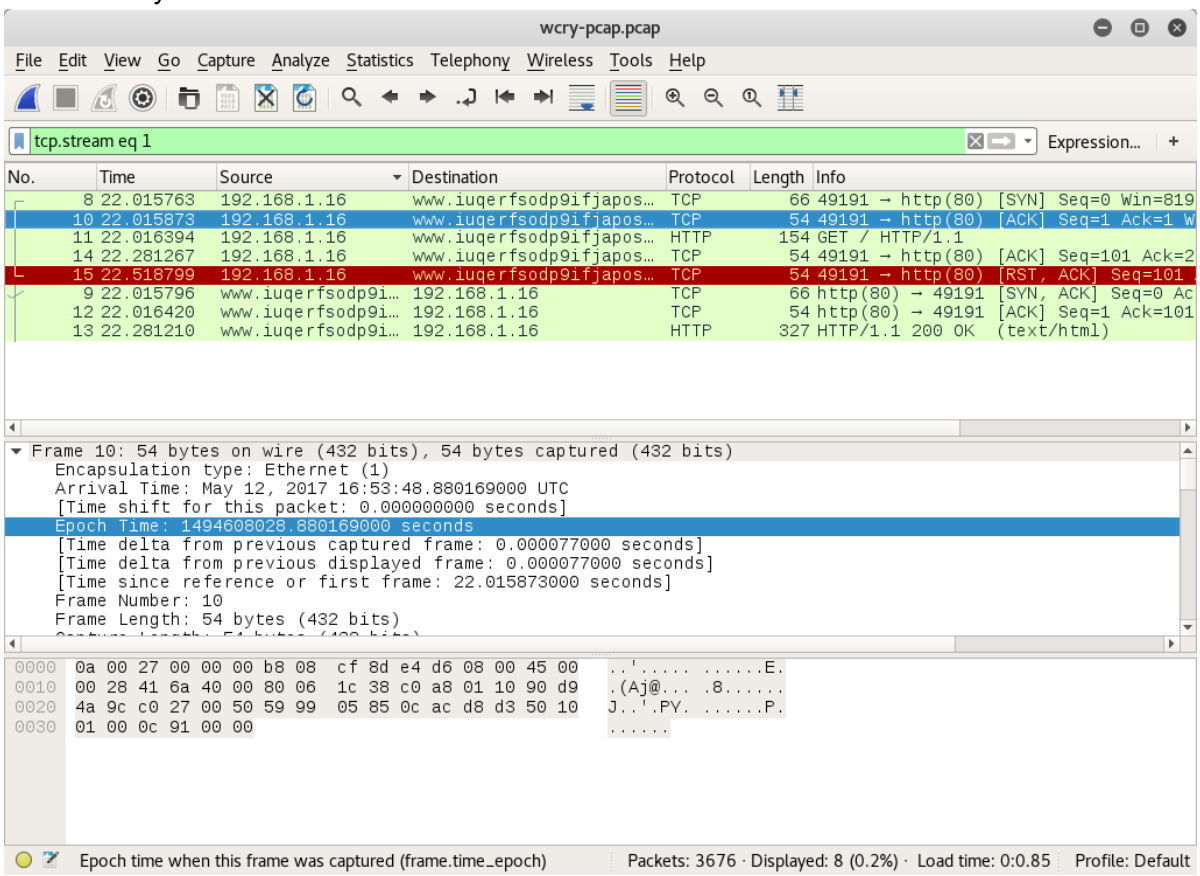
- Foremost carved [X number of FILES], Network Miner shows six. Check the MD5 Checksum of each file and check Virus Total for any matches to known malware.
- [INSERT FILE PAGE HERE]

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



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] Se
18	59.968641	192.168.1.16	digi02080.digicube.fr	TCP	54	49195 → https(443) [ACK] Se
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] Se
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] Se
3589	127.956797	192.168.1.16	digi02080.digicube.fr	TCP	54	49195 → https(443) [ACK] Se
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] Se
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)
```

```
0000  0a 00 27 00 00 00 b8 08  cf 8d e4 d6 08 00 45 00   ..'..... ......E.
0010  00 34 1d de 40 00 80 06  af 34 c0 a8 01 10 5f 82   .4..@... .4...._.
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