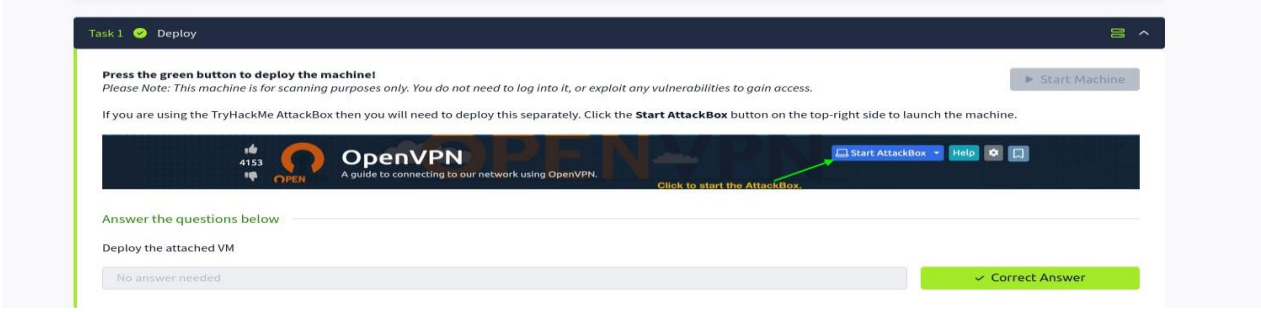
CTF Writeup: Further Nmap

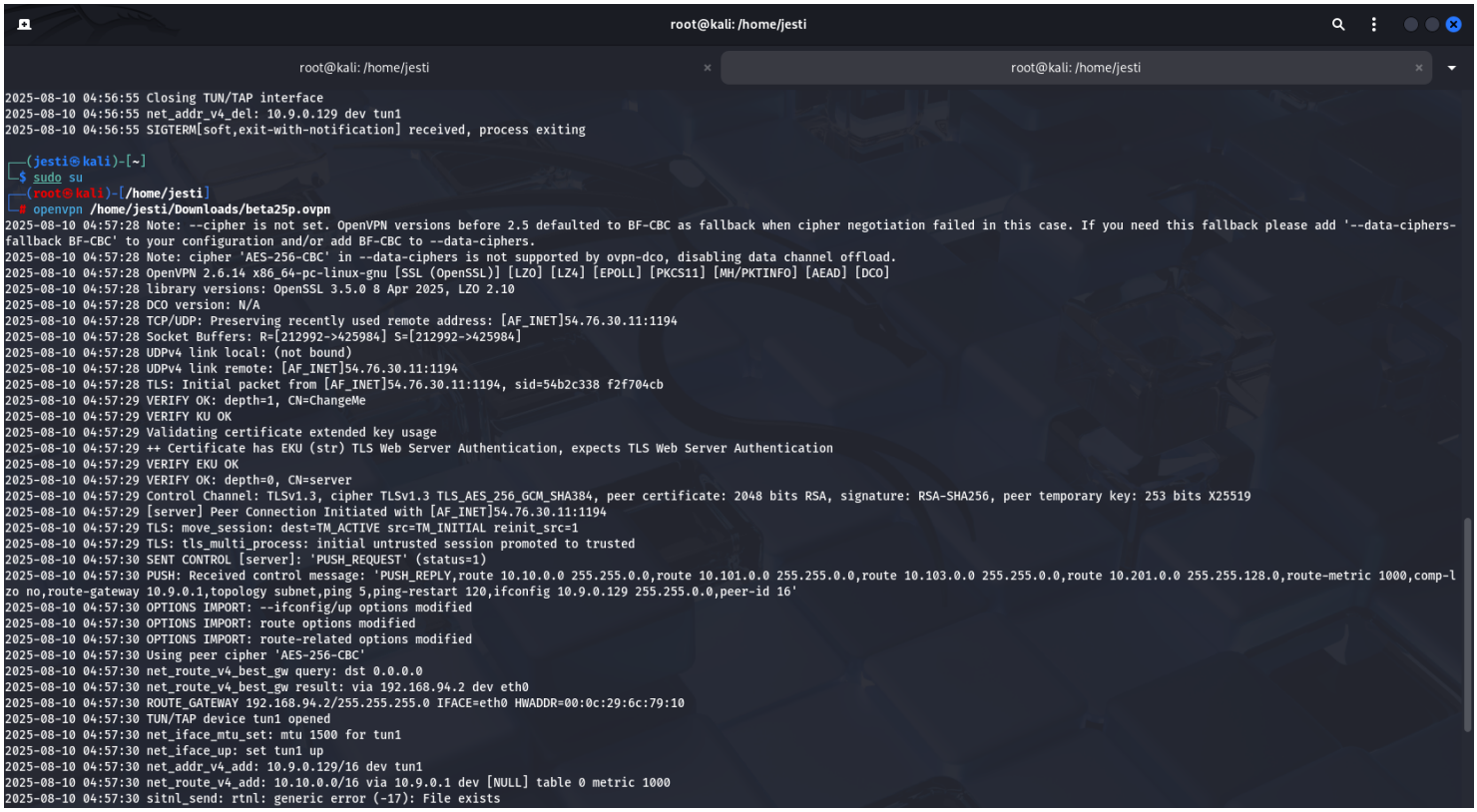
Prepared By: Jestine Thomas Mathew

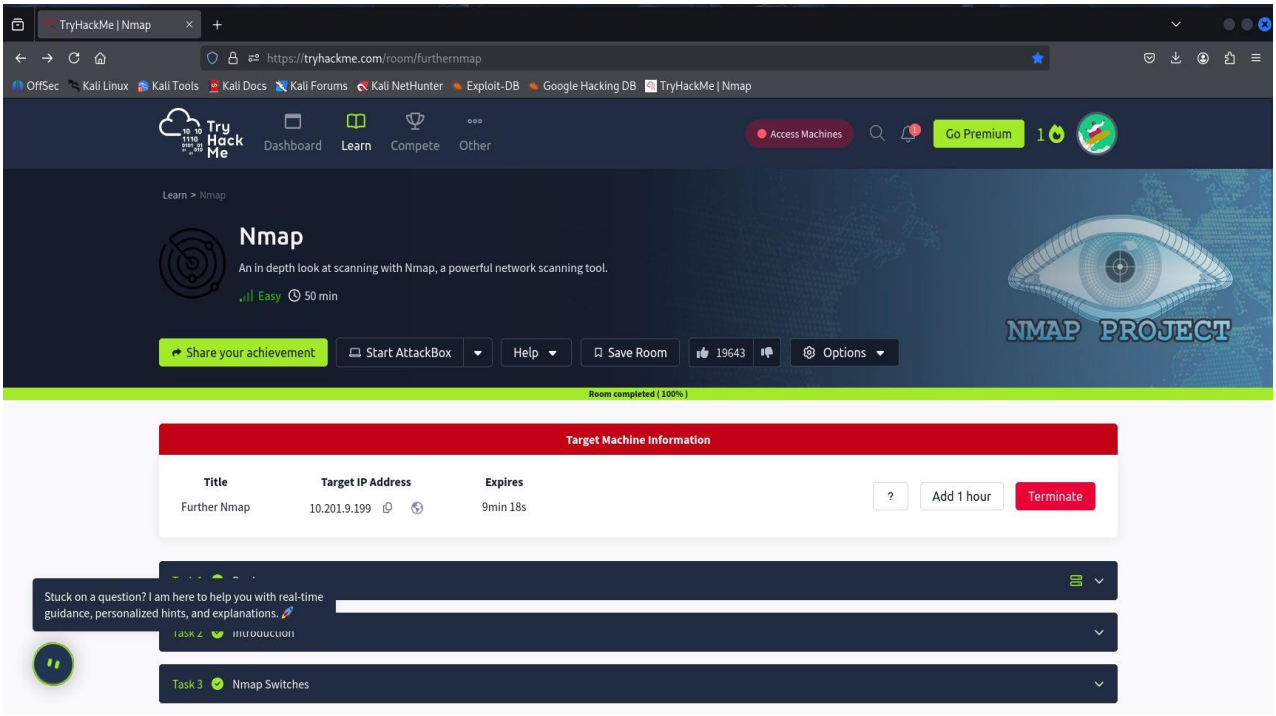
This writeup details the steps taken to solve the 'Further Nmap' room on TryHackMe. This room is categorized as 'Easy' and focuses on advanced Nmap scanning techniques, including host discovery, port scanning, firewall evasion, and the Nmap Scripting Engine (NSE).

1. Introduction & Connecting to the Network

First, I need to connect to the TryHackMe network to access the target machine. This is done by deploying the machine in the room and connecting to the TryHackMe VPN, for example using OpenVPN. Once connected, the target machine's IP address is provided. For this writeup, the target IP is `10.201.9.199`.

