Western Governor's University

**Emerging Technologies in Cybersecurity**

**C844**

**Kern Grant**

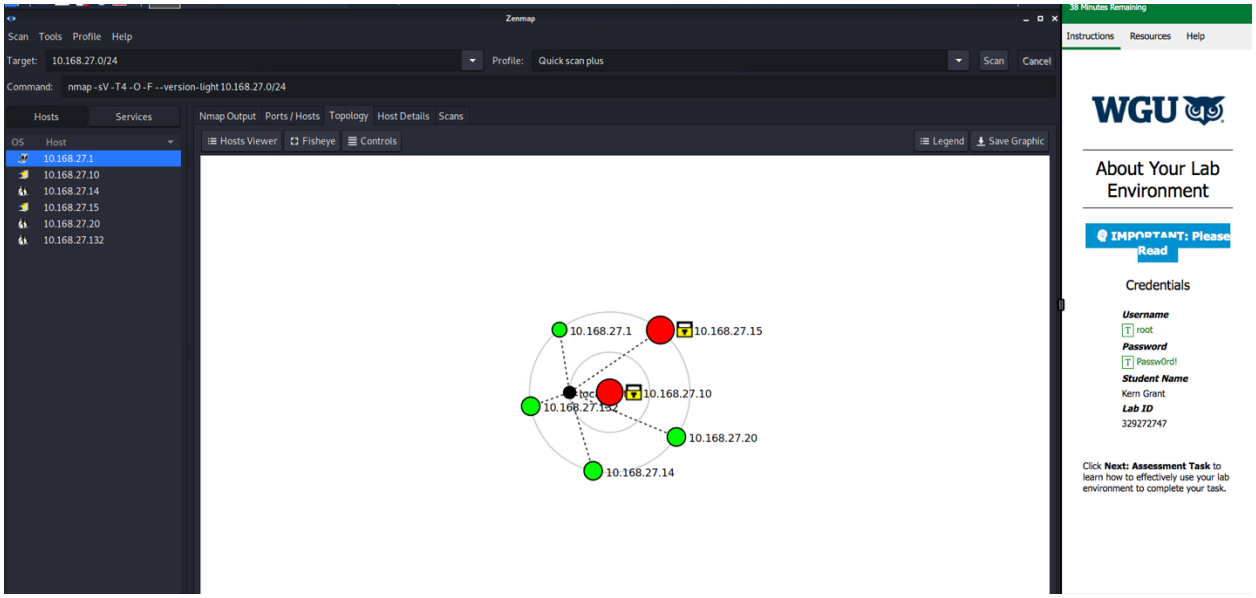
**Task 1: Mapping and Monitoring**

# C844 Task 1: Mapping & Monitoring

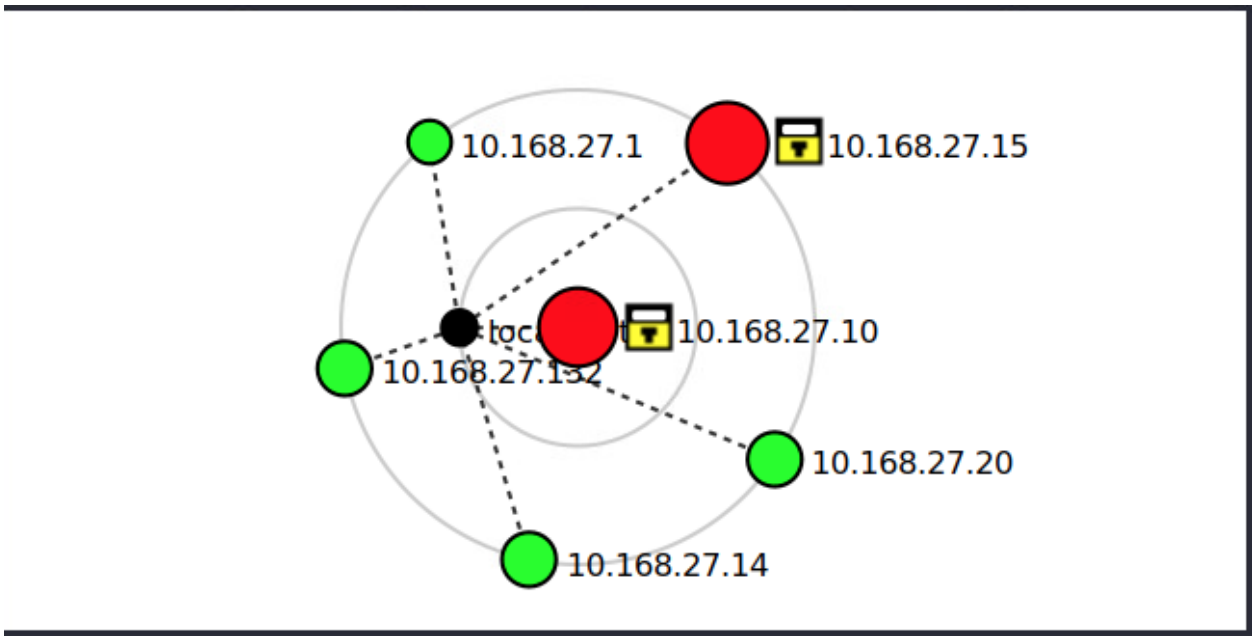
## A. Nmap Topology

My Zen map tool network scan identifies six hosts on a star topology network (Figures 1 & 2). Each host is listed below with their IP address, Operating system, and open port amounts.

