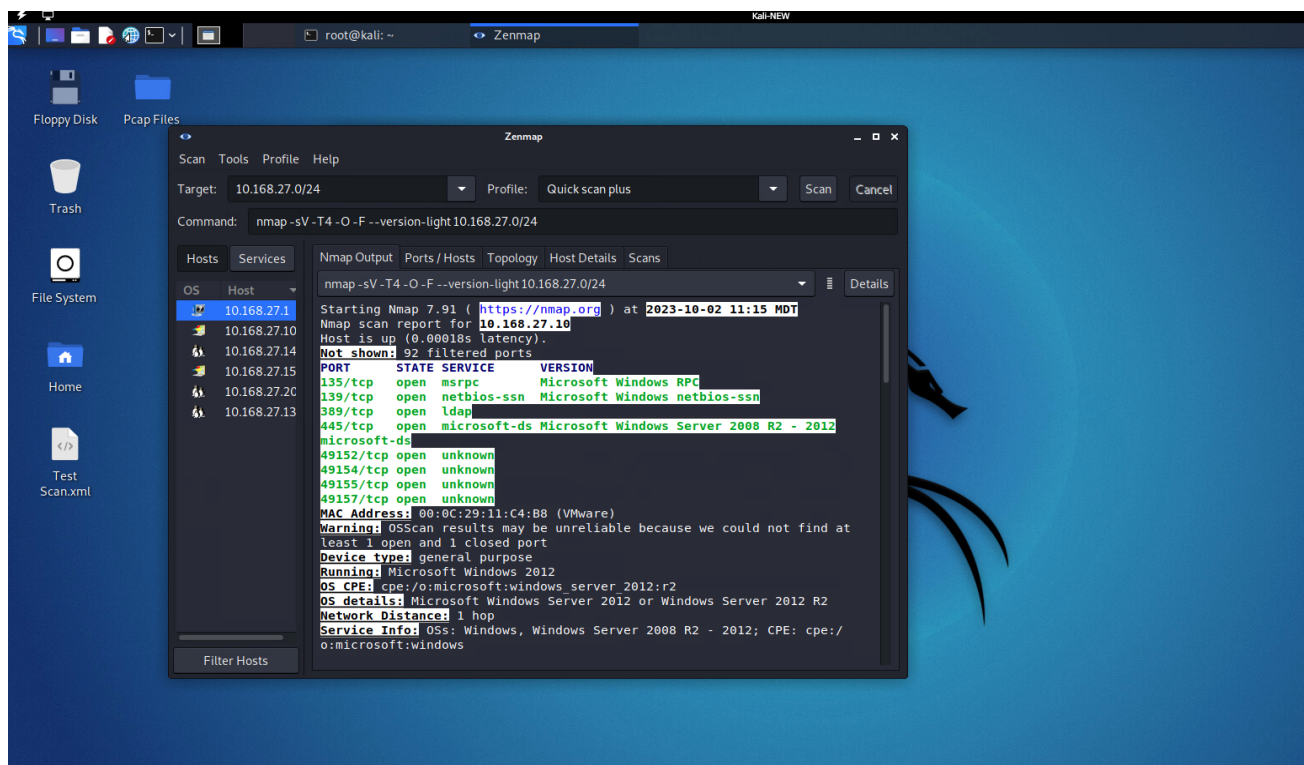
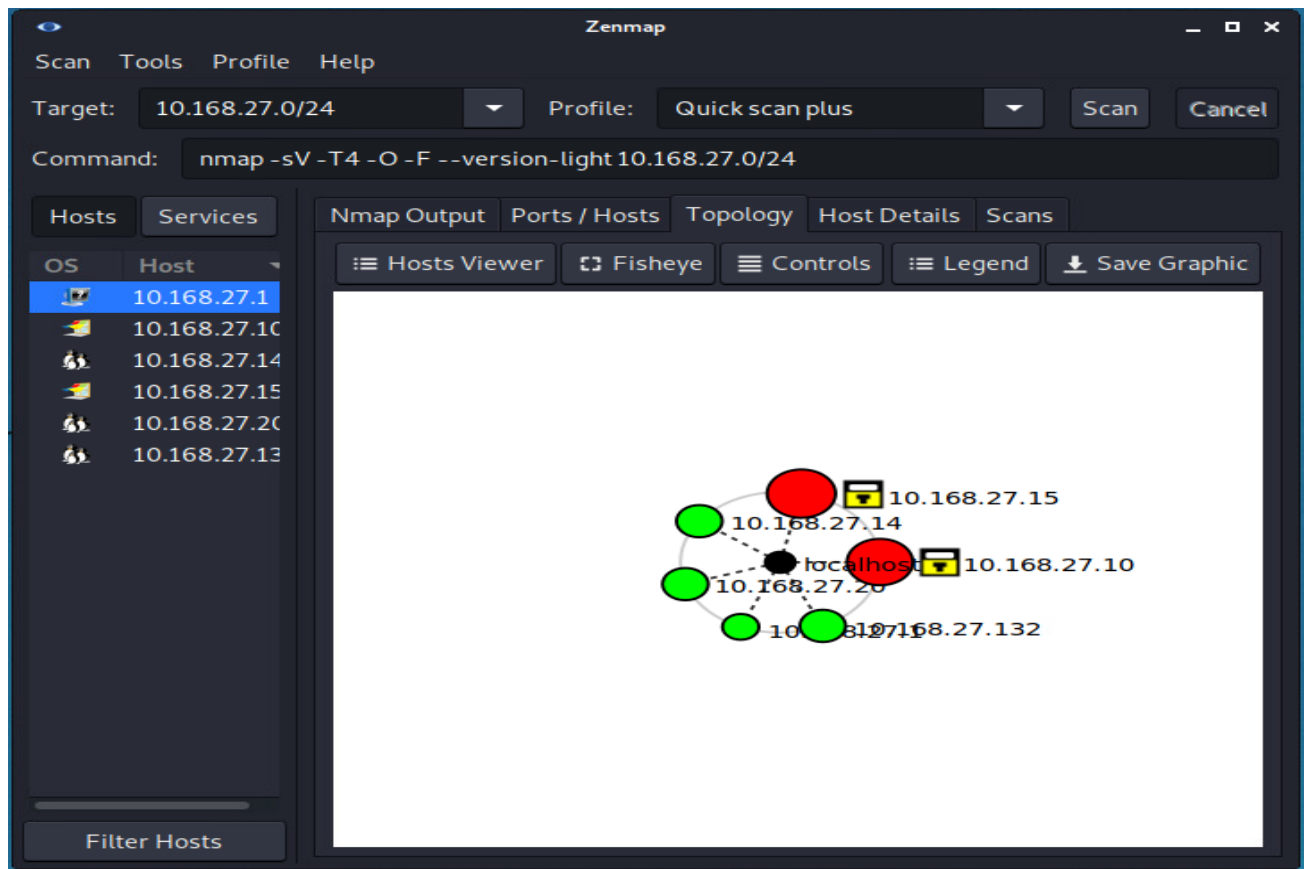


