IDENTIFYING MALWARE ATTACKS USING WIRESHARK

What is Malware?

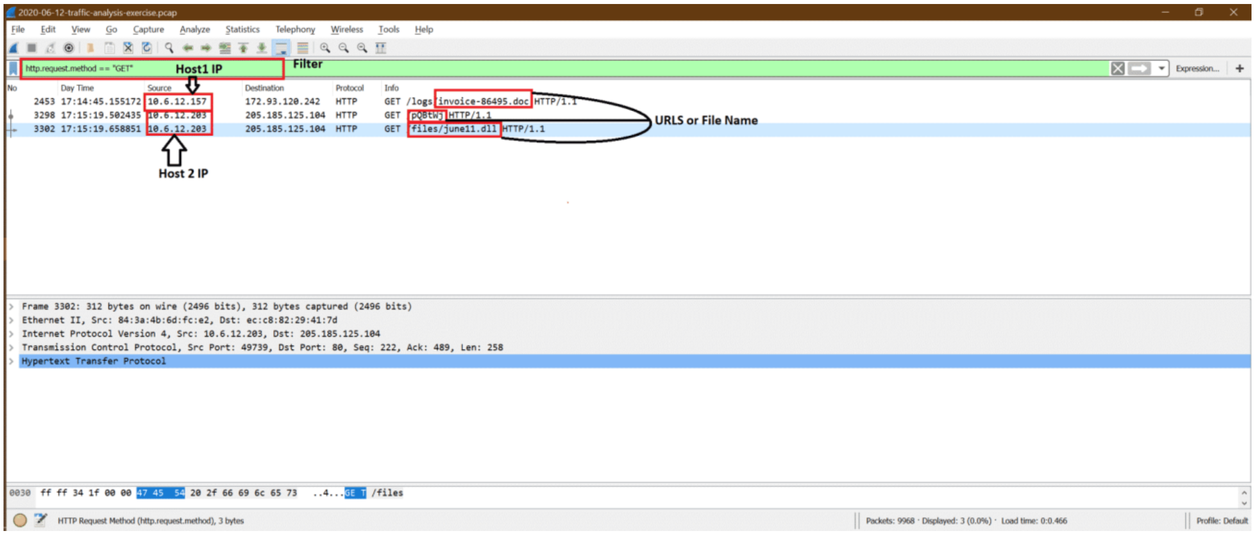
Malware words came from Malicious Software**.** We can think of Malware as a piece of code or software that is designed to do some damage on systems. Trojans, Spyware, Viruses, ransomware are different types of malware.
There are many ways malware gets into the system. We will take one scenario and try to understand it from Wireshark capture.

Scenario:

Here in example capture, we have two windows systems with IP address as 10.6.12.157 and 10.6.12.203. These hosts are communicating with the internet. We can see some HTTP, GET, POST, etc. operations. Let's find out which windows system got infected, or both got infected.

Step1:

Let's see some HTTP communication by these hosts. After using the below the filter, we can see all HTTP GET request in the capture **"http.request.method == GET"** Here is the screenshot to explain the content after the filter.



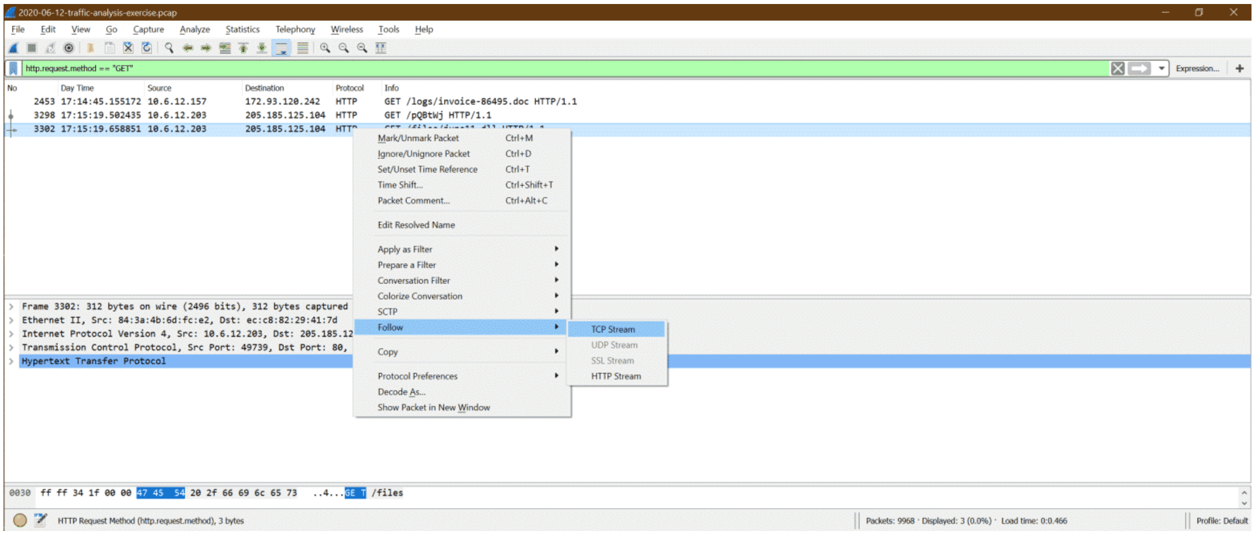
2020-06-12-traffic-analysis-exercise.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...
Host1 IP
Filter
URLS or File Name
Host 2 IP