1. 10.168.27.1 / OS N/A / zero open ports
2. 10.168.27.10 / OS – Microsoft Windows Server 2021 or Windows Server 2021 R2 / 8 open ports
3. 10.168.27.14 / OS- Linux 2.6.32 / 1 open port
4. 10.168.27.15 / OS – Microsoft Windows Server 2008 R2 or Windows 8.1 / 10 open ports.
5. 10.168.27.20 / OS – Linux 2.6.32 / 1 open port
6. 10.168.27.132 / OS - Linux 2.6.32 / 1 open port




(Figure 1)



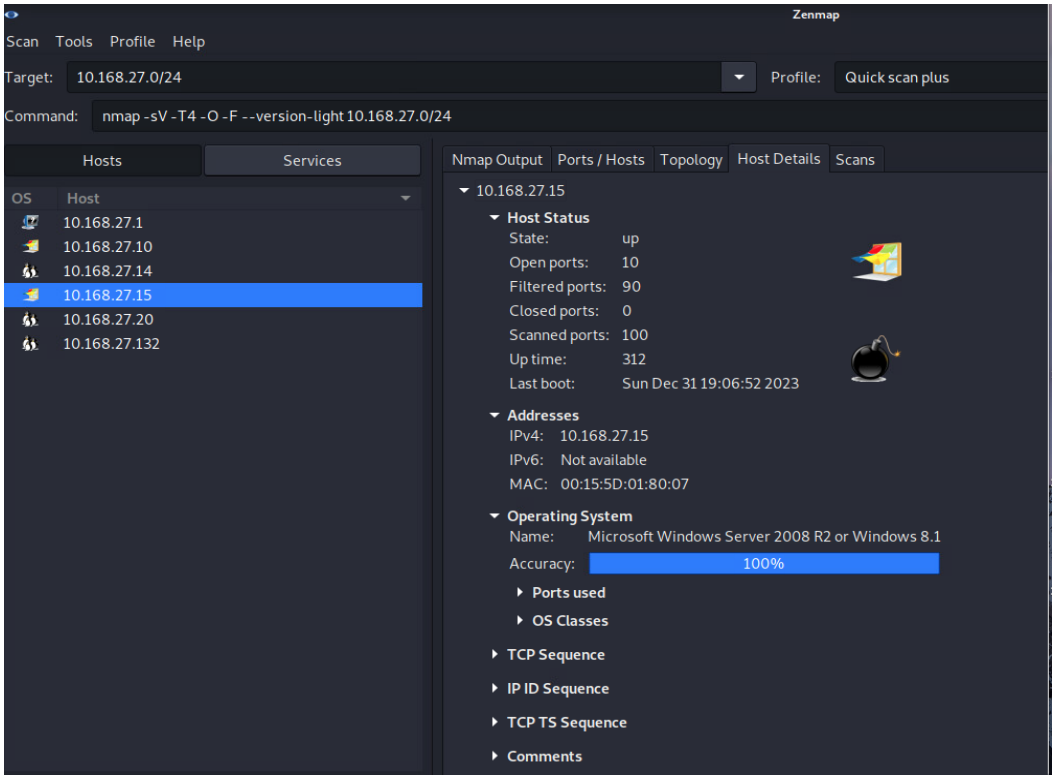

(Figure 2)

# B. Three Vulnerabilities and Implications

Using Zenmap I identified three network vulnerabilities. They are End of life(EOL), Unsecured services with open ports, and a Linux operating system(OS) version with stack-based buffer overflow vulnerability.

## First Zenmap Vulnerability

The first vulnerability is the operating system's end-of-life (EOL)(Figure 3). The host 10.168.27.15 runs either Microsoft Windows Server 2008 R2 or Windows 8.1. Both OS have reached their EOL. It means Microsoft no longer provides customer support or software updates to patch bugs. Therefore, both OS will be highly vulnerable to malware and data breaches from bad actors.




