



NETWORK FORENSICS INVESTIGATION USING WIRESHARK

Made By: Muhammad Faheem Shahid
Under Guidance of: Prof. Naveed Sb



NETWORK FORENSICS INVESTIGATION USING WIRESHARK

Contents

1	Tools Demonstrated in This Lab.....	3
2	Lab Scenario.....	3
3	Introduction	3
4	Lab Objectives	3
5	Lab Environment.....	3
6	Overview of Wireshark.....	4
7	Step-by-Step Procedure	4
8	Task 1: Analyze the PCAP File.....	4
9	Task 2: Identify Evidence of Attack	16
10	Task 3: Timeline Reconstruction	20
11	Task 4: Final Investigation Report	27
12	Conclusion	28

Tools Demonstrated in This Lab

Tool demonstrated in this lab is “**Wireshark**”

Lab Scenario

As a Network Forensic Investigator in FIA Cybercrime Wing, I received a PCAP file captured during suspicious activity. Employees reported slow network and unauthorized access alerts. My task was to analyze the traffic using Wireshark, identify the attacker, extract evidence, and prepare a court-presentable report.

Introduction

In this assignment, I used Wireshark, an open-source packet analyzer, to investigate malicious network activity. I identified port scanning, RDP brute-force, malware download, and C2 beaconing. This hands-on analysis helped me understand how to extract digital evidence from network traffic.

Lab Objectives

- Open and analyze a real attack PCAP file using Wireshark
 - Identify source/destination IPs, protocols, and suspicious packets
 - Detect type of attack and extract evidence (files, URLs, payloads)
 - Reconstruct attack timeline
 - Prepare a professional forensic investigation report
-

Lab Environment

To complete this assignment, I used the following setup:

- **Kali Linux** : My primary system for running Wireshark
 - **Wireshark version** : Latest
 - **Internet Connectivity** : Required to download the PCAP file
 - **PCAP File** : case001.pcap (downloaded from <https://dfirmadness.com/case001/case001-pcap.zip>)
 - **Dedicated IP in PCAP** : Attacker IP 194.61.24.102, Victim IPs 10.42.85.10 & 10.42.85.115
-

Overview of Wireshark

Wireshark is the world's most popular network protocol analyzer. It captures and displays packets in real time.

Key Features of Wireshark

Key Features of Wireshark:

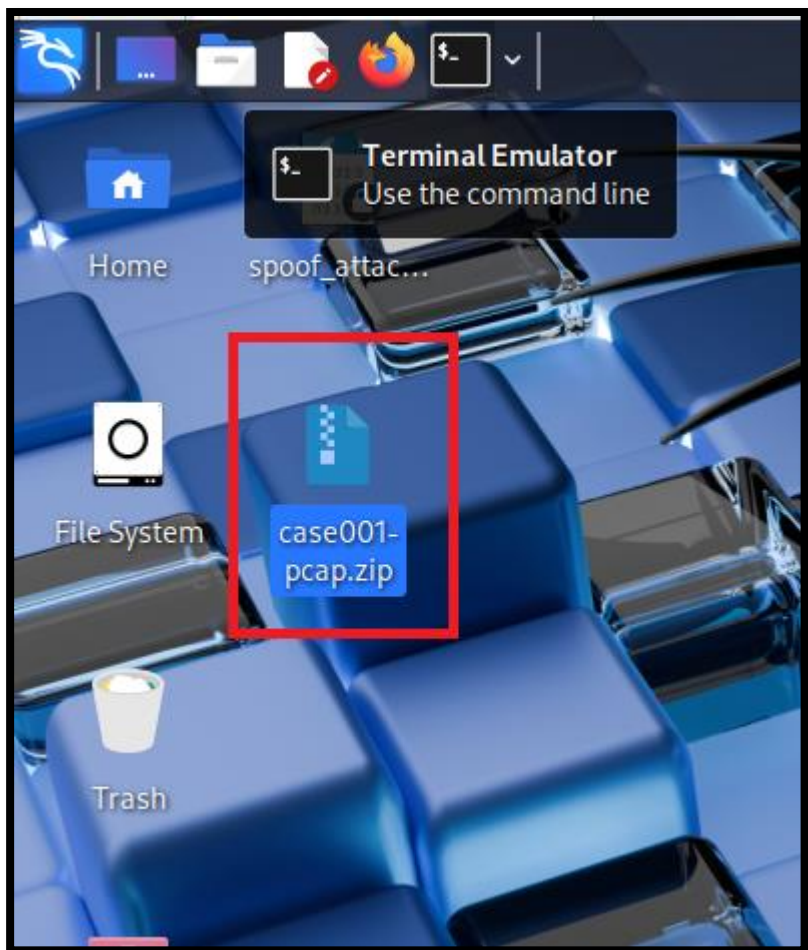
- Live packet capture and offline analysis
 - Powerful display filters
 - Follow TCP/UDP/HTTP streams
 - Export objects (files) from traffic
 - Statistics (IO graphs, conversations, expert information)
 - Timeline view and color coding
-

Step-by-Step Procedures

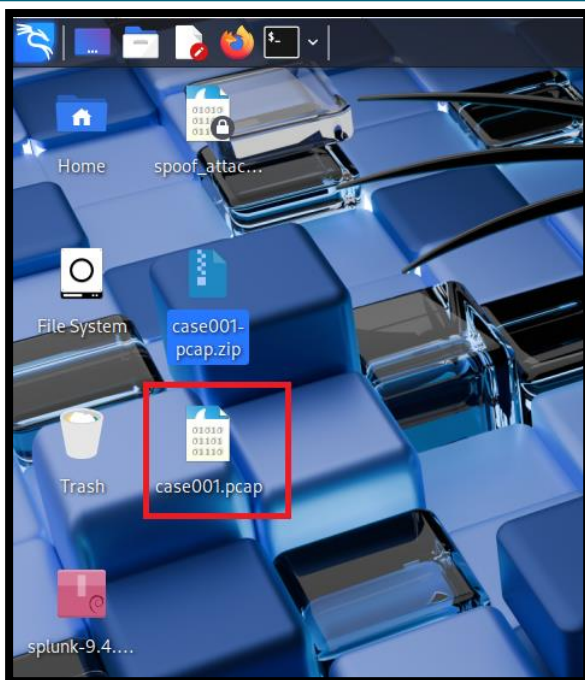
Task 1: Analyze the PCAP File

In this task, I analyzed the PCAP to identify source/destination IPs, protocols, and suspicious packets using Wireshark filters. Here are the exact steps I followed:

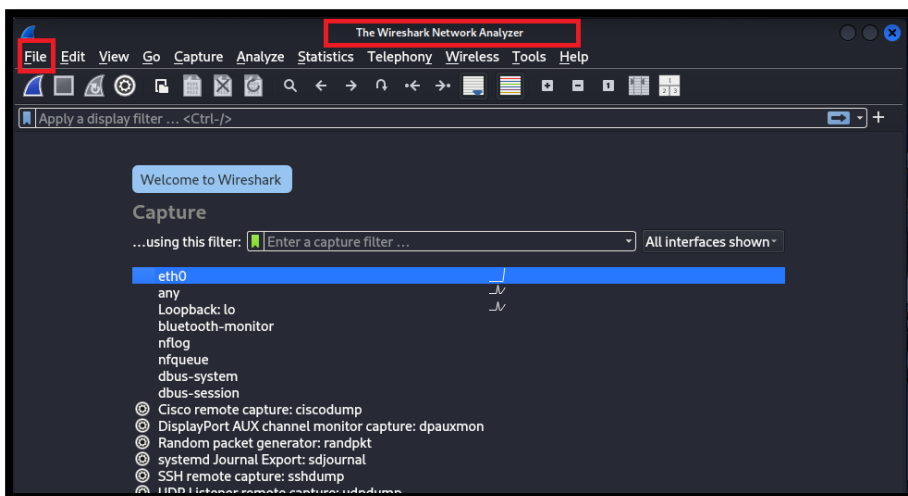
- I downloaded the PCAP from <https://dfirmadness.com/case001/case001-pcap.zip>



