

DIGITAL FORENSICS&INCID. RESP - COMP 4071

Homework Lab 2

First, I started my VMware.

Inside VMware, I turned on my VPN.

Then, I took a screenshot of everything.

After that, I downloaded 5 malware samples from box.com.

I opened Wireshark and ran it using my VPN connection.

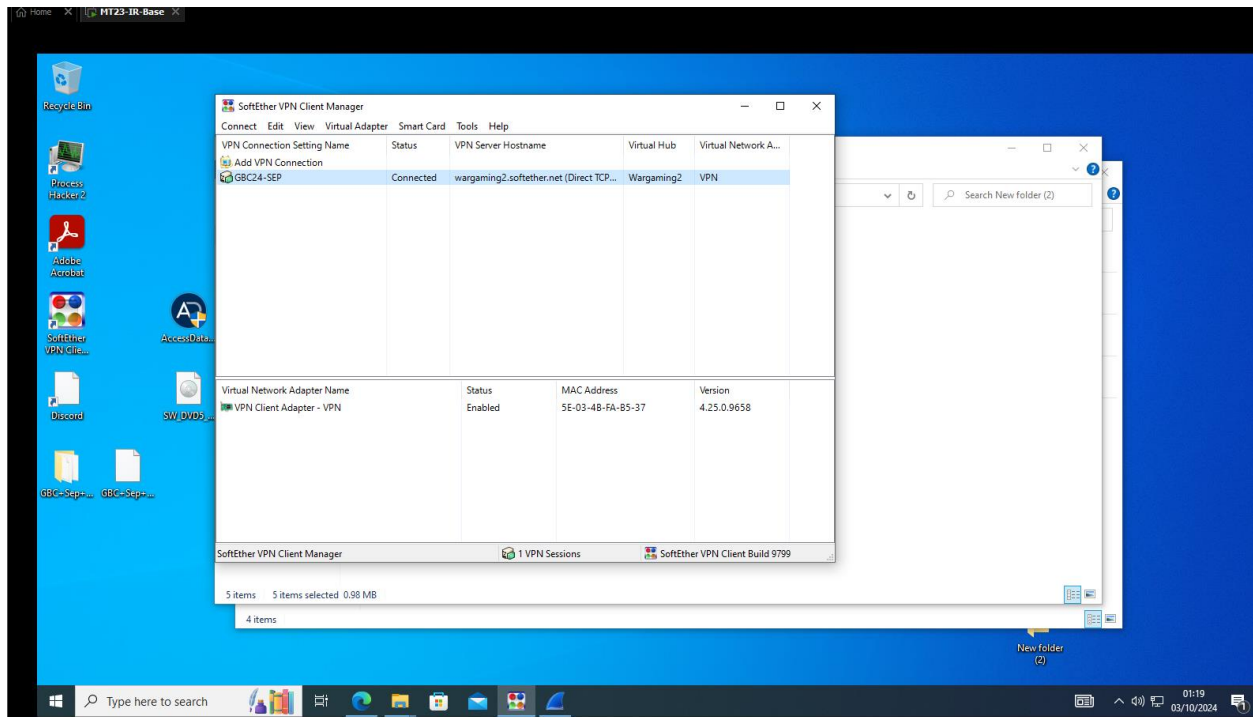
Next, I ran the 5 malware files as administrator.

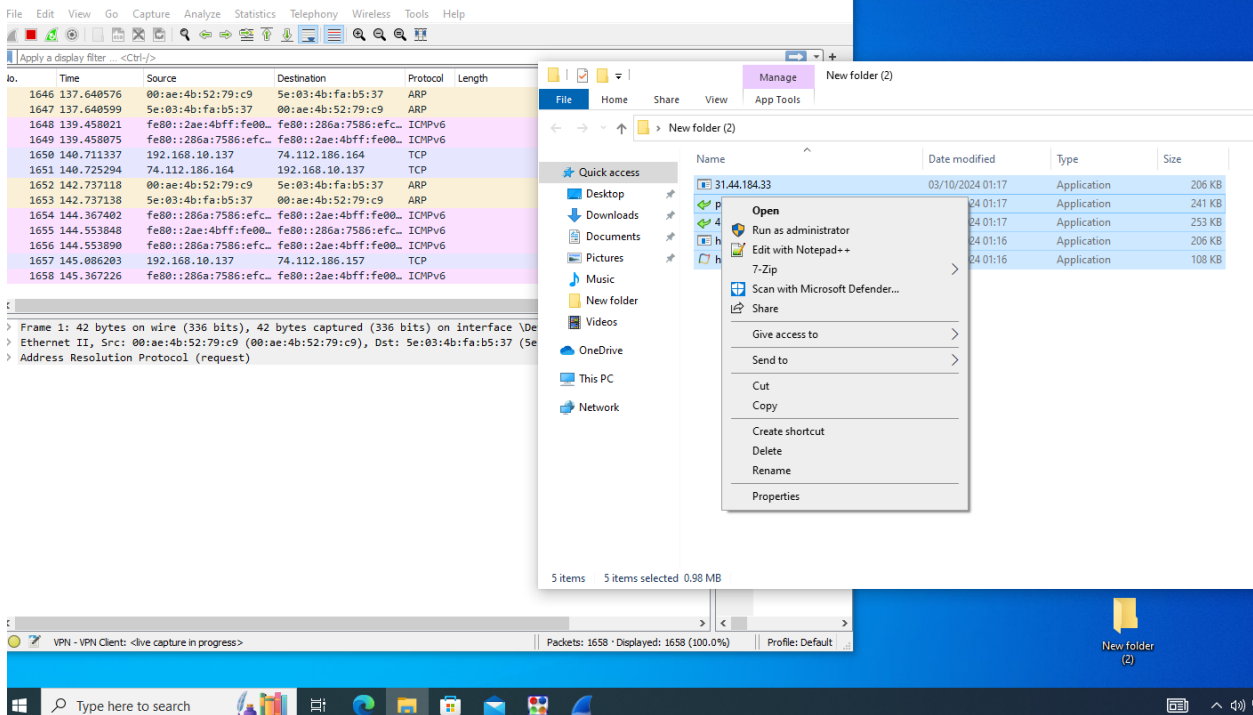
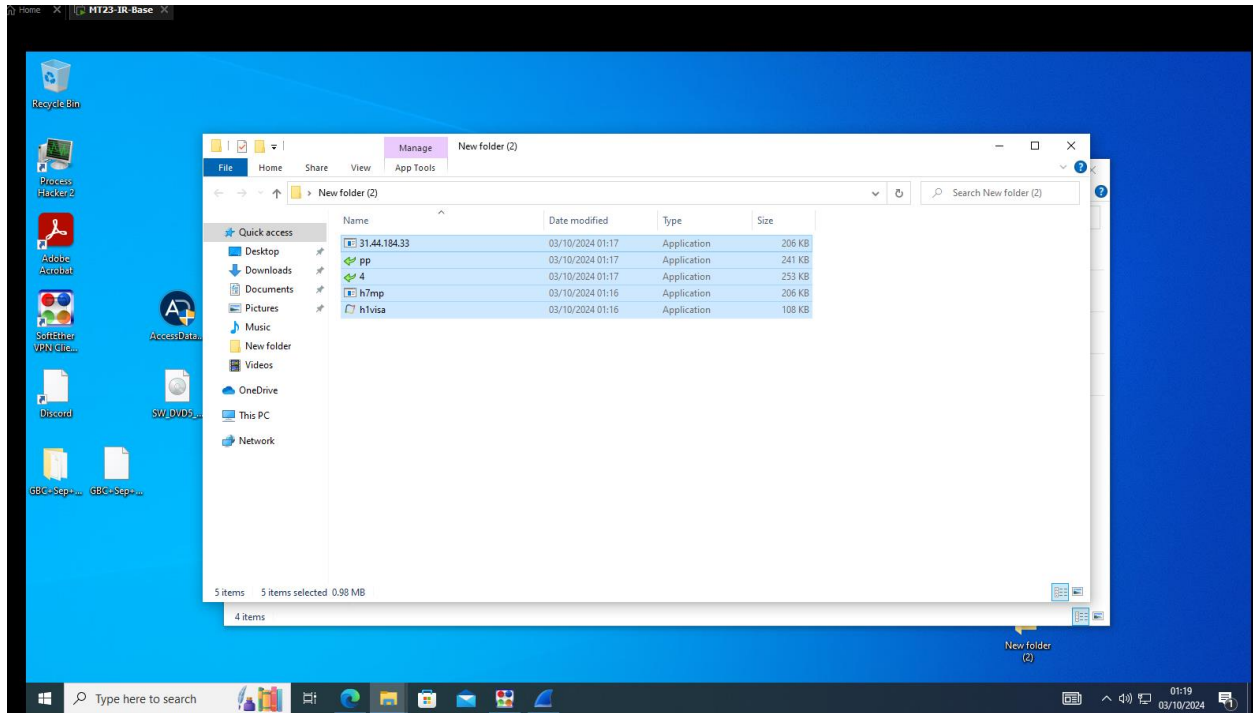
I waited for about 5 minutes.