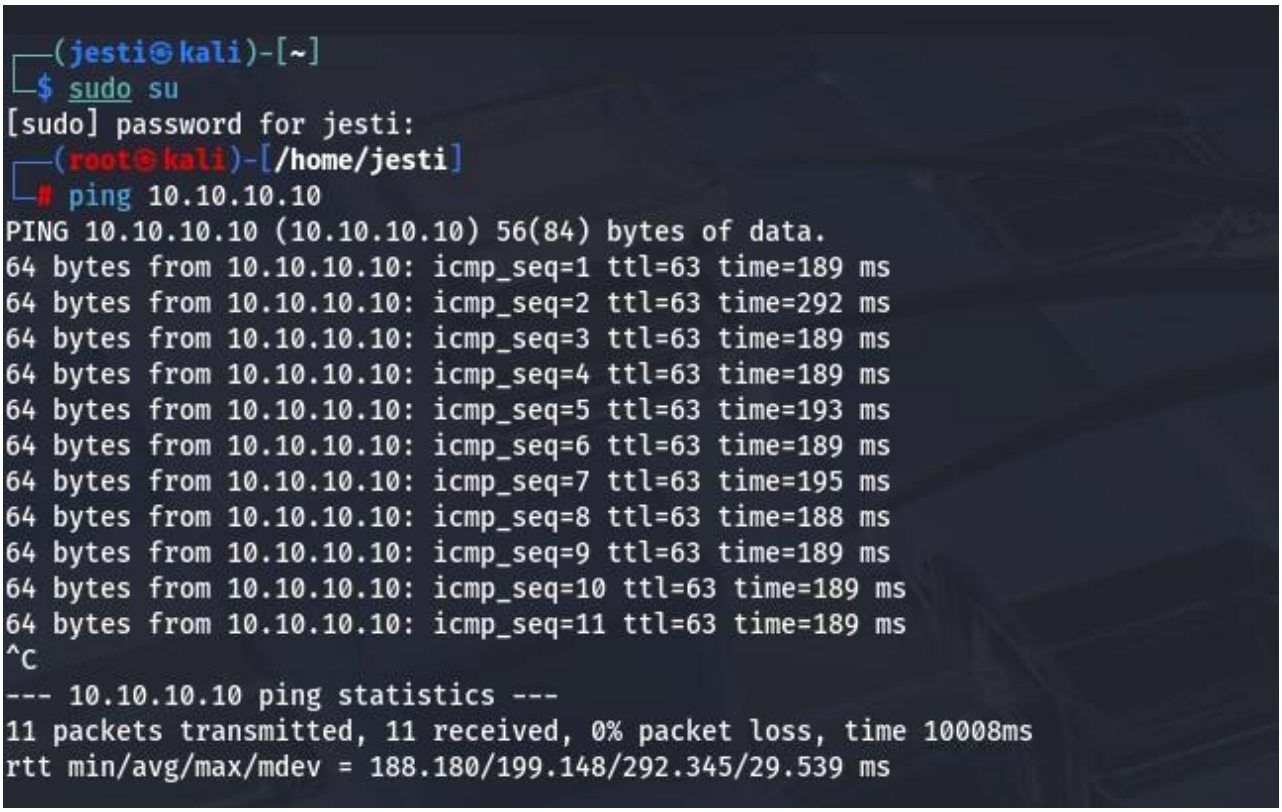






To confirm connectivity, a simple `ping` command can be used.

