Deliverable 1: Scanning and Reporting Using Nessus Vulnerability Scanner

Hammaz Ahmed

Department of Professional Studies, Saint Louis University

CYBR-5220-21- Incident Response and Mitigation

Randy Sliva, Teacher.

November 11th, 2023

Deliverable 1: Scanning and Reporting Using Nessus Vulnerability Scanner

Target Selection

Sanofi is a multinational pharmaceutical company that deals with multiple aspects of healthcare and can include research, development, manufacturing, and distribution of pharmaceuticals. The primary reason for choosing this company is that pharmaceutical companies are among the most targeted industries out there because of their sensitive data, and valuable Personally identifiable information. This company belongs among the mid-sized companies and hence there are more chances of discovering vulnerabilities which will also be more common among smaller-sized companies. This report can also help other lower-level companies by matching the loopholes that are present in this project. A basic network scan was performed using Nessus Essentials and the report and a summary along with recommendations are provided below.

Audience

This report is for the CEO of our organization. It gives the higher-ups an overview of the current vulnerabilities presented in our website "https://www.sanofi.com/en". The report also provides recommendations and steps to take to minimize or completely eradicate the vulnerabilities present.

Figure 1

Visual Analysis of Vulnerabilities Present

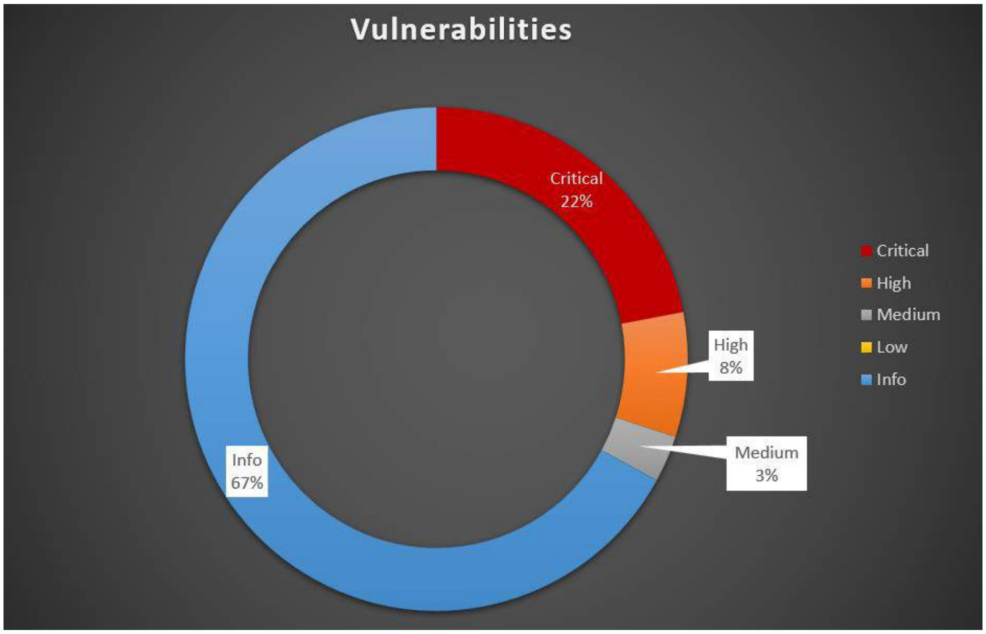


Figure 2

Report From Nessus

Table 1

Top 5 Vulnerabilities Present

CVE	CVSS/Severity	Vulnerability Name	Reason	Solution
CVE-2023-25690	9.8/ Critical	Apache 2.4.x < 2.4.56 Multiple Vulnerabilities	The version of Apache httpd installed on the remote host is prior to 2.4.56	Upgrade to Apache version 2.4.56 or later.
CVE-2021-44224	9.8/Critical	Apache 2.4.x >= 2.4.7 / < 2.4.52 Forward Proxy DoS / SSRF	A version of Apache httpd installed on the remote host is equal to or greater than 2.4.7 and prior to 2.4.52.	Upgrade to Apache version 2.4.52 or later.
CVE-2023-31122	7.5/High	Apache 2.4.x <2 .4.58 Multiple form of Vulnerabilities	The version of Apache httpd installed on the remote host is prior to 2.4.58.	Upgrade to Apache version 2.4.58 or later.
CVE-2021-36160	7.5/High	Apache >= 2.4.30 < 2.4.49 mod_proxy_uwsgi	The version of Apache httpd installed on the remote host is greater than 2.4.30 and is prior to 2.4.49.	Upgrade to Apache version 2.4.49 or later.
CVE-2016-6797	6.5/Medium	HSTS Missing From HTTPS Server (RFC 6797)	Remote web server is not enforcing HSTS, the lack of HSTS allows downgrade attacks, and SSL-stripping man-in-the-middle attacks, and weakens cookie-hijacking protections.	Configure the remote web server to use HSTS.

