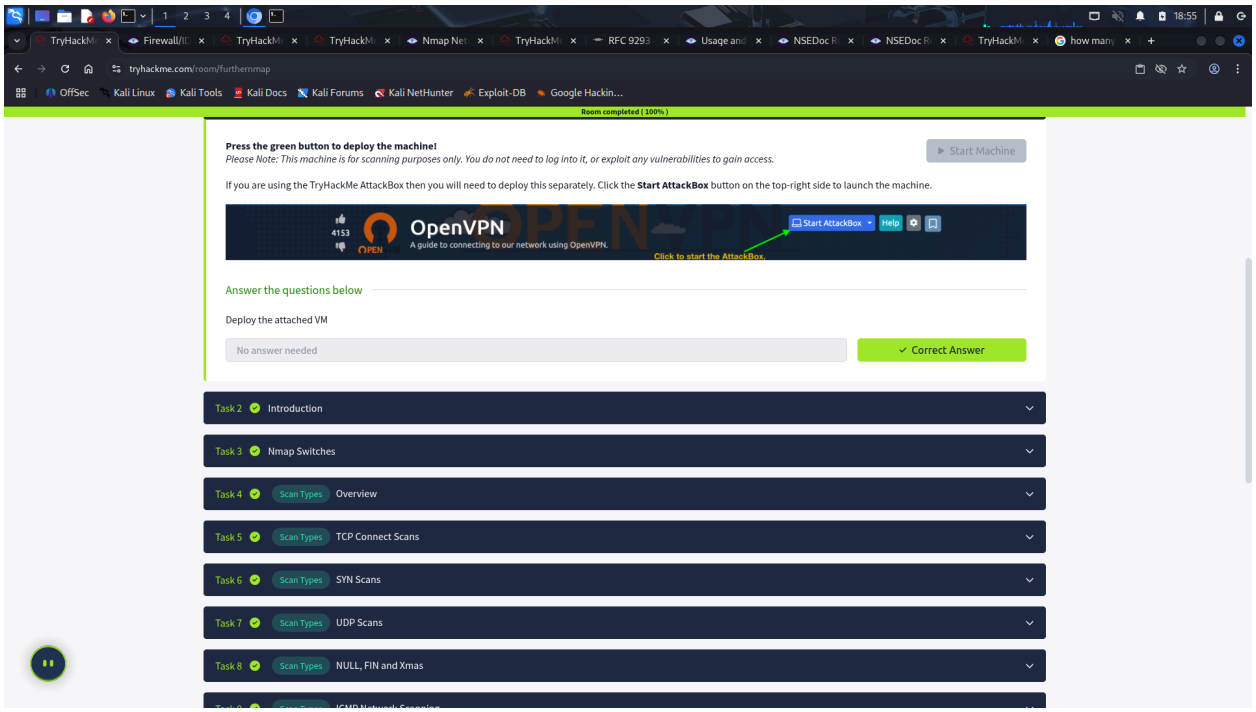
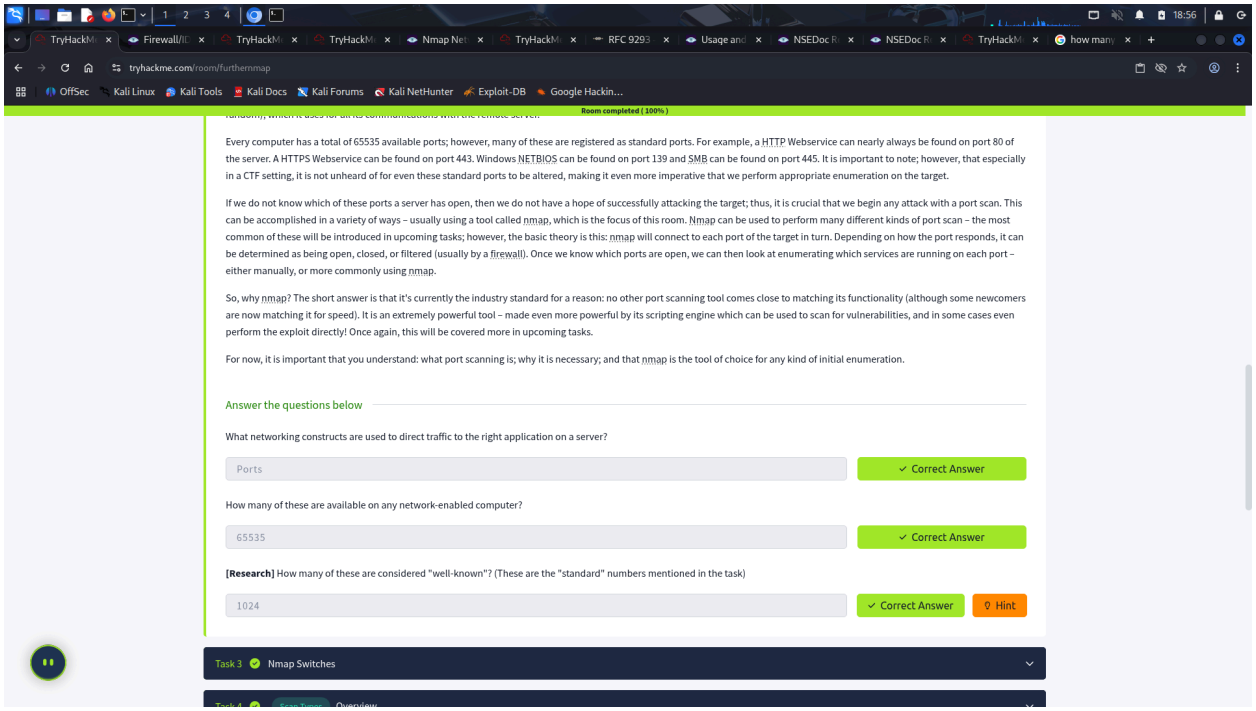
NMAP TryHackMe