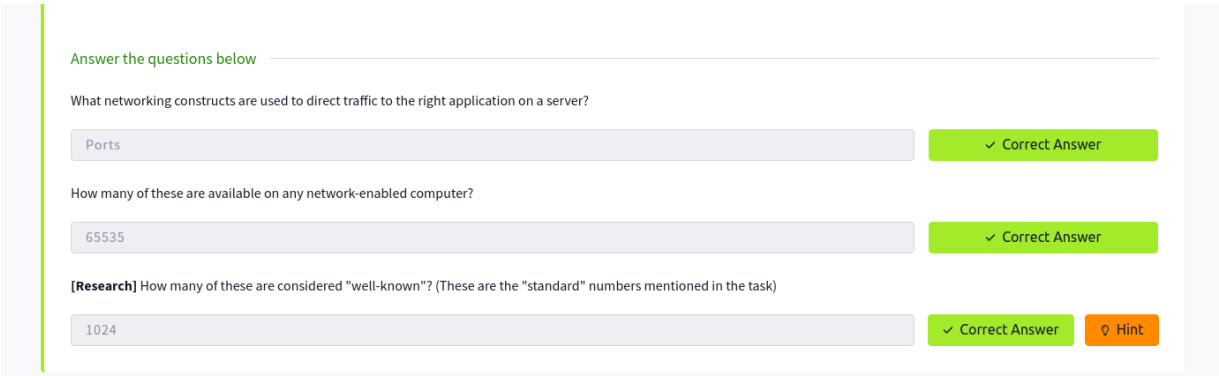
```
ping 10.201.9.199
```



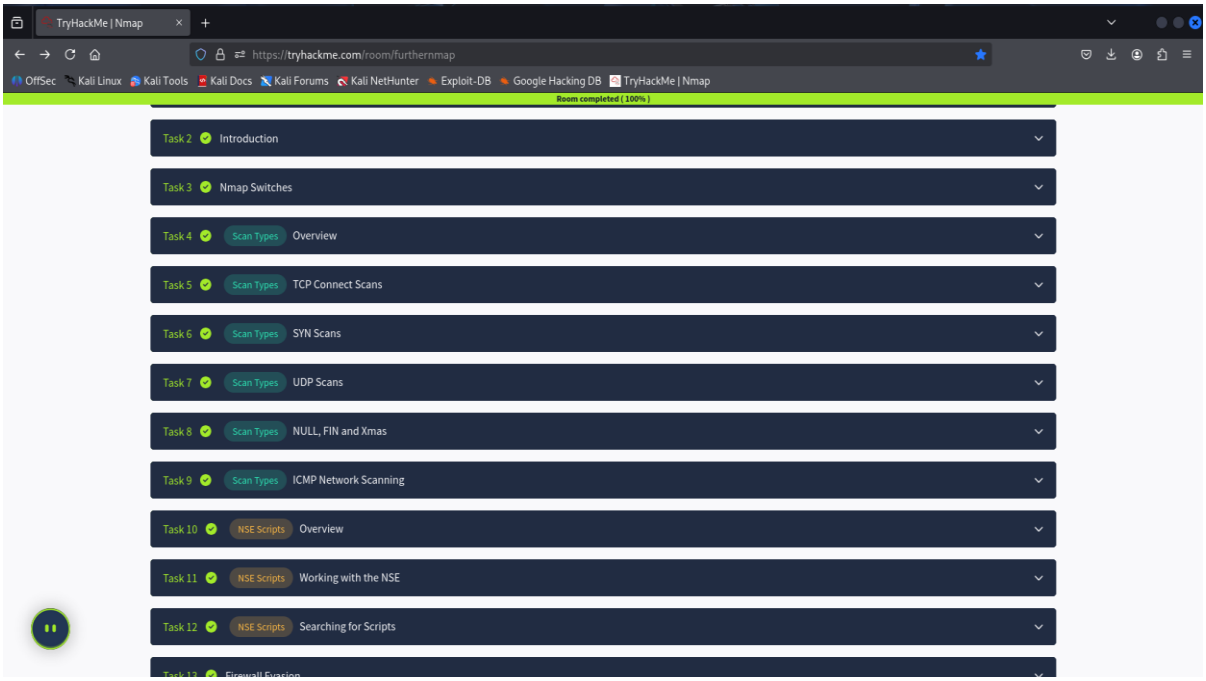
2. Answering the Initial Questions

The initial tasks involve answering theoretical questions about Nmap and networking.

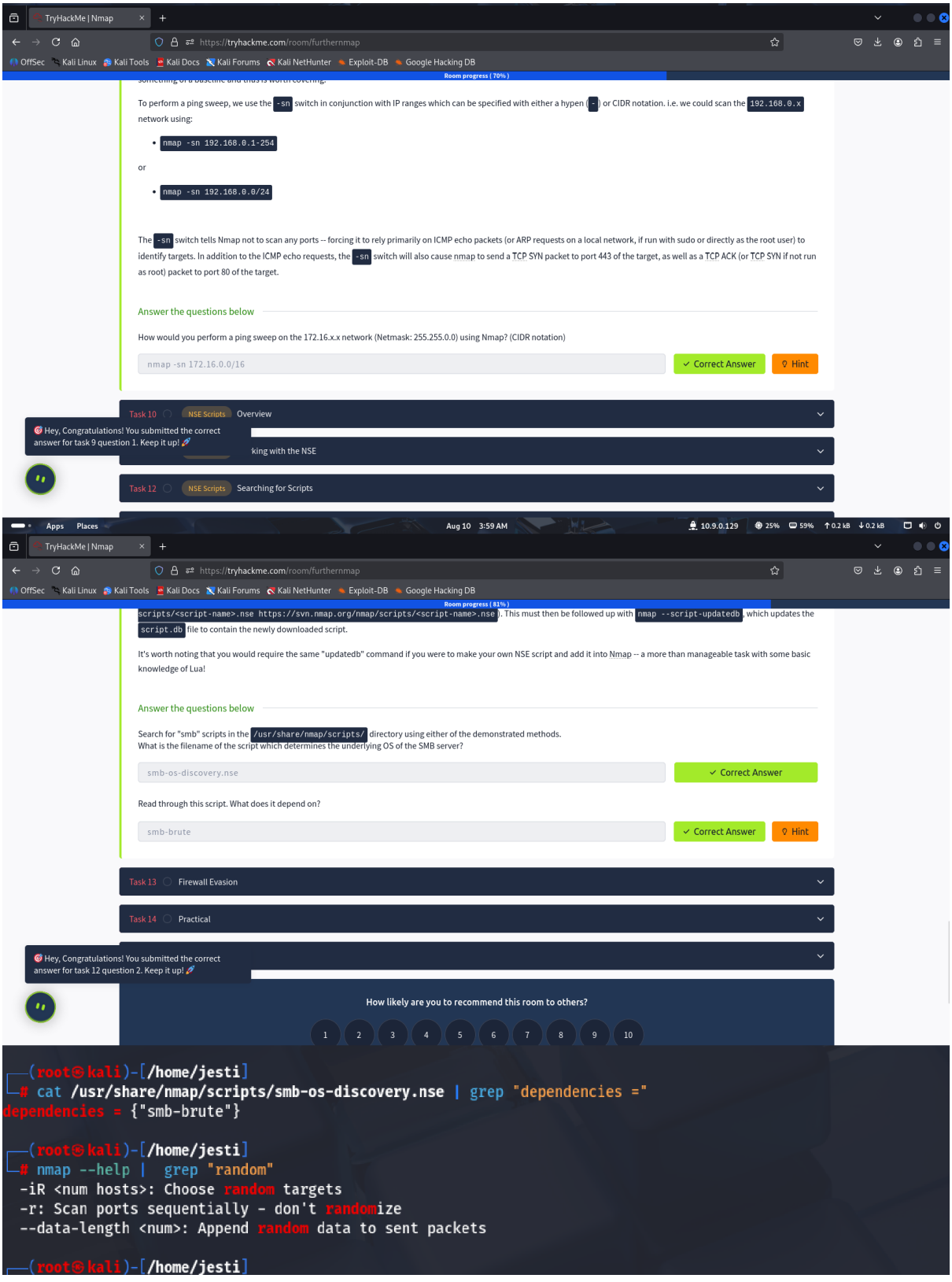
- **Task 2 & 3:** These tasks cover basic networking concepts like ports and the different Nmap switches.



- **Task 4-9:** These tasks go over different scan types like TCP Connect, SYN, UDP, NULL, FIN, and Xmas scans, and ICMP network scanning.

