Introduction

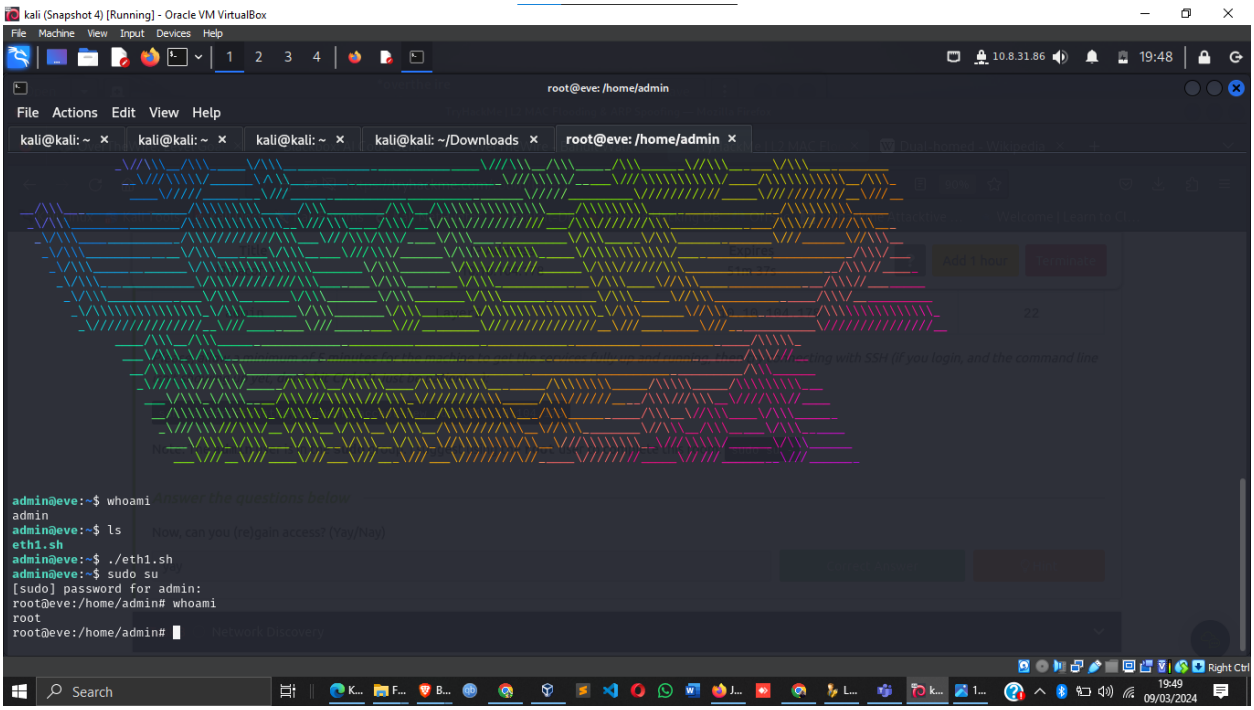
This report outlines the tasks and outcomes of the TryHackMe room L2 MAC Flooding & ARP Spoofing, which delves into MAC flooding and ARP cache poisoning techniques for network manipulation, including sniffing and man-in-the-middle attacks.

https://tryhackme.com/p/Damiano254

Getting Started and Initial Access

After logging in to the target network, I was able to gain initial access to the machine using OpenVPN and use of Ssh.

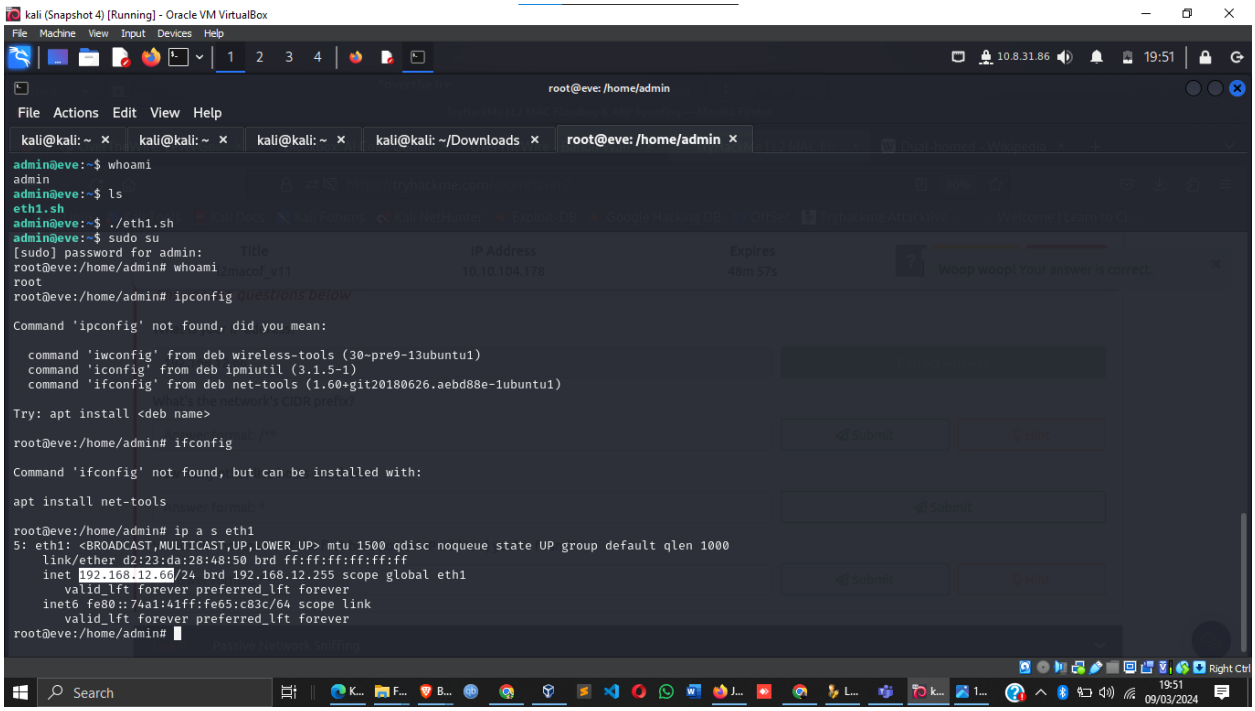
- Can you (re)gain access? (Yay/Nay): **Yay**



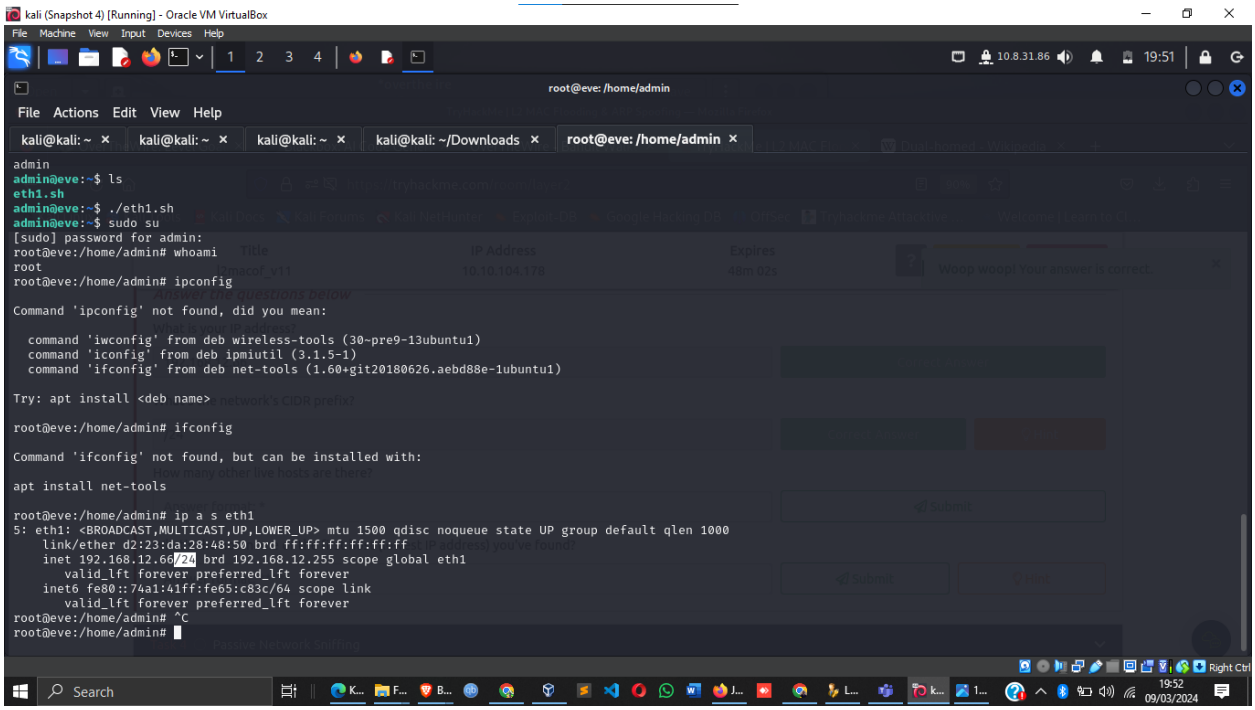
Network Discovery

The user's IP address was 192.168.12.66 with a CIDR prefix of /24. There were two other live hosts on the network, with the first host's (lowest IP address) name being Alice.

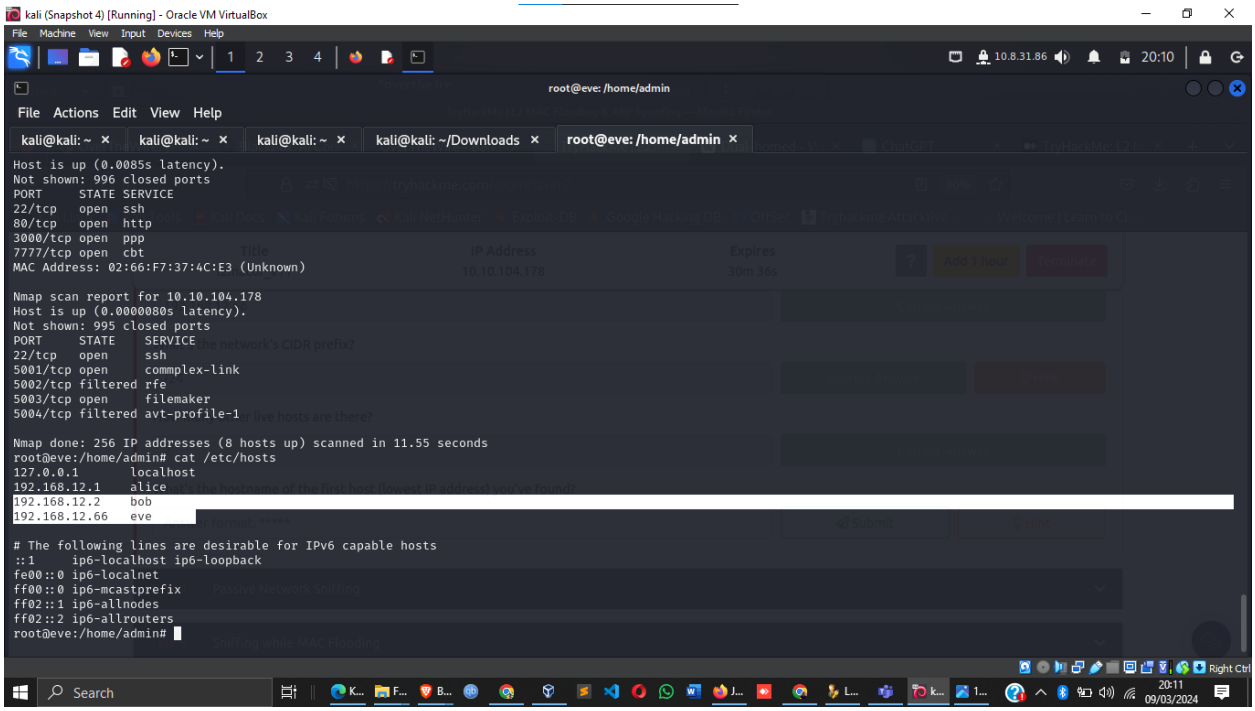
- **What is your IP address**? 192.168.12.66