Name: Nandakumar K.S
Tryhackme username: nandu682015
Date: 10th August 2025

Task 1: deploy the machine



Task 2: Introduction



Every computer has a total of 65535 available ports; however, many of these are registered as standard ports. For example, a HTTP Webservice can nearly always be found on port 80 of the server. A HTTPS Webservice can be found on port 443. Windows NETBIOS can be found on port 139 and SMB can be found on port 445. It is important to note; however, that especially in a CTF setting, it is not unheard of for even these standard ports to be altered, making it even more imperative that we perform appropriate enumeration on the target.

If we do not know which of these ports a server has open, then we do not have a hope of successfully attacking the target; thus, it is crucial that we begin any attack with a port scan. This can be accomplished in a variety of ways – usually using a tool called nmap, which is the focus of this room. Nmap can be used to perform many different kinds of port scan – the most common of these will be introduced in upcoming tasks; however, the basic theory is this: nmap will connect to each port of the target in turn. Depending on how the port responds, it can be determined as being open, closed, or filtered (usually by a firewall). Once we know which ports are open, we can then look at enumerating which services are running on each port – either manually, or more commonly using nmap.

So, why nmap? The short answer is that it's currently the industry standard for a reason: no other port scanning tool comes close to matching its functionality (although some newcomers are now matching it for speed). It is an extremely powerful tool – made even more powerful by its scripting engine which can be used to scan for vulnerabilities, and in some cases even perform the exploit directly! Once again, this will be covered more in upcoming tasks.

For now, it is important that you understand: what port scanning is; why it is necessary; and that nmap is the tool of choice for any kind of initial enumeration.

Answer the questions below

What networking constructs are used to direct traffic to the right application on a server?

Ports — ✓ Correct Answer

How many of these are available on any network-enabled computer?