Assessment

The organization is at serious risk of security breaches due to the presence of critical vulnerabilities in its Apache HTTP Server software. These vulnerabilities could allow attackers to launch Denial of Service (DoS) attacks, which would overwhelm the server and make it unavailable to legitimate users, or Server-Side Request Forgery (SSRF) attacks, which could allow attackers to control the server and execute arbitrary commands. The organization's HTTPS server is also vulnerable to attack because it does not have HSTS (HTTP Strict Transport Security) enabled. HSTS instructs web browsers to always connect to the server over HTTPS, even if the user types in an HTTP URL. This helps to protect against attacks that downgrade security to HTTP, such as man-in-the-middle attacks. Without HSTS, sensitive data could be exposed to potential attackers.

Figure 3

Screenshot from Nessus Dashboard

Sev	CVSS	VPR	Name	Family	Count
MIXED	...	...	11 Apache Httpd (M...	Web Servers	22
MIXED	...	...	4 HTTP (Multiple Is...	Web Servers	5
INFO	...	...	4 SSL (Multiple Issu...	General	4
INFO	...	...	2 IETF Md5 (Multipl...	General	2
INFO	...	...	2 TLS (Multiple Issu...	General	2
INFO			Service Detection	Service detection	3
INFO			Apache HTTP Server V...	Web Servers	2
INFO			Nessus SYN scanner	Port scanners	2
INFO			SolarWinds Server & A...	CGI abuses	2
INFO			Web Server No 404 Er...	Web Servers	2
INFO			Common Platform En...	General	1
INFO			Device Type	General	1

Recommendations

1. **Apache HTTP Server:** Our organization should prioritize updating all Apache servers. This will automatically address critical vulnerabilities mentioned in table 1.

2. **HSTS Configuration:** Upgrading to HSTS (HyperText Strict Transport Security) will better the HTPPS connection, hence preventing man-in-the-middle attack and downgrade attacks.
3. **Regular Vulnerability Scanning:** Regular scans can help identify new security weaknesses as they arise and enable proactive measures to minimize risks.
4. **Patch Management:** Having and implementing robust patch management procedures and rules will reduce the chances of exposure to vulnerabilities.

Remaining Vulnerabilities

As shown in Figure 1 above there are other vulnerabilities as well which I haven't mentioned in my table of top 5 vulnerabilities. For instance, "HTTP/2 Cleartext Detection". This can be avoided by regulating and limiting incoming traffic coming to this port. As for all the other minor vulnerabilities present, as recommended if we update our systems and patch them now and, in the future, then these vulnerabilities will occur at a very low rate and won't pose any threat to our organization.

Conclusion

It is essential to address the critical vulnerabilities found in the Apache HTTP Server and implement HSTS in the organization's web servers to strengthen the security posture. Immediate action should be taken to update and patch vulnerable systems, and ongoing security measures should be implemented to prevent the emergence of new vulnerabilities. A proactive approach to security is crucial to protect the organization's digital assets and data.

Reference

How To: Run Your First Vulnerability Scan with Nessus. (2023, October 31). Tenable®. https://www.tenable.com/blog/how-to-run-your-first-vulnerability-scan-with-nessus

KtechHub. (2019, September 2). *How to do Vulnerability Scanning with Nessus* [Video]. YouTube. https://www.youtube.com/watch?v=35a0VhzIO2Y

Rangapur, A. (2021, December 16). Vulnerability scanning using Nessus Essentials - Security at your desk - Medium. *Medium*. https://medium.com/security-at-your-desk/vulnerability-scanning-using-nessus-essentials-c1a6b71c21f8

Deliverable 2: Unleashing and Defending Against a SYN Flood

Hammaz Ahmed

Department of Professional Studies, Saint Louis University

CYBR-5220-21- Incident Response and Mitigation

Randy Sliva, Teacher.