After logging into the lab and running a network scan on the preselected network address (192.168.27.0/24). I discovered a star topology with 6 IP addresses.

- IP address : 10.168.27.20 , has 1 open port, operating system is Linux 2.6.32
- IP address : 10.168.27.14, has 1 open port, operating system is Linux 2.6.32
- IP address :10.168.27.15 , has 10 open ports, operating system is Microsoft windows 7
- IP address :10.168.27.10, has 8 open ports, operating system is Microsoft Windows Server 2012 R2
- IP address :10.168.27.132, has 1 port open, operating system is Linux 2.6.32
- IP address :10.168.27.1, has 0 ports open, operating system is unknown



Host **10.168.27.15** has a major vulnerability and that **vulnerability is Microsoft Windows 7 operating system**. This operating system is EOL(End of Life) and is no longer supported. Using Windows 7 OS can leave you open to numerous attacks, one of these are TCP/IP hijacking attacks [cvedetails.com](https://cvedetails.com),2023). TCP/IP hijack attacks can lead to data loss, data tampering, denial of service etc.. To resolve this issue the only option is to upgrade windows to a later version that is still supported. According to Microsoft “If you continue to use Windows 7 now that support has ended, your PC will still work, but it will be more vulnerable to security risks and viruses (Microsoft, 2020).” This detail explains that regardless if you were to use windows 7, due to the fact that it’s no longer being being support and is EOL, there are no more possible patches or methods that can be used to fix it’s security vulnerabilities. Therefore the only logical solution is to upgrade to a later windows operating system, preferably windows 10 or 11.