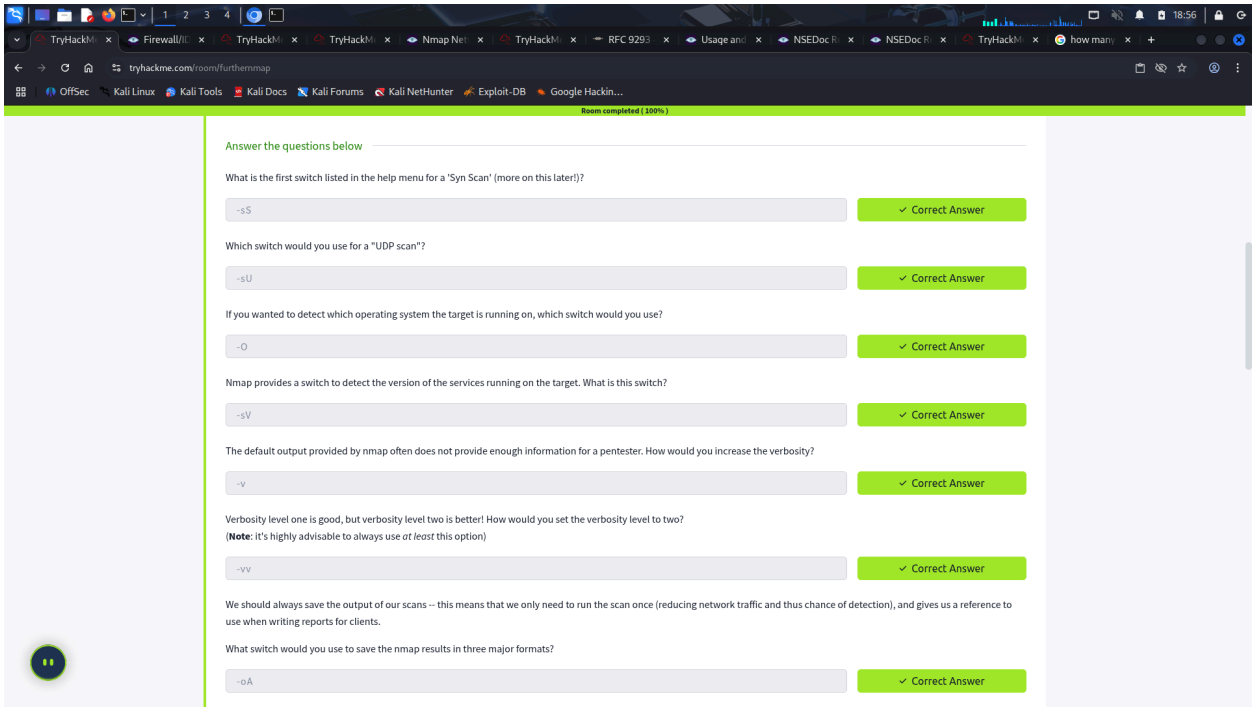
65535 — ✓ Correct Answer

[Research] How many of these are considered "well-known"? (These are the "standard" numbers mentioned in the task)

1024 — ✓ Correct Answer — Hint

Task 3 Nmap Switches

Task 3: Nmap Switches



Answer the questions below

What is the first switch listed in the help menu for a 'Syn Scan' (more on this later!)?

-sS — ✓ Correct Answer

Which switch would you use for a "UDP scan"?

-sU — ✓ Correct Answer

If you wanted to detect which operating system the target is running on, which switch would you use?

-O — ✓ Correct Answer

Nmap provides a switch to detect the version of the services running on the target. What is this switch?

-sV — ✓ Correct Answer

The default output provided by nmap often does not provide enough information for a pentester. How would you increase the verbosity?