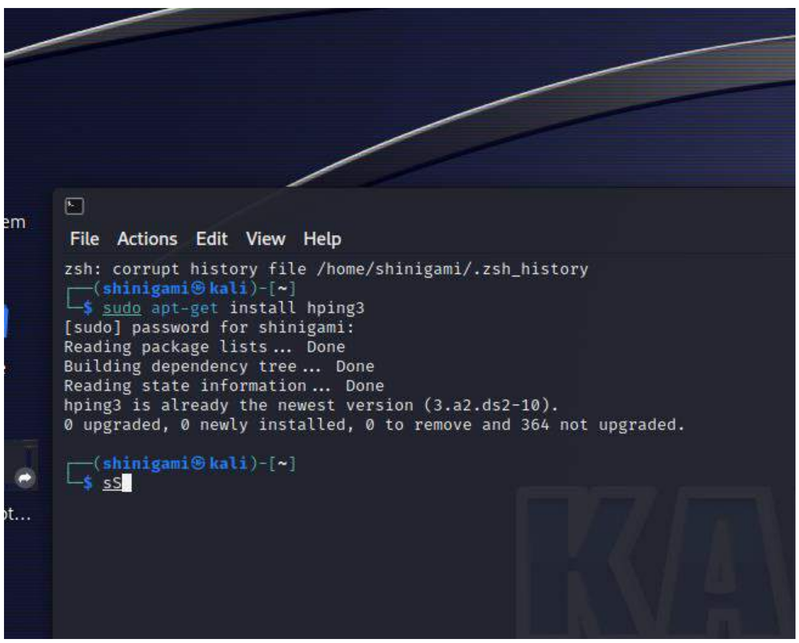
November 10th, 2023

Deliverable 2: Unleashing and Defending Against a SYN Flood

Red side Attack

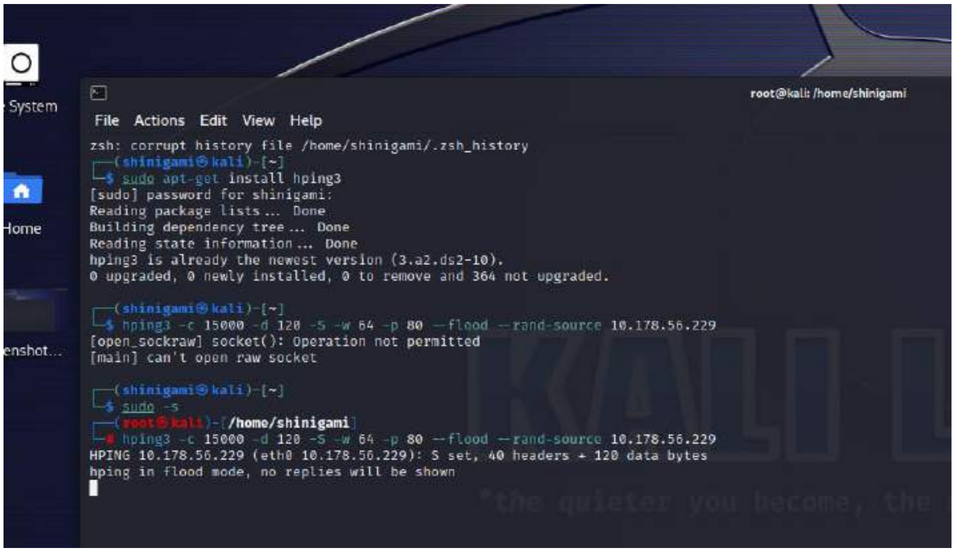
I performed a TCP Synflood attack using Kali Linux on my own computer. I performed the attack using hping3 which is a tool used to send ICMP/UDP/TCP packets. Below are the steps I followed.

1. Using the command prompt and typing "ipconfig" I made a note of my IP Address.
2. Opened Kali Linux using Virtual Box. First, we must install hping3 using the "sudo-apt get install hping3". Refer to the screenshot below.