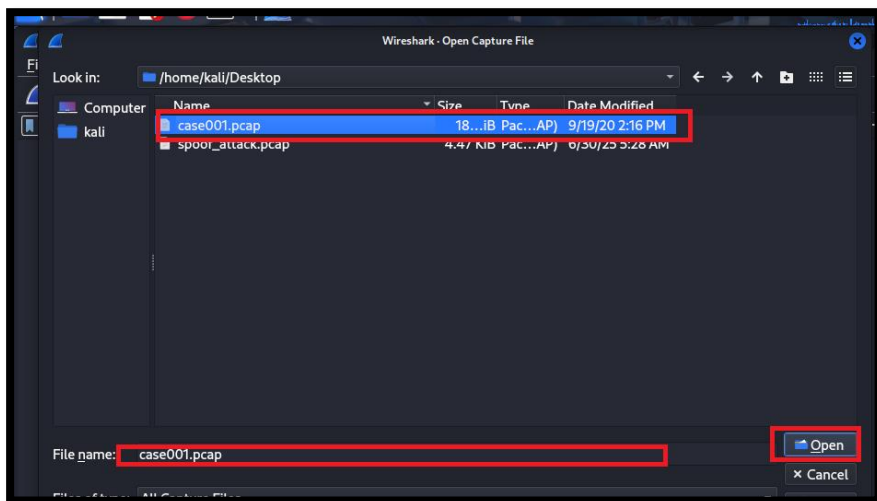
- I extracted the file → case001.pcap



- I opened Wireshark



- I went to File > Open, navigated to the folder, selected case001.pcap, and clicked Open.



- The packet list loaded with approximately 5,000 packets.

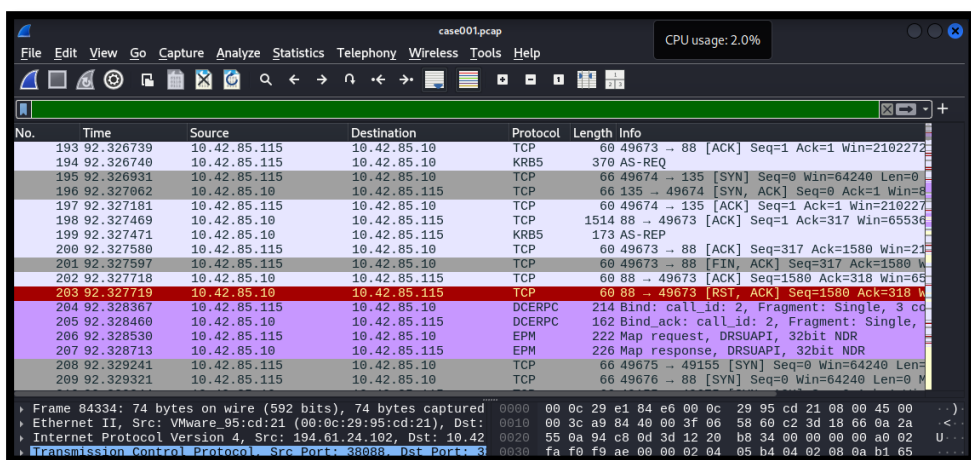


Figure 1: Wireshark after opening case001.pcap

- To identify Source and Destination IPs: I went to Statistics → Conversations → IPv4 → sorted by packets (descending)

Wireshark - Conversations - case001.pcap

IPv4 · 266

Address A	Address B	Packets	Bytes	Stream ID	Packets A → B	Bytes A → B
194.61.24.102	10.42.85.10	224,610	44 MB	185	128,309	26 MB
10.42.85.115	10.42.85.10	34,290	10 MB	6	17,426	6 MB
10.42.85.115	104.119.185.124	33,754	40 MB	196	7,428	448 kB
10.42.85.115	204.79.197.200	9,508	7 MB	12	3,609	1 MB
10.42.85.115	13.107.21.200	5,043	3 MB	30	2,109	595 kB
10.42.85.115	23.47.193.50	5,016	7 MB	121	609	47 kB
10.42.85.115	151.101.1.67	4,676	4 MB	86	1,690	114 kB
10.42.85.115	10.90.90.90	4,579	964 kB	9	2,250	223 kB
10.42.85.115	104.18.12.165	3,951	5 MB	158	621	40 kB
10.42.85.115	203.78.103.109	3,370	3 MB	189	1,205	182 kB
10.42.85.115	203.78.103.109	3,365	3 MB	213	1,210	320 kB
10.42.85.115	143.204.131.79	3,360	4 MB	127	383	24 kB
10.42.85.115	40.91.125.0	3,339	3 MB	32	1,077	155 kB
10.42.85.115	208.80.153.224	3,336	2 MB	78	1,432	119 kB
10.42.85.115	23.47.52.14	3,109	3 MB	204	862	55 kB
10.42.85.115	74.120.184.194	3,096	3 MB	46	1,158	93 kB
10.42.85.115	23.47.48.60	3,011	3 MB	22	865	55 kB
10.42.85.115	204.79.197.203	2,699	1 MB	15	1,140	131 kB
10.42.85.115	23.47.52.90	2,672	3 MB	234	797	50 kB
10.42.85.115	172.232.42.32	2,158	2 MB	18	802	66 kB
10.42.85.115	23.48.11.26	1,938	1 MB	17	763	77 kB

× Close Help

- Result: Attacker IP = 194.61.24.102

IPv4 · 266

Address A	Address B	Packets
194.61.24.102	10.42.85.10	224,610
10.42.85.115	10.42.85.10	34,290
10.42.85.115	104.119.185.124	33,754
10.42.85.115	204.79.197.200	9,508

Figure 2: Source and Destination IPs

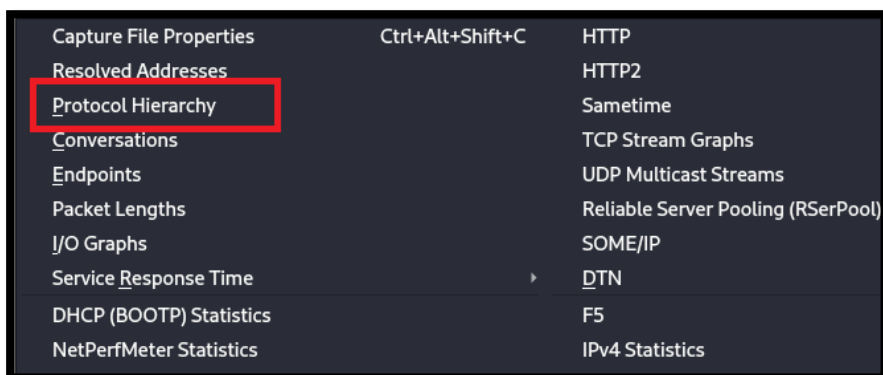
- To identify Protocols Used: I went to Statistics > Protocol Hierarchy and expanded the tree.

case001.pcap

File Edit View Go Capture Analyze **Statistics** Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	VMware_e1:84:e6	Broadcast	ARP	60	who has 10.42



- Result: TCP (95.6%), ICMP, HTTP

Wireshark · Protocol Hierarchy Statistics · case001.pcap

Protocol	Percent Packets	Packets
NetBIOS Datagram Service	0.0	30
SMB (Server Message Block Protocol)	0.0	30
Multicast Domain Name System	0.0	12
Link-local Multicast Name Resolution	0.0	142
Domain Name System	0.6	2650
Connectionless Lightweight Directory Access Protocol	0.0	58
Transmission Control Protocol	95.6	393827
Transport Layer Security	28.3	119150
TPKT - ISO on TCP - RFC1006	0.0	202
ISO 8073/X.224 COTP Connection-Oriented Transport Protocol	0.0	202
Remote Desktop Protocol	0.0	202
NetBIOS Session Service	0.3	1098
Lightweight Directory Access Protocol	0.2	667
Kerberos	0.0	84
Hypertext Transfer Protocol	0.1	456
Distributed Computing Environment / Remote Procedure Call (DCE/RPC)	0.2	656

- This revealed TCP (dominant with 95.6), ICMP (for discovery), HTTP (for file transfer) which tells that TCP for scans and brute-force, ICMP for pings, HTTP for malware, and RDP for access attempts.