http.request.method == "GET"

```
No      Day Time         Source          Destination       Protocol  Info
2453  17:14:45.155172  10.6.12.157     172.93.120.242    HTTP      GET /logs/invoice-86495.doc HTTP/1.1
3298  17:15:19.502435  10.6.12.203     205.185.125.104   HTTP      GET pQBtWj HTTP/1.1
3302  17:15:19.658851  10.6.12.203     205.185.125.104   HTTP      GET files/june11.dll HTTP/1.1

> Frame 3302: 312 bytes on wire (2496 bits), 312 bytes captured (2496 bits)
> Ethernet II, Src: 84:3a:4b:6d:fc:e2, Dst: ec:c8:82:29:41:7d
> Internet Protocol Version 4, Src: 10.6.12.203, Dst: 205.185.125.104
> Transmission Control Protocol, Src Port: 49739, Dst Port: 80, Seq: 222, Ack: 489, Len: 258
> Hypertext Transfer Protocol

0030  ff ff 34 1f 00 00 47 45  54 20 2f 66 69 6c 65 73   ..4...GE T /files

HTTP Request Method (http.request.method), 3 bytes    Packets: 9968 · Displayed: 3 (0.0%) · Load time: 0:0.466    Profile: Default
```

Step2:

Now out of these, the suspicious one is GET request from 10.6.12.203, so we can follow TCP stream [see below screenshot] to find out the more clearly.


2020-06-12-traffic-analysis-exercise.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression…
Mark/Unmark Packet Ctrl+M
Ignore/Unignore Packet Ctrl+D
Set/Unset Time Reference Ctrl+T
Time Shift… Ctrl+Shift+T
Packet Comment… Ctrl+Alt+C
Edit Resolved Name
Apply as Filter
Prepare a Filter
Conversation Filter
Colorize Conversation
SCTP
Follow
Copy
Protocol Preferences
Decode As…
Show Packet in New Window
TCP Stream
UDP Stream
SSL Stream
HTTP Stream

http.request.method == "GET"

```
No      Day Time         Source        Destination      Protocol  Info
   2453 17:14:45.155172 10.6.12.157   172.93.120.242   HTTP      GET /logs/invoice-86495.doc HTTP/1.1
   3298 17:15:19.502435 10.6.12.203   205.185.125.104  HTTP      GET /pQBtWj HTTP/1.1
   3302 17:15:19.658851 10.6.12.203   205.185.125.104  HTTP
```

```
> Frame 3302: 312 bytes on wire (2496 bits), 312 bytes captured
> Ethernet II, Src: 84:3a:4b:6d:fc:e2, Dst: ec:c8:82:29:41:7d
> Internet Protocol Version 4, Src: 10.6.12.203, Dst: 205.185.12
> Transmission Control Protocol, Src Port: 49739, Dst Port: 80,
> Hypertext Transfer Protocol
```