Then, I stopped Wireshark to stop the capture.

After that, I saved the capture file as a backup.

Finally, I used a snapshot reverse to clean my machine.





Process Hacker (GBC-2024-SEP-MO\MjolinirTraining)

Hacker View Tools Users Help

Refresh Options Find handles or DLLs System information Search Processes (Ctrl+K)

Processes Services Network Disk

Name	PID	CPU	I/O total ...	Private b...	User name	Description
Registry	140			12.68 MB		
csrss.exe	592			1.74 MB		Client Server Runtime Process
wininit.exe	672			1.38 MB		Windows Start-Up Application
services.exe	812			5.46 MB		Services and Controller app
svchost.exe	952			10.17 MB		Host Process for Windows Ser...
dllhost.exe	4444			3.37 MB		COM Surrogate
StartMenuExperie...	6596			22.3 MB	GBC-2...\MjolinirTraining	
RuntimeBroker.exe	6672			5.74 MB	GBC-2...\MjolinirTraining	Runtime Broker
SearchApp.exe	6892			157.18 MB	GBC-2...\MjolinirTraining	Search application
RuntimeBroker.exe	7108			11.11 MB	GBC-2...\MjolinirTraining	Runtime Broker
RuntimeBroker.exe	7716			8.45 MB	GBC-2...\MjolinirTraining	Runtime Broker
PhoneExperience...	6928			45.98 MB	GBC-2...\MjolinirTraining	Microsoft Phone Link
ShellExperienceH...	9196			16.25 MB	GBC-2...\MjolinirTraining	Windows Shell Experience Host
RuntimeBroker.exe	9072			2.98 MB	GBC-2...\MjolinirTraining	Runtime Broker
TextInputHost.exe	2096			13.18 MB	GBC-2...\MjolinirTraining	
dllhost.exe	7708			4.04 MB	GBC-2...\MjolinirTraining	COM Surrogate
SearchApp.exe	2156			181.73 MB	GBC-2...\MjolinirTraining	Search application
FileCoAuth.exe	10236			4.56 MB	GBC-2...\MjolinirTraining	Microsoft OneDriveFile Co-Au...
ApplicationFrame...	8004			11.68 MB	GBC-2...\MjolinirTraining	Application Frame Host
SystemSettings.exe	3772			27.34 MB	GBC-2...\MjolinirTraining	Settings
UserOOBEBroker....	10452			1.97 MB	GBC-2...\MjolinirTraining	User OOBEBroker
smartscreen.exe	5684			8.03 MB	GBC-2...\MjolinirTraining	Windows Defender SmartScre...
TiWorker.exe	12080	12.08	10.29 MB/s	231.72 MB		Windows Modules Installer W...

CPU Usage: 33.68% Physical memory: 5.91 GB (36.91%) Processes: 161

Protocol: UDP (17)
Header Checksum: 0xf25b [validation disabled]
[Header checksum status: Unverified]

Instructions

1. **What is the hostname of the infected device?**
GBC-2024-SEP-MO<20>: type NB, class IN

I used nbns (NetBIOS Name Service) in Wireshark. After starting the capture, I looked for nbns traffic, which helps resolve network names. Under the Queries section, I saw the name request for the device, which gave me the hostname. Specifically, I found the hostname in the Additional Records section

The image shows a Wireshark packet capture of a NetBIOS Name Service refresh request. The packet list shows a refresh request from 192.168.10.137 to 192.168.10.2. The packet details pane shows the following information:

- Frame 7087: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on interface Device\NPF... (3E8FE00-85E0-47EA-86CF-DEF2E0700FE), Id 0
- Ethernet II, Src: Se:03:4b:fa:b5:37 (Se:03:4b:fa:b5:37), Dst: VMware e3:29:68 (00:50:56:c3:29:68)
- Internet Protocol Version 4, Src: 192.168.10.137, Dst: 192.168.10.2
- User Datagram Protocol, Src Port: 137, Dst Port: 137
- NetBIOS Name Service
 - Transaction ID: 0x9118
 - Flags: 0x4000, Opcode: Refresh
 - Questions: 1
 - Answer RRs: 0
 - Authority RRs: 0
 - Additional RRs: 1
 - Queries
 - Additional records
 - GBC-2024-SEP-MD<20>: type NB, class IN
 - Name: GBC-2024-SEP-MD<20> (Server service)
 - Type: NB (32)
 - Class: IN (1)
 - Time to live: 3 days, 11 hours, 20 minutes
 - Data length: 6
 - Name flags: 0x0000, ONT: Unknown (H-node, unique)
 - Addr: 192.168.10.137

2. What is the IP of the infected device? 192.168.10.137

The image shows a Wireshark packet capture of a NetBIOS Name Service refresh request. The packet list shows a refresh request from 192.168.10.137 to 192.168.10.2. The packet details pane shows the following information:

- Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface Device\NPF... (3E8FE00-85E0-47EA-86CF-DEF2E0700FE), Id 0
- Ethernet II, Src: Se:03:4b:fa:b5:37 (Se:03:4b:fa:b5:37), Dst: VMware e3:29:68 (00:50:56:c3:29:68)
- Internet Protocol Version 4, Src: 192.168.10.137, Dst: 192.168.10.2
- User Datagram Protocol, Src Port: 137, Dst Port: 137
- NetBIOS Name Service
 - Transaction ID: 0x9118
 - Flags: 0x4000, Opcode: Refresh
 - Questions: 1
 - Answer RRs: 0
 - Authority RRs: 0
 - Additional RRs: 1
 - Queries
 - Additional records
 - GBC-2024-SEP-MD<20>: type NB, class IN
 - Name: GBC-2024-SEP-MD<20> (Server service)
 - Type: NB (32)
 - Class: IN (1)
 - Time to live: 3 days, 11 hours, 20 minutes
 - Data length: 6
 - Name flags: 0x0000, ONT: Unknown (H-node, unique)
 - Addr: 192.168.10.137

