

---

# WIRESHARK- PCAP FILE ANALYSIS ASSIGNMENT NO: 2

**Name: ABHIJITH B**

**Email ID: abhijithbalagopal@gmail.com**

**Submission Date: 16/08/2025**

**Batch: CSA-June-2025**

---

# **INDEX**

- 1. Objective**
- 2. Introduction**
- 3. File 1 Analysis — FILE1.pcap (Unsecured Browsing)**
  - 2.1 Step-by-Step Analysis with Screenshots**
- 4. File 2 Analysis — File2. pcap (TCP SYN Flood)**
  - 3.1 Step-by-Step Analysis with Screenshots**
- 5. Results**
- 6. Recommendations**
- 7. Conclusion**

---

## 1. OBJECTIVE

The objective of this assignment is to analyze captured network traffic using Wireshark and identify:

- The key protocols involved in communication.
- Important conversations and data exchanges between endpoints.
- Potential anomalies or security issues, such as unencrypted credentials and denial-of-service attack patterns.

## 2. INTRODUCTION

Wireshark is a powerful and widely used open-source packet analyzer tool. It allows network administrators, security analysts, and researchers to inspect traffic at the packet level.

Unlike a firewall, which filters, blocks, or allows traffic, Wireshark is passive. It captures and analyzes packets without interfering with traffic flow.

In this assignment, two PCAP files were analysed:

- File 1 shows unsecured browsing with captured credentials.
- File 2 reveals a Distributed Denial of Service (DDoS) attack using a TCP SYN flood.

## 3. File 1 Analysis — FILE1.pcap (Unsecured Browsing)

Key Finding: A cleartext web login was captured in the PCAP file.

➤ Captured Request details:

- Method: POST
- Host: testphp.vulnweb.com
- Path: /userinfo.php
- Content-Type: application/x-www-form-urlencoded
- Content-Length: 20

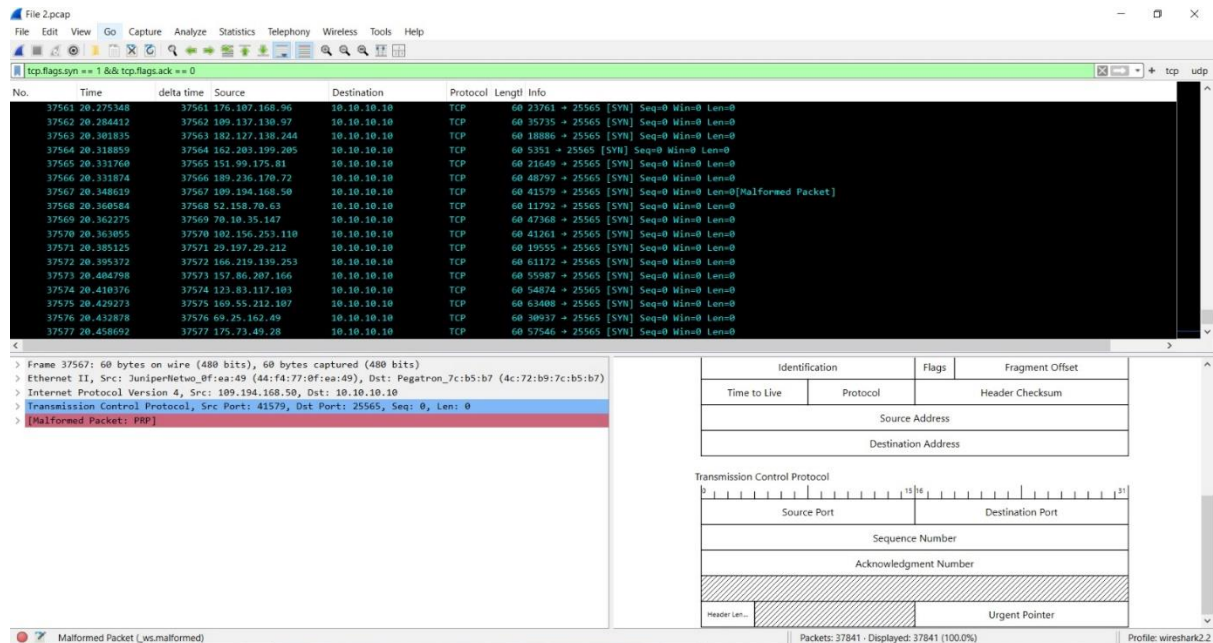
➤ Extracted Credentials:

- `uname=test`
- `pass=test`

Risk: Since credentials were sent over HTTP (without TLS), anyone monitoring the network could intercept and read them.

