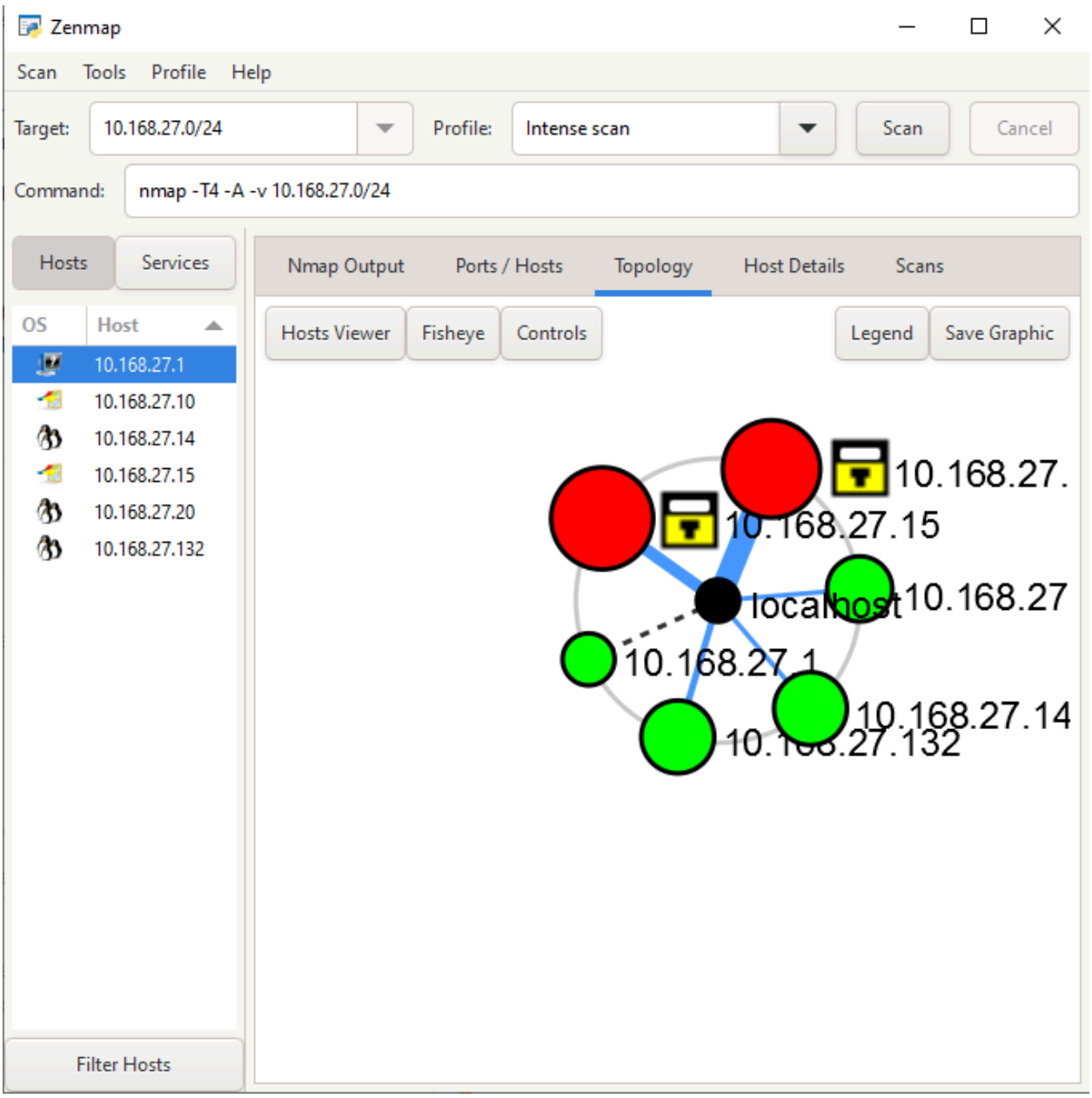
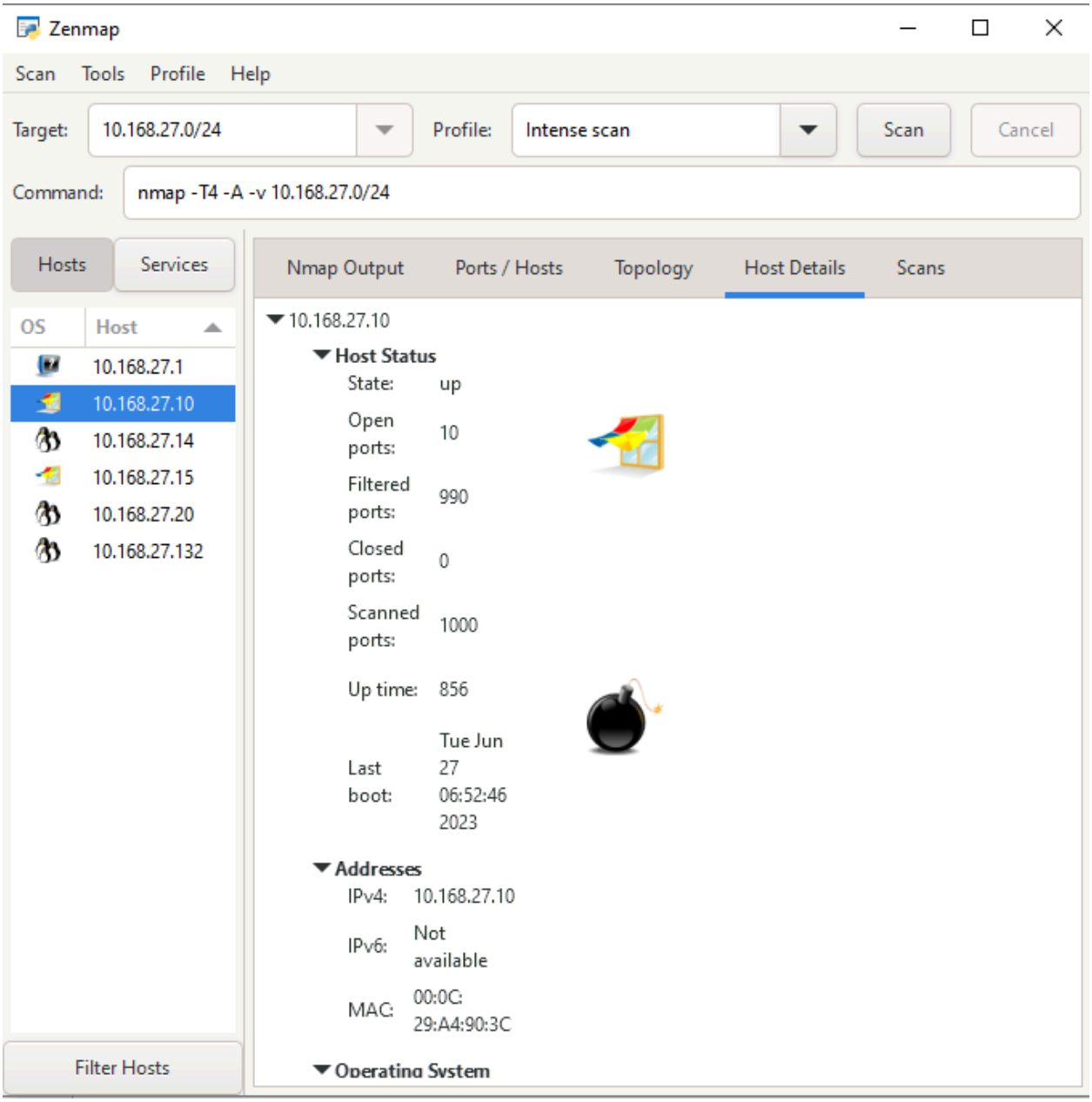
A. Network Topology