HTTP Requests: The infected gadget with IP 192.168.10.137 is making frequent suspicious HTTP requests. GET queries to purportedly random or encoded URLs are included in these requests; these requests may also indicate communication with a command-and-manage (C2) server or the download of harmful payloads. Certain URLs (such as "lowedcertstl.Cab") also point to untrusted or confusing certificates that may be used in malware or infection techniques. **High Volume of Traffic:** As you can see from the "Statistics" page, 192.168.10.137 gets charged for a significant amount of traffic to multiple destinations, which may be a sign of unusual interest. A possible malware contamination can be inferred from the number of connections, bytes exchanged, and length of some of the streams, since these kinds of site visits are frequently linked to C2 communications, malware updates, or data exfiltration. **Destination IPs:** Some of the requests' vacation spot addresses are suspect, including a handful that are associated with well-known CDN (Content Delivery Network) providers. Malware authors may utilize these addresses to conceal their traffic or use trustworthy infrastructure to disseminate

3. Identify all malware(s) with their VT links and VT score:

To find malware collectively with its VirusTotal (VT) URLs and ratings, take the following movements:

Apply the HTTP Filter: In Wireshark, begin by applying the clear out http. With the resource of this clear out, you'll be able to separate out all HTTP traffic and give attention to web-based conversation, which often includes report downloads and server interactions

Find Suspicious IPs: Keep a watch out for any IP addresses showing peculiar or suspicious activity, which include repeated connections, encoded URLs, or connections to uncommon domains. This is probably a sign of viable malware activities, like interacting with a command-and-manage (C2) site or downloading documents which might be dangerous..

Click on "Follow TCP Stream" from the context menu when you see a suspicious IP address. This will allow you to follow the HTTP stream. This displays all of the tool and server communication. A report that might be malware may be seen to be downloading.

Acquire the File: Note the URL or report name if you locate a file in the stream. This

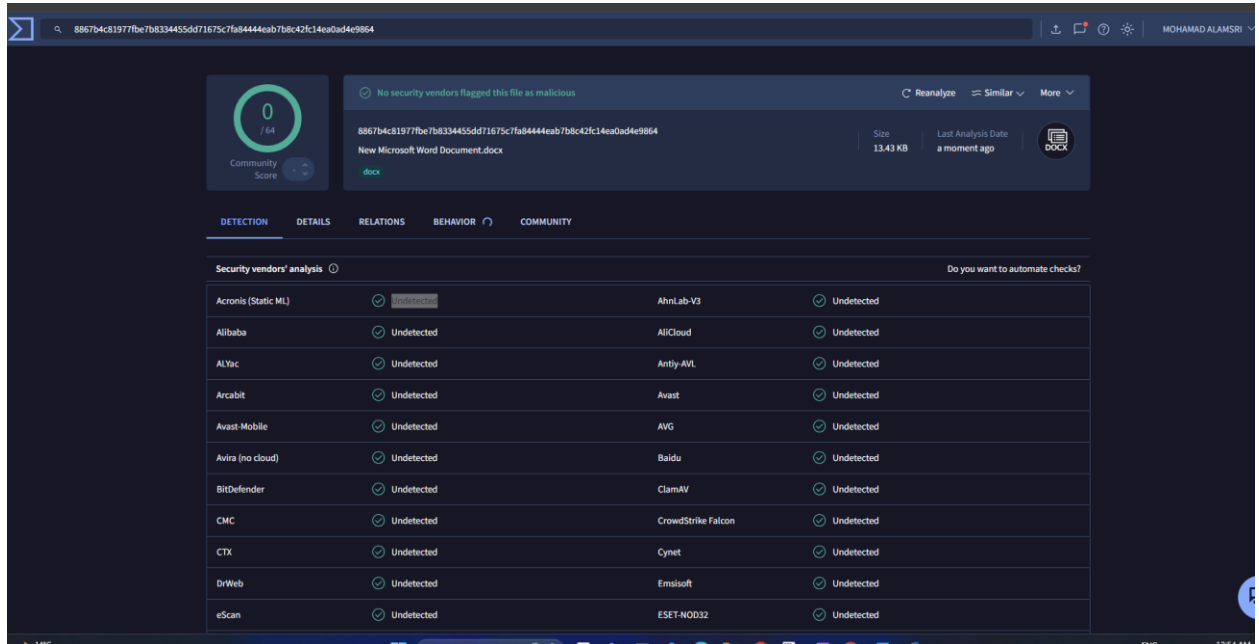
report can be taken out and examined.

Add the document or paste the URL to VirusTotal.Com after visiting the website. It will be examined by VirusTotal using a variety of antivirus programs.

Find the VT Link and Score: VirusTotal will provide a record containing a link and a VT score that indicates the quantity of antivirus programs that deemed the file to be harmful.

The screenshot displays the Wireshark interface with a network capture loaded. The top pane shows a list of captured packets, with the first packet (No. 63) selected. The middle pane shows the details of the selected packet, which is an HTTP GET request from 192.168.10.137 to 192.229.211.100. The bottom pane shows the raw packet data in hexadecimal and ASCII. The ASCII column contains a large block of text, likely a response body or a large data payload, which appears to be a list of file names or a directory listing. The interface includes a menu bar, a toolbar, and a status bar at the bottom.

The image displays two screenshots of Wireshark, a network protocol analyzer. The top screenshot shows a list of network packets in the packet list pane, with packet 6180 selected. The packet details pane shows the structure of the selected packet, including Ethernet II, Internet Protocol Version 4, and Hypertext Transfer Protocol. The packet bytes pane shows the raw data of the packet. The bottom screenshot shows a list of network packets in the packet list pane, with packet 2458 selected. The packet details pane shows the structure of the selected packet, including Ethernet II, Internet Protocol Version 4, and Hypertext Transfer Protocol. The packet bytes pane shows the raw data of the packet. The top pane in both screenshots shows a list of network packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. The bottom pane in both screenshots shows the details of the selected packet, including Ethernet II, Internet Protocol Version 4, and Hypertext Transfer Protocol. The packet bytes pane shows the raw data of the packet.

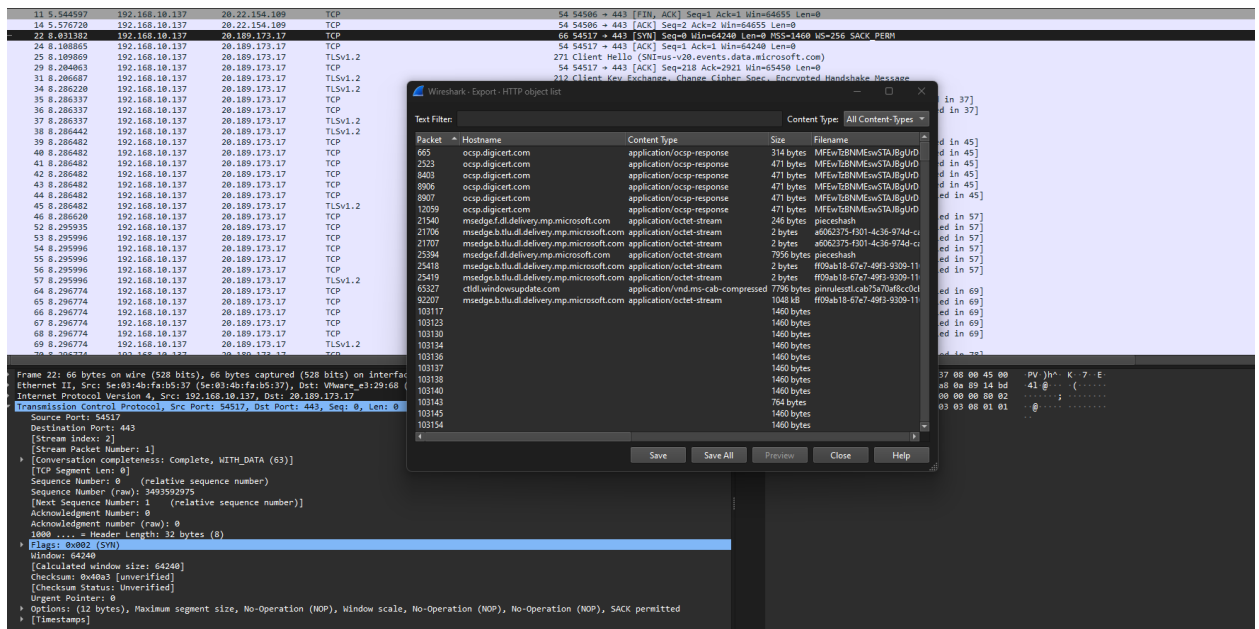


4. Identify all remote hosts which are malicious

To get a list of every external IP that has communicated with your device, navigate to Statistics > Conversations and select the IPv4 or IPv6 tab.