**(Figure 3)**

## Second vulnerability

The second vulnerability I viewed on Zenmap is unsecured services with open ports. These open ports are a pathway for incoming traffic and a gateway to the company network. These services are not encrypted, so data is transferred in plain text. A bad actor can use a network scanning tool like Wireshark to access and leverage sensitive data for various cyber attacks. The unsecured services on the network includes:

1. **LDAP (Lightweight Directory Access Protocol)**

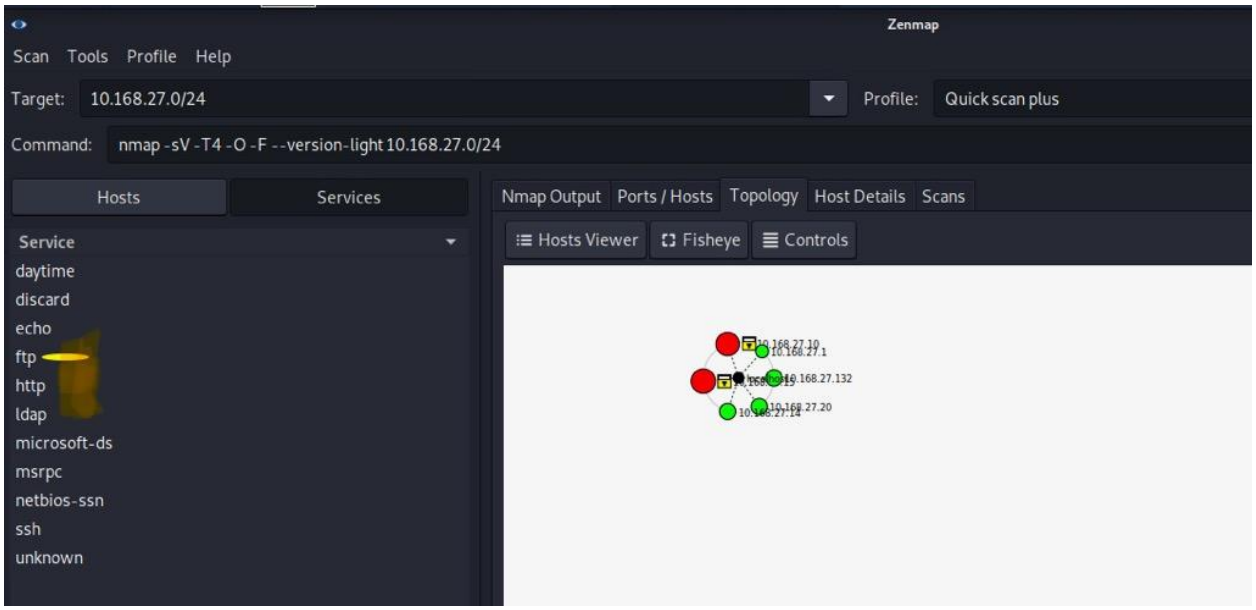
- An unsecured software protocol clients use to access an organization's data.
- It runs on port 389
- Host using the service - 10.168.27.10

2. **HTTP ( Hyper Text Transfer Protocol**

- HTTP is an unsecured protocol for transferring data across networks, such as the Internet.
- Runs on port 80
- Host using the service - 10.168.27.15

3. **FTP (File Transfer Protocol)**

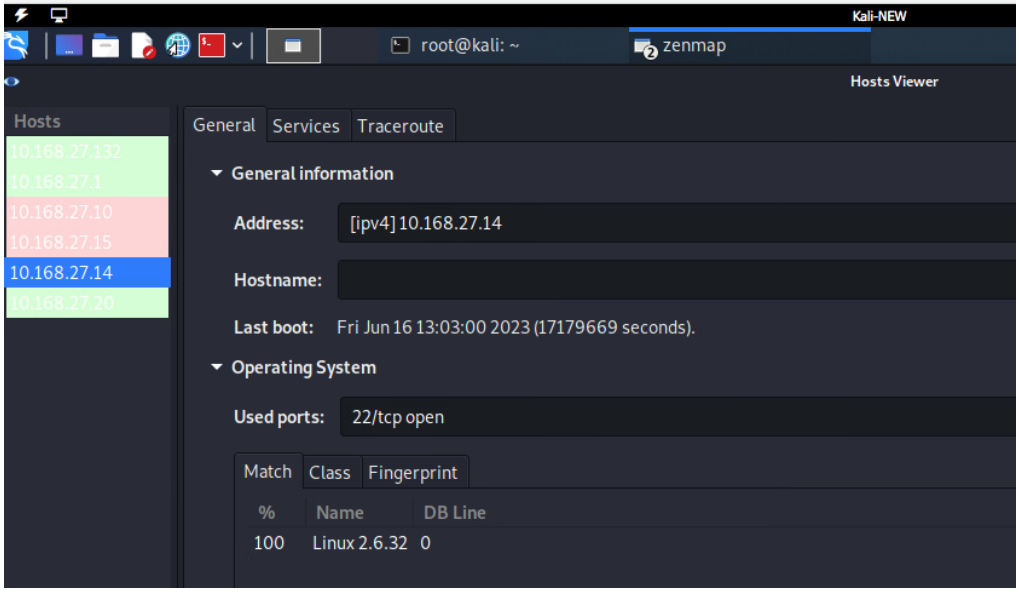
- An unsecured client-server data channel for sharing files.
- Runs on port 21
- Host using the service - 10.168.27.15