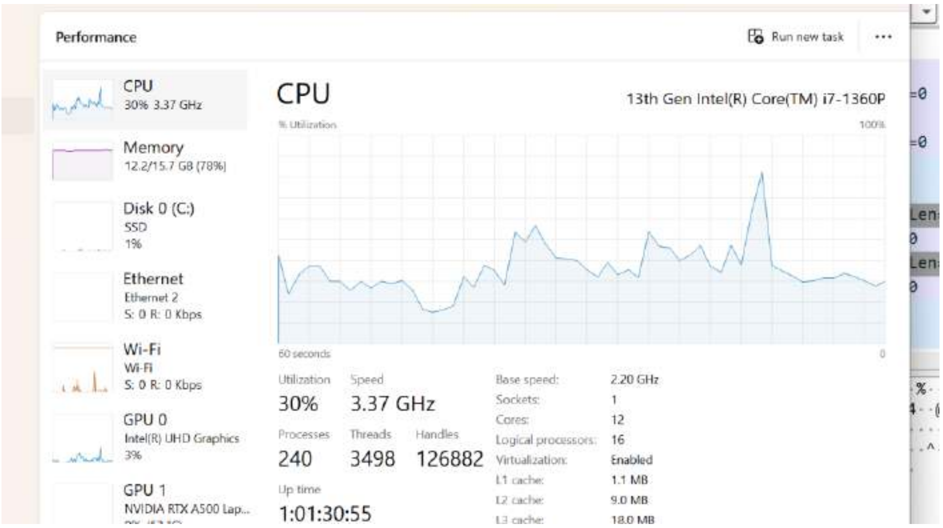
3. Next, I used the following command. hping3 -c 15000 -d 120 -S -w 64 -p 80 --flood --rand-source 10.178.56.229. There might be an error of "can't open raw socket". For this,

we must first use “sudo -s” which is used to gain elevated privilege. Refer to the screenshot below.



4. Let's break down the above command. -c 15000 means we are sending 15000 packets. -d 120 means each packet is 120 bytes. -S says SYN flag is enabled with a TCP window size of 64(-w 64), we are directing the attack on port 80 with -p 80. –flood indicates sending packets as fast as possible. --rand source helps with spoofed IP addresses to disguise the real source.

Note: The below screenshot is from the time the attack was taking place. A sudden rise in the CPU performance was seen, proving the attack started and was successful.



Conclusion:

The above attack was performed using Kali Linux and hping3 where we sent numerous TCP packets to a target which was my own system in this instance. I provided step-by-step instructions along with screenshots for visual presentation. This explains how easily an attacker can inundate a system with a barrage of malicious packets, causing a surge in CPU usage and initiating a successful attack.

<u>Blue Team Analysis:</u>

Incident: ITHUB/2023HP

Date: 9th November 9, 2023.

Incident Title: Suspicious Pcap File Analysis

Person in charge: Hammaz Ahmed. Incident Response Analyst.

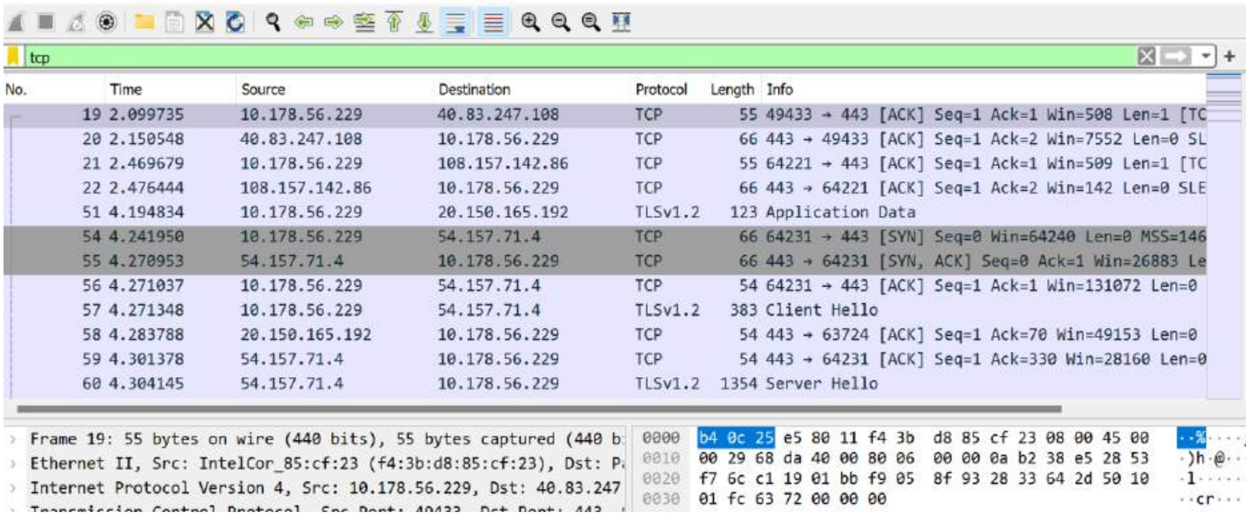
Incident Description: On 10th November 2023 the blue team received a pcap file containing traffic from the previous day. The file had some suspicious activity, and the purpose of this analysis is to find out the severity of this incident and to gather any insights if possible.