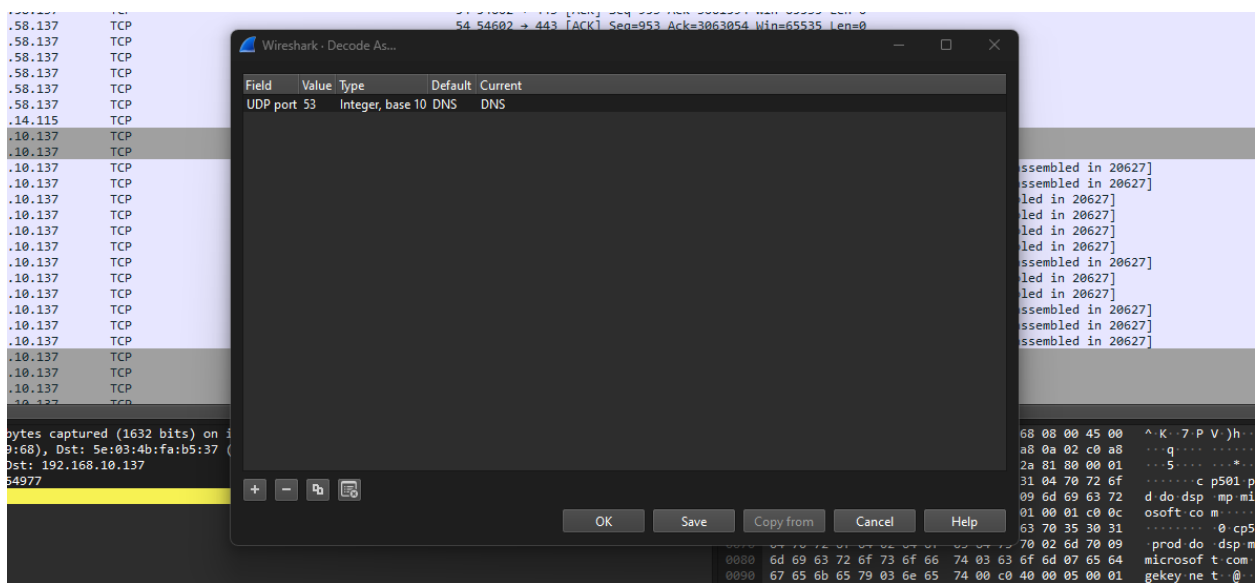
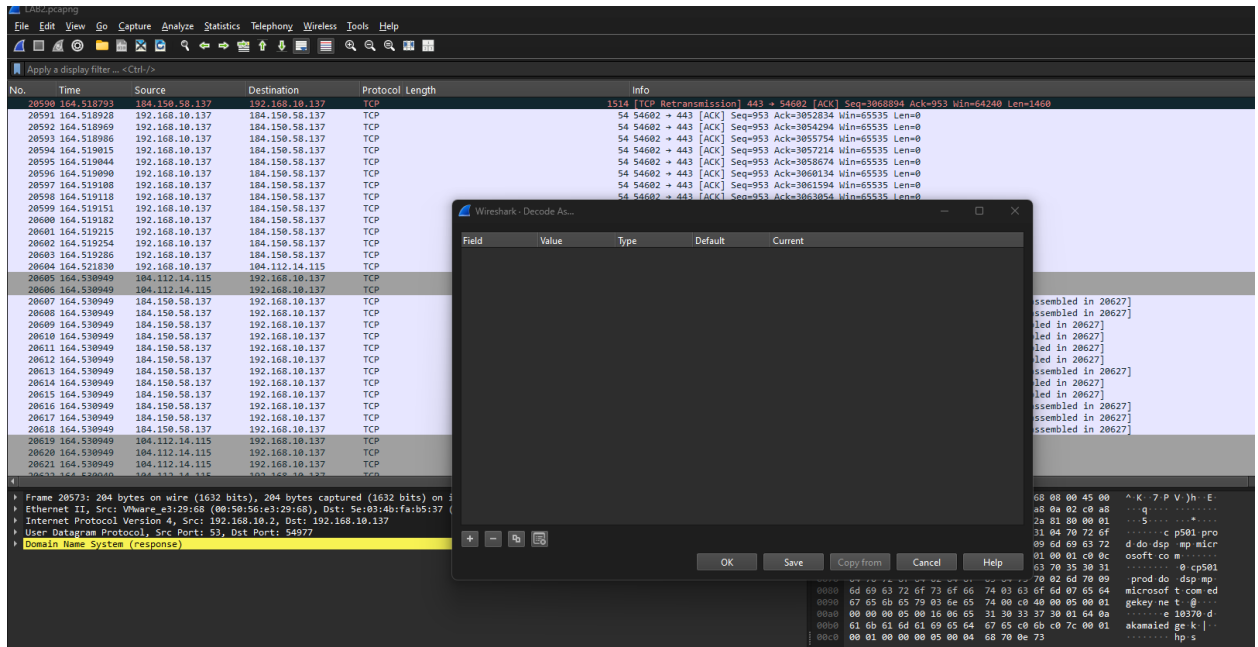
Take the suspicious IP addresses and determine if they are marked as malicious by visiting websites such as VirusTotal or IPVoid.