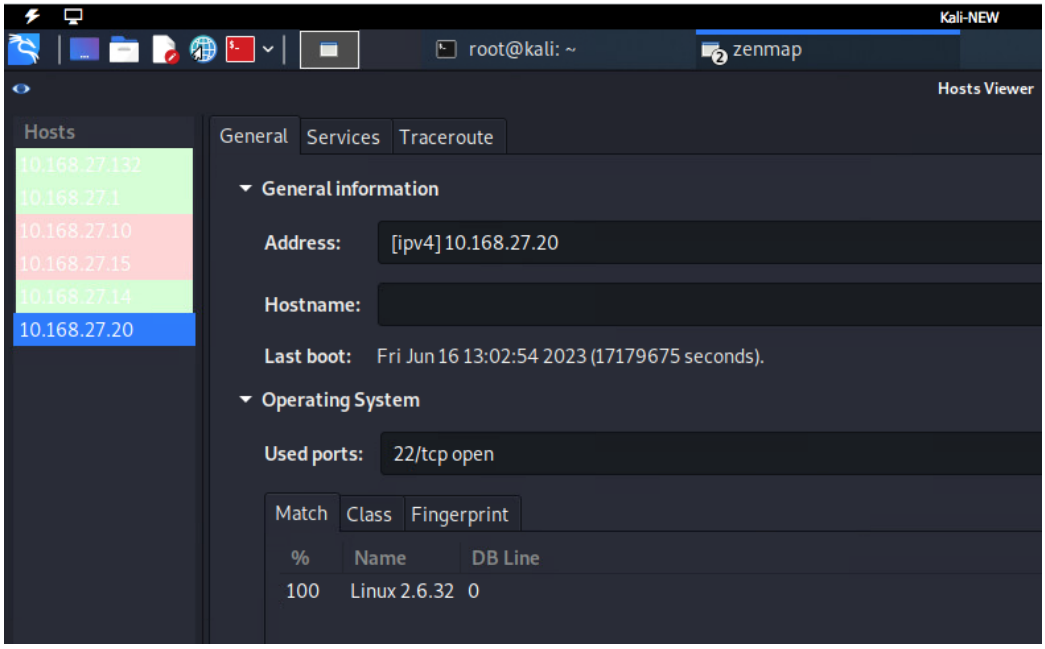
(Figure 4)

## Third vulnerability

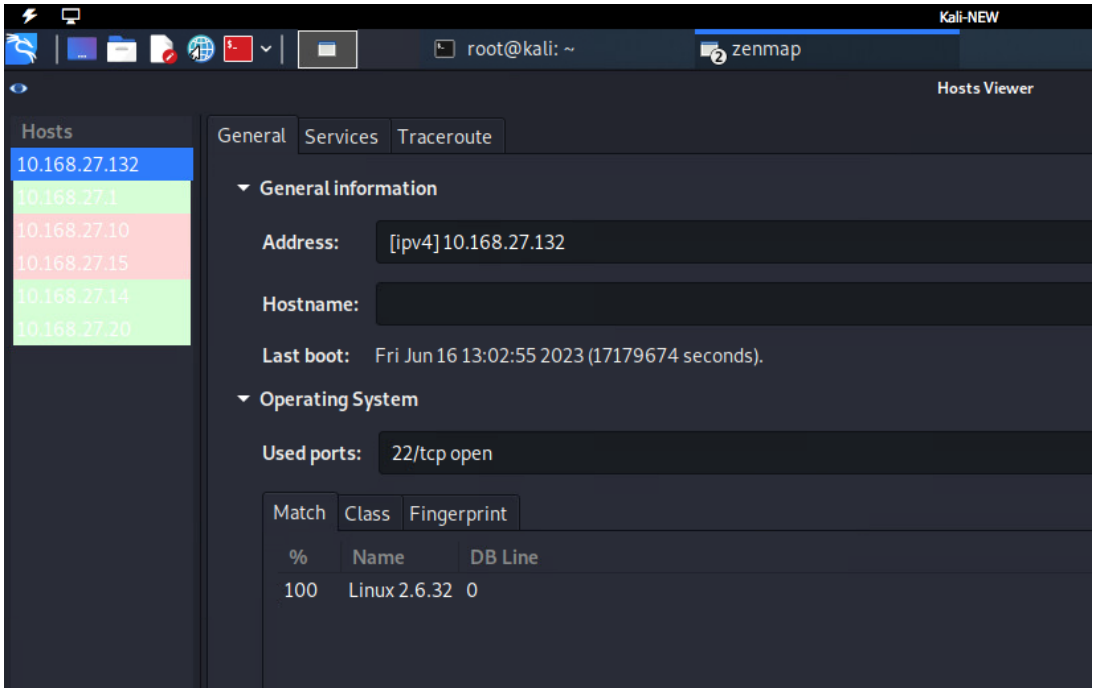
The third vulnerability Is the Linux 2.6.32 OS used by hosts (10.168.27.14, 10.168.27.20, 10.168.27.132) (figure 5-7). The Linux version has a stack-based buffer overflow vulnerability. A bad actor can cause a Denial Of Service(DOS) attack or system crash (Nist, 2023 as cited CVE-2019-14897, para 2).