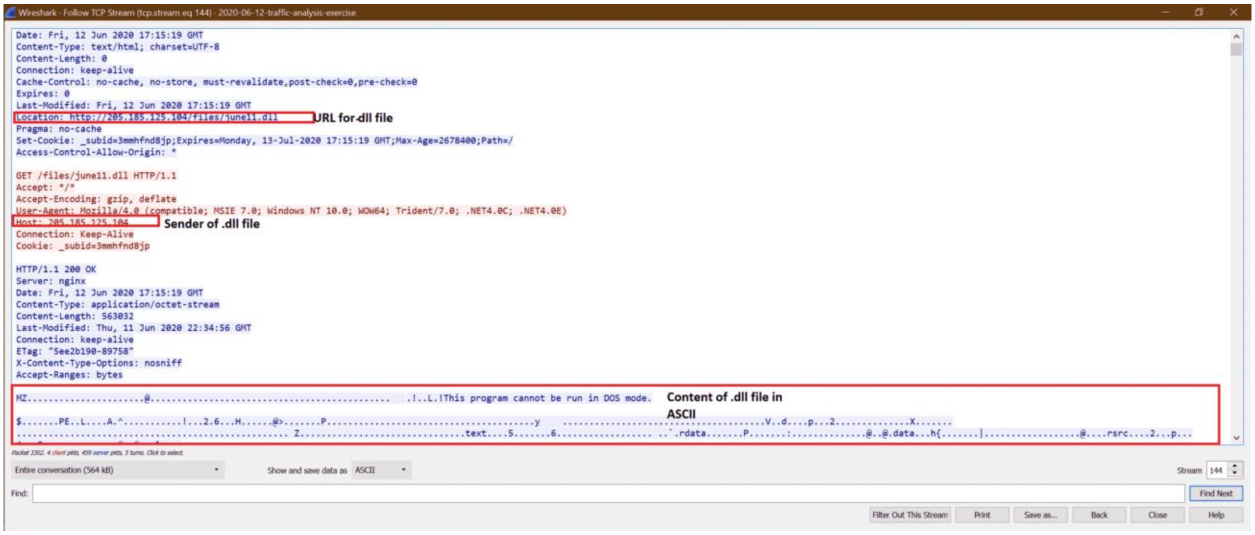
```
0030  ff ff 34 1f 00 00 47 45  54 20 2f 66 69 6c 65 73   ..4...GE T /files
```

HTTP Request Method (http.request.method), 3 bytes

Packets: 9968 · Displayed: 3 (0.0%) · Load time: 0:0.466

Profile: Default

Here are the findings from following TCP stream


Wireshark - Follow TCP Stream (tcp.stream eq 144) - 2020-06-12-traffic-analysis-exercise
Filter Out This Stream
Print
Save as…
Back
Close
Help
Find Next

```
Date: Fri, 12 Jun 2020 17:15:19 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 0
Connection: keep-alive
Cache-Control: no-cache, no-store, must-revalidate,post-check=0,pre-check=0
Expires: 0
Last-Modified: Fri, 12 Jun 2020 17:15:19 GMT
Location: http://205.185.125.104/files/june11.dll
Pragma: no-cache
Set-Cookie: _subid=3mmhfnd8jp;Expires=Monday, 13-Jul-2020 17:15:19 GMT;Max-Age=2678400;Path=/
Access-Control-Allow-Origin: *

GET /files/june11.dll HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 10.0; WOW64; Trident/7.0; .NET4.0C; .NET4.0E)
Host: 205.185.125.104
Connection: Keep-Alive
Cookie: _subid=3mmhfnd8jp

HTTP/1.1 200 OK
Server: nginx
Date: Fri, 12 Jun 2020 17:15:19 GMT
Content-Type: application/octet-stream
Content-Length: 563032
Last-Modified: Thu, 11 Jun 2020 22:34:56 GMT
Connection: keep-alive
ETag: "5ee2b190-89758"
X-Content-Type-Options: nosniff
Accept-Ranges: bytes

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

$.......PE..L....A.^...........!...2.6...H......@>.......P.....................................y    ...................................V..d....p...2..............X.......
.......................................... Z..............................text....5.......6.................. ..`.rdata.......P.......:..............@..@.data...h{.......|.................@....rsrc....2...p...
```

URL for dll file

Sender of .dll file

Content of .dll file in ASCII

Packet 3302. 4 client pkts, 459 server pkts, 5 turns. Click to select.