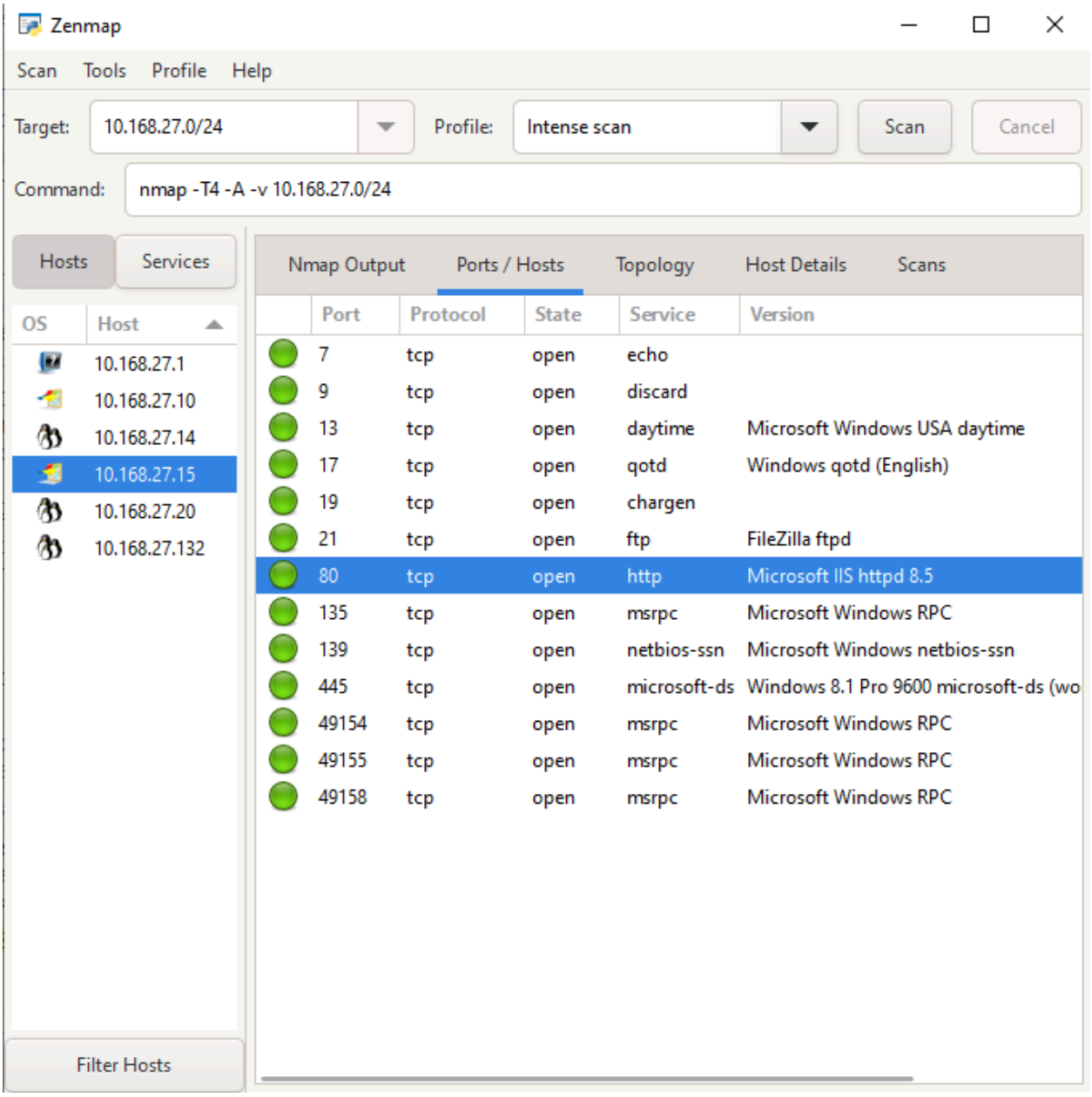
The network uses the STAR topology. There are 6 hosts in total (as seen on the left side): 10.168.27.1 which is running on an unknown operating system, 10.168.127.10 which is running on a Windows OS, 10.168.27.14 which is running on Linux OS, 10.168.27.15 which is also a Windows OS, and the final two 10.168.27.20 and 10.168.27.132 are both Linux OS.