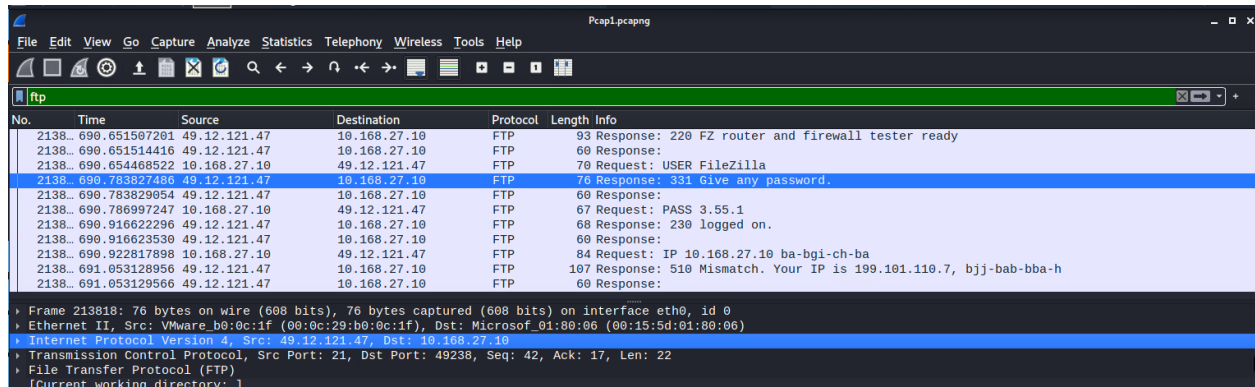
Host **10.168.27.15** has another **vulnerability found that FTP is enabled**, it is an “in the clear” protocol. FTP lacks encryption, uses simple username and password to authenticate, and has no integrity checks. This allows hackers to brute force their way into systems and it allows them to sniff FTP’s plain text transmissions revealing usernames and passwords([cerberusftp.com](https://cerberusftp.com), 2023). This leads to systems/networks being compromised by malicious actors, confidential

data being leaked, data integrity can be tampered with etc.. To remediate this vulnerability, according to [ssh.com](https://ssh.com),” SFTP (SSH File Transfer Protocol) is a secure file transfer protocol. It runs over the **SSH protocol**. It supports the full security and authentication functionality of SSH.([ssh.com](https://ssh.com),2023)” This detail shows that as a secure alternate method you can implement the ssh protocol with ftp, this method encrypts all plaintext that is being transmitted if any data, if any piece of data is intercepted it will be unintelligible to the interceptor.

Hosts **10.168.27.14** , **10.168.27.20**, **10.168.27.132**, all share a common vulnerability in which their operating system runs on linux 2.6.32. One of the many weaknesses that this operating system has is that it is prone to Denial of Service attacks ([cve.mitre.org](https://cve.mitre.org), 2017). With a denial of service attack a target is flooded with network traffic to point where the target is overloaded and it shuts down/crashes. This can lead to service loss, financial loss, resource consumption, negative impact on critical infrastructure, it can even serve as a smoke screen for other malicious activities. In February 2016, Linux 2.6.32 reached end of life([linuxtoday.com](https://linuxtoday.com), 2016). This explains that the only way to remediate these vulnerabilities would be to update this linux operating system to a later version that is still supported.

The wireshark file that I chose to look at was wireshark packets1.

After analyzing the captured data the first anomaly that I would like to talk about are a set of ftp transmissions that I've discovered.