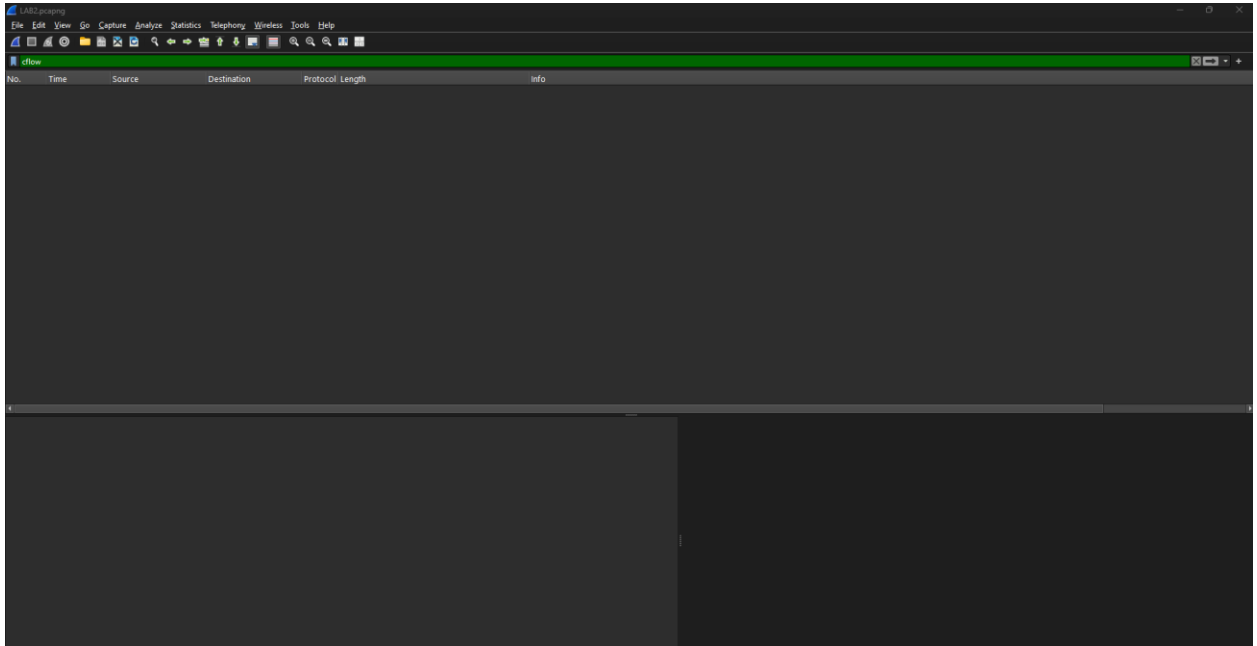
This procedure has the advantage of determining whether your device is corresponding with any risky or unidentified distant servers that might be involved in a malware infection or hack.



Select 'OK' to save the selection. Note flow packets are subsequently denoted as CFLOW in the protocol column:

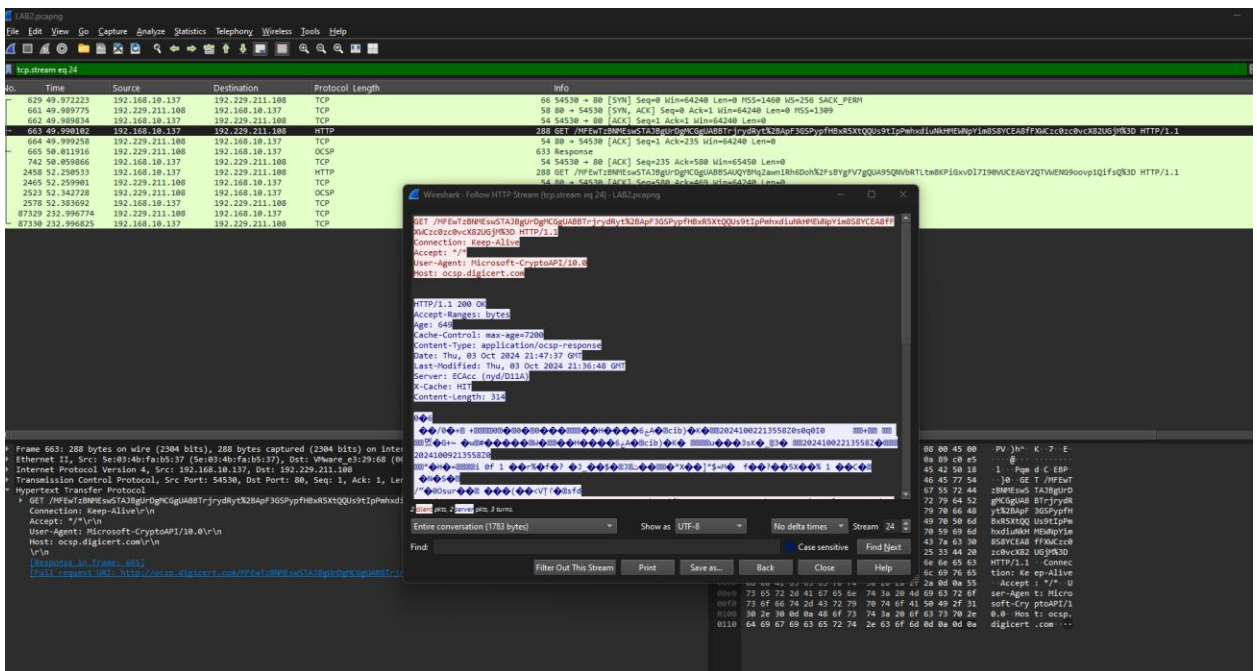
After this, NetFlow packets will appear as cflow in the protocol column. However, no relevant NetFlow traffic was found in the capture.

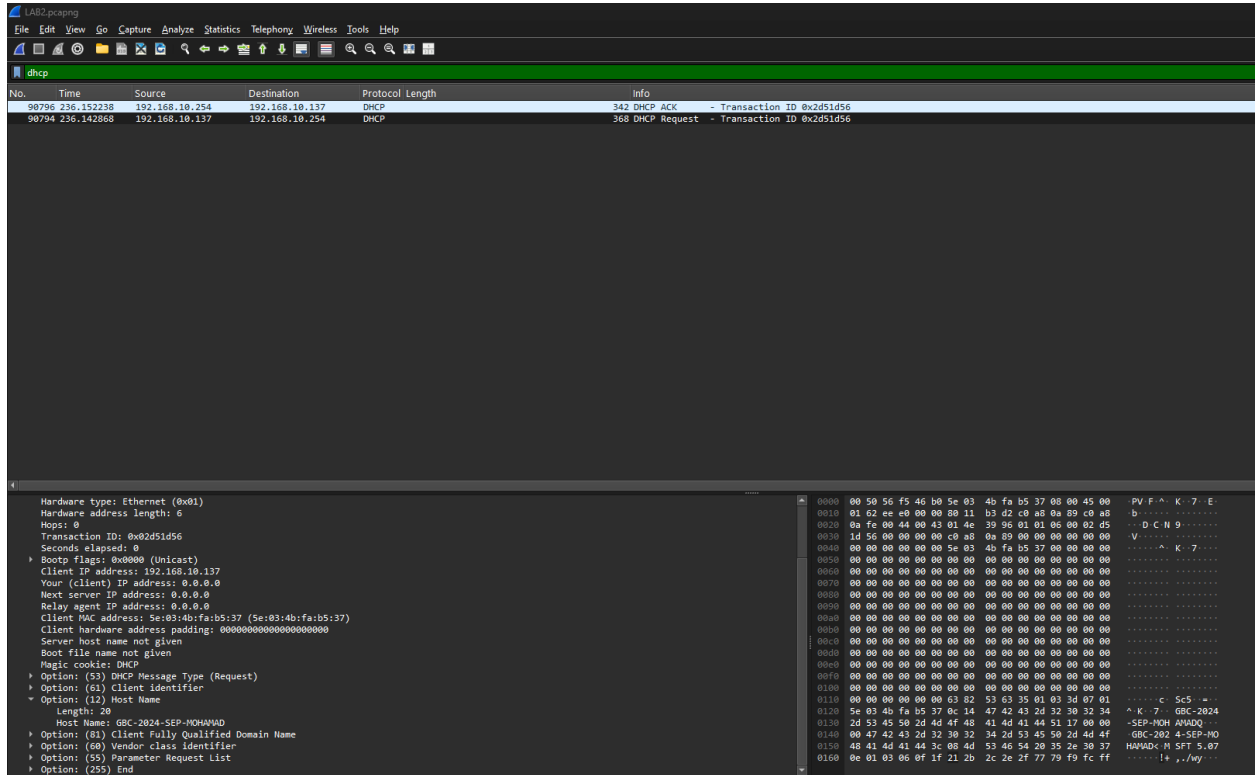




6. Identify usernames :

User-Agent: Microsoft-CryptoAPI/10.0





7. Identify who else is on the network

To identify who else is on the network, follow these steps:

- Go to Statistics: In Wireshark, click on the "Statistics" menu at the top.
- Select Endpoints: Choose "Endpoints" from the drop-down menu. This will show you a list of all devices (endpoints) that have communicated on the network during the capture.
- View the List: The list will display IP addresses, MAC addresses, and other details about each device connected to the network.