We can dive into more details about the host as needed through the Host Details section and individually click on each host on the left side with Zenmap to further find details such as host uptime, how many ports are open, MAC addresses, and other helpful information.

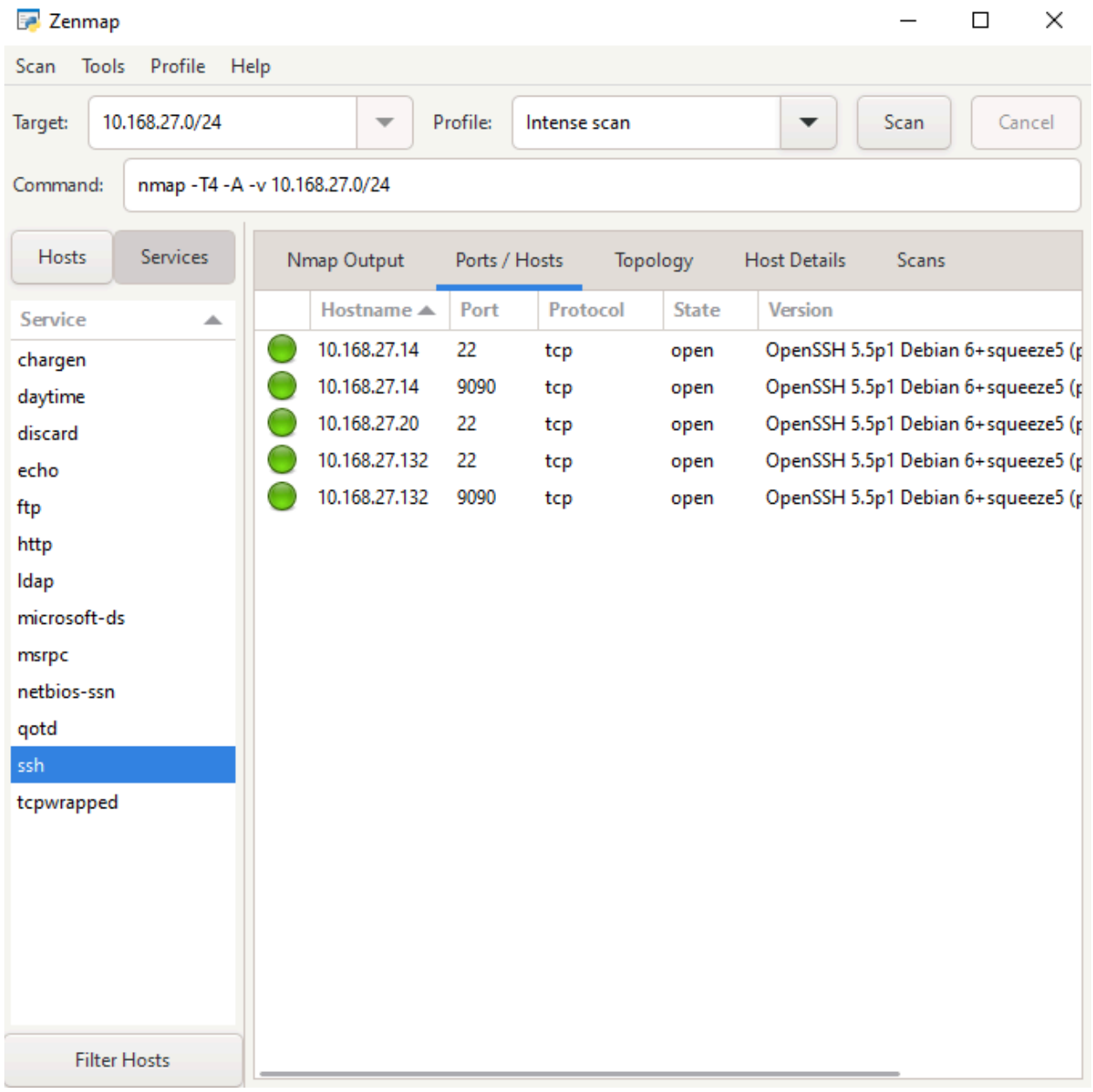


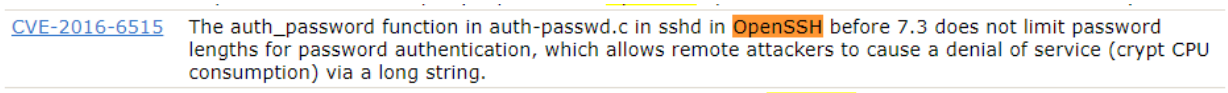
10.168.27.10 has 10 open ports (pictured above), and going through the rest of the hosts we can see that 10.168.27.14 has 2 open ports, host 10.168.27.15 has 13 open ports, 10.168.27.20 has 1 open port, 10.168.27.132 has 2 open ports, and our unknown host 10.168.27.1 has 0 open ports.