-v — ✓ Correct Answer

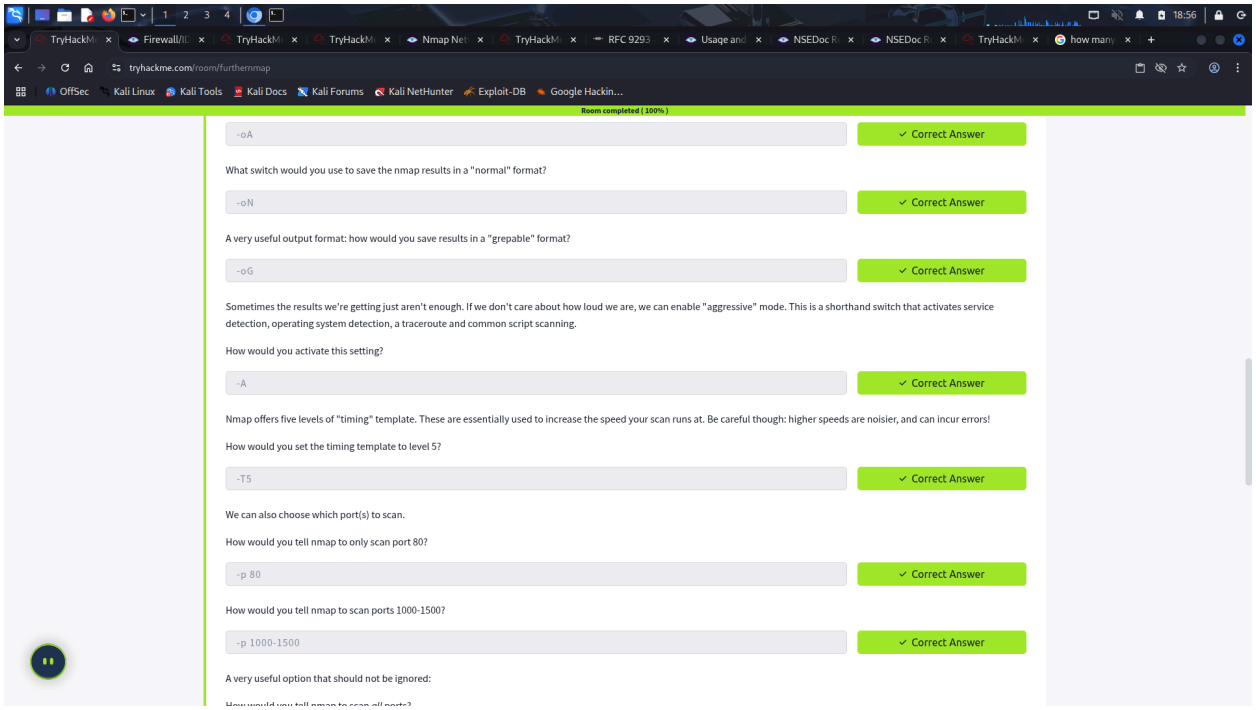
Verbosity level one is good, but verbosity level two is better! How would you set the verbosity level to two?
(**Note**: it's highly advisable to always use *at least* this option)