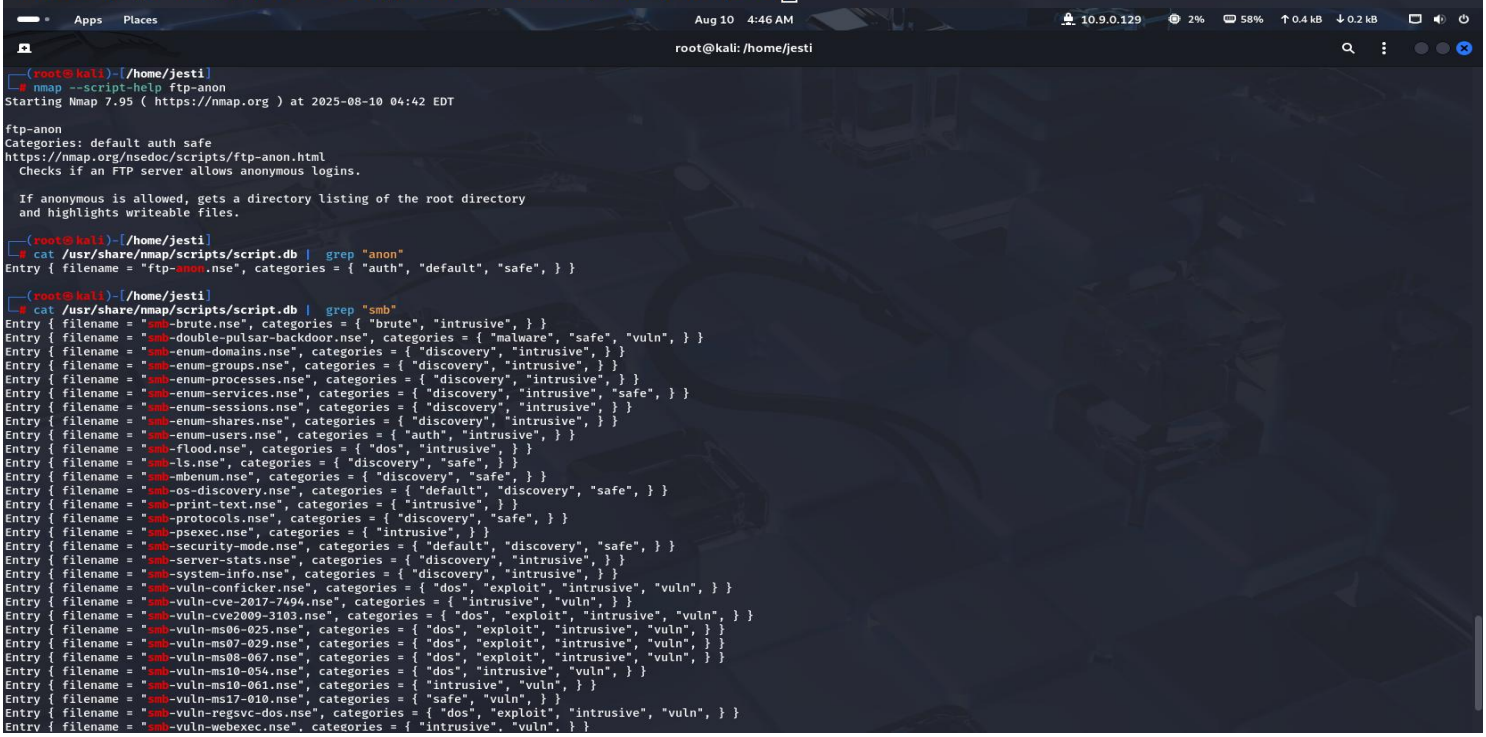


- **T6sk 10-13:** These tasks introduce the Nmap Scripting Engine (NSE), how to work with scripts, and firewall evasion techniques.




```
┌──(root㉿kali)-[/home/jesti]
└─# cat /usr/share/nmap/scripts/smb-os-discovery.nse | grep "dependencies ="
dependencies = {"smb-brute"}

┌──(root㉿kali)-[/home/jesti]
└─# nmap --help |  grep "random"
  -iR <num hosts>: Choose random targets
  -r: Scan ports sequentially - don't randomize
  --data-length <num>: Append random data to sent packets

┌──(root㉿kali)-[/home/jesti]
```