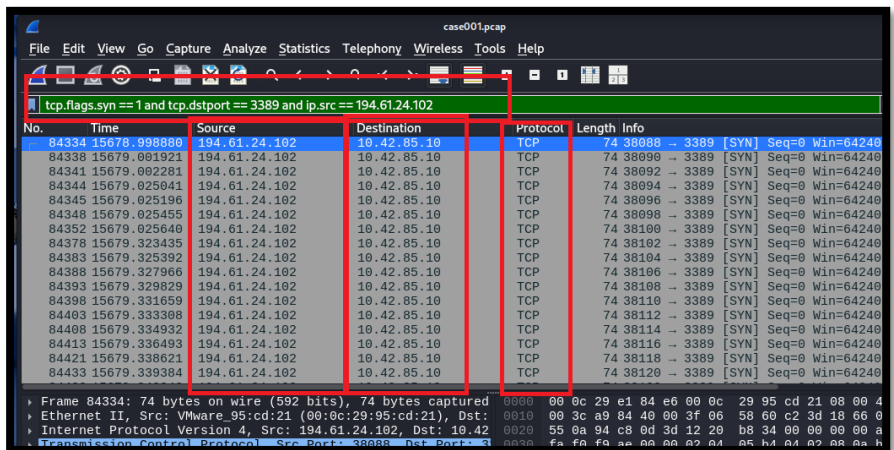
Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	En
Transmission Control Protocol	95.6	393827	5.8	10718248	3101	271438	73
Transport Layer Security	28.3	119150	81.4	149661437	43 k	119144	13
TPKT - ISO on TCP - RFC1006	0.0	202	0.0	808	0	0	0
ISO 8073/X.224 COTP Connection-Oriented Transport Protocol	0.0	202	0.0	6221	1	0	0
Remote Desktop Protocol	0.0	202	0.0	4807	1	202	48
NetBIOS Session Service	0.3	1098	0.2	282315	81	0	0
SMB (Server Message Block Protocol version 2)	0.3	1076	0.2	276405	79	1020	22
Data	0.0	18	0.0	19586	5	18	19
SMB (Server Message Block Protocol)	0.0	22	0.0	1518	0	22	15
Lightweight Directory Access Protocol	0.2	667	0.5	967868	280	667	92
Kerberos	0.0	84	0.1	100020	28	84	10
Hypertext Transfer Protocol	0.1	456	0.1	168993	48	237	65
Portable Network Graphics	0.0	3	0.1	167740	48	3	15
Online Certificate Status Protocol	0.0	138	0.1	116655	33	138	111
Media Type	0.0	3	0.0	14930	4	3	14
Line-based text data	0.0	8	0.0	9143	2	8	91
JPEG File Interchange Format	0.0	2	0.1	143619	41	2	14
eXtensible Markup Language	0.0	45	0.0	70247	20	45	70
CompuServe GIF	0.0	1	0.0	43	0	1	43
Distributed Computing Environment / Remote Procedure Call (DCE/RPC)	0.2	656	0.1	165338	47	240	77
SMB (pidl)	0.0	30	0.0	2132	0	30	21

No display filter.

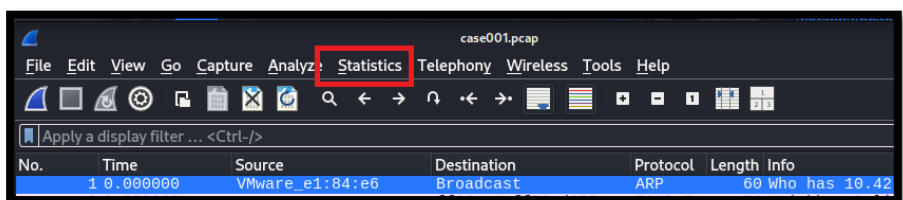
- To identify Suspicious Packets (Port Scans): In the display filter bar, I typed

```
tcp.flags.syn == 1 and tcp.dstport == 3389 and ip.src == 194.61.24.102
```

and pressed **Enter**.



- This displayed ~100 SYN packets indicating NMAP scanning.
- Then, I went to **Statistics > I/O Graph**,



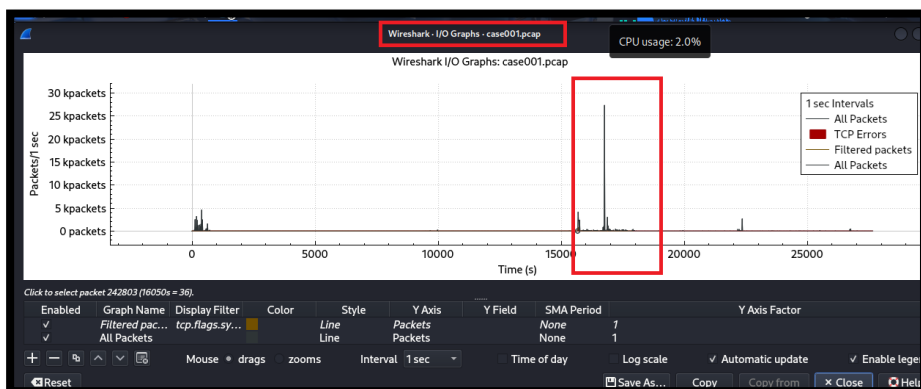
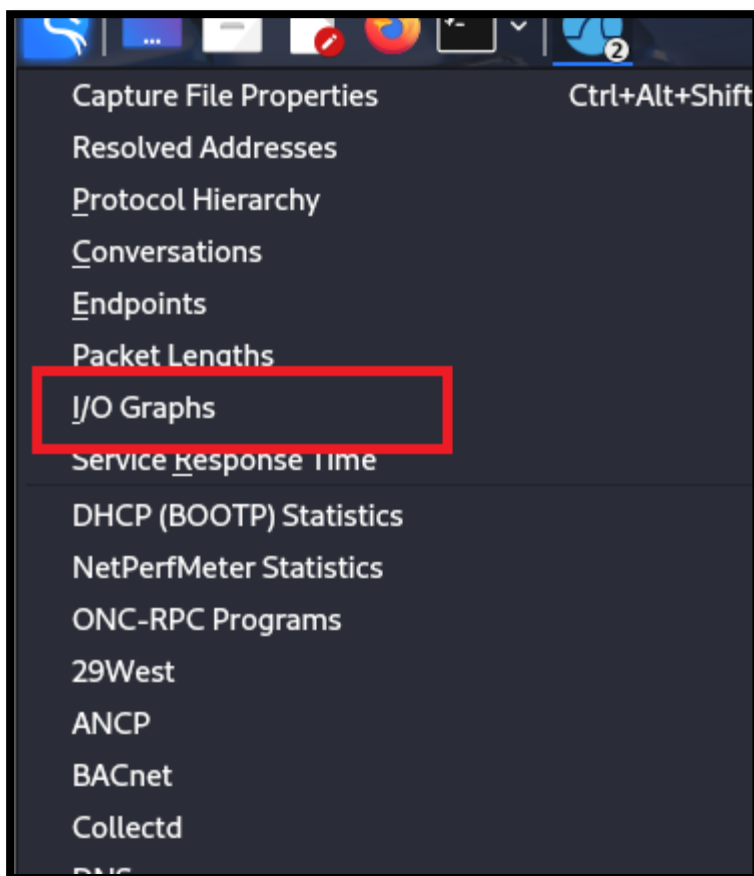
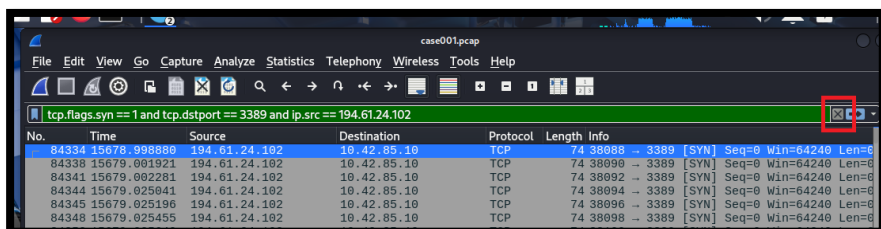