### 3.1 Step-by-Step Analysis with Screenshots

Step 1: Open FILE1.pcapng in Wireshark and locate HTTP requests.



*Wireshark open with applied TCP filter*

Step 2: Follow the TCP stream of the login request to extract credentials.

```
POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
Connection: keep-alive
Content-Length: 20
Cache-Control: max-age=0
Origin: http://testphp.vulnweb.com
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://testphp.vulnweb.com/login.php
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8

uname=test&pass=test
HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Mon, 04 Aug 2025 08:50:19 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Set-Cookie: login=test%2Ftest
Content-Encoding: gzip
```

*TCP Stream view showing captured credentials*

## 4. File 2 Analysis — File 2. Pcap (TCP SYN Flood)

**Key Finding:** The capture shows a Distributed TCP SYN flood attack targeting a Minecraft server at 10.10.10.10:25565.

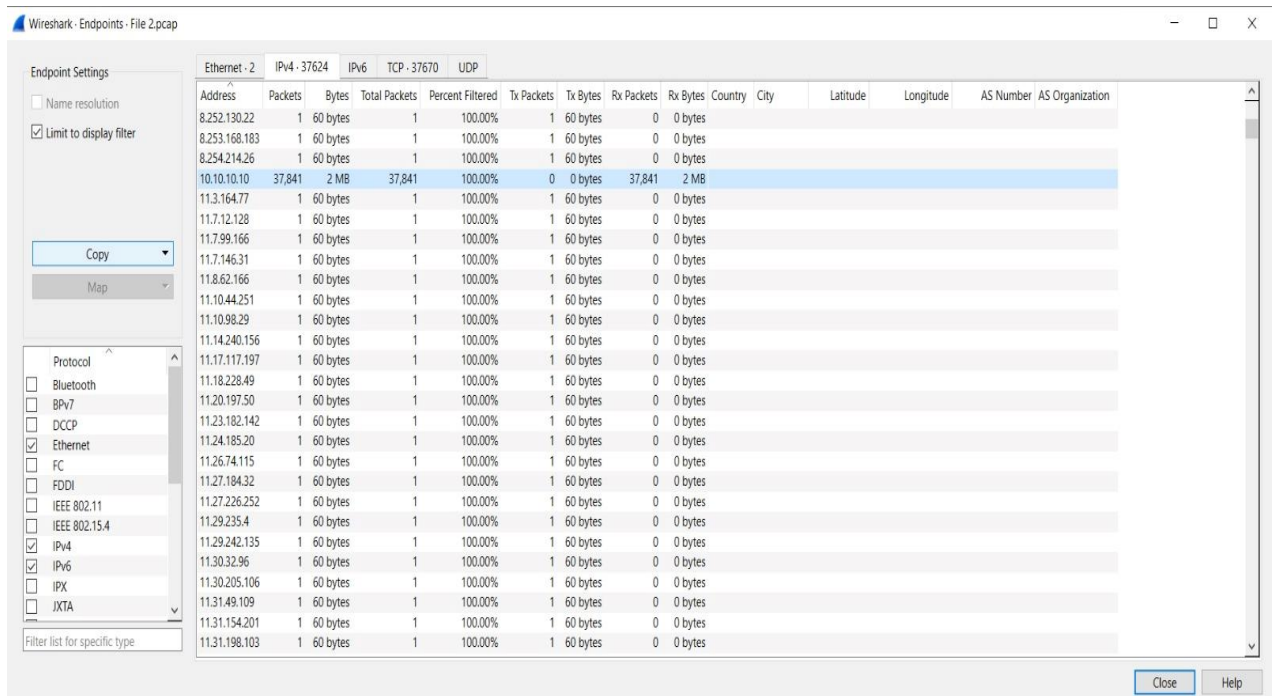
➤ **Capture Statistics:**

- Total packets: ~37,841 (all IPv4/TCP)
- Duration: ~23.68 seconds
- Destination IP: 10.10.10.10
- Destination Port: 25565
- TCP Flags: 37,841 SYN packets, 0 ACKs
- Unique source IPs: ~37,623
- Pattern: Only SYN packets observed, no completed handshakes