The image shows a Wireshark packet capture and the Endpoint Settings window. The packet capture shows several TCP segments between 192.168.10.137 and 192.168.10.137. The Endpoint Settings window shows a list of endpoints with columns for Address, Packets, Bytes, Tx Packets, Tx Bytes, Rx Packets, Rx Bytes, Country, City, Latitude, Longitude, AS Number, and AS Organization. The list is filtered by IP address and shows various endpoints with their respective traffic statistics.

Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Country	City	Latitude	Longitude	AS Number	AS Organization
192.168.10.137	338,442	284 MB	172,014	111 MB	186,428	273 MB						
208.111.183.1	152,120	122 MB	78,626	118 MB	73,494	4 MB						
199.232.210.172	6,316	5 MB	3,419	5 MB	2,897	162 KB						
184.150.58.137	5,096	5 MB	2,927	4 MB	2,769	151 KB						
104.112.13.149	5,651	5 MB	2,949	4 MB	2,702	149 KB						
199.232.214.172	5,171	4 MB	2,718	4 MB	2,453	139 KB						
20.3.187.198	3,070	2 MB	1,515	2 MB	1,495	257 KB						
20.96.184.17	2,992	2 MB	1,731	2 MB	1,261	102 KB						
23.33.4.10	2,951	2 MB	1,560	2 MB	1,391	81 KB						
104.114.72.67	2,078	2 MB	1,200	2 MB	878	62 KB						
20.93.183.109	2,075	2 MB	1,325	2 MB	750	82 KB						
184.150.49.155	1,799	1 MB	939	1 MB	860	50 KB						
184.150.49.113	1,139	856 KB	576	822 KB	563	34 KB						
204.4.229.154	1,003	776 KB	545	732 KB	458	44 KB						
104.112.14.115	898	361 KB	516	306 KB	382	55 KB						
20.189.173.3	845	550 KB	432	29 KB	413	521 KB						
23.33.4.24	790	606 KB	449	582 KB	341	24 KB						
40.126.25.12	593	507 KB	358	205 KB	235	102 KB						
104.112.10.208	573	457 KB	338	439 KB	235	17 KB						
217.20.83.35	564	476 KB	317	461 KB	247	15 KB						
192.168.10.2	527	77 KB	238	51 KB	289	26 KB						
52.183.220.149	496	213 KB	280	184 KB	216	29 KB						
72.21.81.249	493	376 KB	255	361 KB	238	15 KB						
20.189.173.12	400	244 KB	211	23 KB	189	221 KB						
13.85.23.206	391	292 KB	241	174 KB	150	118 KB						
20.189.173.28	381	232 KB	199	20 KB	182	211 KB						
204.79.197.203	325	223 KB	198	208 KB	127	15 KB						
74.112.186.157	279	85 KB	145	24 KB	134	51 KB						
40.79.25.129	250	111 KB	137	57 KB	113	54 KB						
74.112.186.156	224	33 KB	110	18 KB	114	15 KB						
20.189.173.2	202	73 KB	110	34 KB	92	39 KB						
40.79.173.40	182	62 KB	99	29 KB	83	33 KB						
184.150.58.153	162	47 KB	87	39 KB	75	8 KB						
18.102.42.74	160	65 KB	94	49 KB	66	16 KB						
23.101.169.44	159	82 KB	89	69 KB	70	13 KB						
20.189.173.17	157	98 KB	80	9 KB	77	89 KB						
52.183.205.142	138	51 KB	75	30 KB	63	21 KB						
192.168.10.1	136	11 KB	136	11 KB	0	0 bytes						
20.22.207.36	131	37 KB	74	32 KB	57	6 KB						
73.96.180.189	128	56 KB	72	45 KB	56	11 KB						

8. What are the HTTP and DNS requests:

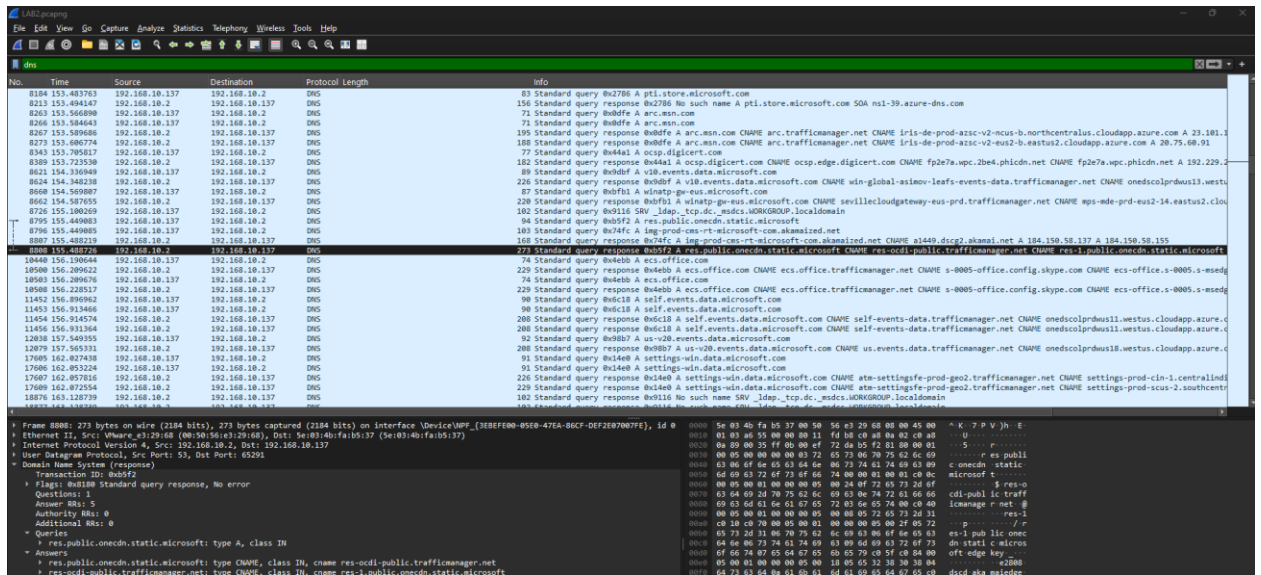
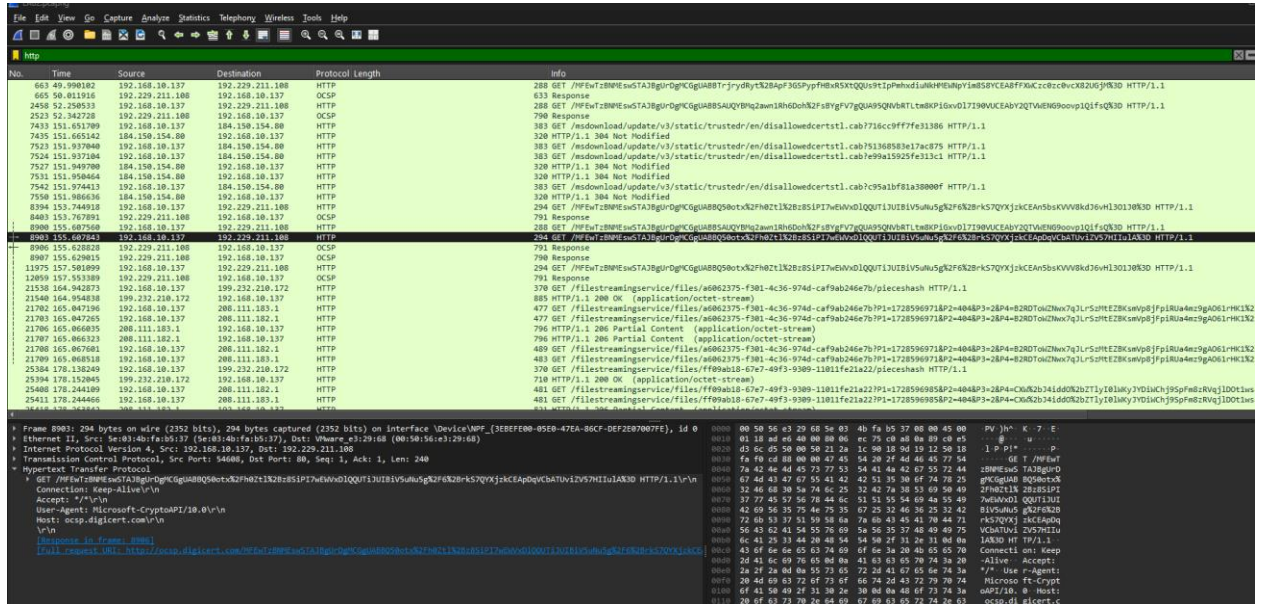
HTTP requests are used when a computer (client) talks to a website's server to ask for something or send information. Different types of requests do different things:

- **GET:** Used to get (retrieve) a webpage or file.
- **POST:** Used to send data, like when you fill out a form.
- **PUT:** Used to update something, like a file or information.
- **DELETE:** Used to remove something from the server.
- **PATCH:** Used to update just part of something.

DNS requests are like asking a phonebook for a phone number. When you type a website's name (like google.com) in your browser, your computer doesn't know where that is right away. So, it sends a **DNS request** to find the **IP address** (like the phone number) for that website.

The **DNS server** then replies with the IP address, and your computer uses that to connect to the website.

In short, DNS requests help your computer find the right address for websites so you can visit them.



9. Identify which malware connected to which remote host

Wireshark

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr == 192.168.10.137

No.	Time	Source	Destination	Protocol	Length	Info
647	0.975901	192.168.10.137	104.40.82.182	TLSv1.2	92	Application Data
648	0.982450	104.40.82.182	192.168.10.137	TCP	54	443 → 54524 [ACK] Seq=5559 Ack=628 Win=64240 Len=0
649	0.982450	104.40.82.182	192.168.10.137	TCP	54	443 → 54524 [ACK] Seq=5559 Ack=660 Win=64240 Len=0
650	0.982450	104.40.82.182	192.168.10.137	TCP	54	443 → 54524 [ACK] Seq=5559 Ack=945 Win=64240 Len=0
651	0.983516	104.40.82.182	192.168.10.137	TCP	54	443 → 54524 [ACK] Seq=5559 Ack=2254 Win=64240 Len=0
652	0.983516	104.40.82.182	192.168.10.137	TCP	54	443 → 54524 [ACK] Seq=5559 Ack=3333 Win=64240 Len=0
653	0.983516	104.40.82.182	192.168.10.137	TCP	54	443 → 54524 [ACK] Seq=5559 Ack=2491 Win=64240 Len=0
654	0.986203	192.168.10.137	104.40.82.182	DNS	113	Standard query query RRDIGS > ocp.dplicort.com CNMHE fqn2ta.wsp.phlcdn.net CNMHE fqn2ta.wsp.phlcdn.net & NS 239.2
655	0.987953	180.40.82.181	192.168.10.137	TCP	56	TCP port 80 [RST] Seq=34525 [ACK] Seq=5559 Ack=945 Win=64240 Len=0
656	0.988096	104.40.82.182	192.168.10.137	TCP	54	443 → 54525 [ACK] Seq=5559 Ack=486 Win=64240 Len=0
657	0.988096	104.40.82.182	192.168.10.137	TCP	54	443 → 54524 [ACK] Seq=5559 Ack=2789 Win=64240 Len=0
658	0.988394	104.40.82.182	192.168.10.137	TCP	54	443 → 54524 [ACK] Seq=5559 Ack=4009 Win=64240 Len=0
659	0.988394	104.40.82.182	192.168.10.137	TCP	54	443 → 54524 [ACK] Seq=5559 Ack=4009 Win=64240 Len=0
660	0.988394	104.40.82.182	192.168.10.137	TCP	54	443 → 54524 [ACK] Seq=5559 Ack=5292 Win=64240 Len=0
661	0.989075	192.229.211.100	192.168.10.137	TCP	56	80 → 54530 [SYN, ACK] Seq=1 Win=64240 Len=0 MSS=1389
662	0.989394	192.168.10.137	104.40.82.182	TCP	54	54530 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
663	0.990172	192.168.10.137	192.229.211.100	HTTP	288	GET /?f=/t/HSwstA3lgGgCqUABtFrJydyLx3kZp3GyPfrH8sAQQUQcIzphnuhLUHmYdyLmsVSECAFffXKczbcvkxZUoJPHSD HTTP/1.1
664	0.990172	192.229.211.100	192.168.10.137	TCP	54	80 → 54530 [ACK] Seq=1 Ack=233 Win=64240 Len=0
665	0.991916	192.229.211.100	192.168.10.137	OSPF	63	Response
666	0.992084	192.168.10.137	104.114.72.67	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
667	0.992015	192.168.10.137	104.114.72.67	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
668	0.992014	192.168.10.137	104.114.72.67	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
669	0.992099	192.168.10.137	104.114.72.67	TLSv1.2	141	Application Data
670	0.992084	192.168.10.137	104.114.72.67	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
671	0.992119	192.168.10.137	104.114.72.67	TLSv1.2	141	Application Data
672	0.992143	192.168.10.137	104.114.72.67	TLSv1.2	198	Application Data
673	0.992178	192.168.10.137	104.114.72.67	TLSv1.2	141	Application Data
674	0.992123	192.168.10.137	104.114.72.67	TLSv1.2	148	Application Data
675	0.992156	192.168.10.137	104.114.72.67	TLSv1.2	141	Application Data
676	0.992160	192.168.10.137	104.114.72.67	TLSv1.2	141	Application Data
677	0.992180	192.168.10.137	104.114.72.67	TLSv1.2	158	Application Data
678	0.992027	104.114.72.67	192.168.10.137	TCP	54	443 → 54527 [ACK] Seq=3724 Ack=326 Win=64240 Len=0