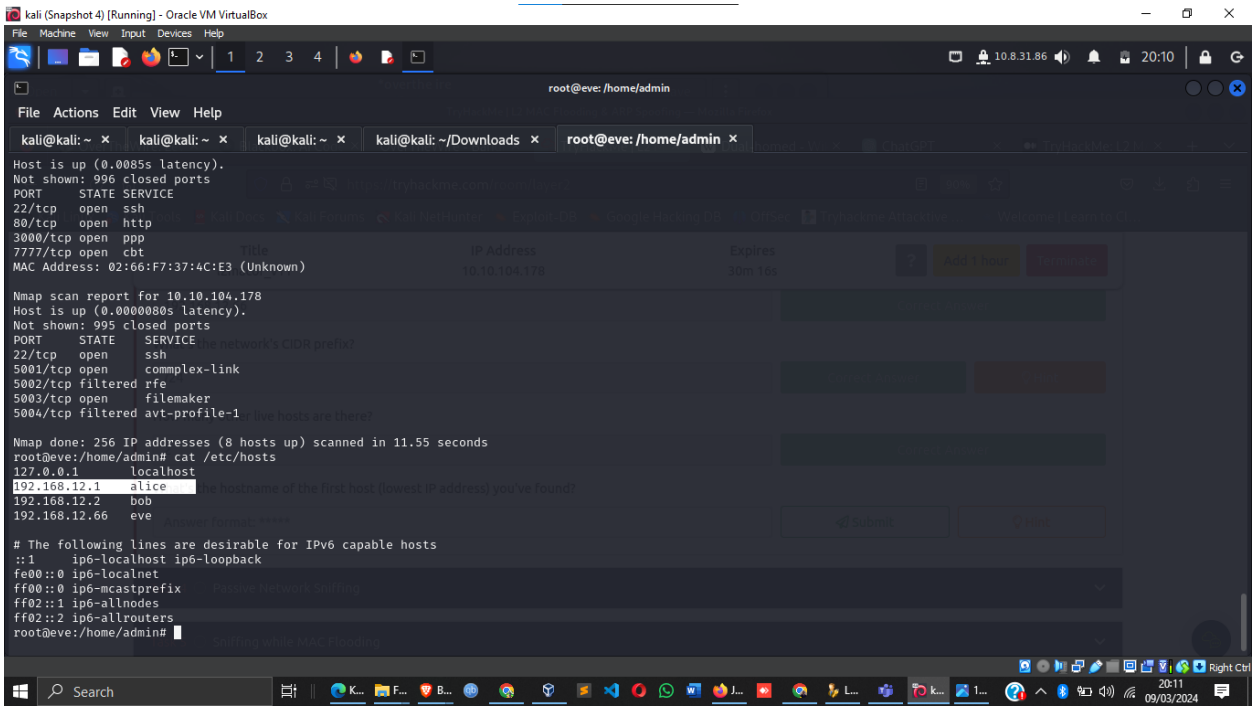
- **What's the network's CIDR prefix**? /24



- **How many other live hosts are there**? 2



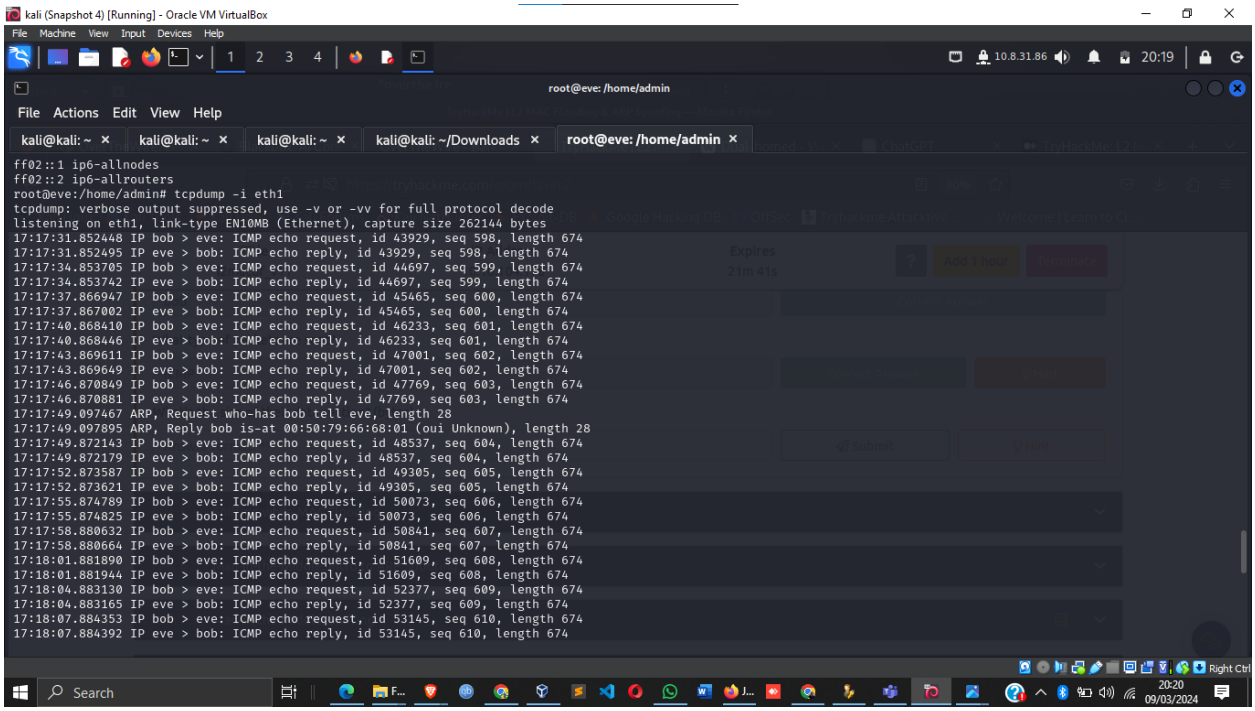
- **What's the hostname of the first host (lowest IP address) you've found?** Alice



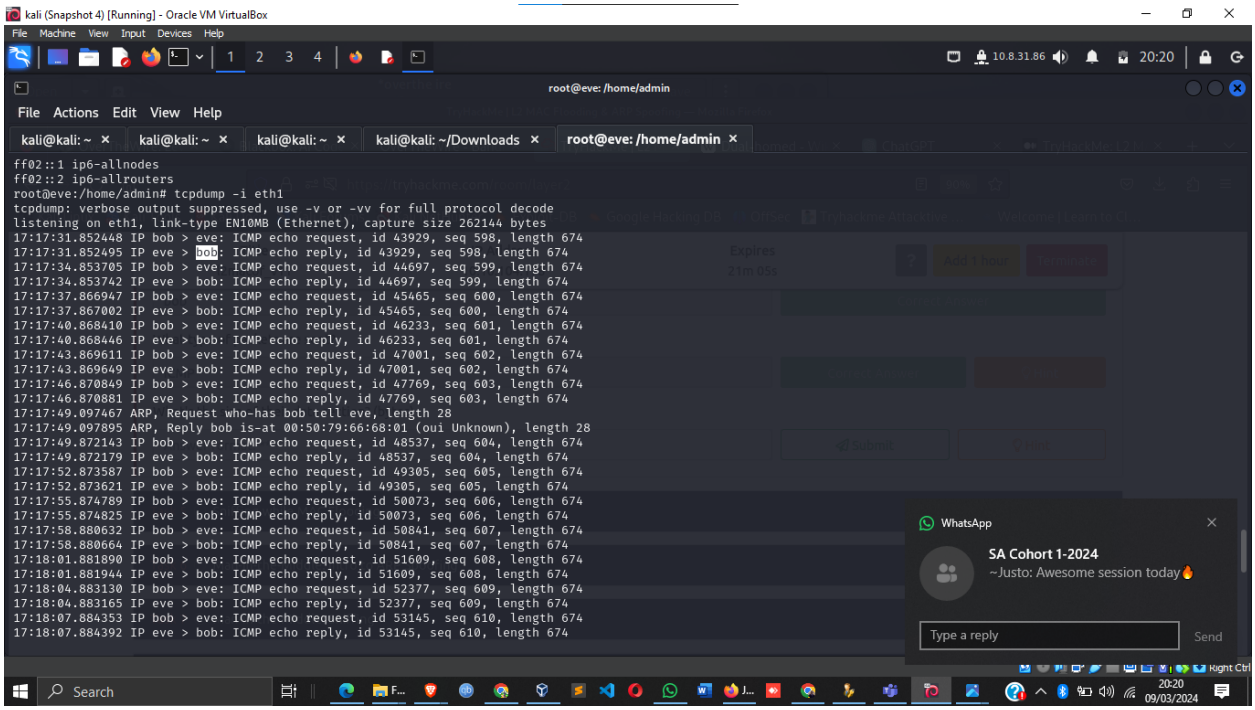
Passive Network Sniffing

I observed traffic from hosts on the network. Specifically, Bob was sending packets to eve with a type of ICMP and a data section size of 666 bytes.

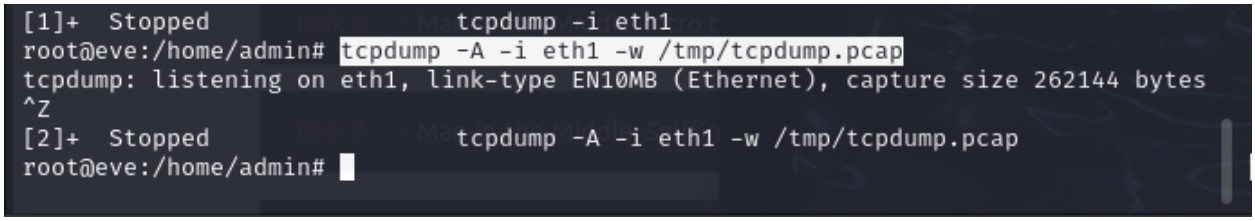
- **Can you see any traffic from those hosts? (Yay/Nay):** Yay