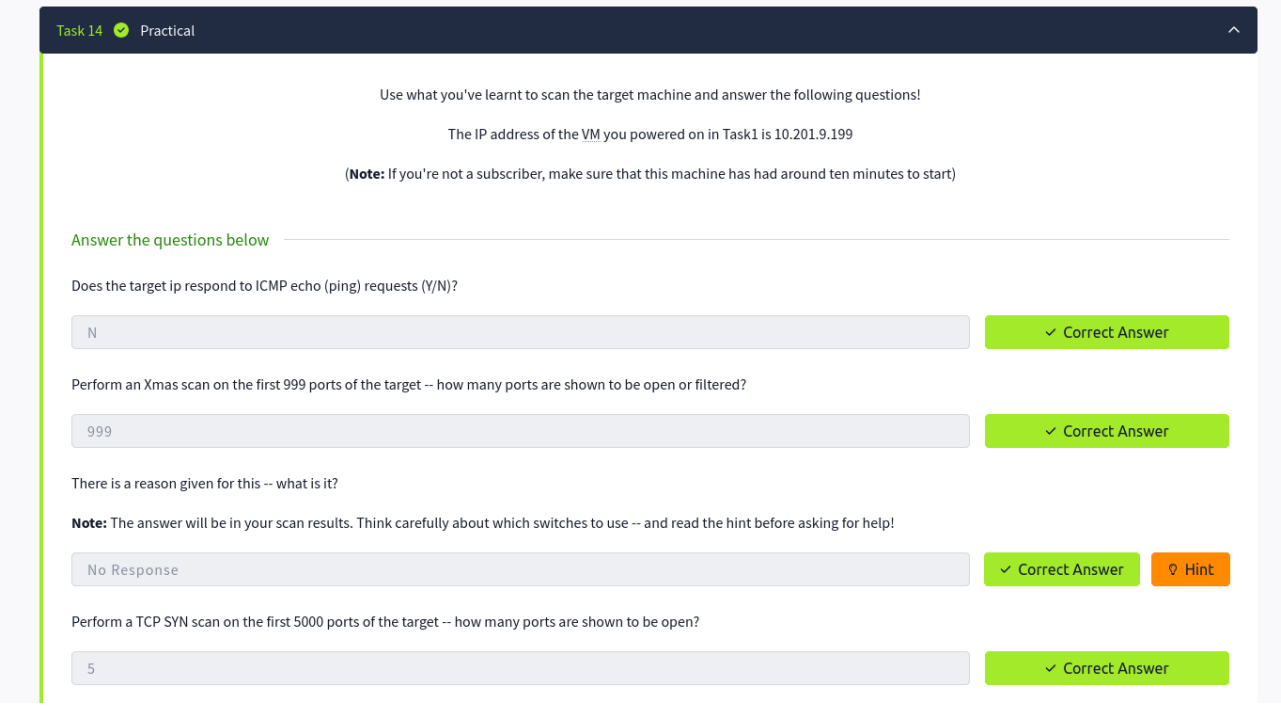
```
  -PO[protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
  --traceroute: Trace hop path to each host
SCAN TECHNIQUES:
  -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
  -sU: UDP Scan
  -sN/sF/sX: TCP Null, FIN, and Xmas scans
  --scanflags <flags>: Customize TCP scan flags
  -sI <zombie host[:probeport]>: Idle scan
  -sY/sZ: SCTP INIT/COOKIE-ECHO scans
  -sO: IP protocol scan
  -b <FTP relay host>: FTP bounce scan
PORT SPECIFICATION AND SCAN ORDER:
  -p <port ranges>: Only scan specified ports
    Ex: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080,S:9
  --exclude-ports <port ranges>: Exclude the specified ports from scanning
  -F: Fast mode - Scan fewer ports than the default scan
  -r: Scan ports sequentially - don't randomize
  --top-ports <number>: Scan <number> most common ports
  --port-ratio <ratio>: Scan ports more common than <ratio>
SERVICE/VERSION DETECTION:
  -sV: Probe open ports to determine service/version info
  --version-intensity <level>: Set from 0 (light) to 9 (try all probes)
  --version-light: Limit to most likely probes (intensity 2)
  --version-all: Try every single probe (intensity 9)
  --version-trace: Show detailed version scan activity (for debugging)
SCRIPT SCAN:
  -sC: equivalent to --script=default
  --script=<Lua scripts>: <Lua scripts> is a comma separated list of
           directories, script-files or script-categories
  --script-args=<n1=v1,[n2=v2,...]>: provide arguments to scripts
  --script-args-file=filename: provide NSE script args in a file
  --script-trace: Show all data sent and received
  --script-updatedb: Update the script database.
  --script-help=<Lua scripts>: Show help about scripts.
           <Lua scripts> is a comma-separated list of script-files or
           script-categories.
OS DETECTION:
  -O: Enable OS detection
  --osscan-limit: Limit OS detection to promising targets
  --osscan-guess: Guess OS more aggressively
TIMING AND PERFORMANCE:
  Options which take <time> are in seconds, or append 'ms' (milliseconds),
  's' (seconds), 'm' (minutes), or 'h' (hours) to the value (e.g. 30m).
Manual page nmap(1) line 81 (press h for help or q to quit)
```



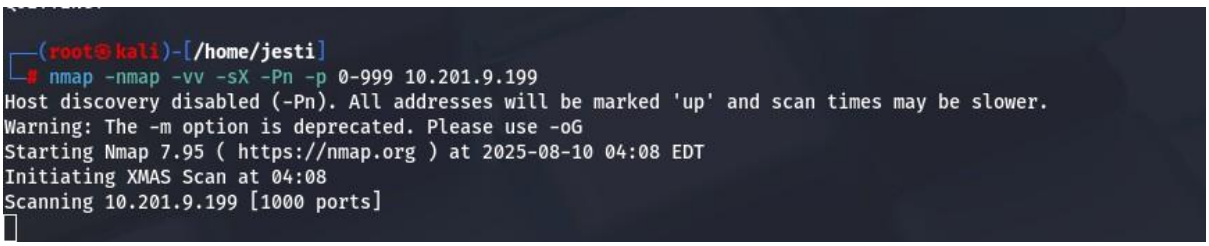
```
```

3. Practical Scanning and Exploitation

Task 14 is the practical part of the room, where I use the learned Nmap techniques to scan the deployed machine and answer a series of questions.