Entire conversation (564 kB)

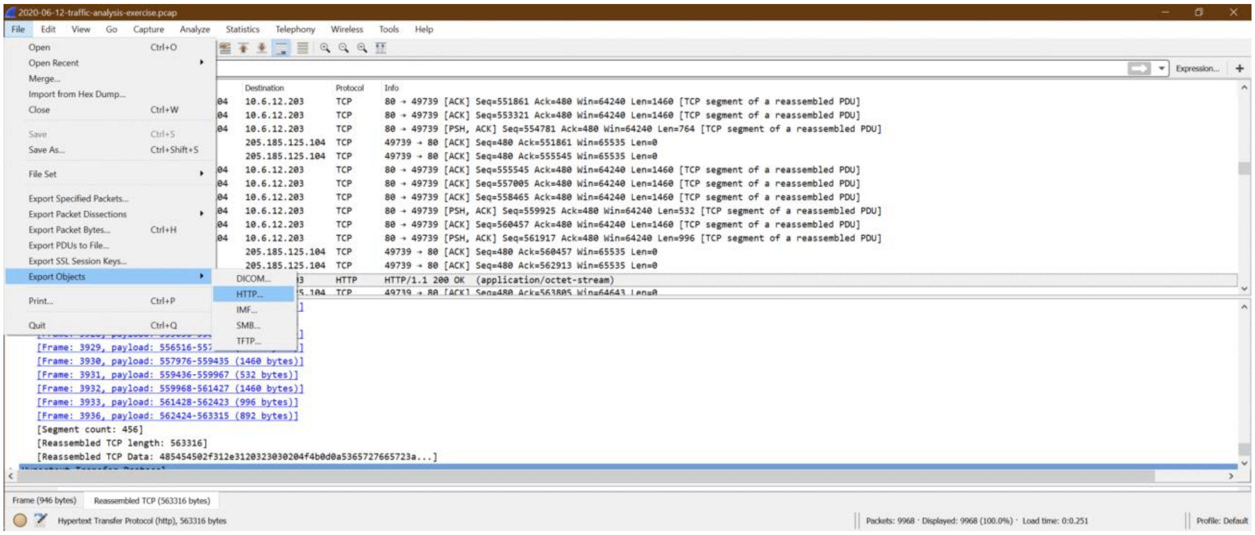
Show and save data as ASCII

Stream 144

Find:

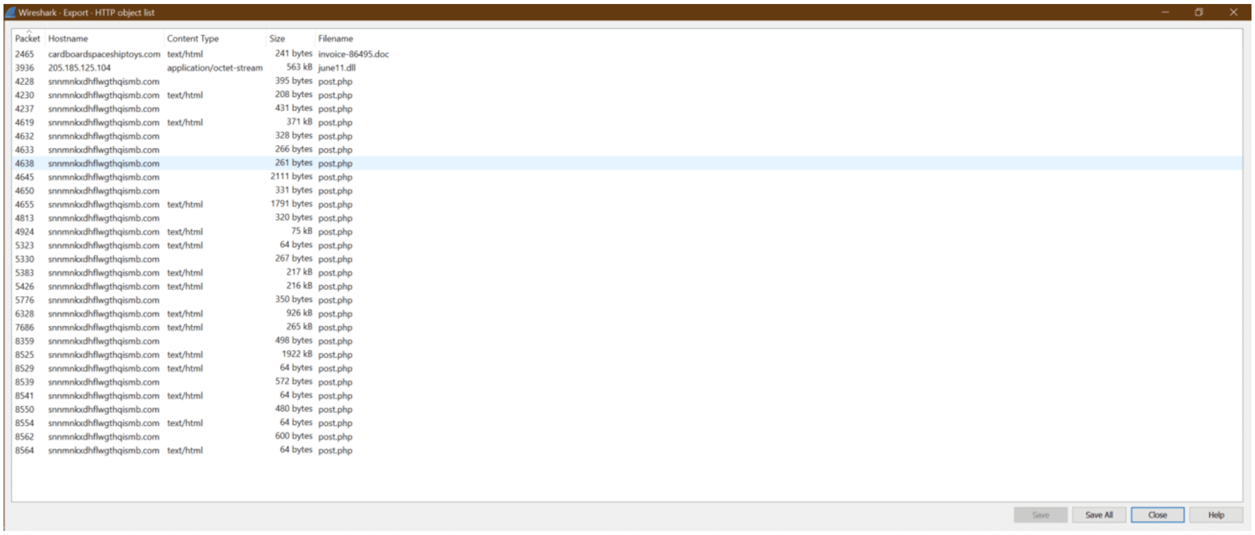
Step3:

Now we can try exporting this **june11.dll** file from pcap. Follow the below screenshot steps a.



```
    Destination      Protocol  Info
04  10.6.12.203      TCP       80 → 49739 [ACK] Seq=551861 Ack=480 Win=64240 Len=1460 [TCP segment of a reassembled PDU]
04  10.6.12.203      TCP       80 → 49739 [ACK] Seq=553321 Ack=480 Win=64240 Len=1460 [TCP segment of a reassembled PDU]
04  10.6.12.203      TCP       80 → 49739 [PSH, ACK] Seq=554781 Ack=480 Win=64240 Len=764 [TCP segment of a reassembled PDU]
    205.185.125.104  TCP       49739 → 80 [ACK] Seq=480 Ack=551861 Win=65535 Len=0
    205.185.125.104  TCP       49739 → 80 [ACK] Seq=480 Ack=555545 Win=65535 Len=0
04  10.6.12.203      TCP       80 → 49739 [ACK] Seq=555545 Ack=480 Win=64240 Len=1460 [TCP segment of a reassembled PDU]
04  10.6.12.203      TCP       80 → 49739 [ACK] Seq=557005 Ack=480 Win=64240 Len=1460 [TCP segment of a reassembled PDU]
04  10.6.12.203      TCP       80 → 49739 [ACK] Seq=558465 Ack=480 Win=64240 Len=1460 [TCP segment of a reassembled PDU]
04  10.6.12.203      TCP       80 → 49739 [PSH, ACK] Seq=559925 Ack=480 Win=64240 Len=532 [TCP segment of a reassembled PDU]
04  10.6.12.203      TCP       80 → 49739 [ACK] Seq=560457 Ack=480 Win=64240 Len=1460 [TCP segment of a reassembled PDU]
04  10.6.12.203      TCP       80 → 49739 [PSH, ACK] Seq=561917 Ack=480 Win=64240 Len=996 [TCP segment of a reassembled PDU]
    205.185.125.104  TCP       49739 → 80 [ACK] Seq=480 Ack=560457 Win=65535 Len=0
    205.185.125.104  TCP       49739 → 80 [ACK] Seq=480 Ack=562913 Win=65535 Len=0
   3                 HTTP      HTTP/1.1 200 OK  (application/octet-stream)
   5.104             TCP       49739 → 80 [ACK] Seq=480 Ack=563805 Win=64643 Len=0

[Frame: 3929, payload: 556516-557
[Frame: 3930, payload: 557976-559435 (1460 bytes)]
[Frame: 3931, payload: 559436-559967 (532 bytes)]
[Frame: 3932, payload: 559968-561427 (1460 bytes)]
[Frame: 3933, payload: 561428-562423 (996 bytes)]
[Frame: 3936, payload: 562424-563315 (892 bytes)]
[Segment count: 456]
[Reassembled TCP length: 563316]
[Reassembled TCP Data: 485454502f312e3120323030204f4b0d0a5365727665723a...]

Hypertext Transfer Protocol (http), 563316 bytes
Packets: 9968 · Displayed: 9968 (100.0%) · Load time: 0:0.251
```