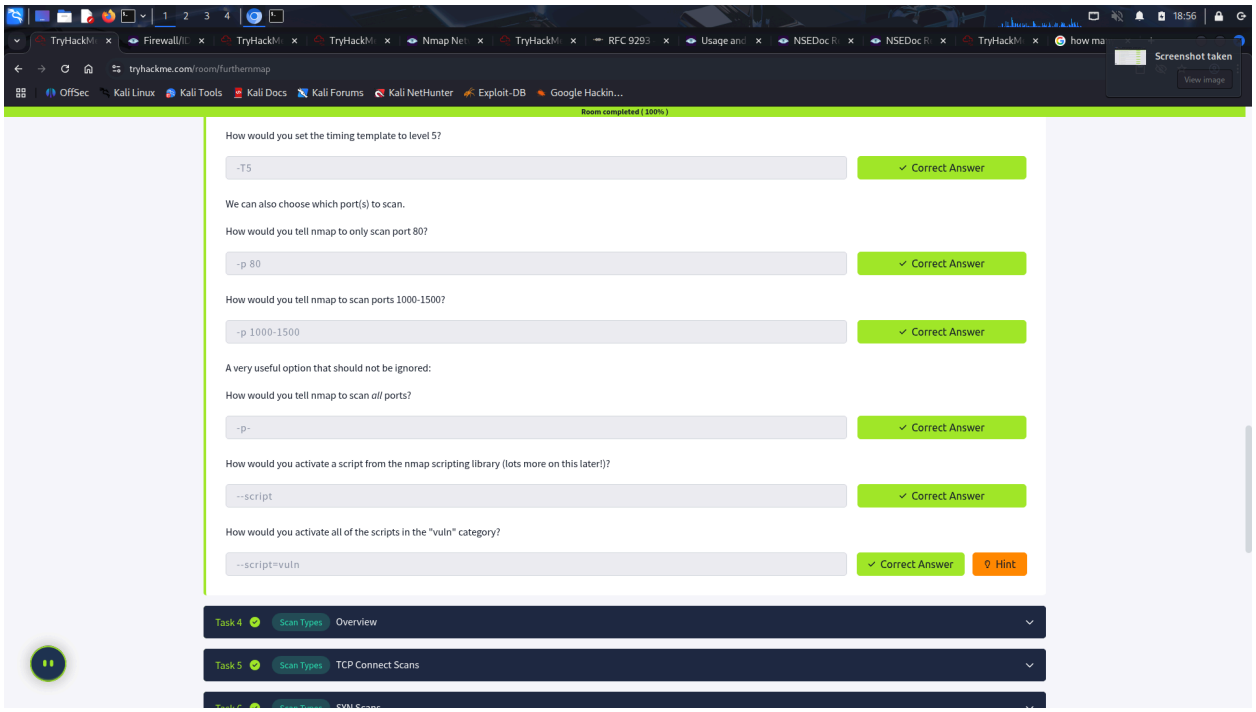
-vv — ✓ Correct Answer

We should always save the output of our scans -- this means that we only need to run the scan once (reducing network traffic and thus chance of detection), and gives us a reference to use when writing reports for clients.

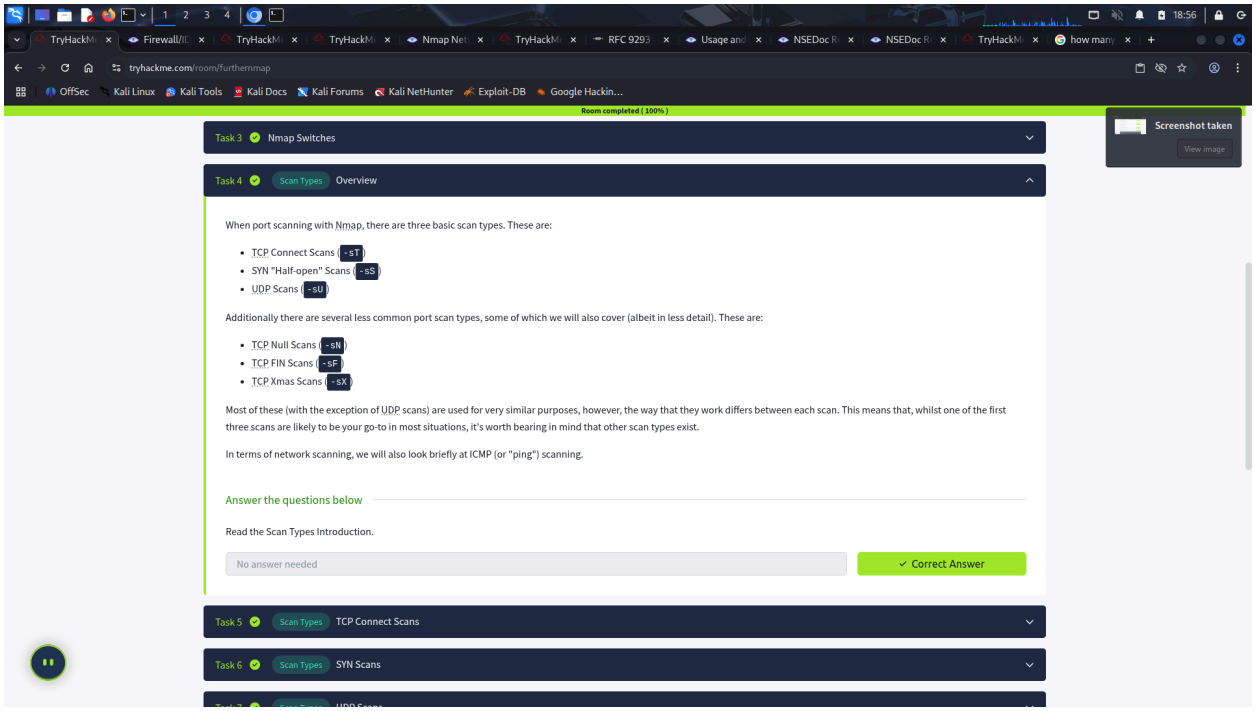
What switch would you use to save the nmap results in three major formats?