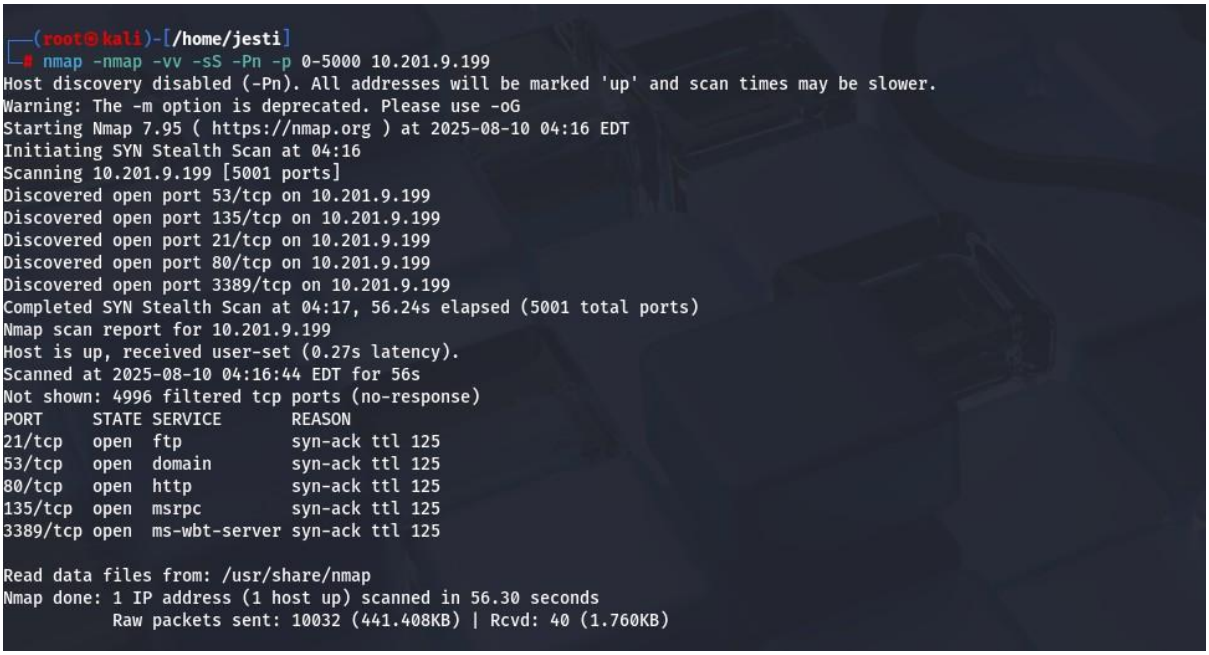
- **Does the target respond to ICMP echo (ping) requests?** The answer is **N** (No), which I can determine because a standard ping fails, or by using `nmap -Pn` to assume the host is up.
- **Perform an Xmas scan on the first 999 ports. How many ports are shown to be open or filtered?** The command `nmap -sX -p 1-999 10.201.9.199` is used. The result shows **999** ports are `open|filtered`.



```
┌──(root㉿kali)-[/home/jesti]
└─# nmap -nmap -vv -sX -Pn -p 0-999 10.201.9.199
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slower.
Warning: The -m option is deprecated. Please use -oG
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-10 04:08 EDT
Initiating XMAS Scan at 04:08
Scanning 10.201.9.199 [1000 ports]
XMAS Scan Timing: About 15.50% done; ETC: 04:12 (0:02:49 remaining)
XMAS Scan Timing: About 30.50% done; ETC: 04:12 (0:02:19 remaining)
XMAS Scan Timing: About 45.00% done; ETC: 04:12 (0:01:51 remaining)
XMAS Scan Timing: About 60.00% done; ETC: 04:12 (0:01:21 remaining)
XMAS Scan Timing: About 75.00% done; ETC: 04:12 (0:00:50 remaining)
Completed XMAS Scan at 04:12, 201.33s elapsed (1000 total ports)
Nmap scan report for 10.201.9.199
Host is up, received user-set.
Scanned at 2025-08-10 04:08:57 EDT for 202s
All 1000 scanned ports on 10.201.9.199 are in ignored states.
Not shown: 1000 open|filtered tcp ports (no-response)

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 201.38 seconds
           Raw packets sent: 2000 (80.000KB) | Rcvd: 0 (0B)

┌──(root㉿kali)-[/home/jesti]
└─#

┌──(root㉿kali)-[/home/jesti]
└─# nmap -nmap -vv -sX -Pn -p 0-999 10.201.9.199
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slower.
Warning: The -m option is deprecated. Please use -oG
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-10 04:08 EDT
Initiating XMAS Scan at 04:08
Scanning 10.201.9.199 [1000 ports]
XMAS Scan Timing: About 15.50% done; ETC: 04:12 (0:02:49 remaining)
XMAS Scan Timing: About 30.50% done; ETC: 04:12 (0:02:19 remaining)
XMAS Scan Timing: About 45.00% done; ETC: 04:12 (0:01:51 remaining)
XMAS Scan Timing: About 60.00% done; ETC: 04:12 (0:01:21 remaining)
XMAS Scan Timing: About 75.00% done; ETC: 04:12 (0:00:50 remaining)
Completed XMAS Scan at 04:12, 201.33s elapsed (1000 total ports)
Nmap scan report for 10.201.9.199
Host is up, received user-set.
Scanned at 2025-08-10 04:08:57 EDT for 202s
All 1000 scanned ports on 10.201.9.199 are in ignored states.
Not shown: 1000 open|filtered tcp ports (no-response)

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 201.38 seconds
           Raw packets sent: 2000 (80.000KB) | Rcvd: 0 (0B)
```