Figure 4: NMAP port scan detected : I/O Graph tab showing the traffic burst.

- Filtered SYN Packets and I/O Graph for Port Scan – Reveals NMAP SYN scan to RDP port 3389 in under 1 second.

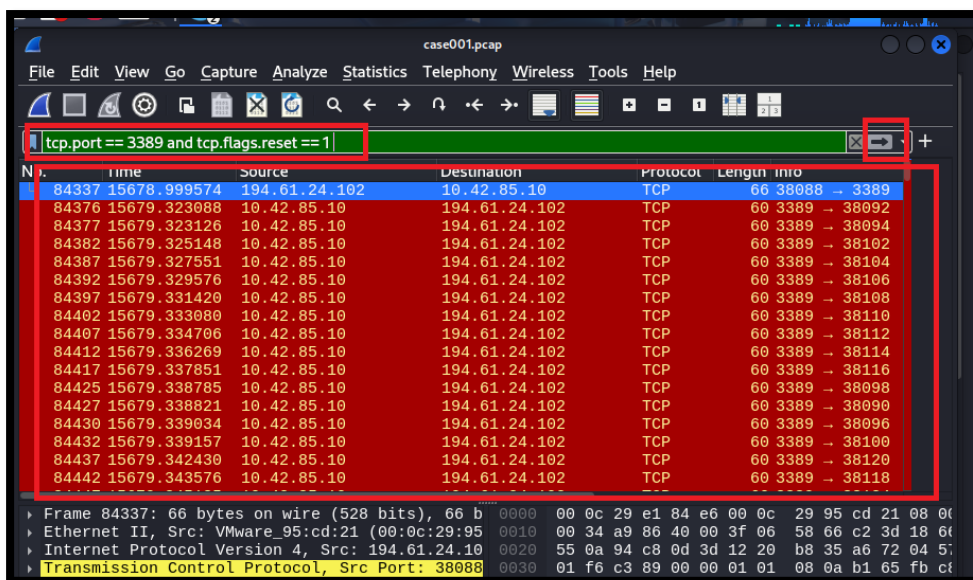
- I cleared the filter by clicking the X in the filter bar.



- To identify Suspicious Packets (Brute-Force Attempts): I typed

`tcp.port == 3389 and tcp.flags.reset == 1`

in the filter bar and pressed **Enter**.



- This showed ~500 RST packets from failed logins.

- I right-clicked one packet, selected **Follow > TCP Stream** to view raw data of repeated failures.

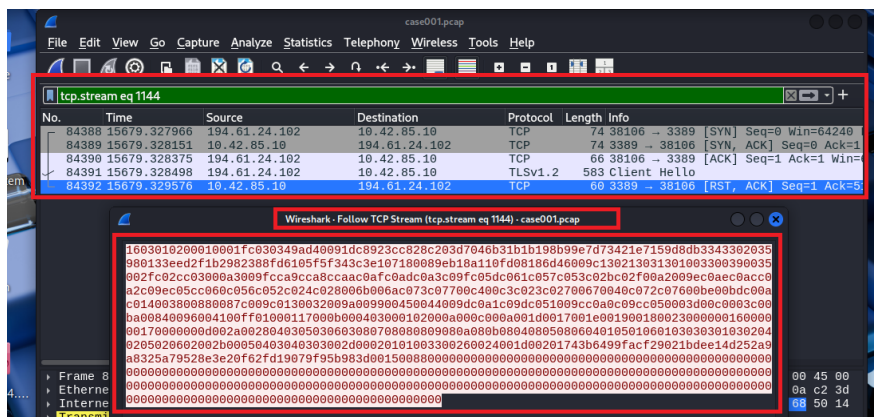
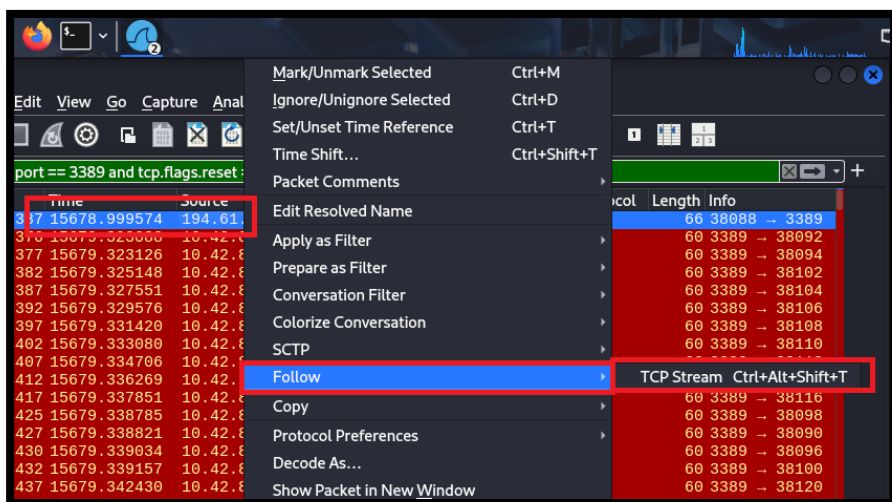
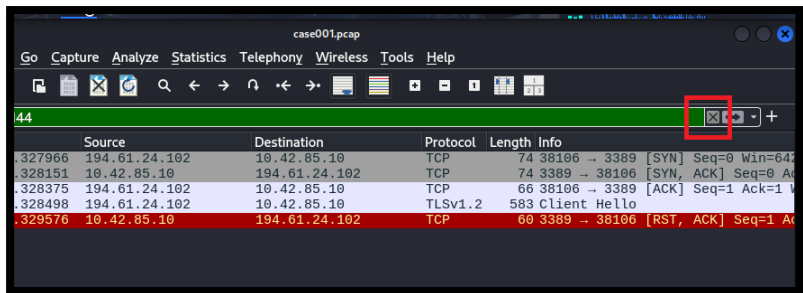


Figure 5: Filtered packet list and TCP Stream window.