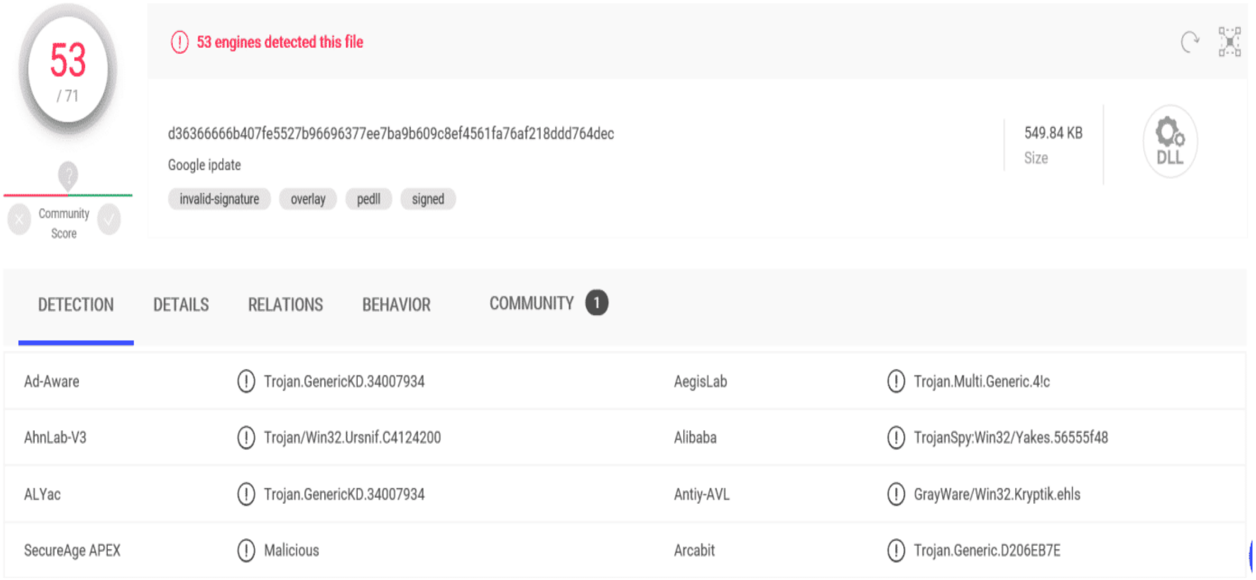
b.



Packet	Hostname	Content Type	Size	Filename
2465	cardboardspaceshiptoys.com	text/html	241 bytes	invoice-86495.doc
3936	205.185.125.104	application/octet-stream	563 kB	june11.dll
4228	snnmnkxdhflwgthqismb.com		395 bytes	post.php
4230	snnmnkxdhflwgthqismb.com	text/html	208 bytes	post.php
4237	snnmnkxdhflwgthqismb.com		431 bytes	post.php
4619	snnmnkxdhflwgthqismb.com	text/html	371 kB	post.php
4632	snnmnkxdhflwgthqismb.com		328 bytes	post.php
4633	snnmnkxdhflwgthqismb.com		266 bytes	post.php
4638	snnmnkxdhflwgthqismb.com		261 bytes	post.php
4645	snnmnkxdhflwgthqismb.com		2111 bytes	post.php
4650	snnmnkxdhflwgthqismb.com		331 bytes	post.php
4655	snnmnkxdhflwgthqismb.com	text/html	1791 bytes	post.php
4813	snnmnkxdhflwgthqismb.com		320 bytes	post.php
4924	snnmnkxdhflwgthqismb.com	text/html	75 kB	post.php
5323	snnmnkxdhflwgthqismb.com	text/html	64 bytes	post.php
5330	snnmnkxdhflwgthqismb.com		267 bytes	post.php
5383	snnmnkxdhflwgthqismb.com	text/html	217 kB	post.php
5426	snnmnkxdhflwgthqismb.com	text/html	216 kB	post.php
5776	snnmnkxdhflwgthqismb.com		350 bytes	post.php
6328	snnmnkxdhflwgthqismb.com	text/html	926 kB	post.php
7686	snnmnkxdhflwgthqismb.com	text/html	265 kB	post.php
8359	snnmnkxdhflwgthqismb.com		498 bytes	post.php
8525	snnmnkxdhflwgthqismb.com	text/html	1922 kB	post.php
8529	snnmnkxdhflwgthqismb.com	text/html	64 bytes	post.php
8539	snnmnkxdhflwgthqismb.com		572 bytes	post.php
8541	snnmnkxdhflwgthqismb.com	text/html	64 bytes	post.php
8550	snnmnkxdhflwgthqismb.com		480 bytes	post.php
8554	snnmnkxdhflwgthqismb.com	text/html	64 bytes	post.php
8562	snnmnkxdhflwgthqismb.com		600 bytes	post.php
8564	snnmnkxdhflwgthqismb.com	text/html	64 bytes	post.php

c. Now click on **Save All** and select destination folder.

d. Now we can upload june11.dll file to **virustotal** site and get the output as below