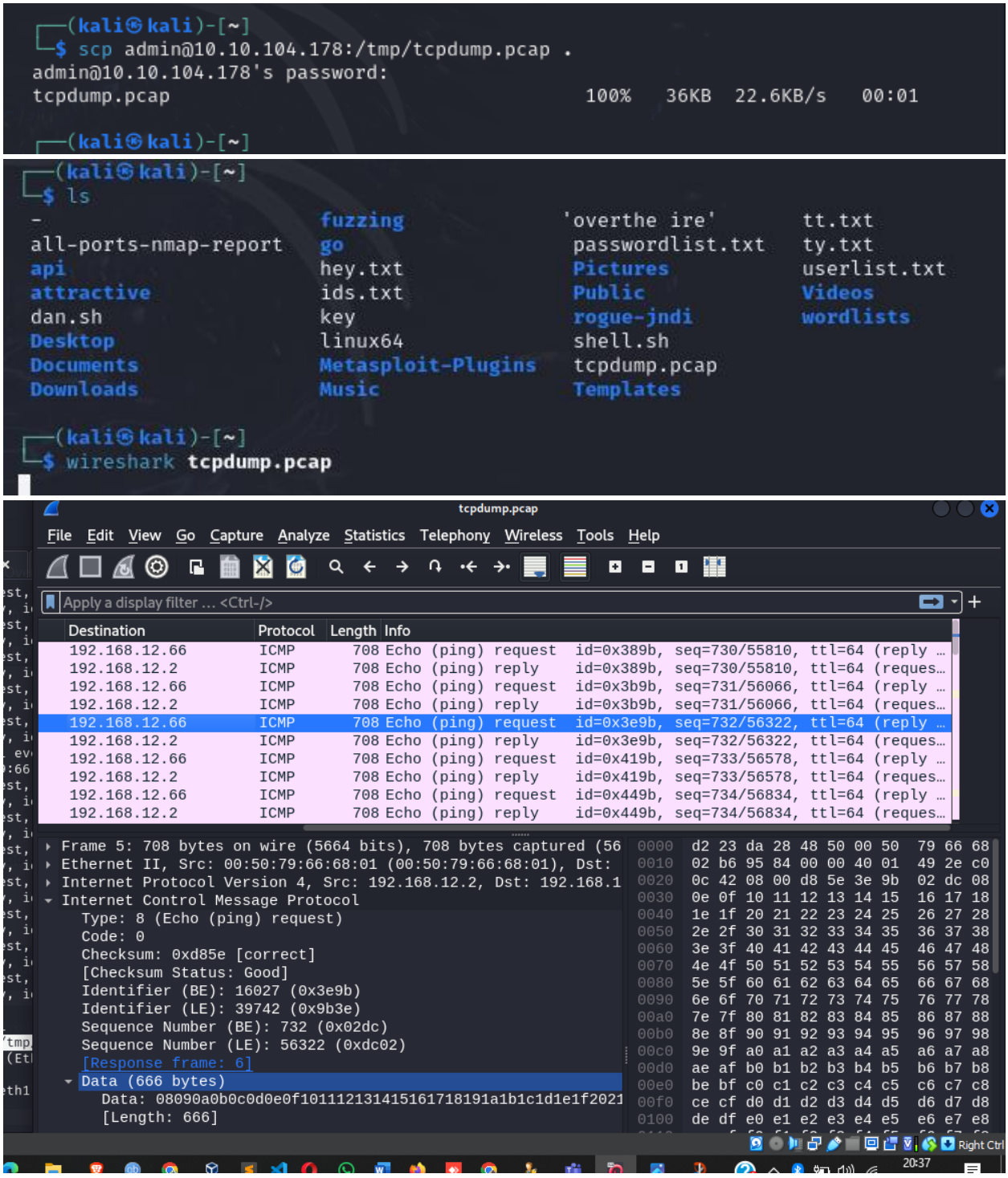


- **Who keeps sending packets to eve?** Bob

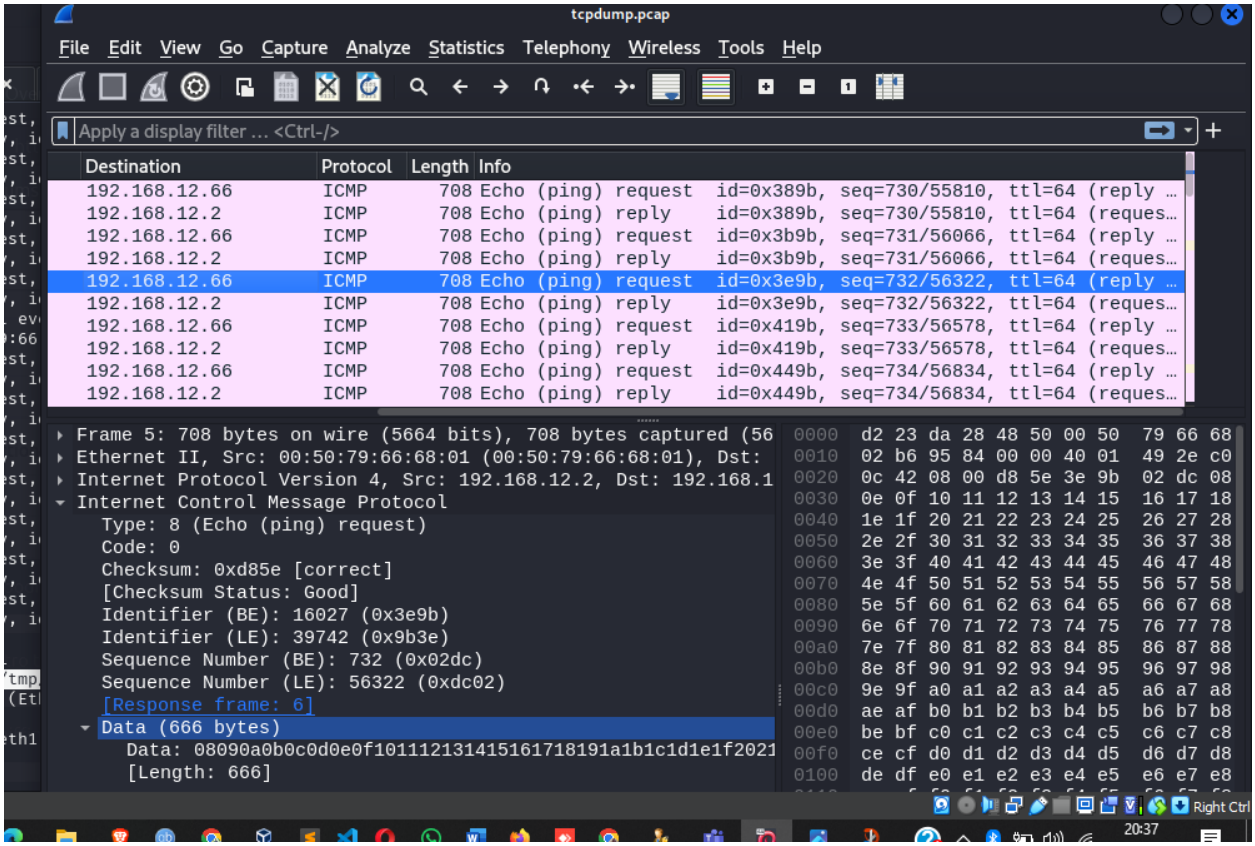


- **What type of packets are sent?** ICMP





- **What's the size of their data section? (bytes)**: 666



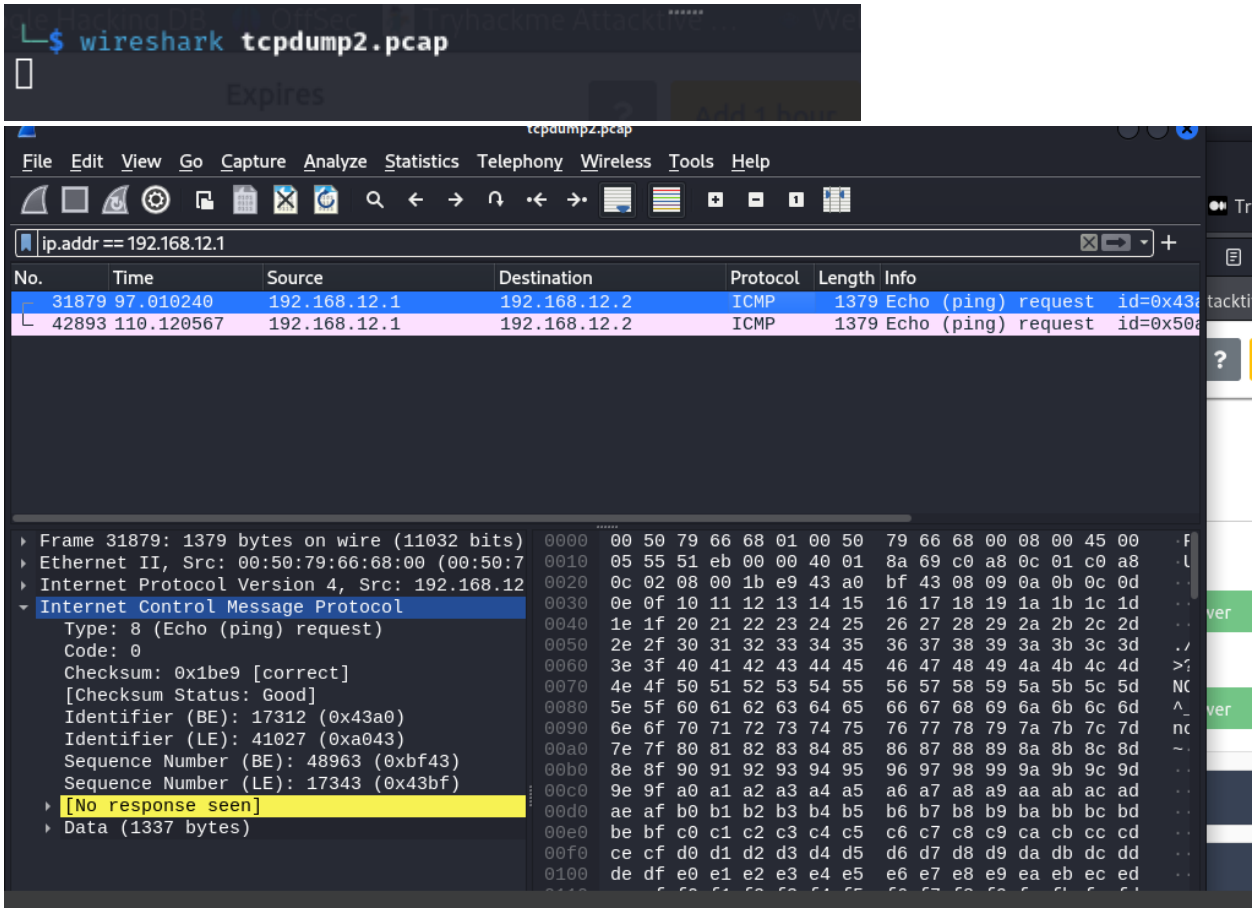
Sniffing while MAC Flooding

Alice was continuously sending ICMP packets to Bob with a data section size of 1337 bytes.

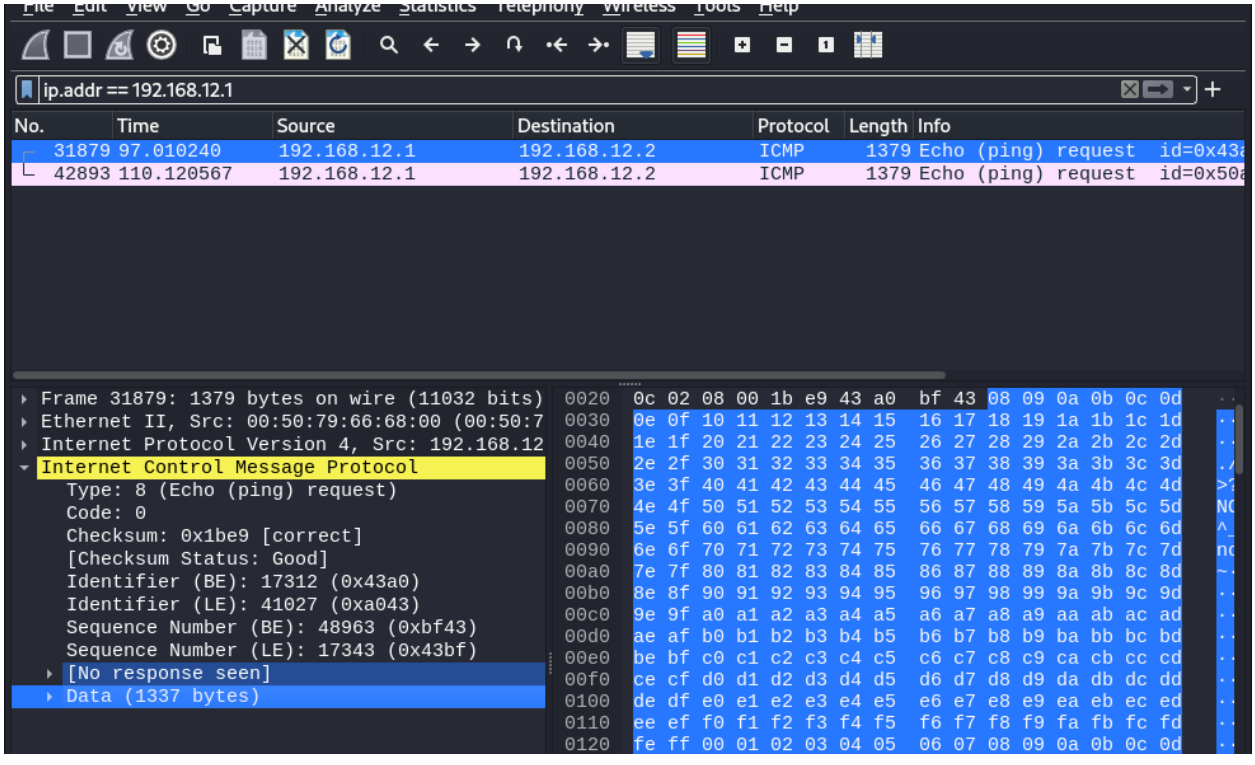
- **What kind of packets is Alice continuously sending to Bob?** ICMP

```
root@eve:/home/admin# tcpdump -A -i eth1 -w /tmp/tcpdump2.pcap
tcpdump: listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
^C48054 packets captured
48054 packets received by filter
0 packets dropped by kernel
root@eve:/home/admin# /tmp/tcpdump2.pcap
```

```
root@eve:/home/admin# tcpdump -A -i eth1 -w /tmp/tcpdump.pcap
tcpdump: /tmp/tcpdump.pcap: Permission denied
root@eve:/home/admin# macof -i eth1
5e:5b:49:25:8b:91 75:6:60:41:11:b6 0.0.0.0.16281 > 0.0.0.0.31439: S 1889638745:188
9638745(0) win 512
95:d:72:1d:ae:90 6b:43:bf:47:3c:4b 0.0.0.0.62451 > 0.0.0.0.36073: S 1082120285:108
2120285(0) win 512
aa:a8:cd:32:61:64 1a:d:ec:40:31:8 0.0.0.0.35655 > 0.0.0.0.33649: S 2064510015:2064
510015(0) win 512
e7:2f:c2:1c:f4:67 27:d:c0:3d:7f:94 0.0.0.0.62238 > 0.0.0.0.56734: S 622708346:6227
08346(0) win 512
9:92:de:5b:9f:16 13:5f:b4:2e:11:f9 0.0.0.0.24374 > 0.0.0.0.54735: S 1870960706:187
0960706(0) win 512
b5:28:e8:6e:3e:63 77:fc:fc:5f:a4:65 0.0.0.0.43009 > 0.0.0.0.30390: S 123433972:123
433972(0) win 512
ce:2:45:2d:f9:c7 9d:8a:a6:6c:fa:7a 0.0.0.0.60354 > 0.0.0.0.20375: S 1302603077:130
2603077(0) win 512
94:5b:1c:71:d0:e0 1b:1d:df:50:77:da 0.0.0.0.21844 > 0.0.0.0.47384: S 1004300744:10
04300744(0) win 512
bf:3c:6c:74:42:bd 47:a:12:36:9f:8d 0.0.0.0.34173 > 0.0.0.0.19176: S 1601913604:160
1913604(0) win 512
d:62:29:78:92:52 30:7f:97:49:7a:51 0.0.0.0.26308 > 0.0.0.0.14672: S 1295296419:129
5296419(0) win 512
64:21:46:6f:88:41 65:1e:19:1c:a4:7e 0.0.0.0.20090 > 0.0.0.0.61779: S 700457913:700
457913(0) win 512
53:b8:1c:16:a4:7e 42:1a:1f:17:c9:e 0.0.0.0.14994 > 0.0.0.0.4546: S 496107934:49610
7934(0) win 512
62:dc:2d:36:67:79 1d:4a:86:65:57:db 0.0.0.0.152 > 0.0.0.0.45477: S 1044543549:1044
543549(0) win 512
3f:f9:13:12:2a:c7 d5:5f:22:57:d1:45 0.0.0.0.55857 > 0.0.0.0.5193: S 933301418:9333
01418(0) win 512
```