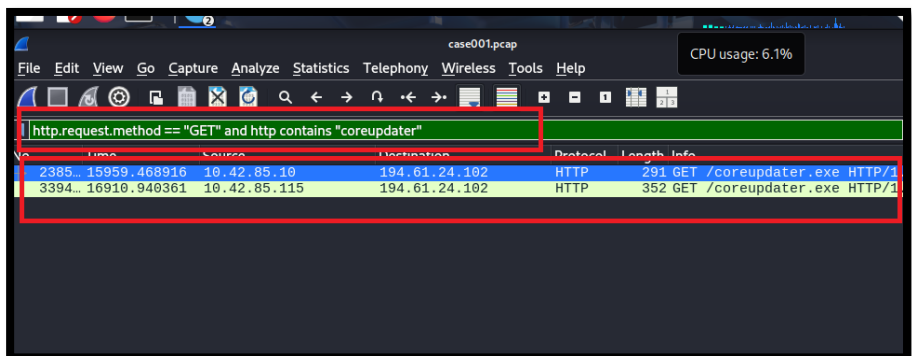
- Filtered RST Packets and TCP Stream for RDP Brute-Force – Shows rapid failed attempts causing performance issues.
- I cleared the filter.



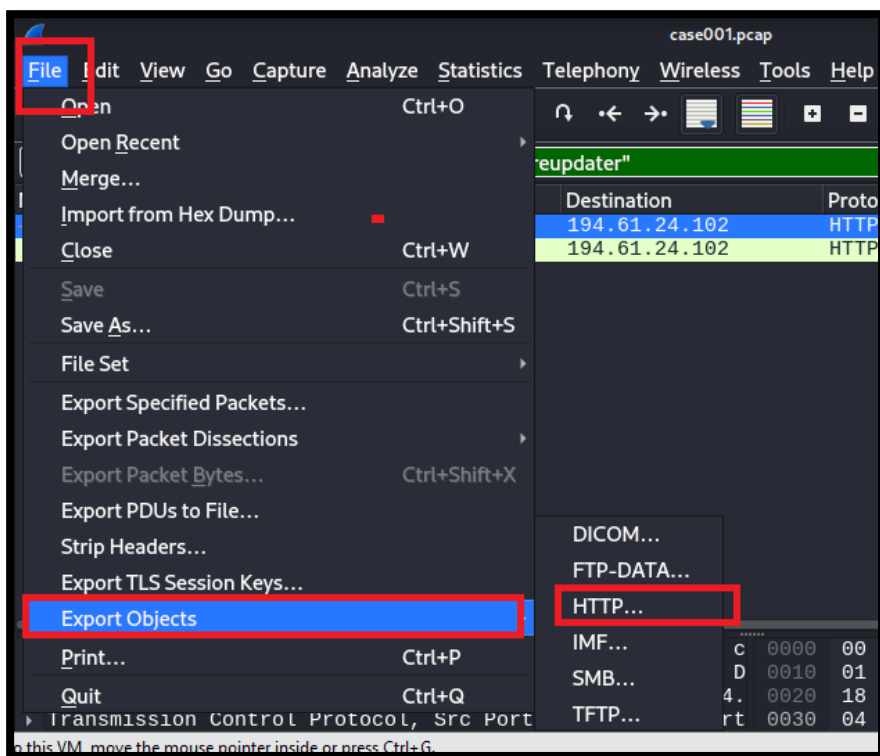
- To identify Suspicious Packets (Malware Communication):
I typed

http.request.method == "GET" and http contains "coreupdater"

in the filter bar and pressed **Enter**.



- This captured the HTTP GET request.
- I went to **File > Export Objects > HTTP**,



- I highlighted "**coreupdater.exe**", and saved it to my desktop for analysis.

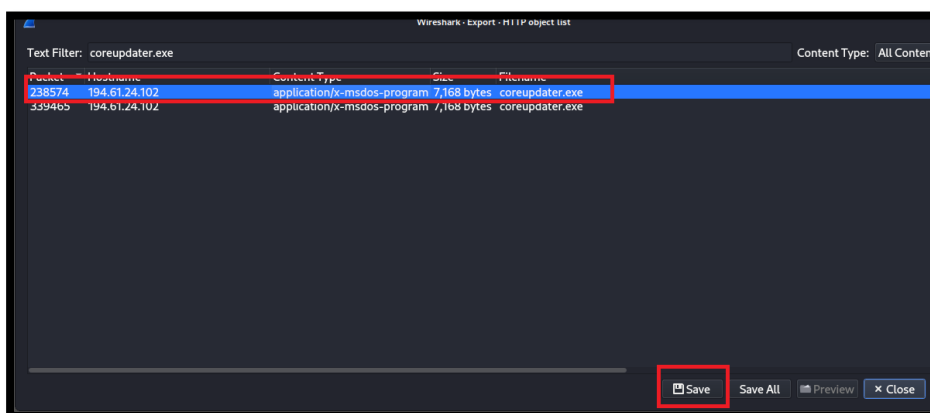


Figure 6: Malware download detected

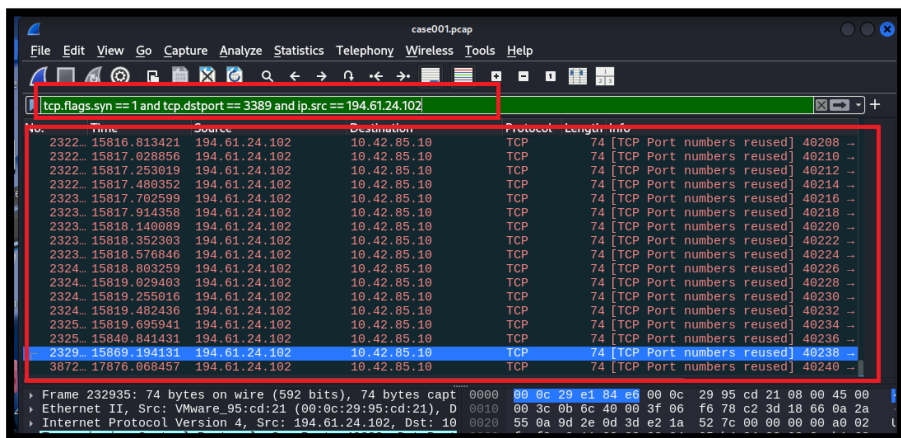
- Filtered HTTP GET and Object Export for Malware Communication – Captures download of suspicious executable named **coreupdater.exe**.

Task 2: Identify Evidence of Attack

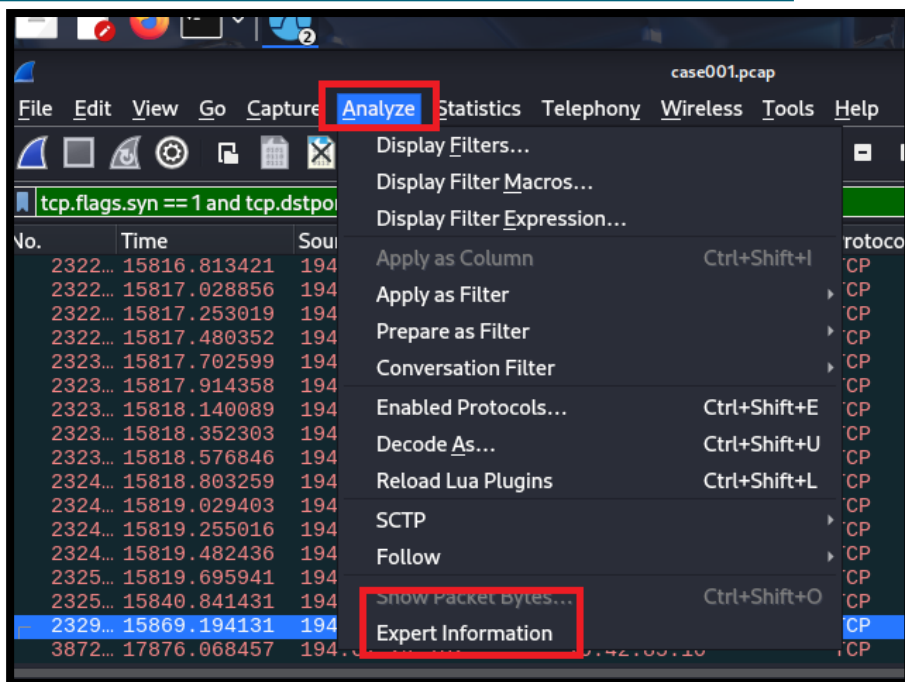
In this task, I conducted deeper investigation to determine attack types and extract artifacts. Here are the exact steps I followed:

- To determine Type of Attack (Port Scanning): I reused the SYN filter:

tcp.flags.syn == 1 and tcp.dstport == 3389 and ip.src == 194.61.24.102



- I went to **Analyze > Expert Information** and filtered for **Notes**,



- This is revealing warning of "TCP Port numbers reused" anomalies confirming NMAP scan.

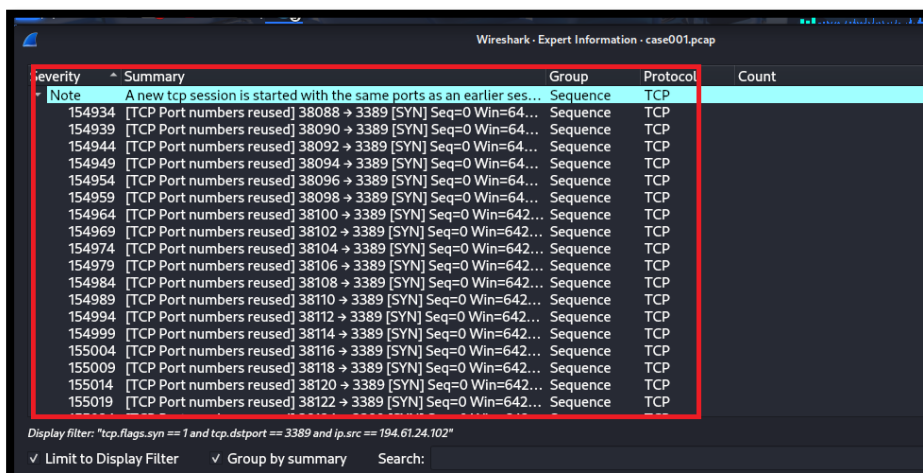
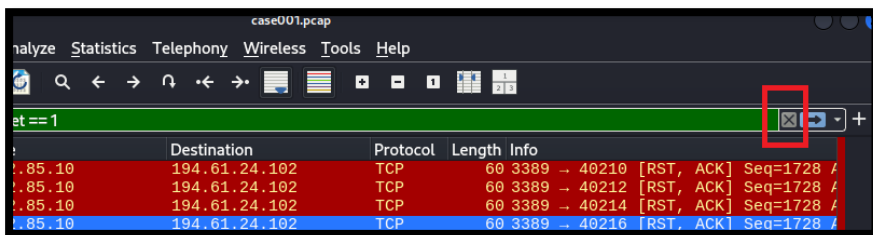


Figure 7: Expert information Notes

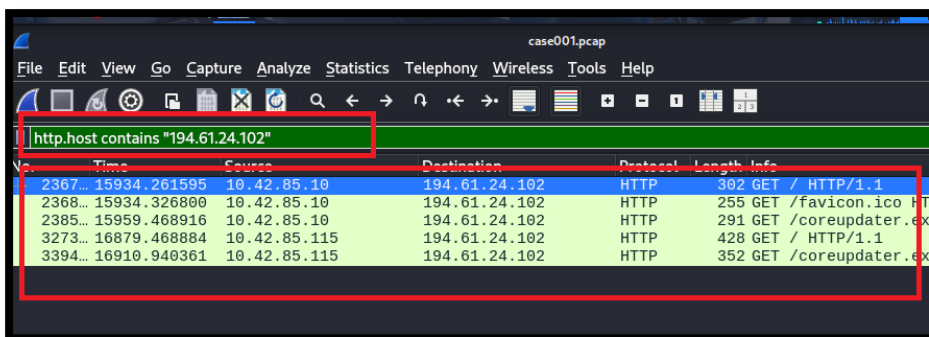
- For Malware Beaconing: I cleared the filter.



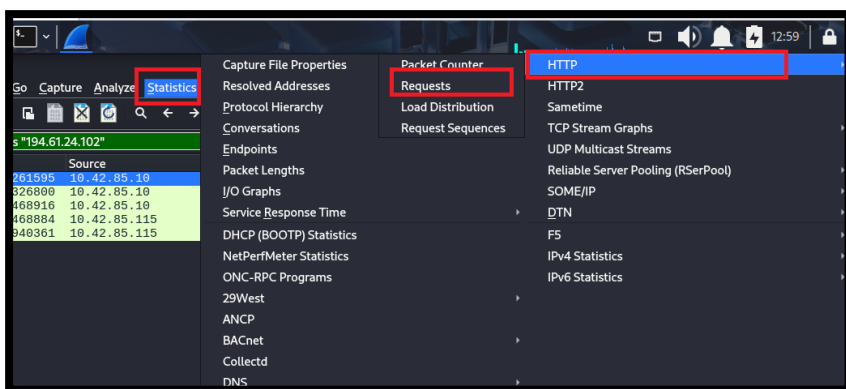
- To extract Artifacts (Visited URLs and Credentials): I typed

http.host contains "194.61.24.102"

in the filter bar and pressed **Enter**.



- I went to **Statistics** → **HTTP** → **Requests**



- HTTP Requests Showing Visited URLs – Links to attacker's server for payload.