Artifact Listing: 

Tools utilized: Wireshark.

Action and Analysis

1. First, I took a scroll glance at the whole pcap file. It had some red flags for the TCP stream. Hence, I filtered by typing “tcp” in the filter column. Not much can be drawn from just this.



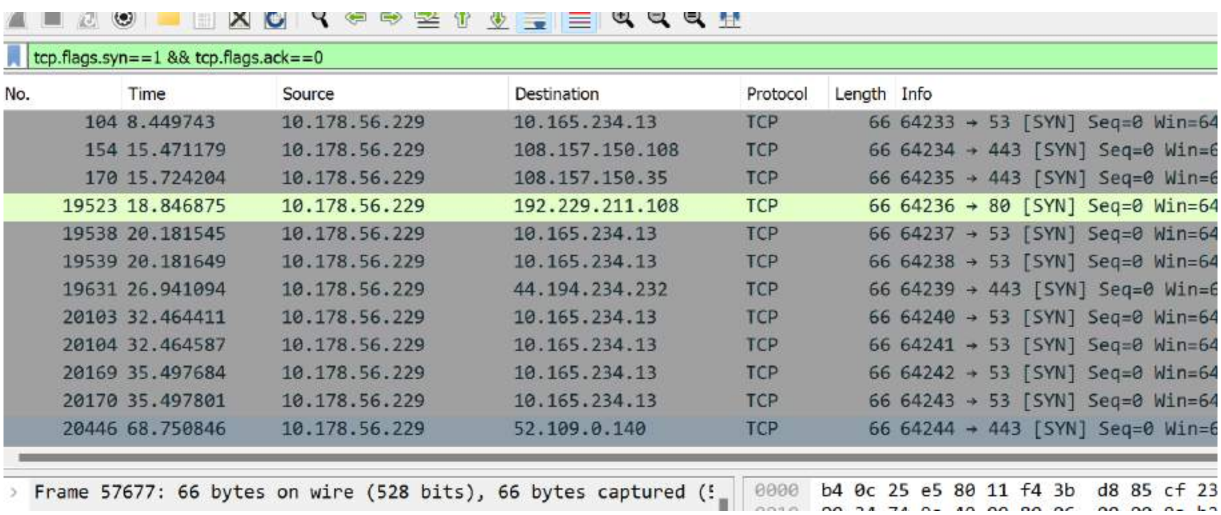
tcp

No.	Time	Source	Destination	Protocol	Length	Info
19	2.099735	10.178.56.229	40.83.247.108	TCP	55	49433 → 443 [ACK] Seq=1 Ack=1 Win=508 Len=1 [TC
20	2.150548	40.83.247.108	10.178.56.229	TCP	66	443 → 49433 [ACK] Seq=1 Ack=2 Win=7552 Len=0 SL
21	2.469679	10.178.56.229	108.157.142.86	TCP	55	64221 → 443 [ACK] Seq=1 Ack=1 Win=509 Len=1 [TC
22	2.476444	108.157.142.86	10.178.56.229	TCP	66	443 → 64221 [ACK] Seq=1 Ack=2 Win=142 Len=0 SLE
51	4.194834	10.178.56.229	20.150.165.192	TLSv1.2	123	Application Data
54	4.241950	10.178.56.229	54.157.71.4	TCP	66	64231 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=146
55	4.270953	54.157.71.4	10.178.56.229	TCP	66	443 → 64231 [SYN, ACK] Seq=0 Ack=1 Win=26883 Le
56	4.271037	10.178.56.229	54.157.71.4	TCP	54	64231 → 443 [ACK] Seq=1 Ack=1 Win=131072 Len=0
57	4.271348	10.178.56.229	54.157.71.4	TLSv1.2	383	Client Hello
58	4.283788	20.150.165.192	10.178.56.229	TCP	54	443 → 63724 [ACK] Seq=1 Ack=70 Win=49153 Len=0
59	4.301378	54.157.71.4	10.178.56.229	TCP	54	443 → 64231 [ACK] Seq=1 Ack=330 Win=28160 Len=0
60	4.304145	54.157.71.4	10.178.56.229	TLSv1.2	1354	Server Hello