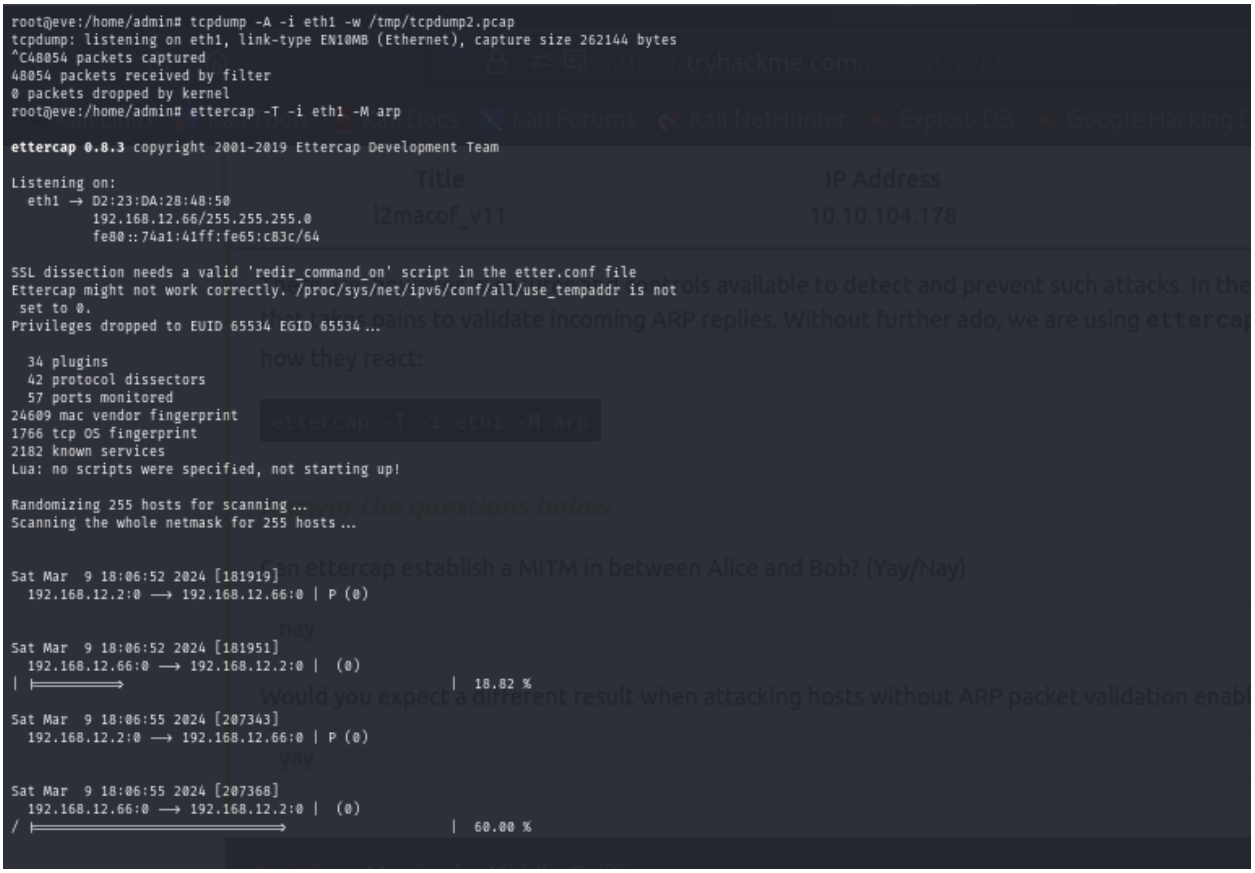
- **What's the size of their data section? (bytes):** 1337



Man-in-the-Middle: Intro to ARP Spoofing

The user attempted an ARP spoofing attack between Alice and Bob, but it was unsuccessful. However, if the target hosts had ARP packet validation disabled, the user would have expected a different result.

- **Can ettercap establish a MITM in between Alice and Bob? (Yay/Nay):** Nay

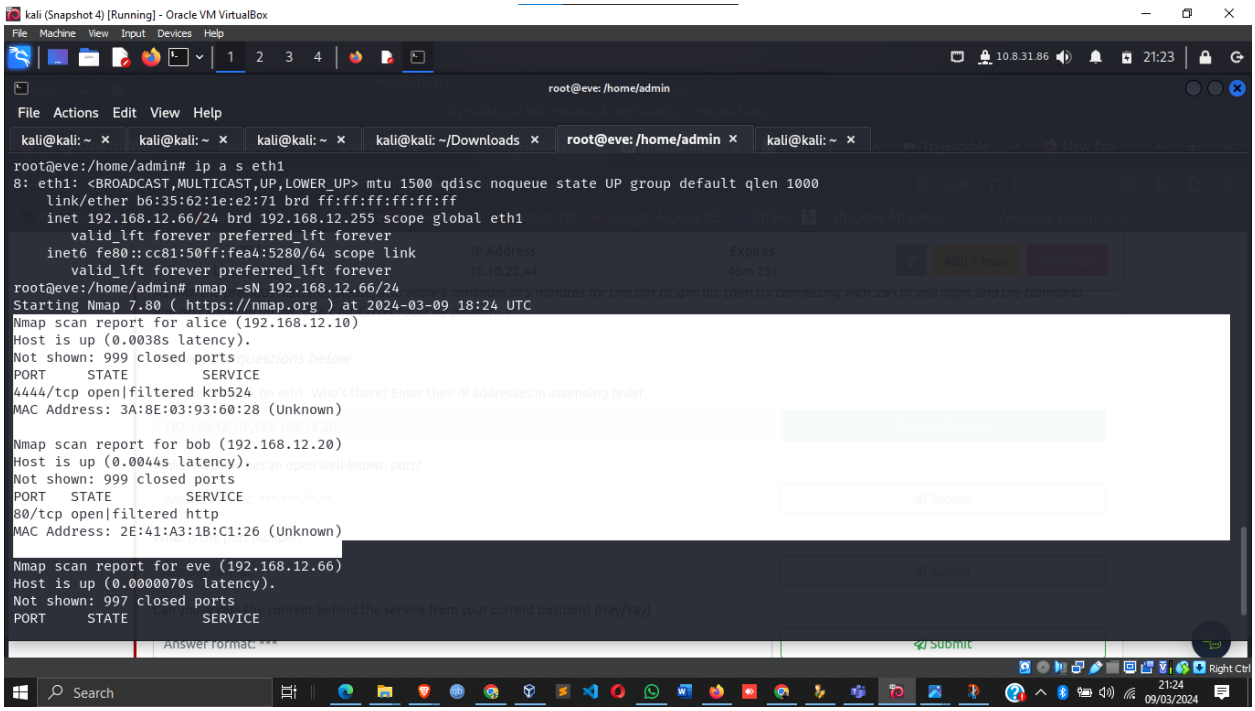


- **Would you expect a different result when attacking hosts without ARP packet validation enabled? (Yay/Nay):** Yay

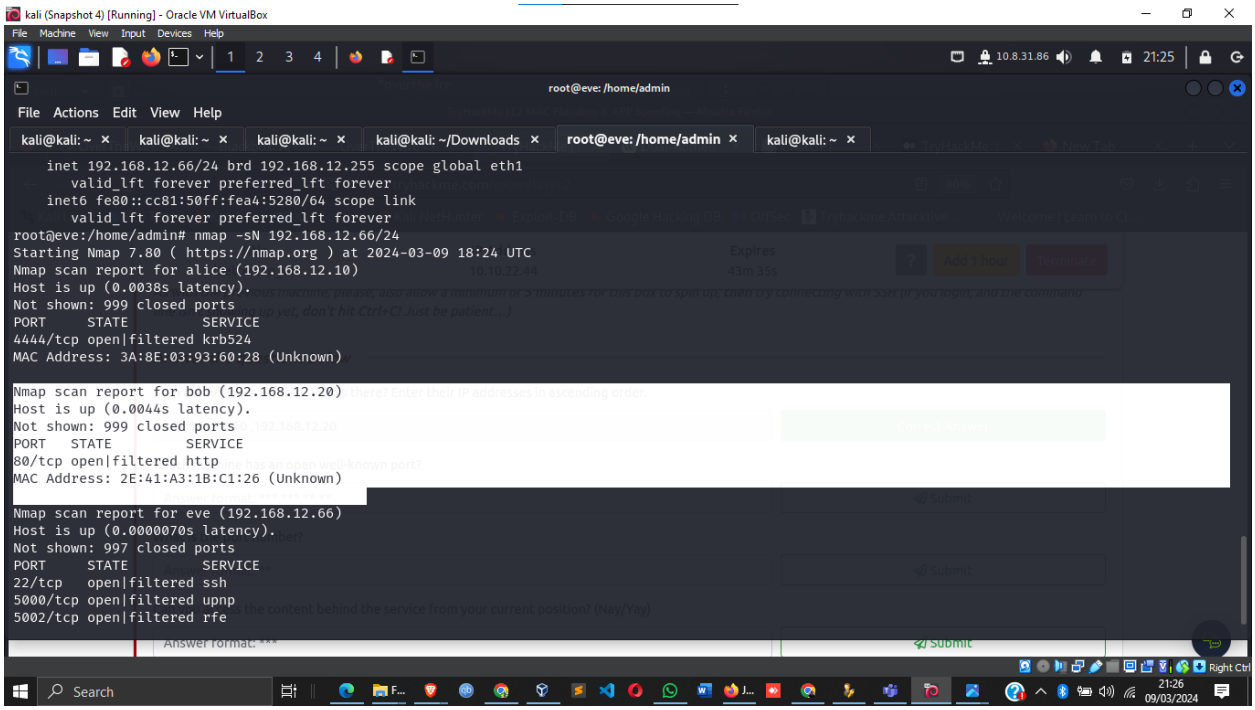
Man-in-the-Middle: Sniffing

The two live hosts were 192.168.12.10 and 192.168.12.20. Only one machine, 192.168.12.20, had an open well-known port, port 80. 1 could not access the content behind the service from their current position. However, after launching the ARP spoofing attack, i observed interesting traffic, including requests for the file test.txt and authentication credentials (admin:s3cr3t_P4zz). i was then able to access the content behind the service using the obtained credentials.

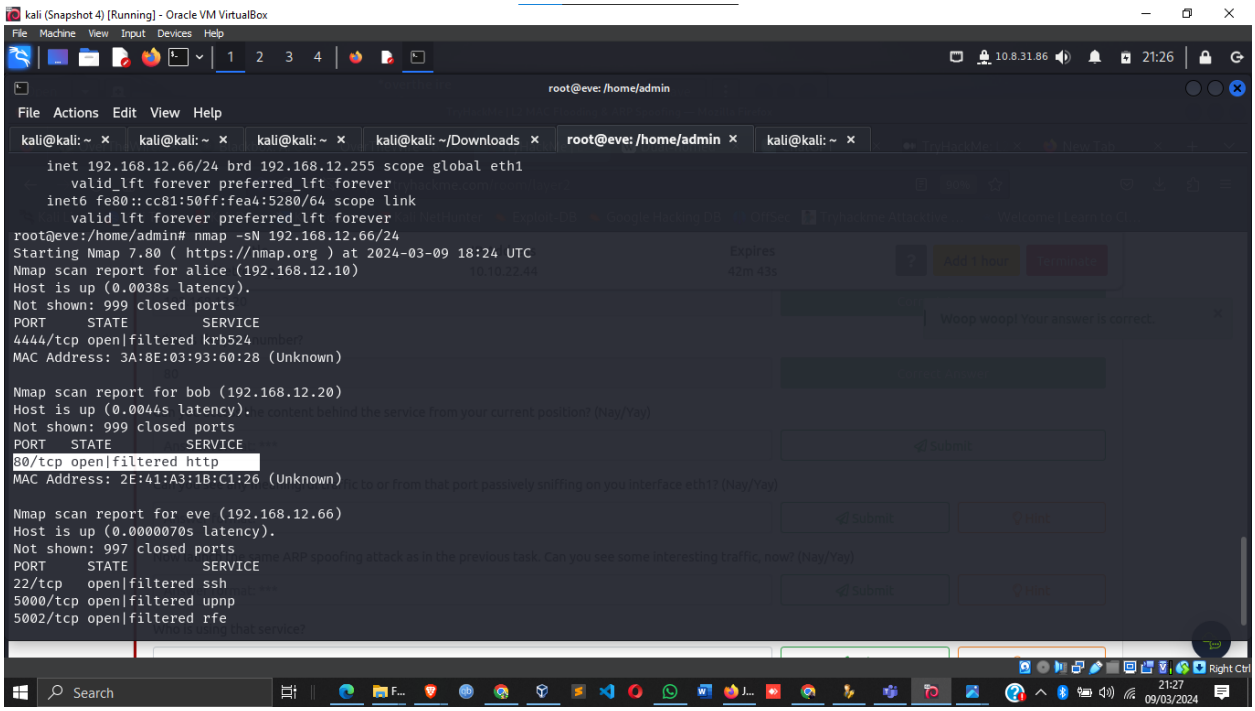
- **Scan the network on eth1. Who's there? Enter their IP addresses in ascending order:** 192.168.12.10, 192.168.12.20