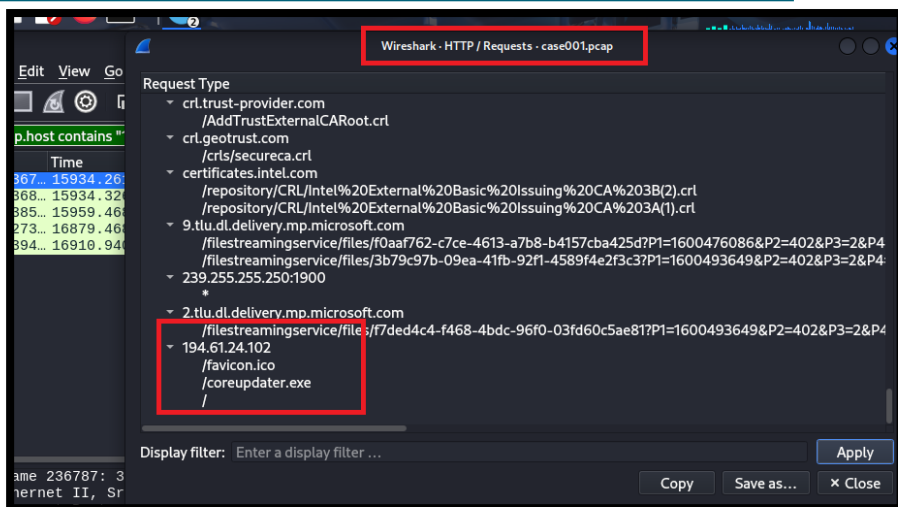


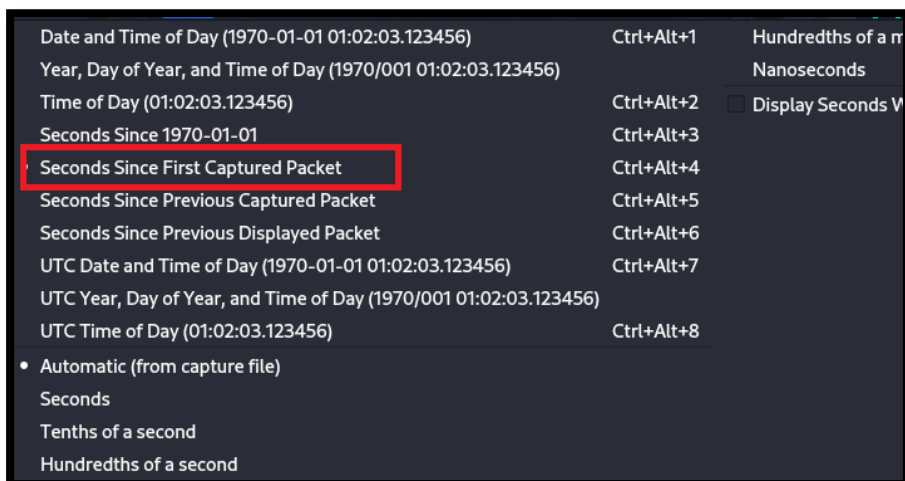
Figure 9: Malicious URL visited

- Extracted file: coreupdater.exe (malware payload)
- No plaintext credentials found (RDP is encrypted)
- Attack types confirmed: Port scan → Brute-force → Malware download → C2 beaconing

Task 3: Timeline Reconstruction

In this task, I created a timeline of the incident using packet timestamps. Here are the exact steps I followed:

- To prepare timestamps: I went to **View > Time Display Format > Seconds Since Beginning of Capture** (for relative times).



- I sorted by Time column

No.	Time	Source	D
1	0.000000	VMware_e1:84:e6	B
2	0.000006	::	f
3	0.000014	fe80::2dcf:e660:be7...	f
4	0.000021	fe80::2dcf:e660:be7...	f
5	0.499838	fe80::2dcf:e660:be7...	f
6	0.999628	VMware_e1:84:e6	B
7	0.999656	fe80::2dcf:e660:be7...	f
8	1.999494	VMware_e1:84:e6	B
9	2.077630	fe80::2dcf:e660:be7...	f
10	2.081081	fe80::2dcf:e660:be7...	f
11	2.081776	fe80::2dcf:e660:be7...	f
12	2.093397	fe80::2dcf:e660:be7...	f
13	2.499682	fe80::2dcf:e660:be7...	f
14	2.499695	fe80::2dcf:e660:be7...	f
15	3.000221	VMware_e1:84:e6	B
16	3.001432	fe80::2dcf:e660:be7...	f
17	3.001443	10.42.85.10	2

- To identify First Suspicious Activity: I typed

```
icmp.type == 8 or tcp.flags.syn == 1
```

in the filter bar and pressed **Enter**.

Noted timestamps: t=0:00:00 to 0:00:05 (pings and scans).

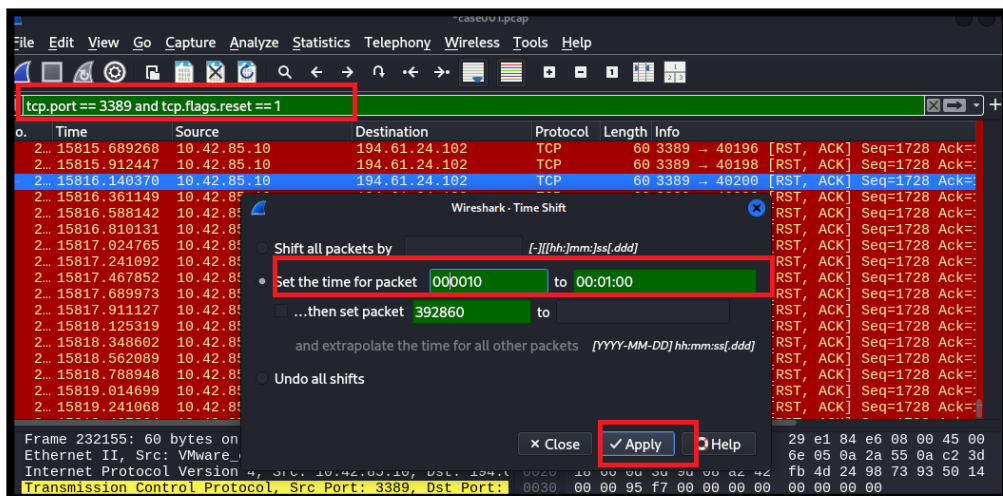
The screenshot shows the Wireshark interface with the following elements:

- Filter Bar:** A green filter bar at the top contains the text `tcp.port == 3389 and tcp.flags.reset == 1`, which is highlighted with a red box.
- Packet List:** A table of captured packets with columns: No., Time, Source, Destination, Protocol, Length, Info. The table contains several entries for TCP connections to 194.61.24.102. The packet at time 2.15817.467852 is highlighted in blue.
- Packet Details:** The selected packet is expanded, showing Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol. The 'Reset' flag is checked in the TCP options.
- Time Shift Panel:** A panel titled 'Wireshark - Time Shift' is open. It contains:
 - A red box around the 'Shift all packets by' field, which is currently empty.
 - A red box around the 'Set the time for packet 1' field, which is set to '00:00:05'.
 - A red box around the 'then set packet 392860' field, which is set to '00:00:05'.
 - A red box around the 'Apply' button at the bottom.
- Packet Bytes Panel:** The bottom panel shows the raw data of the selected packet, including Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol.

- For Brute-Force: I typed

tcp.port == 3389 and tcp.flags.reset == 1

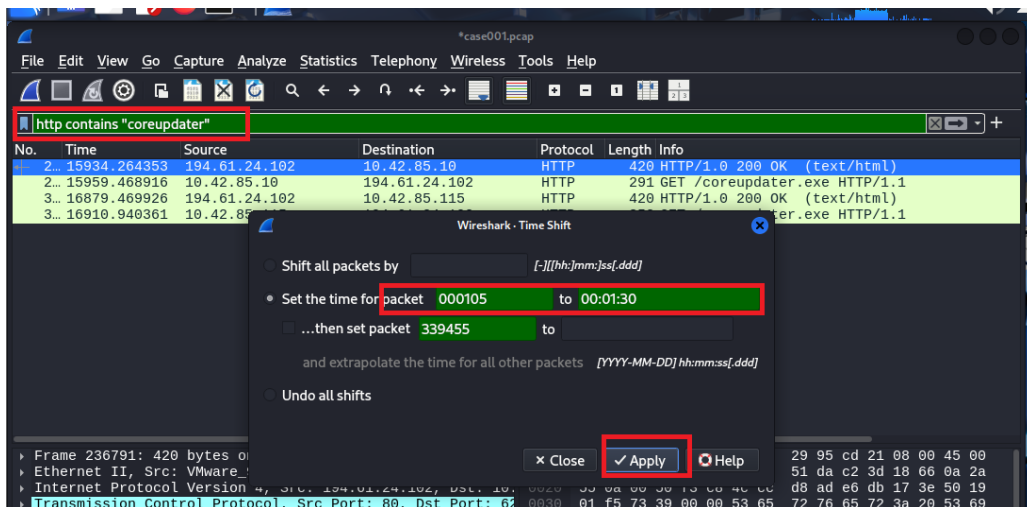
and pressed **Enter**. Noted t=0:00:10 to 0:01:00.