**(Figure 5)**




**(Figure 6)**


Kali-NEW
root@kali: ~
zenmap
Hosts Viewer
Hosts
10.168.27.132
10.168.27.1
10.168.27.10
10.168.27.15
10.168.27.14
10.168.27.20
General
Services
Traceroute
General information
Address:
[ipv4] 10.168.27.132
Hostname:
Last boot:
Fri Jun 16 13:02:55 2023 (17179674 seconds).
Operating System
Used ports:
22/tcp open
Match
Class
Fingerprint
%
Name
DB Line
100
Linux 2.6.32
0

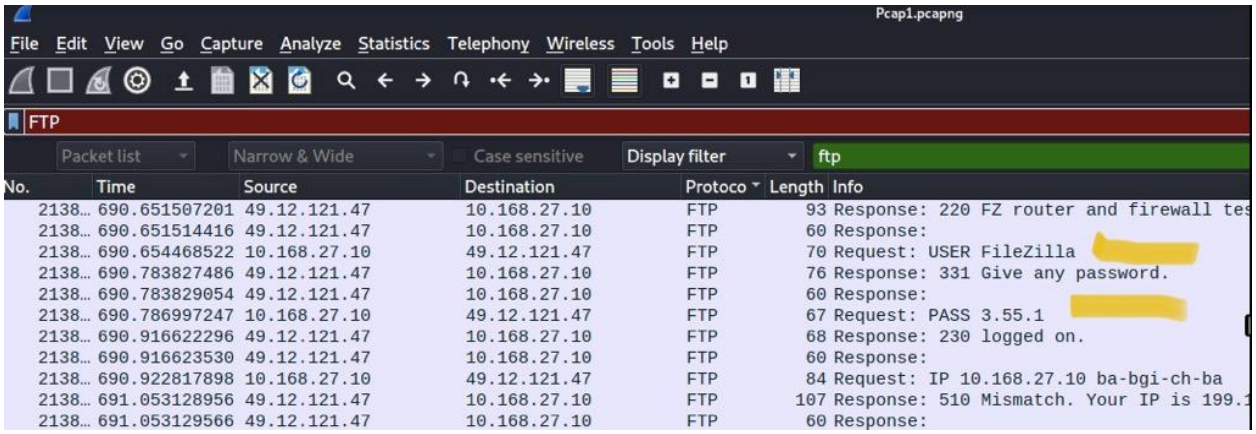
**(Figure 7)**

# C. Network Wireshark Anomalies

I identified three network anomalies on Wireshark. They are File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP), and TCP reset, acknowledgment (RST, ACK).

## First Wireshark Anomaly

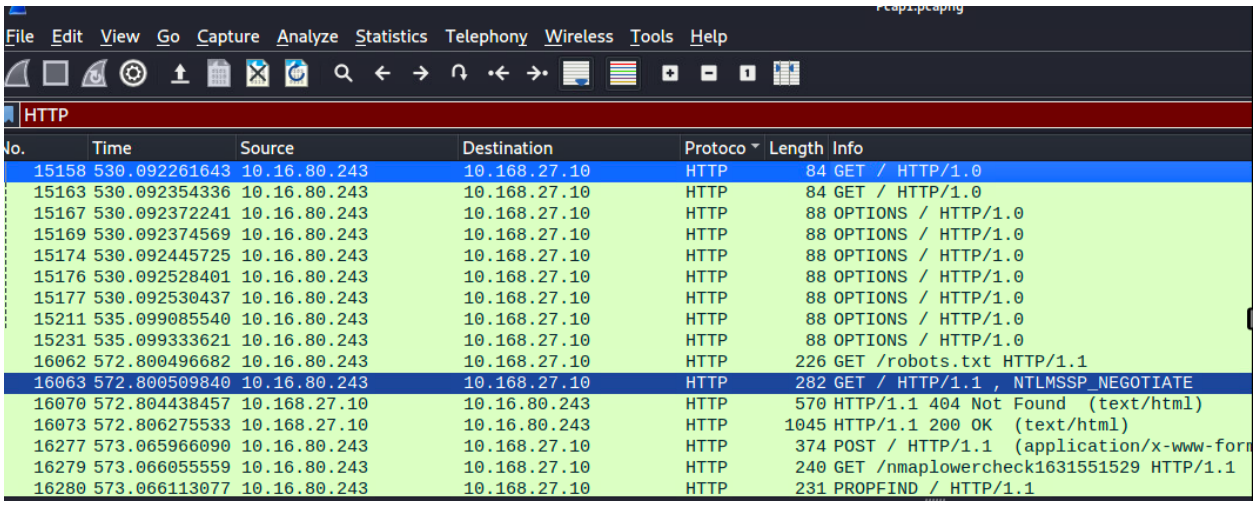
The first network anomaly I recognized using Wireshark is the File Transfer Protocol (FTP) service. FTP is an unsecured client-server data channel for transferring files. The source IP address 10.168.27.10 to destination IP 49.12.121.47 FTP info column displayed the username (FileZilla) and password (3.55.1) login credentials (figure 8).