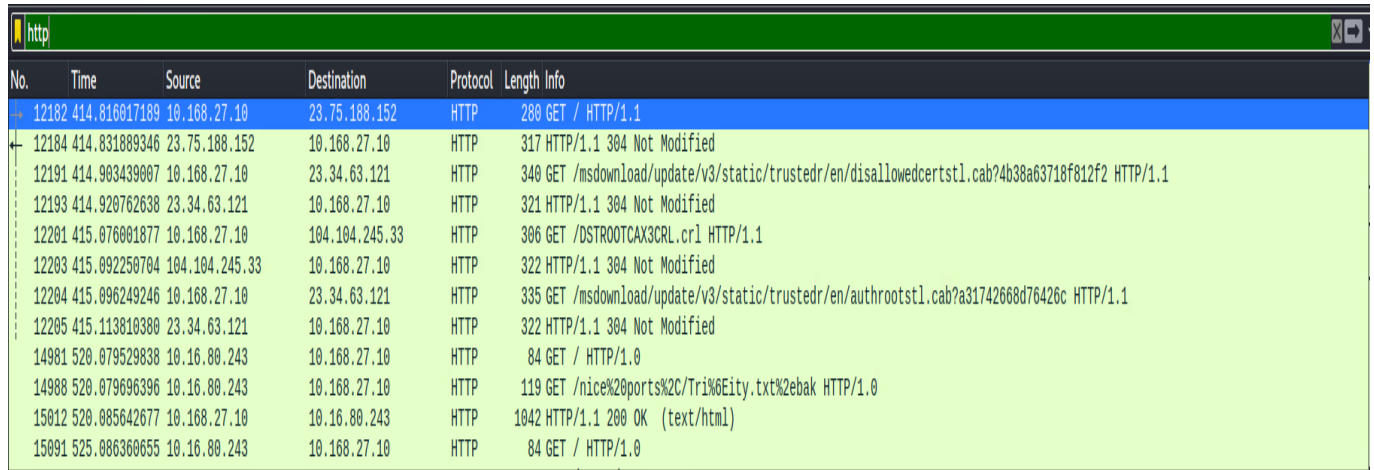


No.	Time	Source	Destination	Protocol	Length	Info
2138...	690.651507201	49.12.121.47	10.168.27.10	FTP	93	Response: 220 FZ router and firewall tester ready
2138...	690.651514416	49.12.121.47	10.168.27.10	FTP	60	Response:
2138...	690.654468522	10.168.27.10	49.12.121.47	FTP	70	Request: USER FileZilla
2138...	690.783827486	49.12.121.47	10.168.27.10	FTP	76	Response: 331 Give any password.
2138...	690.783829054	49.12.121.47	10.168.27.10	FTP	60	Response:
2138...	690.786997247	10.168.27.10	49.12.121.47	FTP	67	Request: PASS 3.55.1
2138...	690.916622296	49.12.121.47	10.168.27.10	FTP	68	Response: 230 logged on.
2138...	690.916623530	49.12.121.47	10.168.27.10	FTP	60	Response:
2138...	690.922817898	10.168.27.10	49.12.121.47	FTP	84	Request: IP 10.168.27.10 ba-bgi-ch-ba
2138...	691.053128956	49.12.121.47	10.168.27.10	FTP	107	Response: 510 Mismatch. Your IP is 199.101.110.7, bjj-bab-bba-h
2138...	691.053129566	49.12.121.47	10.168.27.10	FTP	60	Response:

Frame 2138(18): 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface eth0, id 0  
Ethernet II, Src: VMware\_b0:0c:1f (00:0c:29:b0:0c:1f), Dst: Microsof\_01:80:06 (00:15:5d:01:80:06)  
Internet Protocol Version 4, Src: 49.12.121.47, Dst: 10.168.27.10  
Transmission Control Protocol, Src Port: 21, Dst Port: 49238, Seq: 42, Ack: 17, Len: 22  
File Transfer Protocol (FTP)  
[Current working directory: ]

In this image above it shows a range of ftp transmissions. These transmissions can cause harm or disruption to a host or network. In the highlighted line it shows that passwords/ credentials can be read in plain text when using ftp. The reason passwords are able to be read in plain text is because ftp provides no encryption(ciodive.com,2018). This can lead to malicious actors gaining access to your systems, networks or host computers which poses even worse risks such as, compliance issues, data tampering, data loss etc. SFTP, or Secure Shell File Transfer Protocol, is a secure means of transferring files. It operates on the foundation of the SSH (Secure Shell) protocol, thus inheriting its robust security features and authentication capabilities (ssh.com, 2023). This information underscores that by employing the SSH protocol alongside FTP, you ensure that any transmitted plaintext data is encrypted. This encryption renders intercepted data indecipherable to any unauthorized party.