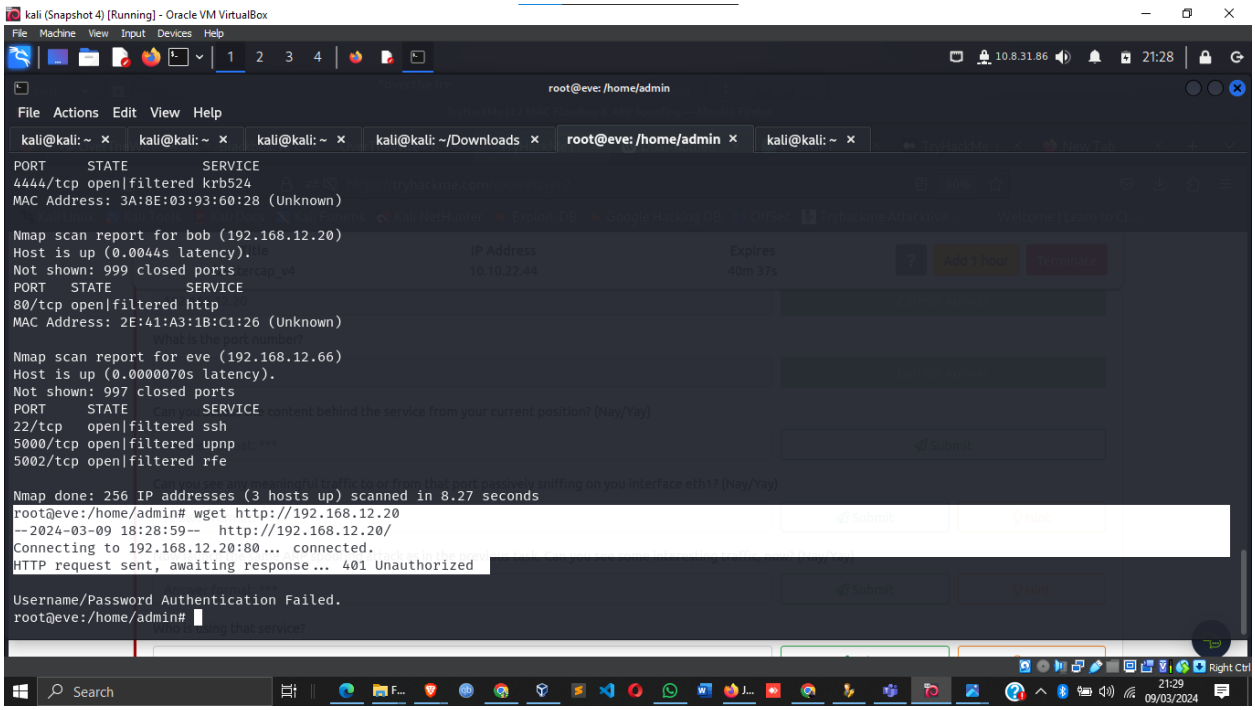
- **Which machine has an open well-known port?** 192.168.12.20



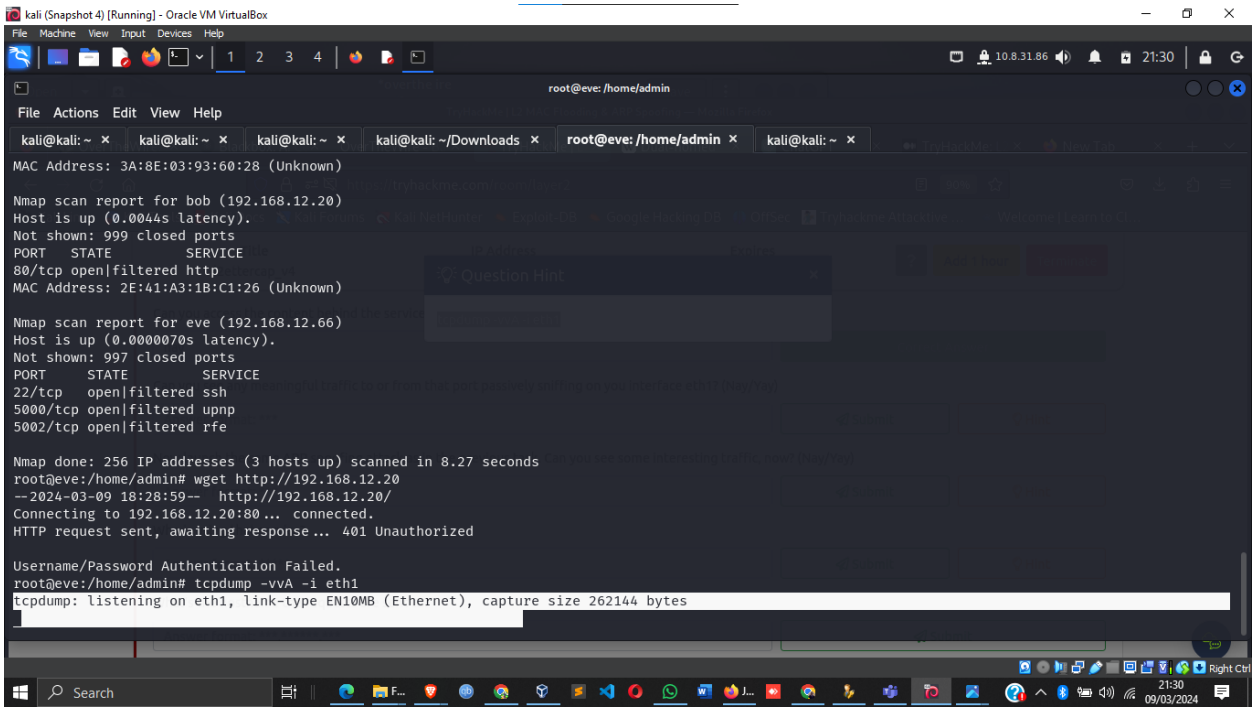
- **What is the port number?** 80



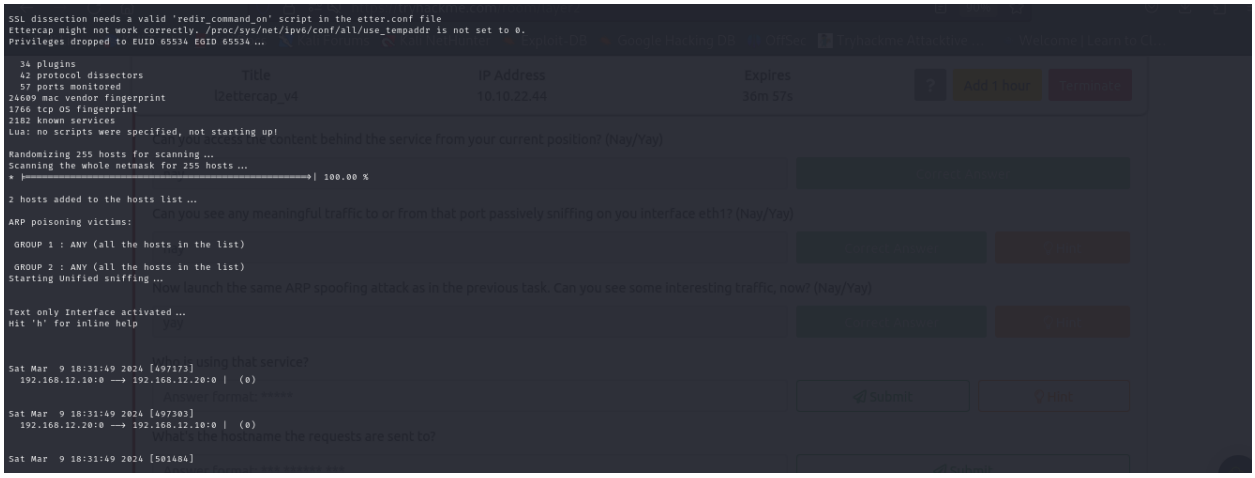
- **Can you access the content behind the service from your current position? (Nay/Yay):** Nay

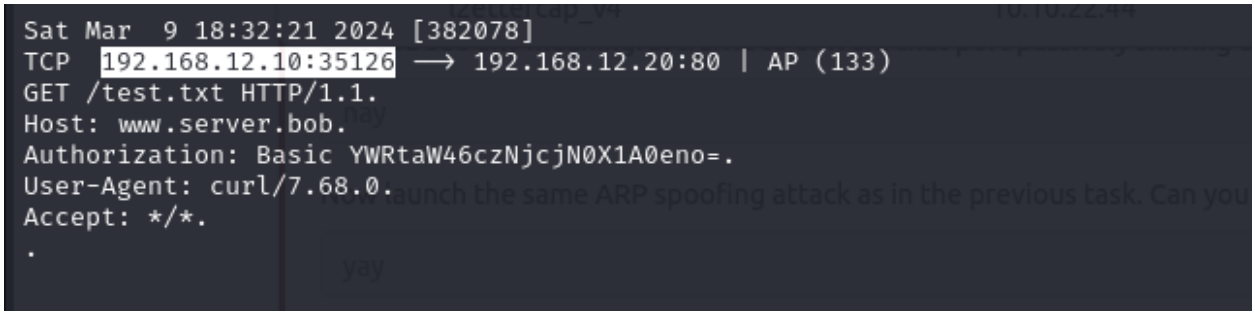


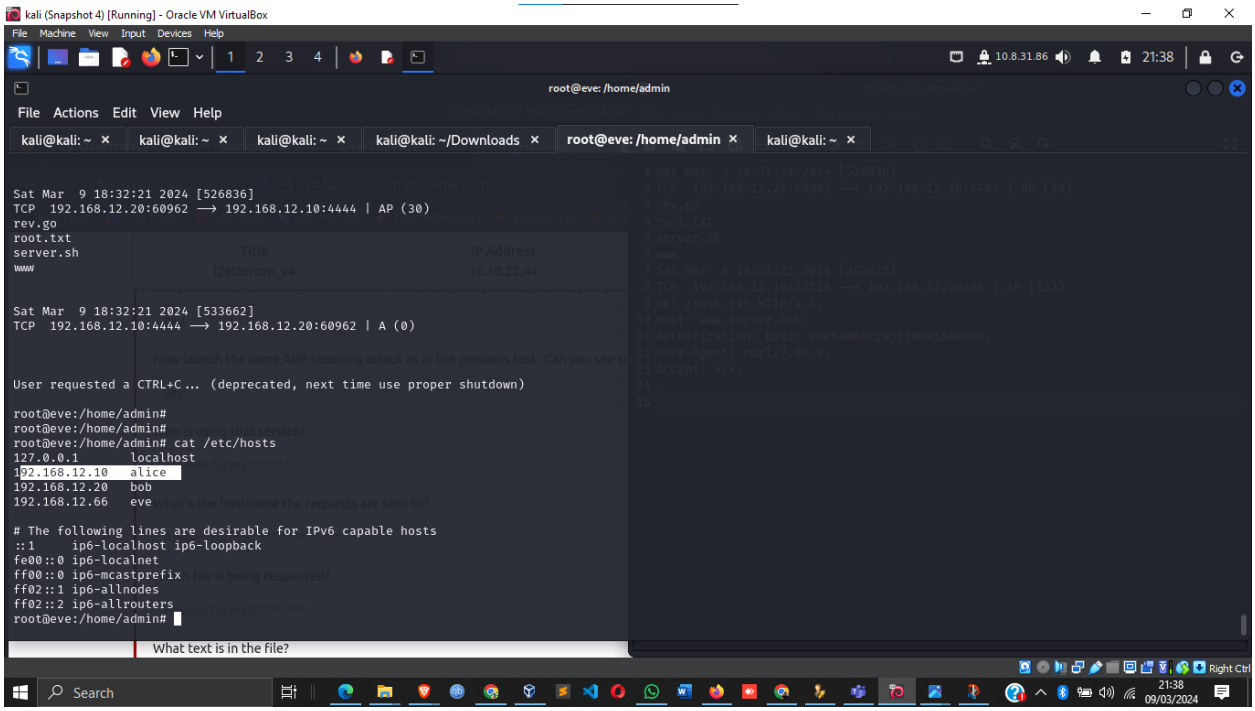
- **Can you see any meaningful traffic to or from that port passively sniffing on your interface eth1? (Nay/Yay):** Nay