B. Vulnerabilities

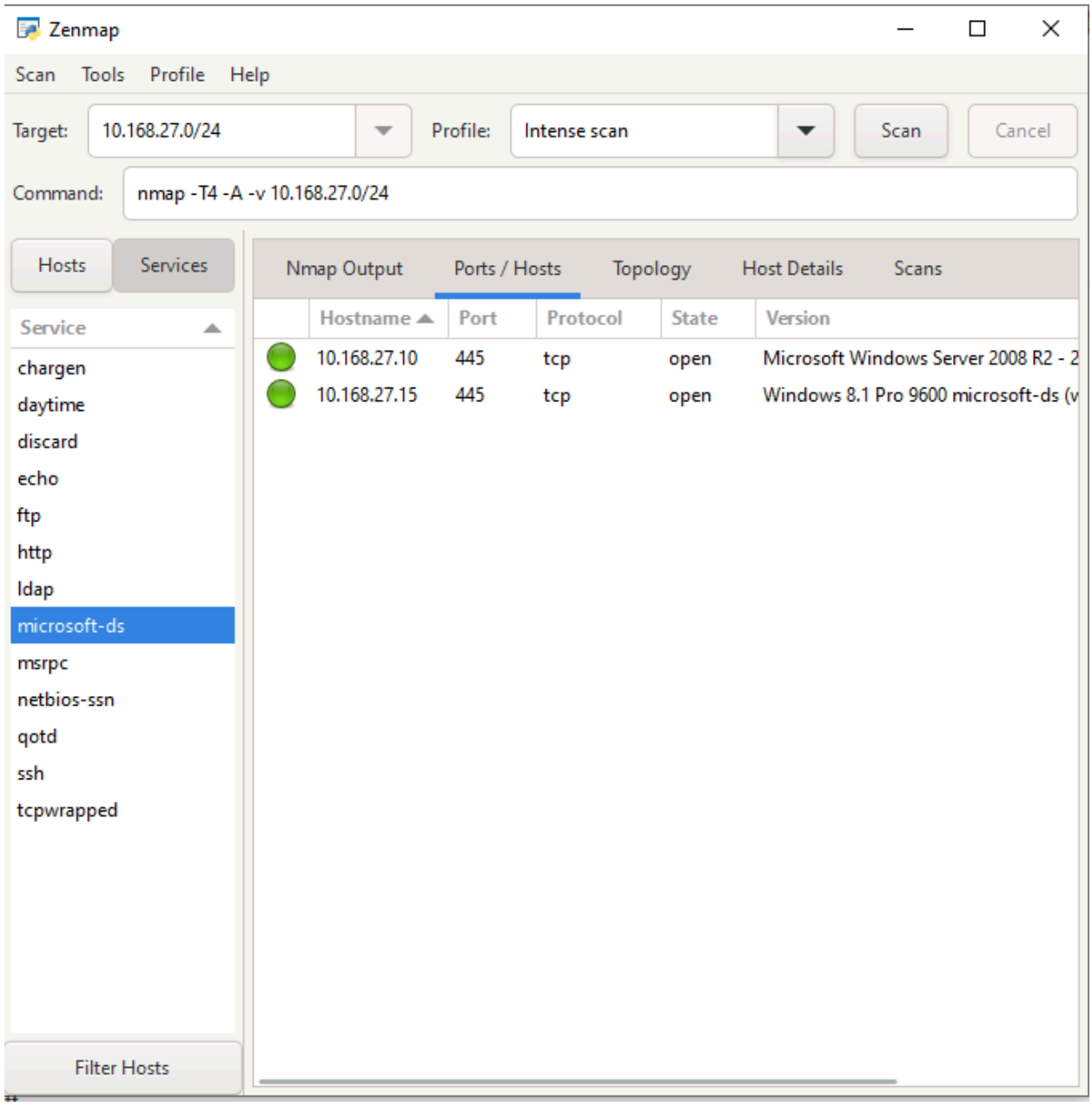


The computer with the IP 10.168.27.15 (pictured above) is using port 80 HTTP which is not a secure protocol as information is not encrypted and allows web traffic to be viewed in the clear. If a hacker is able to perform an on-path attack, this would be a risk to any user using this computer especially if they type in usernames, passwords, credit card information, or any other sensitive data that should be encrypted over the internet.





The version of OpenSSH 5.5p.1 has several vulnerabilities such as causing a DOS through invalid input validation with CVE-2016-6515 which does not limit password