Another anomaly that I detected was a range of websites being accessed using the HTTP protocol. See below:



No.	Time	Source	Destination	Protocol	Length	Info
12182	414.816017189	10.168.27.10	23.75.188.152	HTTP	280	GET / HTTP/1.1
12184	414.831889346	23.75.188.152	10.168.27.10	HTTP	317	HTTP/1.1 304 Not Modified
12191	414.903439007	10.168.27.10	23.34.63.121	HTTP	340	GET /msdownload/update/v3/static/trustedr/en/disallowedcertstl.cab?4b38a63718f812f2 HTTP/1.1
12193	414.920762638	23.34.63.121	10.168.27.10	HTTP	321	HTTP/1.1 304 Not Modified
12201	415.076001877	10.168.27.10	104.104.245.33	HTTP	306	GET /DSTROOTCAX3CRL.crl HTTP/1.1
12203	415.092250704	104.104.245.33	10.168.27.10	HTTP	322	HTTP/1.1 304 Not Modified
12204	415.096249246	10.168.27.10	23.34.63.121	HTTP	335	GET /msdownload/update/v3/static/trustedr/en/authrootstl.cab?a31742668d76426c HTTP/1.1
12205	415.113810300	23.34.63.121	10.168.27.10	HTTP	322	HTTP/1.1 304 Not Modified
14981	520.079529838	10.16.00.243	10.168.27.10	HTTP	84	GET / HTTP/1.0
14988	520.079696396	10.16.00.243	10.168.27.10	HTTP	119	GET /nice%20ports%2C/Tri%6Eity.txt%2ebak HTTP/1.0
15012	520.085642677	10.168.27.10	10.16.00.243	HTTP	1042	HTTP/1.1 200 OK (text/html)
15091	525.086360655	10.16.00.243	10.168.27.10	HTTP	84	GET / HTTP/1.0

HTTP is an insecure protocol, information that clients use on HTTP websites are transmitted in plain text. Therefore data in motion can be captured/intercepted and used by malicious actors. These plain text transmissions can contain information that contain PII, financial data, top secret credentials or information (gcore.com,2023). Malicious actors can use this information for personal gain. Some implications can be data loss, data tampering, compliance issues, financial loss, and the compromisation of data confidentiality. To solve this issue you would need to upgrade to HTTPS instead. According to cloudflare, “With HTTPS, data is encrypted in transit in both directions: going to and coming from the origin server. The protocol keeps communications secure so that malicious parties can't observe what data is being sent. As a result usernames and passwords can't be stolen in

transit when users enter them into a form.(cloudflare.com, 2022)” Therefore this explains that the only way to remediate http would be to swap to HTTPS.

ICMP anomalies were detected here in this range below:

icmp						
No.	Time	Source	Destination	Protocol	Length	Info
12490	511.504865047	10.16.80.243	10.168.27.10	ICMP	120	Destination unreachable (Port unreachable)
12494	511.505068328	10.16.80.243	10.16.80.2	ICMP	120	Destination unreachable (Port unreachable)
13469	513.114014036	10.16.80.243	10.168.27.10	ICMP	120	Destination unreachable (Port unreachable)
13470	513.114015437	10.16.80.243	10.16.80.2	ICMP	120	Destination unreachable (Port unreachable)
14741	514.086304173	10.16.80.243	10.168.27.10	ICMP	120	Destination unreachable (Port unreachable)
14742	514.086309032	10.16.80.243	10.16.80.2	ICMP	120	Destination unreachable (Port unreachable)
14755	514.732998764	10.16.80.243	10.168.27.10	ICMP	120	Destination unreachable (Port unreachable)
14756	514.733050481	10.16.80.243	10.16.80.2	ICMP	120	Destination unreachable (Port unreachable)
14759	515.608910097	10.16.80.243	10.168.27.10	ICMP	120	Destination unreachable (Port unreachable)
14760	515.609040870	10.16.80.243	10.16.80.2	ICMP	120	Destination unreachable (Port unreachable)
14769	517.204507244	10.16.80.243	10.168.27.10	ICMP	120	Destination unreachable (Port unreachable)
14770	517.204527714	10.16.80.243	10.16.80.2	ICMP	120	Destination unreachable (Port unreachable)