53 engines detected this file

d36366666b407fe5527b96696377ee7ba9b609c8ef4561fa76af218ddd764dec

Google ipdate

invalid-signature overlay pedll signed

549.84 KB
Size

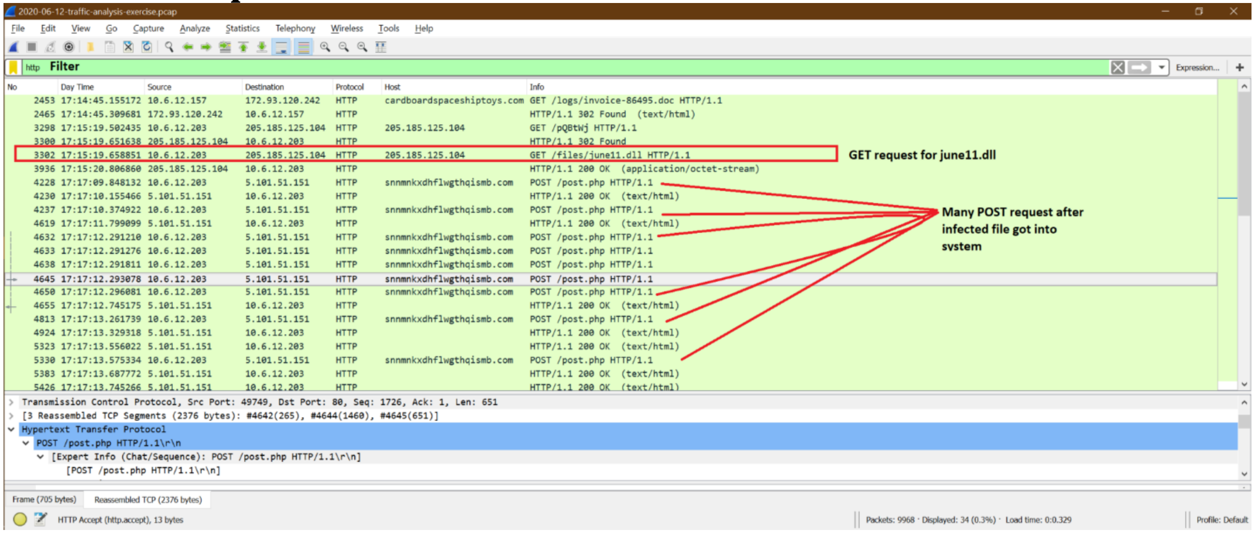
Ad-Aware	Trojan.GenericKD.34007934	AegisLab	Trojan.Multi.Generic.4!c
AhnLab-V3	Trojan/Win32.Ursnif.C4124200	Alibaba	TrojanSpy:Win32/Yakes.56555f48
ALYac	Trojan.GenericKD.34007934	Antiy-AVL	GrayWare/Win32.Kryptik.ehls
SecureAge APEX	Malicious	Arcabit	Trojan.Generic.D206EB7E

This confirms that **june11.dll** is a malware that got downloaded to the system [10.6.12.203].

Step4:
We can use the below filter to see all http packets.

Used Filter: "http"
Now, after this june11.dll got into the system we can see there is multiple **POST** from 10.6.12.203 system to **snnmnkxdhflwgthqismb.com**. The user did not do this POST, but the downloaded malware started doing this. It's very difficult to catch this type of issue on run time. One more point to be noticed that the POST are simple HTTP packets instead of HTTPS, but most of the time, ZLoader packets are HTTPS. In that case, it's quite impossible to see it, unlike HTTP.

This is HTTP post-infection traffic for ZLoader malware.