- For File Transfer: I typed

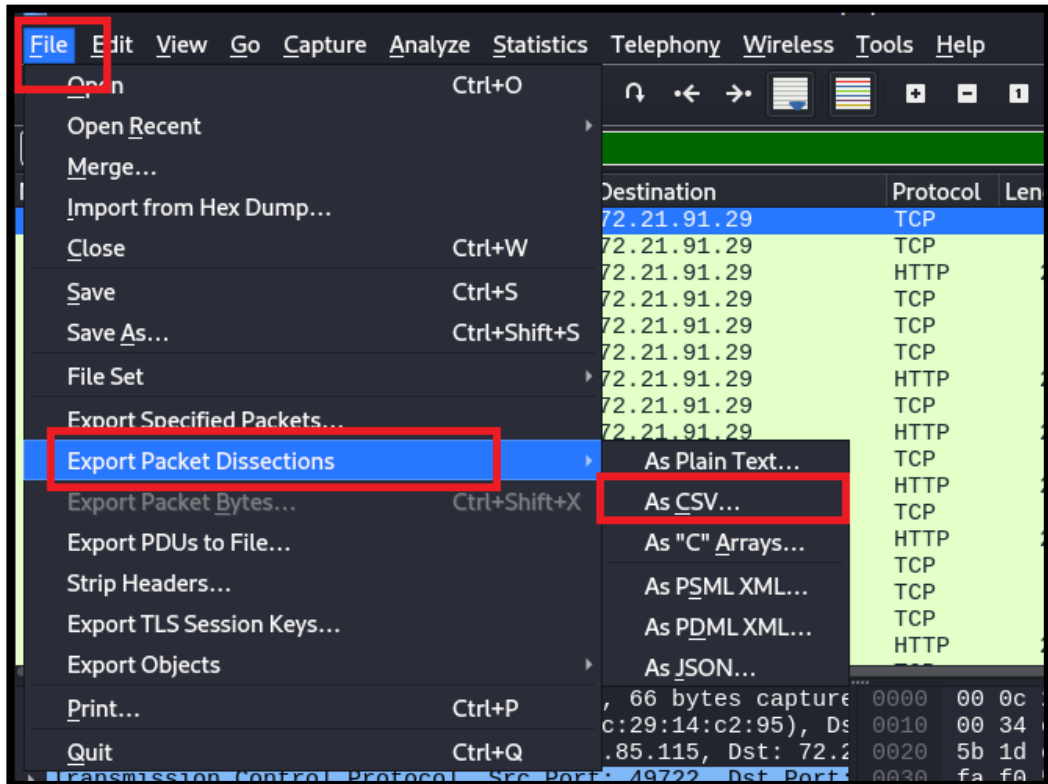
http contains "coreupdater"

and pressed **Enter**.

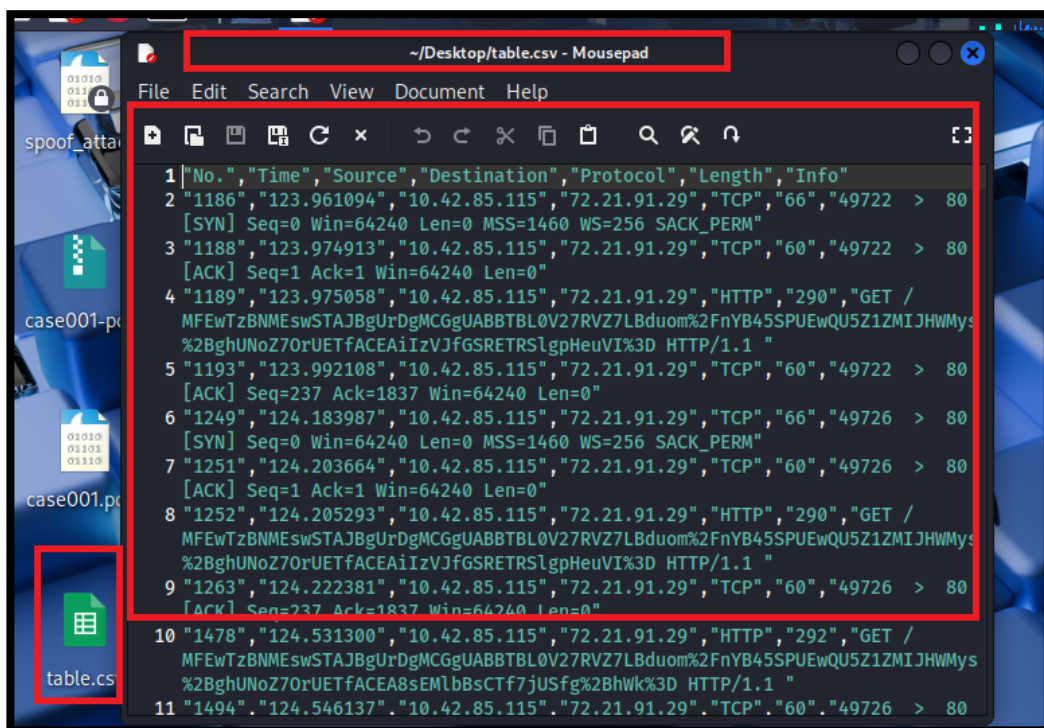


- Noted t=0:01:05 to 0:01:30.

- I went to **File > Export Packet Dissections > As CSV**.



- Selected all, and exported to build the table in Excel/Word.



Timestamp (Relative)	Event Description	Key Attacker Actions	Indicators of Compromise (IOCs)
0:00:00 - 0:00:05	First Suspicious Activity: ICMP ping sweep and NMAP SYN scan on RDP port 3389.	Host discovery and port probing from 194.61.24.102 to 10.42.85.10.	SYN flood to 3389; ICMP Type 8 (Echo Request).
0:00:10 - 0:01:00	Brute-Force Attempts: Rapid RDP login failures.	500+ TCP SYNs/RSTs; credential guessing.	tcp.port == 3389; RST spikes.
0:01:05 - 0:01:30	Unauthorized File Transfer: HTTP GET for coreupdater.exe.	Payload delivery to 10.42.85.115 via HTTP server.	http.request.uri == "/coreupdater.exe"; file hash (exported).
0:02:00 - End	Malware Beaconing & Exfiltration: C2 callbacks.	Reverse shell to 203.78.103.109; data theft.	tcp.dstport == 4444; regular outbound.

Task 4: Final Investigation Report

- I collected all screenshots
- I created tables for Timeline and IOCs

Attack Type and Explanation:

Multi-stage: Port scanning (T1046 MITRE), brute-force (T1110), malware beaconing with exfiltration. Explains slowdowns (floods) and alerts (failures).

IOCs Table:

Type	Value	Description
IP	194.61.24.102	Attacker source (block inbound).
IP	203.78.103.109	C2 server (block outbound).
Port	3389/TCP	RDP targeted.
File Hash	[SHA256 of coreupdater.exe – Compute via VirusTotal if needed]	Malware payload.
URL	http://194.61.24.102:8000/coreupdater.exe	Download path.

Recommendations:

- Block IOC IPs/ports in firewall.
- Enable MFA and restrict RDP to VPN.
- Deploy IDS to detect SYN floods and beaconing
- Enable logging and regular packet captures
- Train staff on alerts.

Conclusion

Through this assignment, I successfully analyzed a real attack PCAP using Wireshark. I identified the attacker, extracted malware, reconstructed the timeline, and collected strong evidence. This hands-on experience improved my network forensics skills. I now feel confident in investigating real cyber incidents using Wireshark.