lengths (*CVE*). There are many vulnerabilities with OpenSSH before version X.X, it is important to make sure that the services that the computers are using are updated using patch management throughout the network environment.



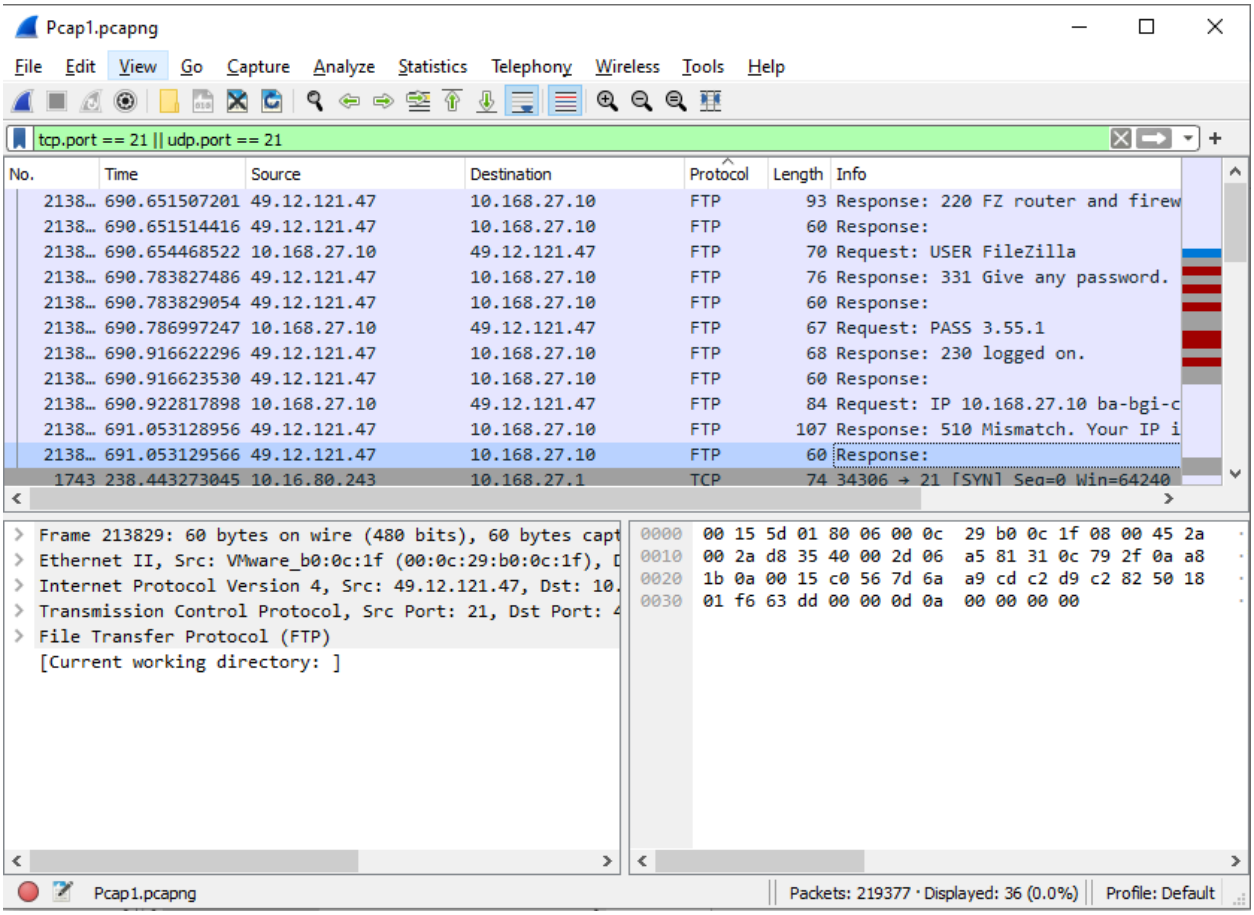
CVE-2019-0543	An elevation of privilege vulnerability exists when Windows improperly handles authentication requests, aka "Microsoft Windows Elevation of Privilege Vulnerability." This affects Windows 7, Windows Server 2012 R2, Windows RT 8.1, Windows Server 2008, Windows Server 2019, Windows Server 2012, Windows 8.1, Windows Server 2016, Windows Server 2008 R2, Windows 10, Windows 10 Servers.