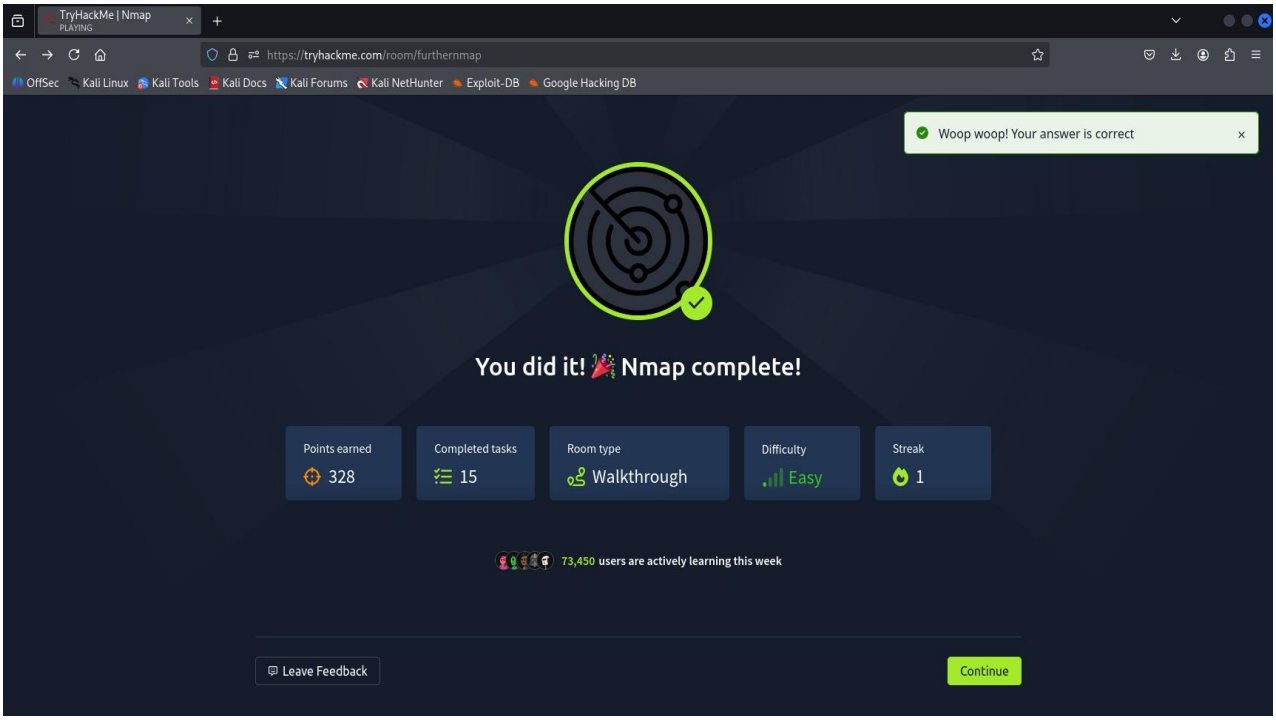
- **There is a reason given for this – what is it?** The reason given in the Nmap output is **no-response**.
- **Perform a TCP SYN scan on the first 5000 ports. How many ports are shown to be open?** Using the command `nmap -sS -p 1-5000 -Pn 10.201.9.199`, I found **5** open ports.

```
┌──(root㉿kali)-[/home/jesti]
└─# nmap -nmap -vv -sS -Pn -p 0-5000 10.201.9.199
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slower.
Warning: The -m option is deprecated. Please use -oG
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-10 04:14 EDT
Initiating SYN Stealth Scan at 04:14
Scanning 10.201.9.199 [5001 ports]
Discovered open port 21/tcp on 10.201.9.199
Discovered open port 53/tcp on 10.201.9.199
Discovered open port 135/tcp on 10.201.9.199
Discovered open port 80/tcp on 10.201.9.199
Discovered open port 3389/tcp on 10.201.9.199
```



4. Conclusion

The "Further Nmap" room provides a solid, hands-on understanding of Nmap's more advanced capabilities. It effectively demonstrates how different scan types can be used to bypass firewall rules and how to leverage the Nmap Scripting Engine for deeper enumeration. The practical challenge reinforces the importance of choosing the right Nmap switches for a given scenario.


TryHackMe | Nmap
PLAYING
https://tryhackme.com/room/furthernmap
OffSec
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
Woop woop! Your answer is correct
You did it! Nmap complete!
Points earned
328
Completed tasks
15
Room type
Walkthrough
Difficulty
Easy
Streak
1
73,450 users are actively learning this week
Leave Feedback
Continue