**(Figure 8)**

## Second Wireshark Anomaly

The second network anomaly is the Hypertext Transfer Protocol (HTTP) used for network communication(figure 9). HTTP is an unsecured protocol used for transferring data across networks.  Therefore, hosts on the network is sending data over the internet browser without authentication or encryption. According to Cloudflare, ".... modern Internet authentication is essential (How does HTTPS help authenticate web servers ? – Section, para 1)." Yet HTTP does not provide that security feature.



File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

HTTP

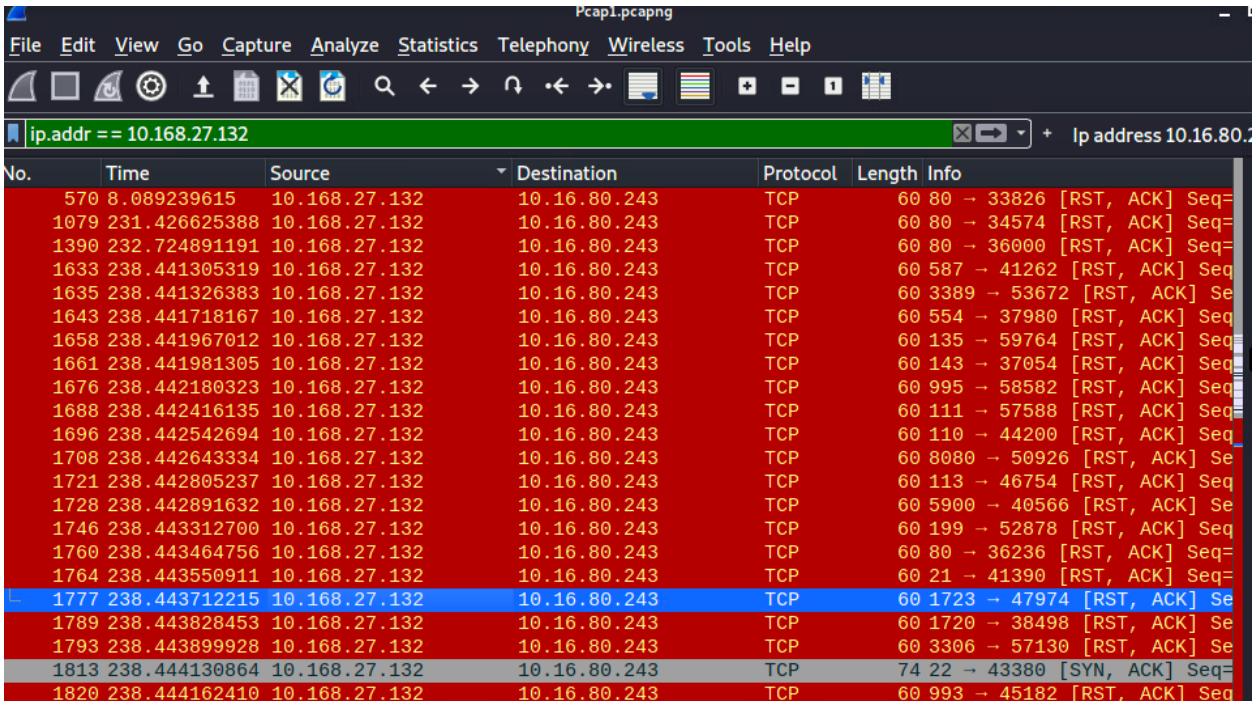
| No. | Time | Source | Destination | Protoco | Length | Info |
|---|---|---|---|---|---|---|
| 15158 | 530.092261643 | 10.16.80.243 | 10.168.27.10 | HTTP | 84 | GET / HTTP/1.0 |
| 15163 | 530.092354336 | 10.16.80.243 | 10.168.27.10 | HTTP | 84 | GET / HTTP/1.0 |
| 15167 | 530.092372241 | 10.16.80.243 | 10.168.27.10 | HTTP | 88 | OPTIONS / HTTP/1.0 |
| 15169 | 530.092374569 | 10.16.80.243 | 10.168.27.10 | HTTP | 88 | OPTIONS / HTTP/1.0 |
| 15174 | 530.092445725 | 10.16.80.243 | 10.168.27.10 | HTTP | 88 | OPTIONS / HTTP/1.0 |
| 15176 | 530.092528401 | 10.16.80.243 | 10.168.27.10 | HTTP | 88 | OPTIONS / HTTP/1.0 |
| 15177 | 530.092530437 | 10.16.80.243 | 10.168.27.10 | HTTP | 88 | OPTIONS / HTTP/1.0 |
| 15211 | 535.099085540 | 10.16.80.243 | 10.168.27.10 | HTTP | 88 | OPTIONS / HTTP/1.0 |
| 15231 | 535.099333621 | 10.16.80.243 | 10.168.27.10 | HTTP | 88 | OPTIONS / HTTP/1.0 |
| 16062 | 572.800496682 | 10.16.80.243 | 10.168.27.10 | HTTP | 226 | GET /robots.txt HTTP/1.1 |
| 16063 | 572.800509840 | 10.16.80.243 | 10.168.27.10 | HTTP | 282 | GET / HTTP/1.1 , NTLMSSP_NEGOTIATE |
| 16070 | 572.804438457 | 10.168.27.10 | 10.16.80.243 | HTTP | 570 | HTTP/1.1 404 Not Found  (text/html) |
| 16073 | 572.806275533 | 10.168.27.10 | 10.16.80.243 | HTTP | 1045 | HTTP/1.1 200 OK  (text/html) |
| 16277 | 573.065966090 | 10.16.80.243 | 10.168.27.10 | HTTP | 374 | POST / HTTP/1.1  (application/x-www-form |
| 16279 | 573.066055559 | 10.16.80.243 | 10.168.27.10 | HTTP | 240 | GET /nmaplowercheck1631551529 HTTP/1.1 |
| 16280 | 573.066113077 | 10.16.80.243 | 10.168.27.10 | HTTP | 231 | PROPFIND / HTTP/1.1 |