The Microsoft website mentions that these versions of OS running on both Windows computers are at the end-of-life. 8.1 reached end-of-life a few months ago in January. Microsoft Windows Server 2008 R2 reached end-of-life in 2020. There is a risk in using end-of-life products that are no longer getting vendor support. These systems are vulnerable to such things as privilege elevation attacks, CVE-2019-0543 (CVE). As part of its patch management, this company should ensure that they are using products that are still receiving vendor support and upgrade the operating systems on their Windows computers. The NVD mentions that the best way to

prevent this specific CVE is to apply patches from the vendor which on the Microsoft Windows Server 2008 R2 is not possible since the patch was not released until 2022 (2 years after support ended for this specific Window's OS) (NVD).

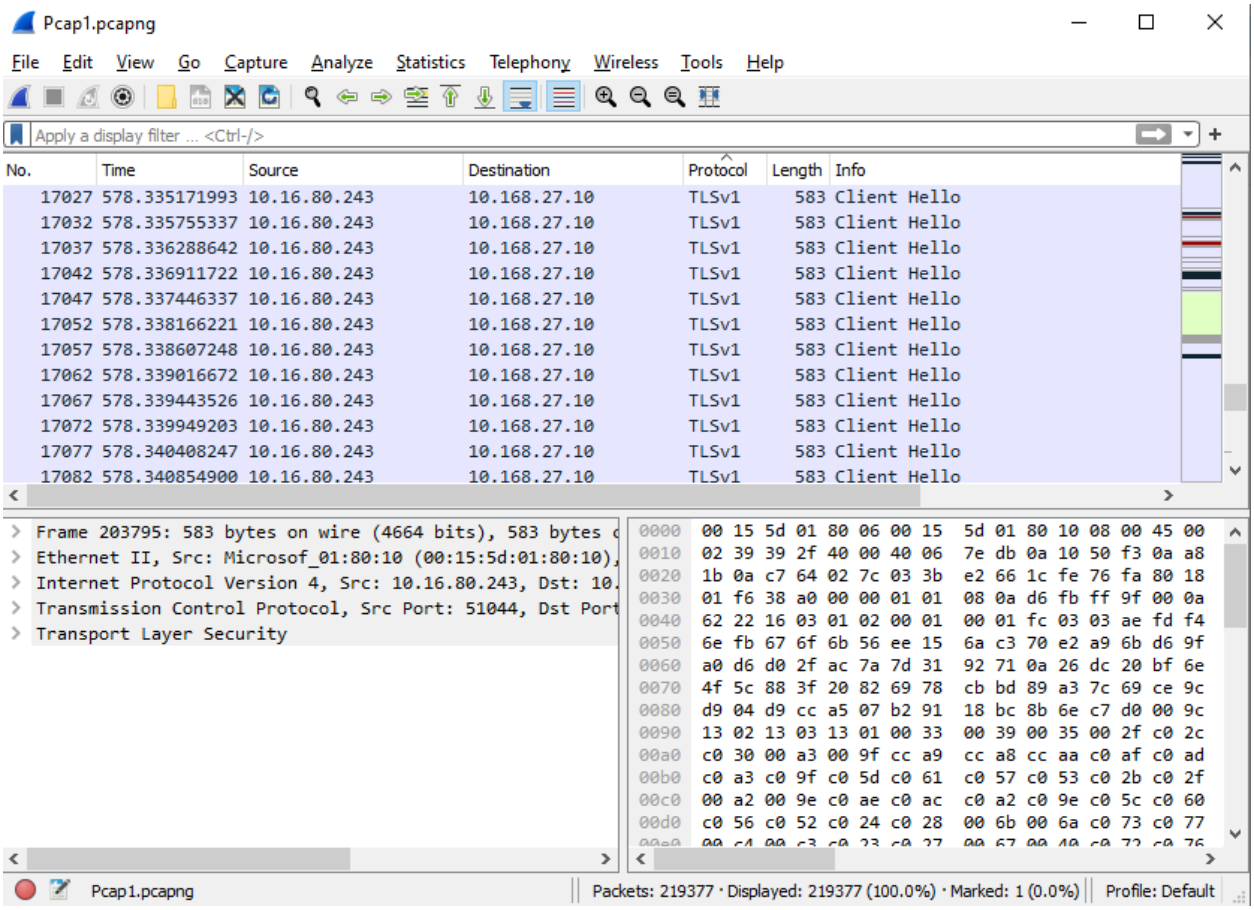
C. Wireshark Anomalies (using Pcap1)