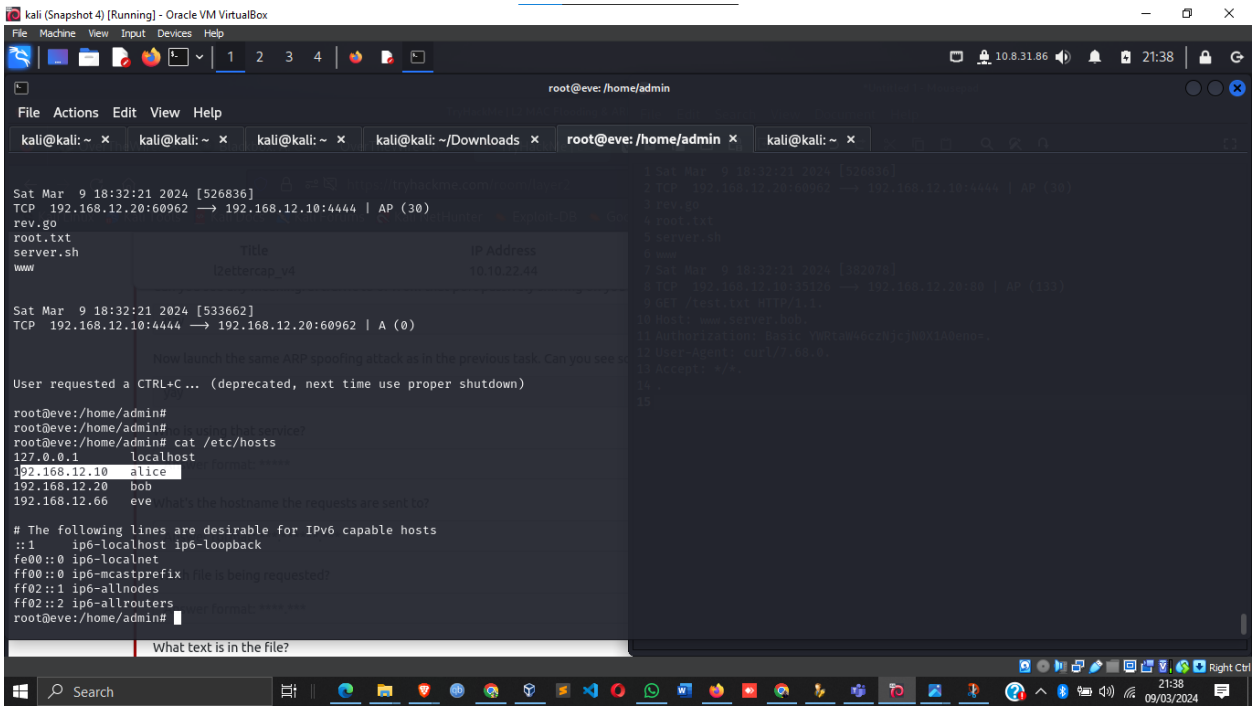
- **Now launch the same ARP spoofing attack as in the previous task. Can you see some interesting traffic, now? (Nay/Yay):** Yay



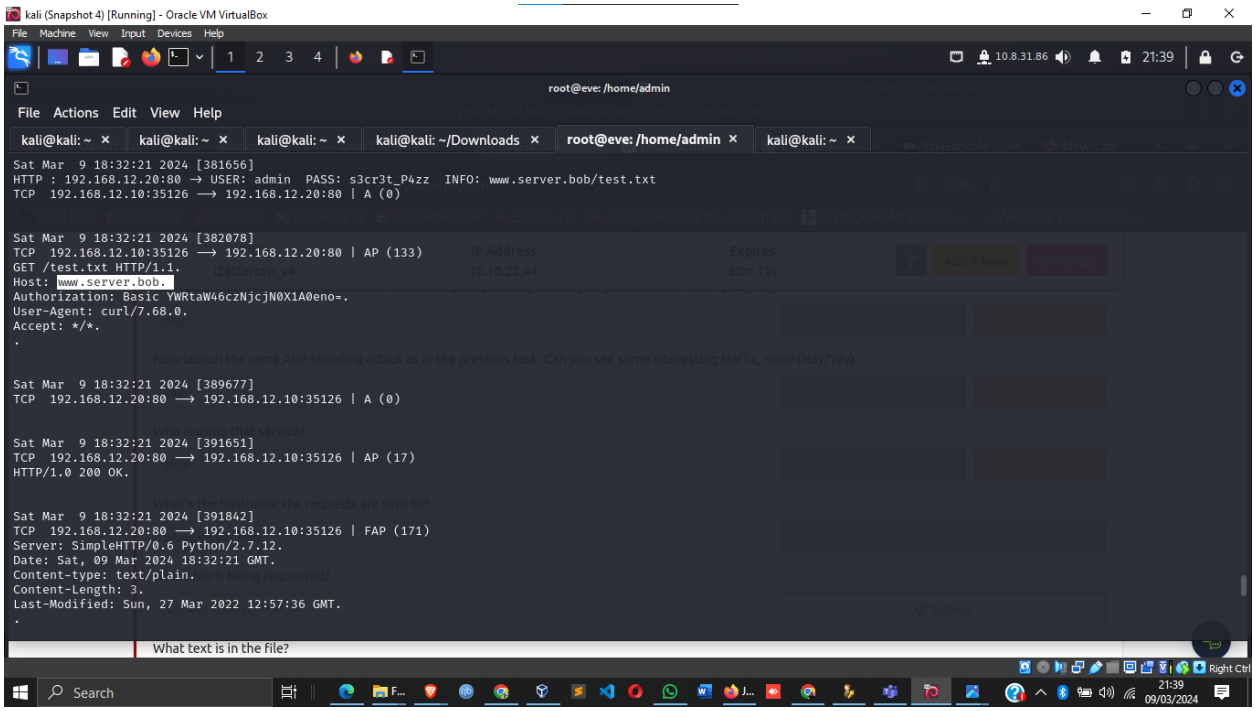




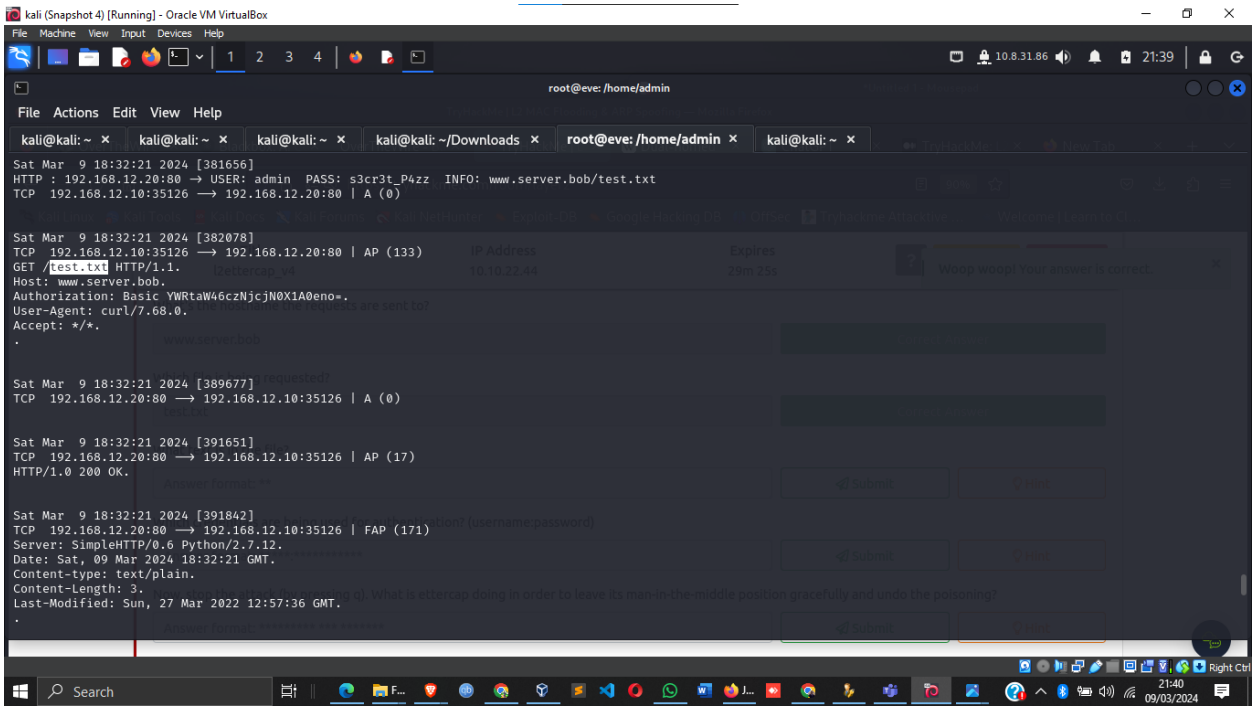
- **Who is using that service? Alice**



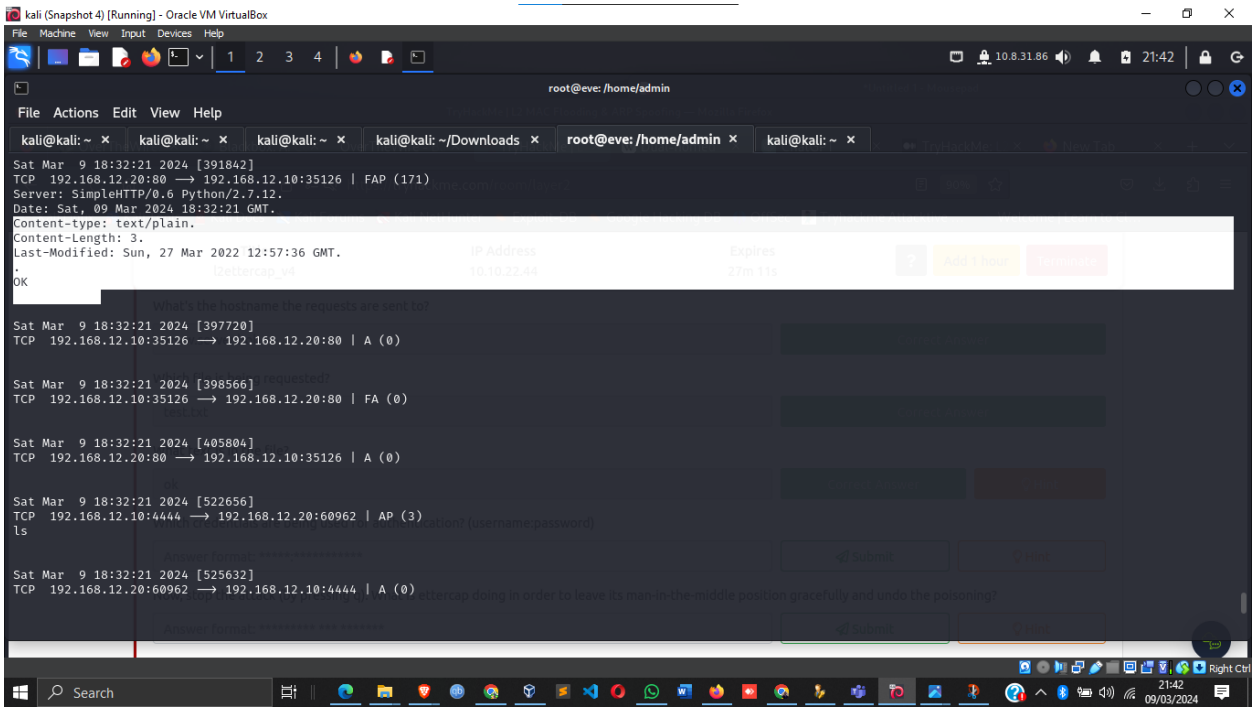
- **What's the hostname the requests are sent to? www.server.bob**