2020-06-12-traffic-analysis-exercise.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http Filter
Expression...
Frame (705 bytes)
Reassembled TCP (2376 bytes)
HTTP Accept (http.accept), 13 bytes
Packets: 9968 · Displayed: 34 (0.3%) · Load time: 0:0.329
Profile: Default

```
No     Day Time        Source          Destination     Protocol Host                         Info
  2453 17:14:45.155172 10.6.12.157     172.93.120.242  HTTP     cardboardspaceshiptoys.com   GET /logs/invoice-86495.doc HTTP/1.1
  2465 17:14:45.309681 172.93.120.242  10.6.12.157     HTTP                                  HTTP/1.1 302 Found  (text/html)
  3298 17:15:19.502435 10.6.12.203     205.185.125.104 HTTP     205.185.125.104              GET /pQBtWj HTTP/1.1
  3300 17:15:19.651638 205.185.125.104 10.6.12.203     HTTP                                  HTTP/1.1 302 Found
  3302 17:15:19.658851 10.6.12.203     205.185.125.104 HTTP     205.185.125.104              GET /files/june11.dll HTTP/1.1
  3936 17:15:20.806860 205.185.125.104 10.6.12.203     HTTP                                  HTTP/1.1 200 OK  (application/octet-stream)
  4228 17:17:09.848132 10.6.12.203     5.101.51.151    HTTP     snnmnkxdhflwgthqismb.com     POST /post.php HTTP/1.1
  4230 17:17:10.155466 5.101.51.151    10.6.12.203     HTTP                                  HTTP/1.1 200 OK  (text/html)
  4237 17:17:10.374922 10.6.12.203     5.101.51.151    HTTP     snnmnkxdhflwgthqismb.com     POST /post.php HTTP/1.1
  4619 17:17:11.799099 5.101.51.151    10.6.12.203     HTTP                                  HTTP/1.1 200 OK  (text/html)
  4632 17:17:12.291210 10.6.12.203     5.101.51.151    HTTP     snnmnkxdhflwgthqismb.com     POST /post.php HTTP/1.1
  4633 17:17:12.291276 10.6.12.203     5.101.51.151    HTTP     snnmnkxdhflwgthqismb.com     POST /post.php HTTP/1.1
  4638 17:17:12.291811 10.6.12.203     5.101.51.151    HTTP     snnmnkxdhflwgthqismb.com     POST /post.php HTTP/1.1
  4645 17:17:12.293078 10.6.12.203     5.101.51.151    HTTP     snnmnkxdhflwgthqismb.com     POST /post.php HTTP/1.1
  4650 17:17:12.296081 10.6.12.203     5.101.51.151    HTTP     snnmnkxdhflwgthqismb.com     POST /post.php HTTP/1.1
  4655 17:17:12.745175 5.101.51.151    10.6.12.203     HTTP                                  HTTP/1.1 200 OK  (text/html)
  4813 17:17:13.261739 10.6.12.203     5.101.51.151    HTTP     snnmnkxdhflwgthqismb.com     POST /post.php HTTP/1.1
  4924 17:17:13.329318 5.101.51.151    10.6.12.203     HTTP                                  HTTP/1.1 200 OK  (text/html)
  5323 17:17:13.556022 5.101.51.151    10.6.12.203     HTTP                                  HTTP/1.1 200 OK  (text/html)
  5330 17:17:13.575334 10.6.12.203     5.101.51.151    HTTP     snnmnkxdhflwgthqismb.com     POST /post.php HTTP/1.1
  5383 17:17:13.687772 5.101.51.151    10.6.12.203     HTTP                                  HTTP/1.1 200 OK  (text/html)
  5426 17:17:13.745266 5.101.51.151    10.6.12.203     HTTP                                  HTTP/1.1 200 OK  (text/html)
```

```
> Transmission Control Protocol, Src Port: 49749, Dst Port: 80, Seq: 1726, Ack: 1, Len: 651
> [3 Reassembled TCP Segments (2376 bytes): #4642(265), #4644(1460), #4645(651)]
v Hypertext Transfer Protocol
  v POST /post.php HTTP/1.1\r\n
    v [Expert Info (Chat/Sequence): POST /post.php HTTP/1.1\r\n]
        [POST /post.php HTTP/1.1\r\n]
```

GET request for june11.dll

Many POST request after infected file got into system

Summary of malware analysis:

We can say 10.6.12.203 got infected because of downloading **june11.dll** but did not get any more information about 10.6.12.157 after this host downloaded **invoice-86495.doc** file**.**

This is an example of one type of malware, but there may be different types of malware which work in a different style. Each has a different pattern to damage systems.

Conclusion:

In conclusion, we can say there many types of network attacks. It's not an easy job to learn everything in detail for all attacks, but we can get the pattern for famous attacks discussed in this chapter. In summary, here are the points we should know step by step to get the primary hints for any attack.

1. Know basic knowledge of the OSI/ TCP-IP layer and understand the role of each layer. There are multiple fields in each layer, and it carries some information. We should be aware of these.

2. Know the basics of Wireshark and get comfortable using it. Because there are some Wireshark options that help us to get the expected information easily.

3. Get an idea for attacks discussed here and try to match the pattern with your real Wireshark capture data.