FTP -



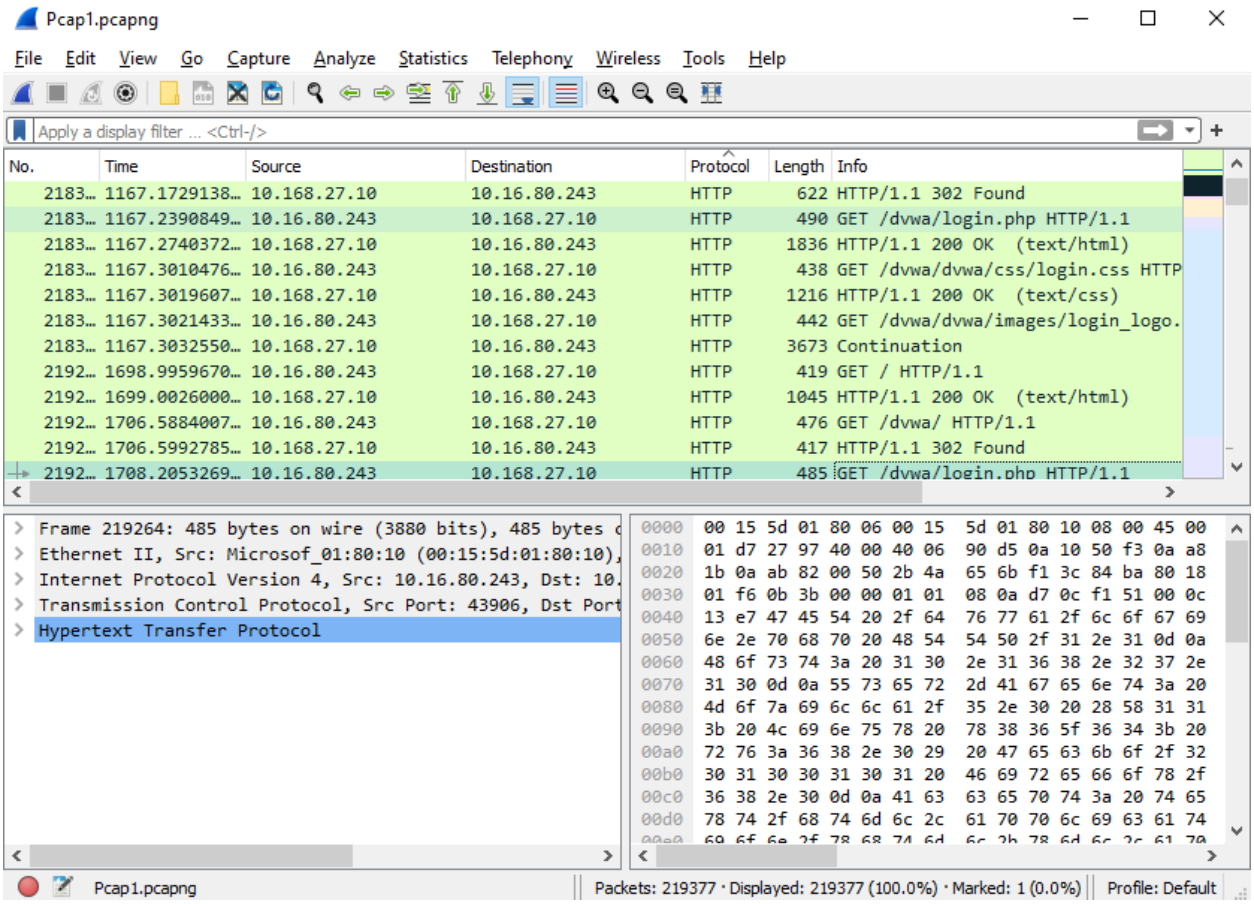
In this packet capture from Wireshark, there is a host that is not part of the network topology from our viewing of Nmap/Zenmap. 49.12.121.47 seems to potentially be spoofing itself as 10.168.27.10 to create an account and password and logging into the FTP server.