Frame 19: 55 bytes on wire (440 bits), 55 bytes captured (440 b
Ethernet II, Src: IntelCor_85:cf:23 (f4:3b:d8:85:cf:23), Dst: P
Internet Protocol Version 4, Src: 10.178.56.229, Dst: 40.83.247

```
0000  b4 0c 25 e5 80 11 f4 3b  d8 85 cf 23 08 00 45 00
0010  00 29 68 da 40 00 80 06  00 00 0a b2 38 e5 28 53
0020  f7 6c c1 19 01 bb f9 05  8f 93 28 33 64 2d 50 10
0030  01 fc 63 72 00 00 00
```

2. I used the filter “**tcp.flags.syn==1 && tcp.flags.ack==0**”. This filter helps in TCP packets that are part of the 3-way handshake. Missing a proper 3-way handshake can give us a hint of synflood attacks. In fact, we do see loads of **synchronization packets** but no sign of a **complete handshake**.



tcp.flags.syn==1 && tcp.flags.ack==0

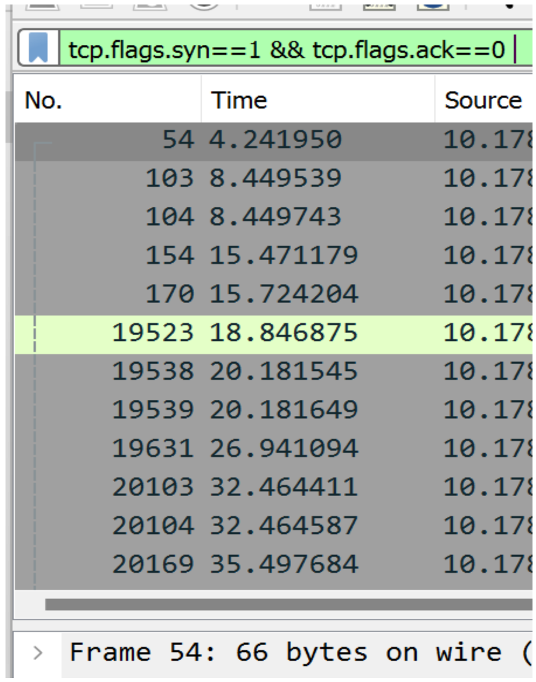
No.	Time	Source	Destination	Protocol	Length	Info
104	8.449743	10.178.56.229	10.165.234.13	TCP	66	64233 → 53 [SYN] Seq=0 Win=64
154	15.471179	10.178.56.229	108.157.150.108	TCP	66	64234 → 443 [SYN] Seq=0 Win=6
170	15.724204	10.178.56.229	108.157.150.35	TCP	66	64235 → 443 [SYN] Seq=0 Win=6
19523	18.846875	10.178.56.229	192.229.211.108	TCP	66	64236 → 80 [SYN] Seq=0 Win=64
19538	20.181545	10.178.56.229	10.165.234.13	TCP	66	64237 → 53 [SYN] Seq=0 Win=64
19539	20.181649	10.178.56.229	10.165.234.13	TCP	66	64238 → 53 [SYN] Seq=0 Win=64
19631	26.941094	10.178.56.229	44.194.234.232	TCP	66	64239 → 443 [SYN] Seq=0 Win=6
20103	32.464411	10.178.56.229	10.165.234.13	TCP	66	64240 → 53 [SYN] Seq=0 Win=64
20104	32.464587	10.178.56.229	10.165.234.13	TCP	66	64241 → 53 [SYN] Seq=0 Win=64
20169	35.497684	10.178.56.229	10.165.234.13	TCP	66	64242 → 53 [SYN] Seq=0 Win=64
20170	35.497801	10.178.56.229	10.165.234.13	TCP	66	64243 → 53 [SYN] Seq=0 Win=64
20446	68.750846	10.178.56.229	52.109.0.140	TCP	66	64244 → 443 [SYN] Seq=0 Win=6