**(Figure 9)**

## Third Anamonly

The third anomaly Is the TCP reset, acknowledgment [RST, ACK] from source IP address 10.168.27.132 to destination address

10.16.80.243(figure 10). It reflects a high traffic volume from alternating/various port services with the same packet size, resulting in [RST, ACK]. It suggests a Reset (RST) attack. According to NordVPN, (n.d.) an RST attack is "A TCP reset attack is a type of denial-of-service attack that aims to terminate an established TCP connection between two parties using fake TCP reset packets" (TCP reset attack, para. 1).




**(Figure 10)**

# D. Implications of each Wireshark Anomaly

## First Anamoly Implication

FTP service is unsecured because it transmits data without encryption. Therefore, FTP is compromising this company network. The lack of encryption implies a bad actor can intercept data. In this case, an attacker using a network sniffing tool like Wireshark can view the login credentials in plain text. That sensitive data is the gateway to the company network. Unauthorized access to sensitive data puts this network data and the company's reputation at risk for cybersecurity attacks. According to IBM (2023), "If you use your system as an FTP server on the internet, it is accessible to the entire world (Securing File Transfer Protocol, para. 2)."

## Second Anamoly Implication

The network Wireshark traffic shows HTTP being used to communicate between networks like the internet. The implications of not taking action to resolve the HTTP anomaly can cause cyberattacks like on-path or malware. A bad actor can use a network sniffing tool like Wireshark to intercept and leverage data for attacks. Sensitive data can be eavesdropped, compromised, or leaked. An unsecure network is vulnerable and can damage a business's reputation.

## Third Anamoly Implications

The implications of an RST attack can deny port services to the network. According to Myers, R (n.d) attackers can, "… continually send TCP RST packets to a target IP and port number which will effectively prevent any communication on that port (P. 6)." My findings from the network Wireshark results imply such actions. Therefore, a bad actor can cause a Denial Of Service(DOS) attack to make authorized users unable to access network resources.

# C. Recommended Solution

## First Vulnerability Solution

I recommend host 10.168.27.15 update its OS to a recent version with customer support and security coverages. Microsoft (2020) states," Once a product reaches the end of support, or a service retires, there will be no new security updates, non-security updates, or assisted support. Customers are encouraged to migrate to the latest version of the product or service. (Overview - Product End of Support & Retirements, para. 2)". Therefore the features offered with OS updates make the network less vulnerable.

## Second Vulnerability Solution

My second vulnerability solution is adding Transfer Layer Security (TSL) / Secure Socket Layer (SSL) as encryption and authentication protocols to all network services. According to Amazon (n.d), " Both SSL and TLS are communication protocols that encrypt data between servers, applications, users and systems. They authenticate two parties connected over a network so they can exchange data securely(What are the Similarities between SSL and TLS? section, para 1)".