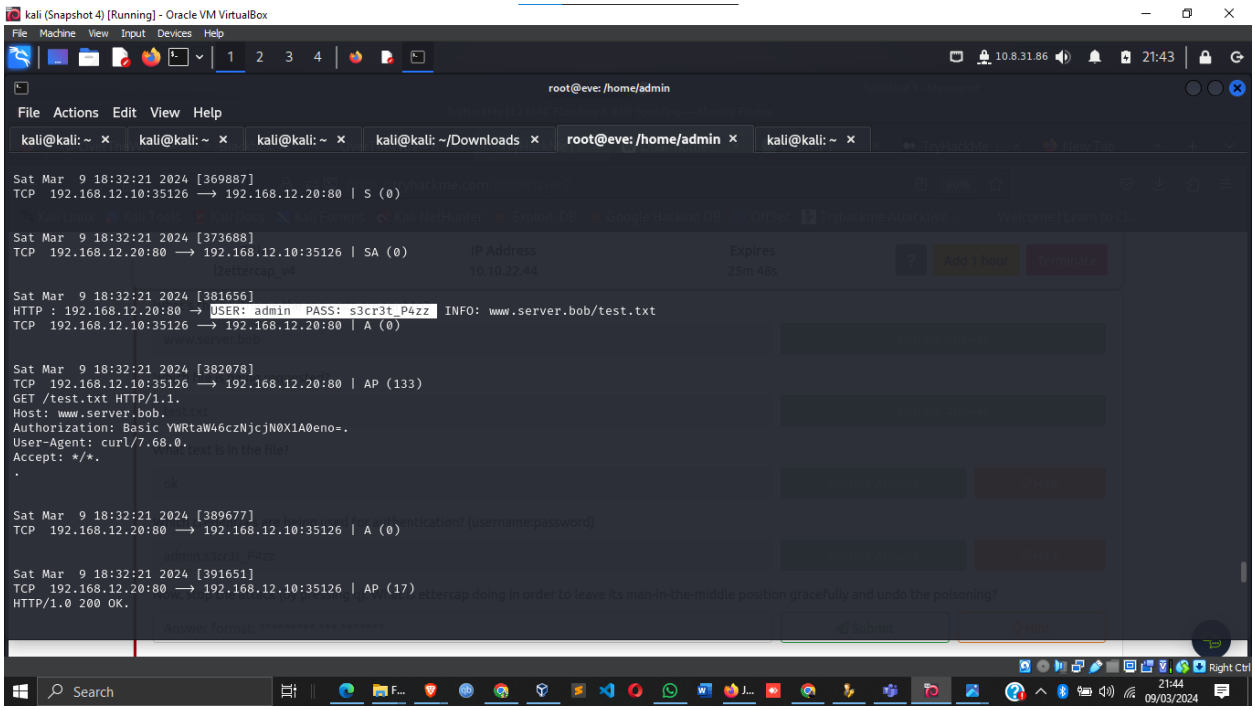
- **Which file is being requested?** test.txt



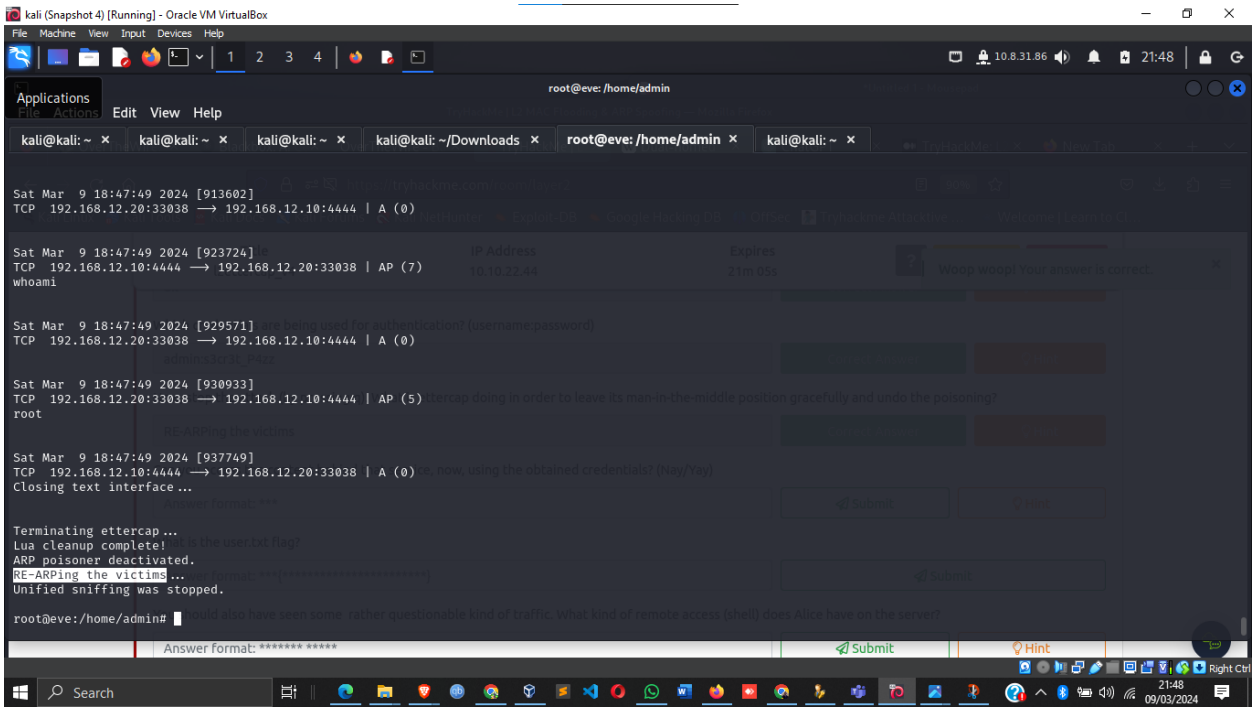
- **What text is in the file?** OK



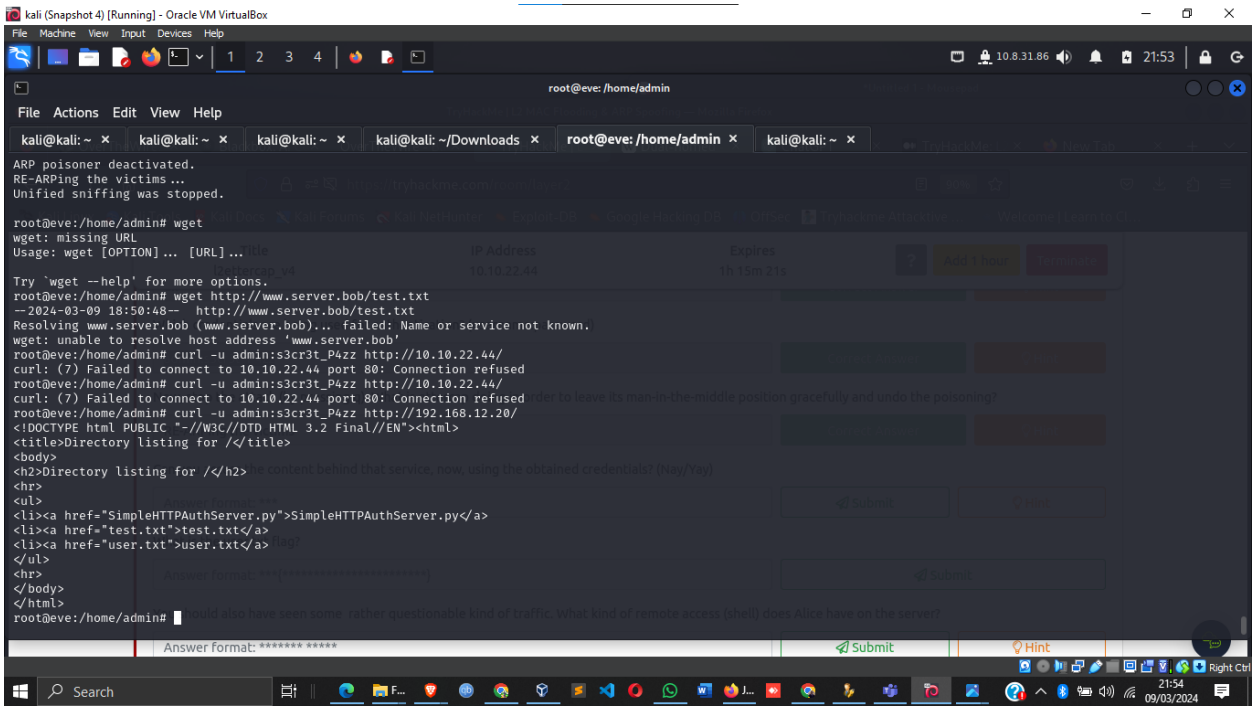
- **Which credentials are being used for authentication? (username:password)** admin:s3cr3t_P4zz



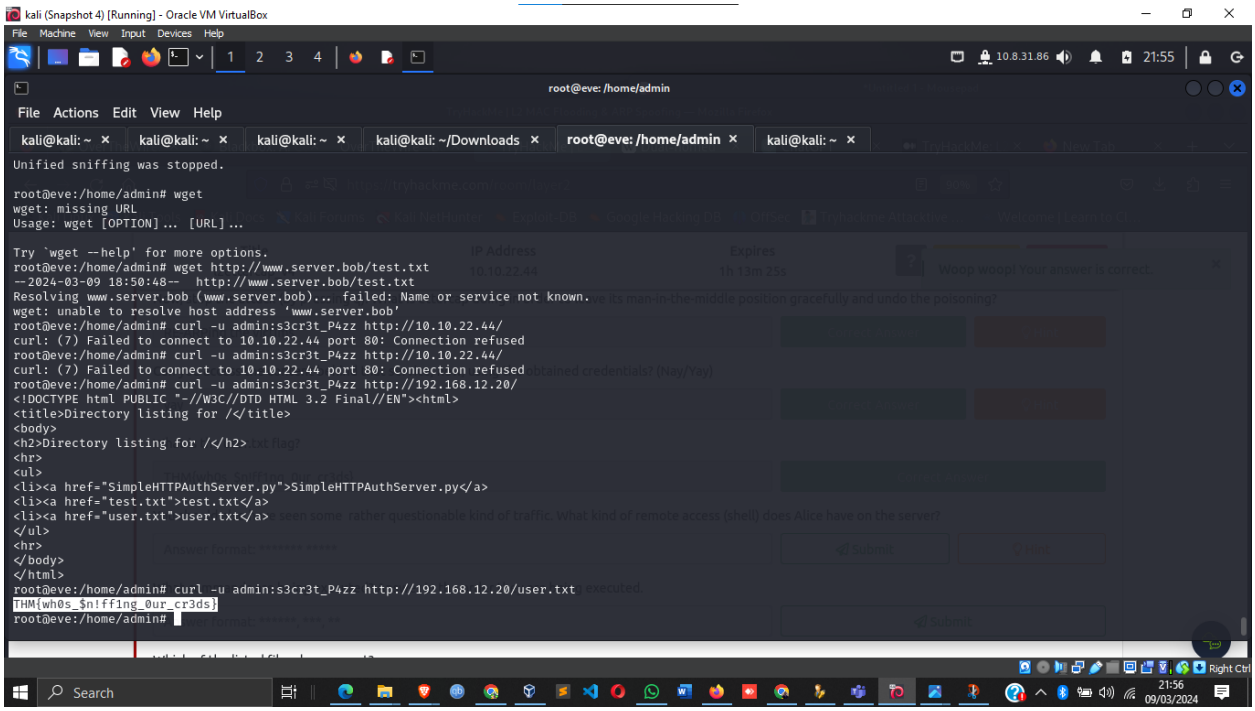
- **Now, stop the attack (by pressing q). What is ettercap doing in order to leave its man-in-the-middle position gracefully and undo the poisoning?** RE-ARPing the victims



- **Can you access the content behind that service, now, using the obtained credentials? (Nay/Yay):** Yay



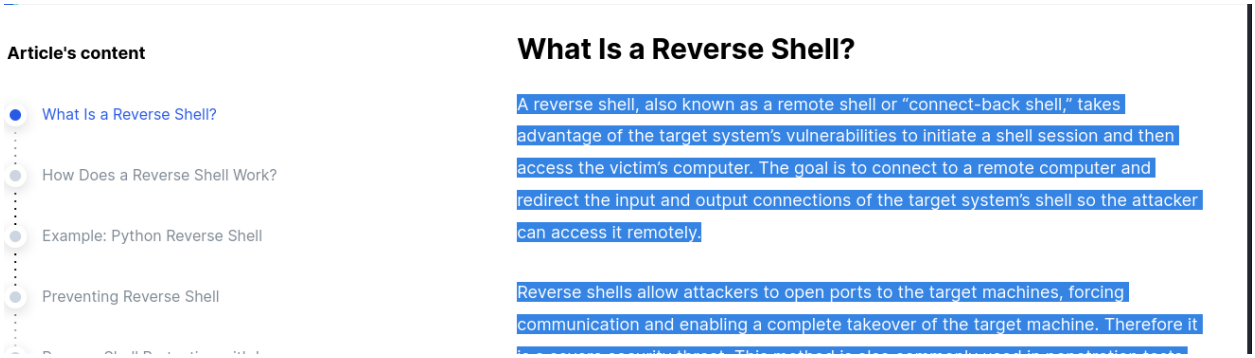
- **What is the user.txt flag?** THM{wh0s_$n!ff1ng_0ur_cr3ds}



Man-in-the-Middle: Manipulation

The user found the root.txt flag (THM{wh4t_an_ev1l_M!tM_u_R}). Alice had a reverse shell on the server, and the user saw the commands being executed in order: whoami, pwd, and ls. The user then read the contents of the root.txt file.