Allowing the ICMP protocol is a major vulnerability, allowing ICMP packets to be transmitted on a network can cause systems and networks to be susceptible to DoS(Denial of Service) attacks(cyberstanc.com,2023). Attacks such as Ping of Death or ICMP flood , in these attacks ICMP packets are sent to a targeted machine to crash and overwhelm its input buffers. As the name reads ICMP DoS attacks result in service outages, making networks and machines inaccessible to its intended users. To remediate this issue you will need to disable the ICMP protocol on the network, according to netscout.com,”Preventing an ICMP flood DDoS attack can be accomplished by disabling the ICMP functionality of the targeted router, computer or other device.(netscout.com,2023)”

## References

- What is an ICMP flood attack?*. NETSCOUT. (n.d.).  
[https://www.netscout.com/what-is-ddos/icmp-flood#:~:text=An%20Internet%20Control%20Message%20Protocol,echo%2Drequests%20\(pings\).](https://www.netscout.com/what-is-ddos/icmp-flood#:~:text=An%20Internet%20Control%20Message%20Protocol,echo%2Drequests%20(pings).)
- Admin. (2023, March 20). *Pinging our way to remote code execution: The new ICMP vulnerability you need to know about!*. Cyberstanc Blog.  
<https://cyberstanc.com/blog/pinging-our-way-to-remote-code-execution-the-new-icmp-vulnerability-you-need-to-know-about/#:~:text=An%20attacker%20can%20send%20a,a%20critical%20level%20of%20severity>
- Why use HTTPS?*  
(n.d.).<https://www.cloudflare.com/learning/ssl/why-use-https/#:~:text=With%20HTTPS%2C%20data%20is%20encrypted,enter%20them%20into%20a%20form>
- “Why Is HTTP Not Secure? The Difference between HTTP and HTTPS - Gcore.”  
*Gcore*, <https://www.facebook.com/gcorelabscom>,  
<https://gcore.com/learning/http-vs-https-security-comparison/>. Accessed 7 Oct. 2023.
- Real risks of using file transfer protocol( Chandra Shekhar)  
<https://www.ciodive.com/news/real-risks-of-using-file-transfer-protocol/528881/>



*Microsoft Windows 7 : Security vulnerabilities, CVEs.* (n.d.). Microsoft Windows 7 : Security Vulnerabilities, CVEs.

[https://www.cvedetails.com/vulnerability-list/vendor\\_id-26/product\\_id-17153/Microsoft-Windows-7.html?page=1&order=1&trc=2369&sha=24cc2fec22a2ef0d7f178a8657832bffd6f301](https://www.cvedetails.com/vulnerability-list/vendor_id-26/product_id-17153/Microsoft-Windows-7.html?page=1&order=1&trc=2369&sha=24cc2fec22a2ef0d7f178a8657832bffd6f301)

*Windows 7 support ended on January 14, 2020 - Microsoft Support.* (n.d.).

Windows 7 Support Ended on January 14, 2020 - Microsoft Support.

<https://support.microsoft.com/en-us/windows/windows-7-support-ended-on-january-14-2020-b75d4580-2cc7-895a-2c9c-1466d9a53962>

Young, T. (2019, February 8). *Secure FTP - how to mitigate the risks and keep data secure.* Cerberus FTP Server.

<https://www.cerberusftp.com/blog/how-secure-is-ftp-how-you-can-mitigate-the-risks-of-using-file-transfer-protocol/>

*SSH File Transfer Protocol (SFTP): Secure File Transfer Protocol.* (n.d.). SSH File Transfer Protocol (SFTP): Secure File Transfer Protocol.

<https://www.ssh.com/academy/ssh/sftp-ssh-file-transfer-protocol>

*CVE -CVE-2017-1000407.* (n.d.). CVE -CVE-2017-1000407.

<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-1000407>

*Linux Kernel 2.6.32 LTS Reaches End of Life on February 2016 | Linux Today.*

(2016, January 30). Linux Today.

<https://www.linuxtoday.com/news/linux-kernel-2-6-32-lts-reaches-end-of-life-on-february-2016/>