TLSv1 -



Using TLSv1 is an out-of-date TLS encryption version as there is TLS1.1, 1.2, and 1.3 versions. Using this out-of-date protocol for encryption over the internet can cause the computer to be at risk.

HTTP -



It seems that HTTP is being used to log in. HTTP is sent in-the-clear, so if a on-path attack happens all of the user's credentials (username and password) would be seen by the attacker which would allow them to perform several attacks.

D. Implications of not addressing the anomalies

FTP -

FTP is an unsecure protocol that sends information in-the-clear. This allows an attacker that is able to capture the network traffic access to view any file being transferred, or username and passwords being sent. Using some form of secure FTP protocol such as SFTP (SSH FTP) or FTPS (FTP Secure) would encrypt the traffic being sent and disallow an attack from viewing anything sent over the FTP's traffic.

TLSv1 -

CVE-2014-3566	The SSL protocol 3.0, as used in OpenSSL through 1.0.1i and other products, uses nondeterministic CBC padding, which makes it easier for man-in-the-middle attackers to obtain cleartext data via a padding-oracle attack, aka the "POODLE" issue.

TLSv1 is able to fall back to SSL 3.0 which has vulnerabilities due to a POODLE downgrade attack (Bhattacharya). The attacker can use an on-path attack and obtain cleartext by downgrading the TLS encryption to use the SSL 3.0 protocol, CVE-2014-3566 (CVE).

HTTP -
HTTP is an insecure protocol that sends information in-the-clear. This means that if you are on a website that requires you to use your credit card or log in with a username and password, this information is being sent as plaintext and can be read by an attacker that is able to capture your network traffic that is sent over HTTP.

E. Recommend solutions for eliminating or minimizing vulnerabilities or anomalies from Wireshark and Nmap.

FTP -
Using a secure method of transferring files by using SSH with FTP (SFTP) via port 22 instead of port 21. For SFTP the user would need proper authentication in order to connect and use the FTP all while encrypting the information exchanged between the two systems. The user connecting would need the correct info to connect such as a username, password, and SSH key which would ensure that unauthorized users are unable to connect to the FTP server. (Cyber Defense Magazine)

TLSv1 -
TLSv1 is vulnerable to a POODLE downgrade attack that makes TLSv1 fall back to SSL 3.0 which opens up vulnerabilities on the outdated SSL encryption. There are newer versions 1.2 and 1.3 of TLS that are not vulnerable to this POODLE attack so simply upgrading the TLS version would remove this specific vulnerability and greatly improve the strength of the encryption used over the web. (Bhattacharya)

HTTP -
HTTP sends traffic in-the-clear and makes a user vulnerable to several attacks such as on-path attacks, or allowing the attack to redirect your traffic to malicious websites. HTTPS securely sends HTTP traffic over the web with encryption which prevents these types of attacks. HTTPS text would be in Ciphertext (encrypted) instead of using Plaintext (in-the-clear) that HTTP uses which would not allow an attacker to view your web requests. (Cloudflare)

Works Cited

CVE, cve.mitre.org/. Accessed 27 June 2023.

NVD, nvd.nist.gov/. Accessed 27 June 2023.

Bhattacharya, Anish. “Why Older TLS PROTOCOLS ARE UNSAFE FOR YOUR ORGANIZATION?” *Encryption Consulting*, 17 Nov. 2022, www.encryptionconsulting.com/why-should-organizations-avoid-older-tls-protocols#:~:text=TLS%201.0%20and%201.1%20are,a%20server%20for%20MITM%20attacks.

Why Is HTTP Not Secure? | *HTTP vs. HTTPS* | *Cloudflare*, www.cloudflare.com/learning/ssl/why-is-http-not-secure/. Accessed 27 June 2023.

“Forget FTP: 4 Modern Protocols You Should Use Instead.” *Cyber Defense Magazine*, 26 July 2017, www.cyberdefensemagazine.com/forget-ftp-4-modern-protocols-you-should-use-instead/.