**Risk:** This is a SYN flood attack, designed to exhaust the server's backlog and deny service to legitimate users.

### 4.1 Step-by-Step Analysis with Screenshots

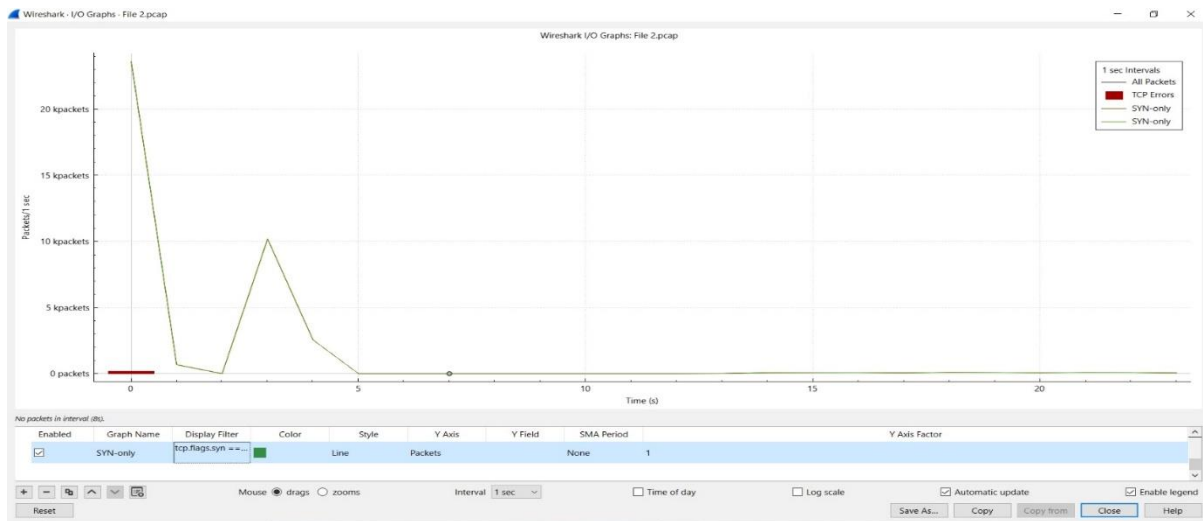
Step 1: Open File 2. pcap in Wireshark and apply a SYN flood filter.



Address	Packets	Bytes	Total Packets	Percent Filtered	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Country	City	Latitude	Longitude	AS Number	AS Organization
8.252.130.22	1	60 bytes	1	100.00%	1	60 bytes	0	0 bytes						
8.253.168.183	1	60 bytes	1	100.00%	1	60 bytes	0	0 bytes						
8.254.214.26	1	60 bytes	1	100.00%	1	60 bytes	0	0 bytes						
10.10.10.10	37,841	2 MB	37,841	100.00%	0	0 bytes	37,841	2 MB						
11.3.164.77	1	60 bytes	1	100.00%	1	60 bytes	0	0 bytes						
11.7.12.128	1	60 bytes	1	100.00%	1	60 bytes	0	0 bytes						
11.7.99.166	1	60 bytes	1	100.00%	1	60 bytes	0	0 bytes						
11.7.146.31	1	60 bytes	1	100.00%	1	60 bytes	0	0 bytes						
11.8.62.166	1	60 bytes	1	100.00%	1	60 bytes	0	0 bytes						
11.10.44.251	1	60 bytes	1	100.00%	1	60 bytes	0	0 bytes						
11.10.98.29	1	60 bytes	1	100.00%	1	60 bytes	0	0 bytes						
11.14.240.156	1	60 bytes	1	100.00%	1	60 bytes	0	0 bytes						
11.17.117.197	1	60 bytes	1	100.00%	1	60 bytes	0	0 bytes						
11.18.228.49	1	60 bytes	1	100.00%	1	60 bytes	0	0 bytes						
11.20.197.50	1	60 bytes	1	100.00%	1	60 bytes	0	0 bytes						
11.23.182.142	1	60 bytes	1	100.00%	1	60 bytes	0	0 bytes						
11.24.185.20	1	60 bytes	1	100.00%	1	60 bytes	0	0 bytes						
11.26.74.115	1	60 bytes	1	100.00%	1	60 bytes	0	0 bytes						
11.27.184.32	1	60 bytes	1	100.00%	1	60 bytes	0	0 bytes						
11.27.226.252	1	60 bytes	1	100.00%	1	60 bytes	0	0 bytes						
11.29.235.4	1	60 bytes	1	100.00%	1	60 bytes	0	0 bytes						
11.29.242.135	1	60 bytes	1	100.00%	1	60 bytes	0	0 bytes						
11.30.32.96	1	60 bytes	1	100.00%	1	60 bytes	0	0 bytes						
11.30.205.106	1	60 bytes	1	100.00%	1	60 bytes	0	0 bytes						
11.31.49.109	1	60 bytes	1	100.00%	1	60 bytes	0	0 bytes						
11.31.154.201	1	60 bytes	1	100.00%	1	60 bytes	0	0 bytes						
11.31.198.103	1	60 bytes	1	100.00%	1	60 bytes	0	0 bytes						