Frame 57677: 66 bytes on wire (528 bits), 66 bytes captured (

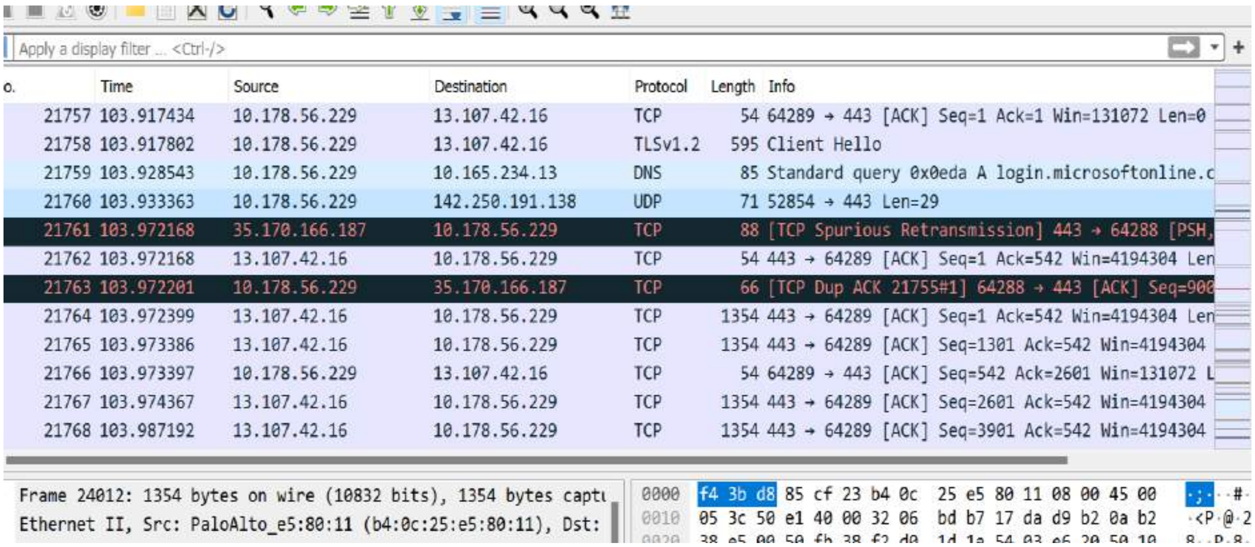
```
0000  b4 0c 25 e5 80 11 f4 3b  d8 85 cf 23
```

3. I can also see a **sudden rise in packets** and continuously receiving similar ones. This is also an indicator of a synflood attack.

4. If we look at the time interval in which the packets are coming, it is suspicious since the time gap is very low. That is an extremely large number of requests occurring in a brief interval of time.



5. I can also see an increase in "**TCP Spurious Transmission**". The receiver is receiving a retransmitted segment even before the ACK packet is sent. This can be an indicator of a synflood attack. The below screenshot also highlights "**TCP dup ACK**" which shows the arrival of multiple ACK packets. This is usually due to network congestion, or packet loss (another indicator of a SYN Flood Attack)



Conclusion

In conclusion, there's a clear sign of a SYN Flood Attack. To summarize:

- Sudden Increase in SYN Packets.
- Incomplete 3-way Handshakes.
- TCP Spurious Transmissions and TCP dup ACK.
- Large number of similar traffic within a small time frame.

After critically investigating the pcap file it is clear that there has been a SYN Flood Attack (Discussed above) since there are several indicators. Noticing the severity and unambiguousness of this event, it will be reassigned to the **Incident Response Team Manager**. The incident was analyzed but needs further investigation and clarification to point out the severity and if any denial of service happened. A follow-up report will be generated for this event.

Recommendation

This type of attack overwhelms the network by sending tons of connection requests, potentially leading to disruption of service. While our team is actively monitoring the situation, I recommend some countermeasures.

- Firewalls: Configure your firewall to detect and block malicious SYN flood traffic. This may involve setting up rules to block traffic from specific source IP addresses or implementing heuristics to identify abnormal traffic patterns.
- TCP Timeout Adjustment: Adjust the TCP timeout values on your server. By tweaking these values, you can potentially reduce the impact of SYN flood attacks by releasing half-open connections more quickly.
- Syn Cookies: Enable SYN cookies on our server. SYN cookies are a technique that allows the server to validate connection requests without maintaining a full connection state until the three-way handshake is complete. This can help mitigate the impact of a SYN flood attack.
- Improve Network Monitoring: Strengthen our network monitoring capabilities to promptly detect and respond to any unusual patterns or irregularities in our traffic.
- Review Security Policies: Regularly review and update our security policies to ensure they align with the latest best practices and are effective against evolving threats.
- Utilize load balancers to distribute incoming traffic across multiple servers. This can help distribute the impact of a SYN flood attack, making it more difficult for the attacker to overwhelm a single server.