- **You should also have seen some rather questionable kind of traffic. What kind of remote access (shell) does Alice have on the server?** Reverse shell

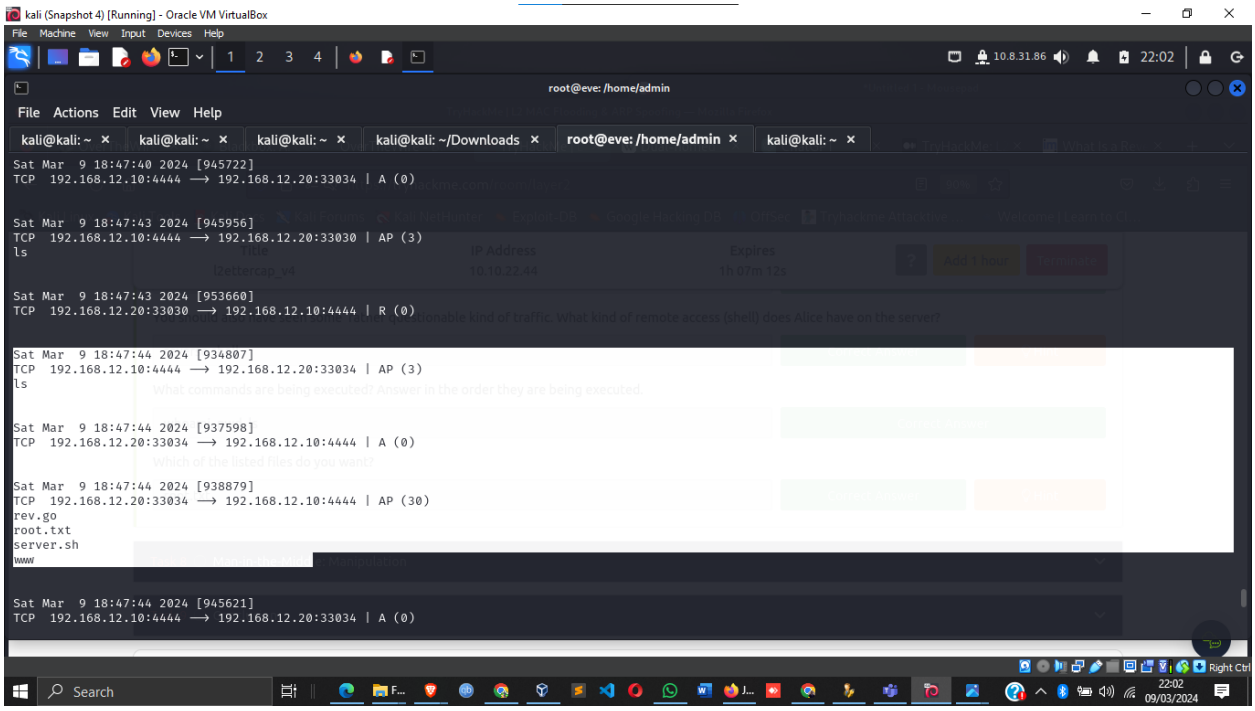


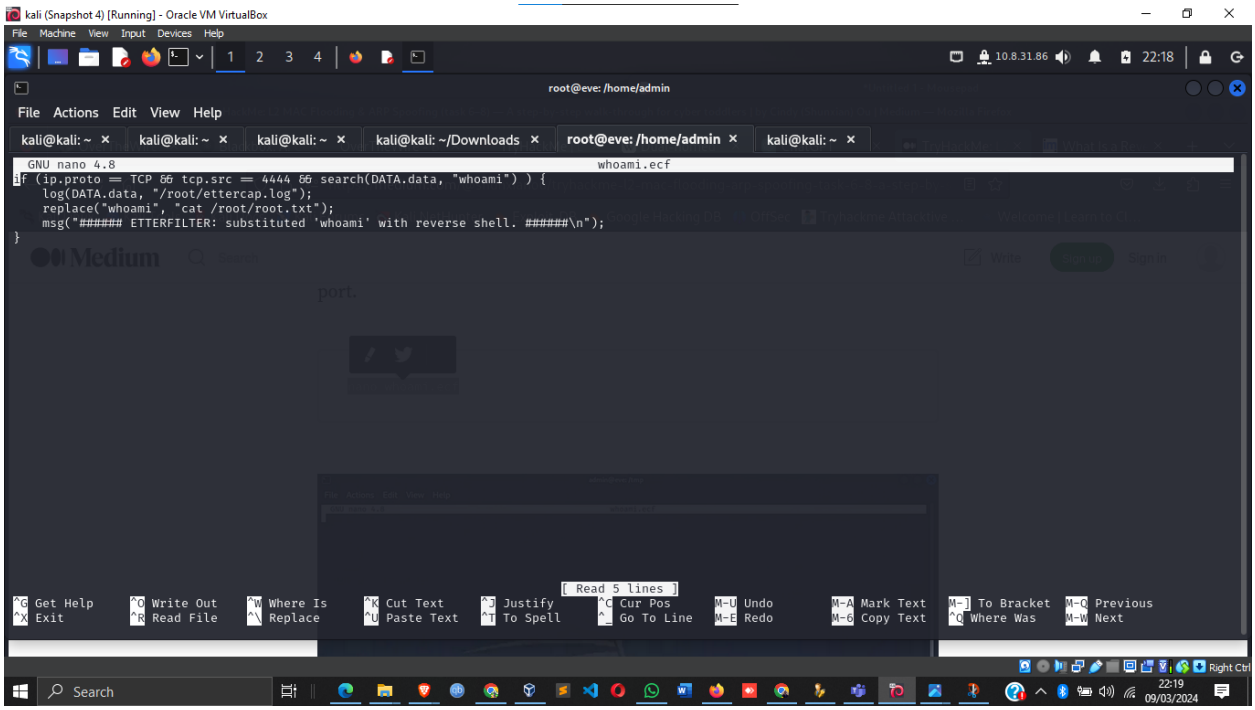
- **What commands are being executed? Answer in the order they are being executed:** whoami, pwd, ls

```
Sat Mar  9 18:47:43 2024 [945956]
TCP  192.168.12.10:4444 ⟶ 192.168.12.20:33030 | AP (3)
ls
```

```
Sat Mar  9 18:47:40 2024 [933754]
TCP  192.168.12.10:4444 ⟶ 192.168.12.20:33034 | AP (4)
pwd
```

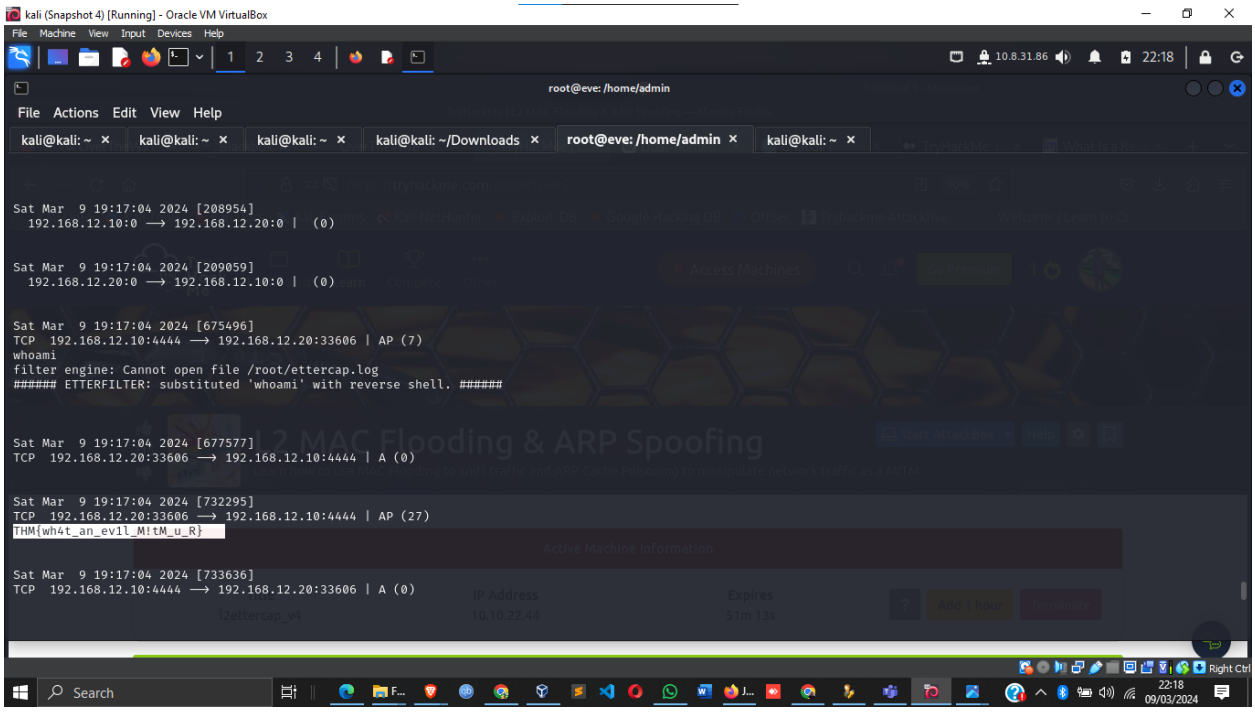
- **Which of the listed files do you want?** root.txt





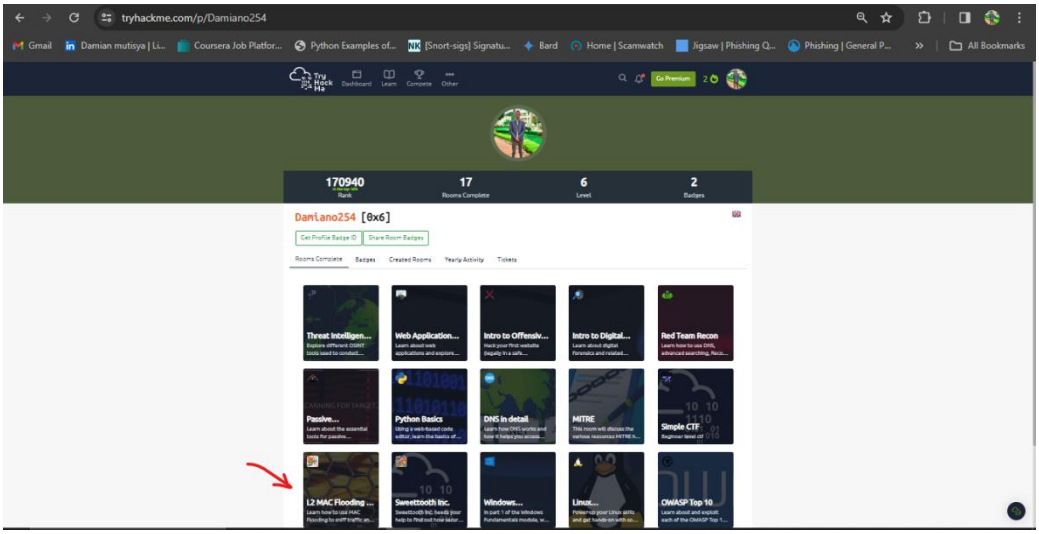
```
root@eve:/home/admin# nano whoami.ecf
root@eve:/home/admin# etterfilter whoami.ecf -o whoami.ef

etterfilter 0.8.3 copyright 2001-2019 Ettercap Development Team


 14 protocol tables loaded:
       DECODED DATA udp tcp esp gre icmp ipv6 ip arp wifi fddi tr eth

 13 constants loaded:
       VRRP OSPF GRE UDP TCP ESP ICMP6 ICMP PPTP PPPOE IP6 IP ARP

 Parsing source file 'whoami.ecf'  done.

 Unfolding the meta-tree  done.

 Converting labels to real offsets  done.

 Writing output to 'whoami.ef'  done.

 → Script encoded into 9 instructions.


root@eve:/home/admin# ls
USER:  whoami.ecf  whoami.ef
root@eve:/home/admin# ettercap -T -i eth1 -M arp -F whoami.ef
```

- **What is the root.txt flag?** THM{wh4t_an_ev1l_M!tM_u_R}



Conclusion

I successfully completed the L2 MAC Flooding & ARP Spoofing room on TryHackMe, mastering techniques like MAC flooding and ARP spoofing for network manipulation. Through practical exercises, I gained insights into network vulnerabilities and the importance of robust security measures. I accessed the target network, discovered live hosts, and obtained credentials to access restricted content. This experience emphasized the importance of proactive network security practices.


tryhackme.com/p/Damiano254
Gmail
Damian mutisya | Li...
Coursera Job Platfor...
Python Examples of...
[Snort-sigs] Signatu...
Bard
Home | Scamwatch
Jigsaw | Phishing Q...
Phishing | General P...
All Bookmarks
Dashboard
Learn
Compete
Other
Go Premium
170940
Rank
17
Rooms Complete
6
Level
2
Badges
DamIano254 [0x6]
Get Profile Badge ID
Share Room Badges
Rooms Complete
Badges
Created Rooms
Yearly Activity
Tickets
Threat Intelligen...
Web Application...
Intro to Offensiv...
Intro to Digital...
Red Team Recon
Passive...
Python Basics
DNS in detail
MITRE
Simple CTF
L2 MAC Flooding ...
Sweettooth Inc.
Windows...
Linux...
OWASP Top 10