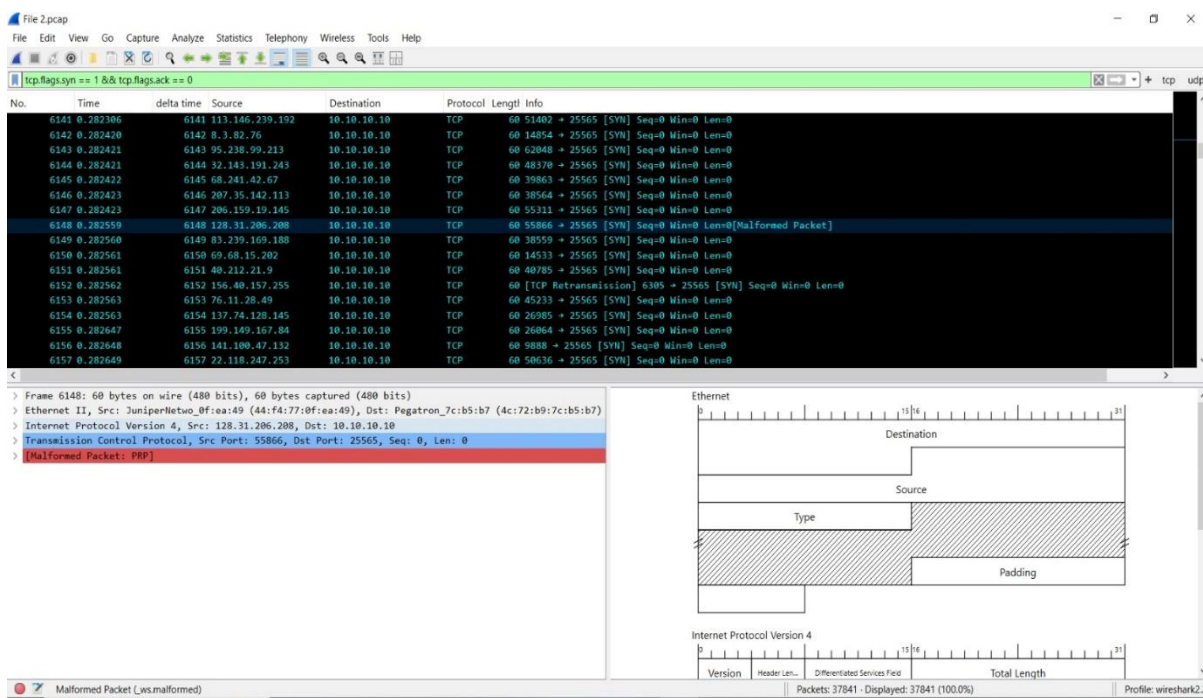
*Endpoints IPv4 showing high number of unique sources*

## Step 2: Visualize the attack using I/O Graphs.



*I/O Graph showing SYN flood spike*

## Step 3: Inspect TCP Conversations to confirm incomplete handshakes.



*TCP Conversations window showing SYN only, no ACKs*

## 5. RESULTS

- Wireshark successfully captured and decoded multiple protocols.
- Packet inspection showed how communication occurs step by step.
- Filtering made it easy to focus on specific protocols.

---

## 6. RECOMMENDATIONS

To mitigate the identified risks:

- For File 1 (cleartext login):
  - Enforce HTTPS/TLS to protect credentials.
  - Educate users to avoid insecure logins.
- For File 2 (SYN flood attack):
  - Enable SYN cookies on the server.
  - Apply firewall rules for rate-limiting or IP reputation filtering.
  - Use DDoS protection services (e.g., reverse proxy, cloud-based filters).
  - Restrict access with allow-lists when possible.

## 7. CONCLUSION

This assignment demonstrates how Wireshark can be used to analyze real-world network captures. From intercepting cleartext credentials to identifying DDoS patterns, packet analysis provides crucial insights for network security.

Wireshark remains an invaluable tool for learning, investigation, and defense in cybersecurity.