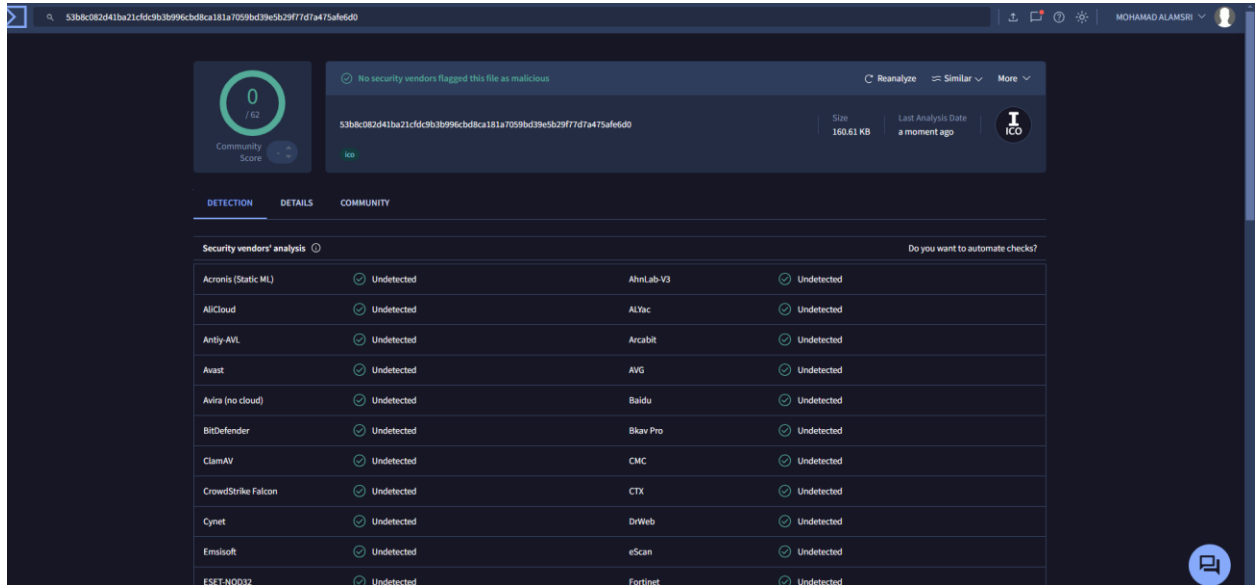
```

Frame 663: 288 bytes on wire (2304 bits), 288 bytes captured (2304 bits) on interface WFP_{D8EF0E00-95E0-47EA-86CF-DE2DF0707FE}, ID 0
    Ethernet II, Src: SenbAbFa1e61372 (08:00:27:00:00:13:72), Dst: VMware_03:2B:6F (08:00:56:01:29:6F)
    Internet Protocol Version 4, Src: 192.168.10.137, Dst: 192.229.211.100
    Transmission Control Protocol, Src Port: 54530, Dst Port: 80, Seq: 1, Len: 234
    Hypertext Transfer Protocol
     
```

The image shows a Wireshark packet capture analysis of a TCP stream. The top pane displays a list of packets, with packet 44 selected. The middle pane shows the raw packet data in hexadecimal and ASCII. The bottom pane displays the packet details, including Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol. The packet details pane is expanded to show the TCP segment, which includes the source and destination ports, sequence number, and flags.

Packet List:

No.	Time	Source	Destination	Protocol	Length	Info
22	0.03382	192.168.10.137	20.189.273.17	TCP	66	66 54517 → 443 [SYN] Seq=0 win=64240 len=0 MSS=1460 SACK_PERM=1
23	0.10009	20.189.273.17	192.168.10.137	TCP	54	54 443 → 54517 [ACK] Seq=1 Ack=3 win=64240 len=0 MSS=1380
24	0.10065	192.168.10.137	20.189.273.17	TCP	54	54 54517 → 443 [ACK] Seq=1 Ack=3 win=64240 len=0
25	0.10069	192.168.10.137	20.189.273.17	TLV5.1.2	271	Client hello [SSLv3] (len=271) from 192.168.10.137 to 20.189.273.17
26	0.120005	20.189.273.17	192.168.10.137	TCP	54	54 443 → 54517 [ACK] Seq=1 Ack=18 win=64240 len=0
27	0.203795	20.189.273.17	192.168.10.137	TCP	1514	443 → 54517 [ACK] Seq=1 Ack=218 win=64240 len=1460 [TCP PDU reassembled in 30]
28	0.204041	20.189.273.17	192.168.10.137	TCP	1514	443 → 54517 [ACK] Seq=1 Ack=218 win=64240 len=1460 [TCP PDU reassembled in 30]
29	0.206785	192.168.10.137	20.189.273.17	TCP	54	54 54517 → 443 [ACK] Seq=1 Ack=218 win=64240 len=0
30	0.204330	20.189.273.17	192.168.10.137	TLV5.1.2	1369	Server hello, [SSLv3] (len=1369) from 20.189.273.17 to 192.168.10.137
31	0.206687	192.168.10.137	20.189.273.17	TLV5.1.2	212	Client key exchange, [SSLv3] (len=212) from 20.189.273.17 to 192.168.10.137
32	0.212458	20.189.273.17	192.168.10.137	TCP	54	54 443 → 54517 [ACK] Seq=1 Ack=218 win=64240 len=0
33	0.204793	20.189.273.17	192.168.10.137	TLV5.1.2	174	Change cipher spec, [SSLv3] (len=174) from 20.189.273.17 to 192.168.10.137
34	0.206220	192.168.10.137	20.189.273.17	TLV5.1.2	141	Application data, [SSLv3] (len=141) from 20.189.273.17 to 192.168.10.137
35	0.206337	192.168.10.137	20.189.273.17	TCP	1363	54517 → 443 [ACK] Seq=1 Ack=218 win=64240 len=0
36	0.206337	192.168.10.137	20.189.273.17	TCP	1363	54517 → 443 [ACK] Seq=1 Ack=218 win=64240 len=0
37	0.206337	192.168.10.137	20.189.273.17	TLV5.1.2	928	Application data, [SSLv3] (len=928) from 20.189.273.17 to 192.168.10.137
38	0.206442	192.168.10.137	20.189.273.17	TLV5.1.2	92	Application data, [SSLv3] (len=92) from 20.189.273.17 to 192.168.10.137
39	0.206442	192.168.10.137	20.189.273.17	TCP	1363	54517 → 443 [ACK] Seq=1 Ack=218 win=64240 len=0
40	0.206442	192.168.10.137	20.189.273.17	TCP	1363	54517 → 443 [ACK] Seq=1 Ack=218 win=64240 len=0
41	0.206442	192.168.10.137	20.189.273.17	TCP	1363	54517 → 443 [ACK] Seq=1 Ack=218 win=64240 len=0
42	0.206442	192.168.10.137	20.189.273.17	TCP	1363	54517 → 443 [ACK] Seq=1 Ack=218 win=64240 len=0
43	0.206442	192.168.10.137	20.189.273.17	TCP	1363	54517 → 443 [ACK] Seq=1 Ack=218 win=64240 len=0
44	0.206442	192.168.10.137	20.189.273.17	TLV5.1.2	438	Application data, [SSLv3] (len=438) from 20.189.273.17 to 192.168.10.137
45	0.206500	192.168.10.137	20.189.273.17	TCP	1363	54517 → 443 [ACK] Seq=1 Ack=218 win=64240 len=0
47	0.205885	20.189.273.17	192.168.10.137	TCP	54	443 → 54517 [ACK] Seq=1 Ack=218 win=64240 len=0
48	0.205885	20.189.273.17	192.168.10.137	TCP	54	443 → 54517 [ACK] Seq=1 Ack=218 win=64240 len=0
49	0.205885	20.189.273.17	192.168.10.137	TCP	54	443 → 54517 [ACK] Seq=1 Ack=218 win=64240 len=0
50	0.205885	20.189.273.17	192.168.10.137	TCP	54	443 → 54517 [ACK] Seq=1 Ack=218 win=64240 len=0
51	0.205885	20.189.273.17	192.168.10.137	TCP	54	443 → 54517 [ACK] Seq=1 Ack=218 win=64240 len=0
52	0.205935	20.189.273.17	192.168.10.137	TCP	1363	54517 → 443 [



10. In a minimum of 1500 words, explain what was captured in the pcap chronologically. Your explanation must include full screen screenshots with labels on them.

This is an independent activity, do not work in groups to solve the answer. Everyone needs to run their own malware, and do their own analysis.