Therefore, the services and their port number will be updated as follows:

1. **LDAPS (Lightweight Directory Access Protocol Secure)**

- A secure software protocol clients use to access an organization's data.
- According to IBM (2023), "Secure LDAP protocol (LDAPS) encrypts the communication between the Access Manager component of Content Manager and the directory server. LDAPS prevents sensitive information in the directory server and the LDAP credentials from being sent as clear text (Enabling secure communication to LDAP Server, para 1)."
- It runs on port 636

2. **HTTPS ( Hyper Text Transfer Protocol Secure )**

- HTTPS is a secured protocol for transferring data across networks like the Internet.
- According to Cloudflare (n.d) , "HTTP requests and responses are sent in plaintext, which means that anyone can read them. HTTPS corrects this problem by using TLS/SSL encryption (Why is HTTP not secure? | HTTP vs. HTTPS, para 1)." -
- Runs on port 443

3. **FTPS (File Transfer Protocol Secure)**

- It is a secure protocol for sharing files between a client-server data channel.
- According to IBM (2023), " You can use Transport Layer Security (TLS) or Secure Sockets Layer (SSL) connections to encrypt data transferred over File Transfer Protocol (FTP) control and data connections (Securing File Transfer Protocol, para. 6)."
- It runs on port 990

## Third Vulnerability Solution

My third vulnerability solution for stack-based buffer attacks is to deploy preventative measures to mitigate and harden the network security. Therefore decreasing the likelihood of bad actors successfully performing such attacks. I would implement preventive measures, using operating system runtime protection (Address Space Layout Randomization and Data Execution Prevention), keeping devices patched, and using the principle of least prevailed (Cobb, 2021, How to prevent buffer overflow attacks - section, para 1). For this scenario, the most important measure is to update Linux to the current version 6.6.8. The latest Linux version may have patches to fix stack-based buffer attack vulnerabilities,

## First Anomaly Solution

The solution for this network anomaly is adding secure protocols socket layer (SSL) / Transport Layer Security (TLS) (SSL) to FTP. In addition to the secure protocol, FTP service becomes FTPS. The S at the end of FTPS represents secure. According to IBM (2023), "With Secure Socket Layer (SSL), you can eliminate the exposure of transmitting passwords and data in the clear when using the File Transfer Protocol (FTP) server with an FTP client that also uses SSL (Securing File Transfer Protocol, para. 5)."

## Second Anomaly Solution

According to CloudFlare (n.d), "If a website uses HTTP instead of HTTPS, all requests and responses can be read by anyone who is monitoring the session (what does a typical HTTP request look like? - section, para 5)." HTTPS uses TLS/SSL protocols over the internet browser network to encrypt plain text and authenticate the user and machine. Therefore, my solution for this Network is using HTTPS to secure their network communication.

## Third Anomaly Solution

To provide a solution to help mitigate RST attacks, I suggest NordVPN, (n.d.) recommendation:

- **Use firewalls and intrusion detection systems** to identify and filter out suspicious network traffic, including forged reset packets.
- **Monitor network traffic for unusual patterns,** such as a sudden surge in reset packets or unexpected terminations of established connections.
- **Implement secure communication protocols(**such as Transport Layer Security) to encrypt TCP connections, making it harder for attackers to tamper with them. **(para. 3).**

# References


1. United States Government. (2019). *CVE-2019-14897 Detail*. National Vulnerability Database. https://nvd.nist.gov/vuln/detail/CVE-2019-14897

2. CloudFlare. (n.d.). *Why is HTTP not secure? | HTTP vs. HTTPS*. CloudFlare. https://www.cloudflare.com/learning/ssl/why-is-http-not-secure/

3. NordVPN. (n.d.). *TCP Reset Attack*. NordVPN. https://nordvpn.com/cybersecurity/glossary/tcp-reset-attack/#:~:text=A%20TCP%20reset%20attack%20is,using%20fake%20TCP%20reset%20packets.

4. IBM. (2023, October 10). *Securing File Transfer Protocol*. IBM. https://www.ibm.com/docs/en/i/7.5?topic=i-securing-ftp

5. Myers, R. (n.d.). *Attacks on TCP/IP Protocols* . University of Tennessee at Chattanooga. https://www.utc.edu/sites/default/files/2021-04/course-paper-5620-attacktcpip.pdf
6. Microsoft . (2023). *Overview - Product End of Support & Retirements*. Microsoft. https://learn.microsoft.com/en-us/lifecycle/overview/product-end-of-support-overview
7. IBM. (2023b, November 11). *Enabling secure communication to the LDAP server*. IBM.

https://www.ibm.com/docs/en/cognos-analytics/11.1.0?topic=ldap-enabling-secure-communication-server

8. Cobb, M. (2021, July). *Buffer Overflow*. Tech Target. https://www.techtarget.com/searchsecurity/definition/buffer-overflow