-oA — ✓ Correct Answer

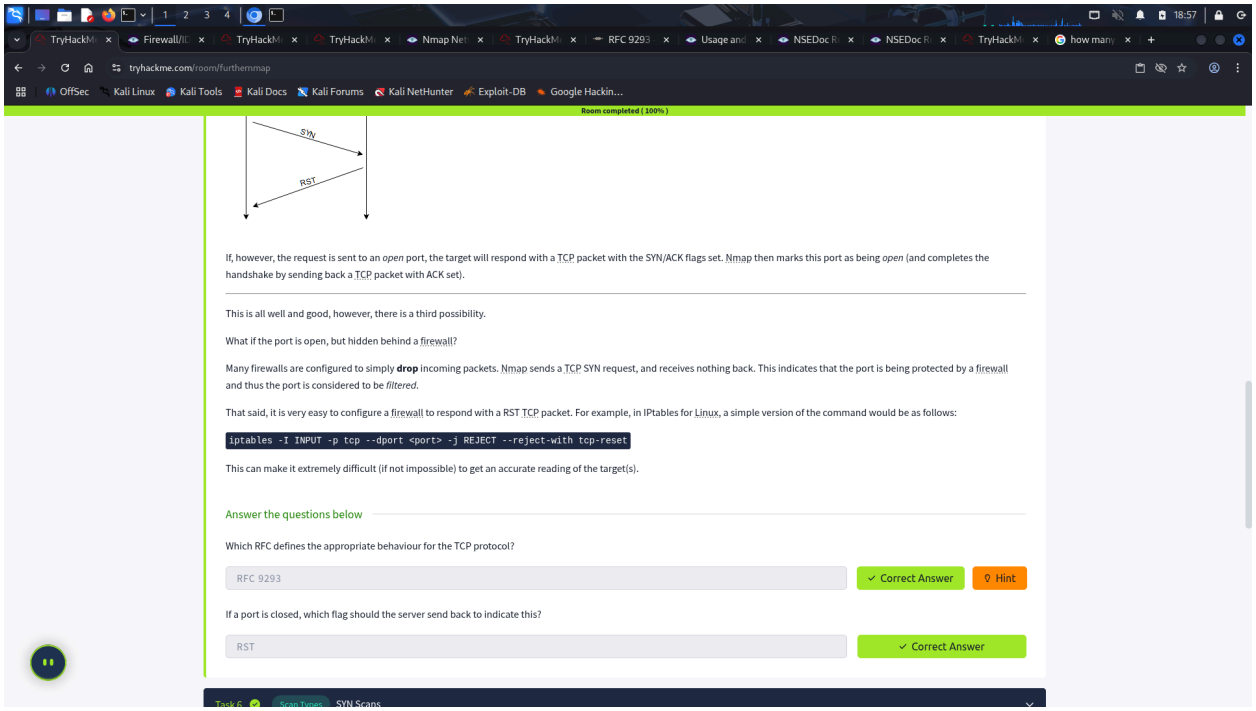

tryhackme.com/room/furthernmap
Room completed (100%)
-oA
Correct Answer
What switch would you use to save the nmap results in a "normal" format?
-oN
Correct Answer
A very useful output format: how would you save results in a "grepable" format?
-oG
Correct Answer
Sometimes the results we're getting just aren't enough. If we don't care about how loud we are, we can enable "aggressive" mode. This is a shorthand switch that activates service detection, operating system detection, a traceroute and common script scanning.
How would you activate this setting?
-A
Correct Answer
Nmap offers five levels of "timing" template. These are essentially used to increase the speed your scan runs at. Be careful though: higher speeds are noisier, and can incur errors!
How would you set the timing template to level 5?
-T5
Correct Answer
We can also choose which port(s) to scan.
How would you tell nmap to only scan port 80?
-p 80
Correct Answer
How would you tell nmap to scan ports 1000-1500?
-p 1000-1500
Correct Answer
A very useful option that should not be ignored:


tryhackme.com/room/furthernmap
Screenshot taken
Room completed (100%)
How would you set the timing template to level 5?
-T5
Correct Answer
We can also choose which port(s) to scan.
How would you tell nmap to only scan port 80?
-p 80
Correct Answer
How would you tell nmap to scan ports 1000-1500?
-p 1000-1500
Correct Answer
A very useful option that should not be ignored:
How would you tell nmap to scan all ports?
-p-
Correct Answer
How would you activate a script from the nmap scripting library (lots more on this later!)?
--script
Correct Answer
How would you activate all of the scripts in the "vuln" category?
--script=vuln
Correct Answer
Hint
Task 4
Scan Types
Overview
Task 5
Scan Types
TCP Connect Scans

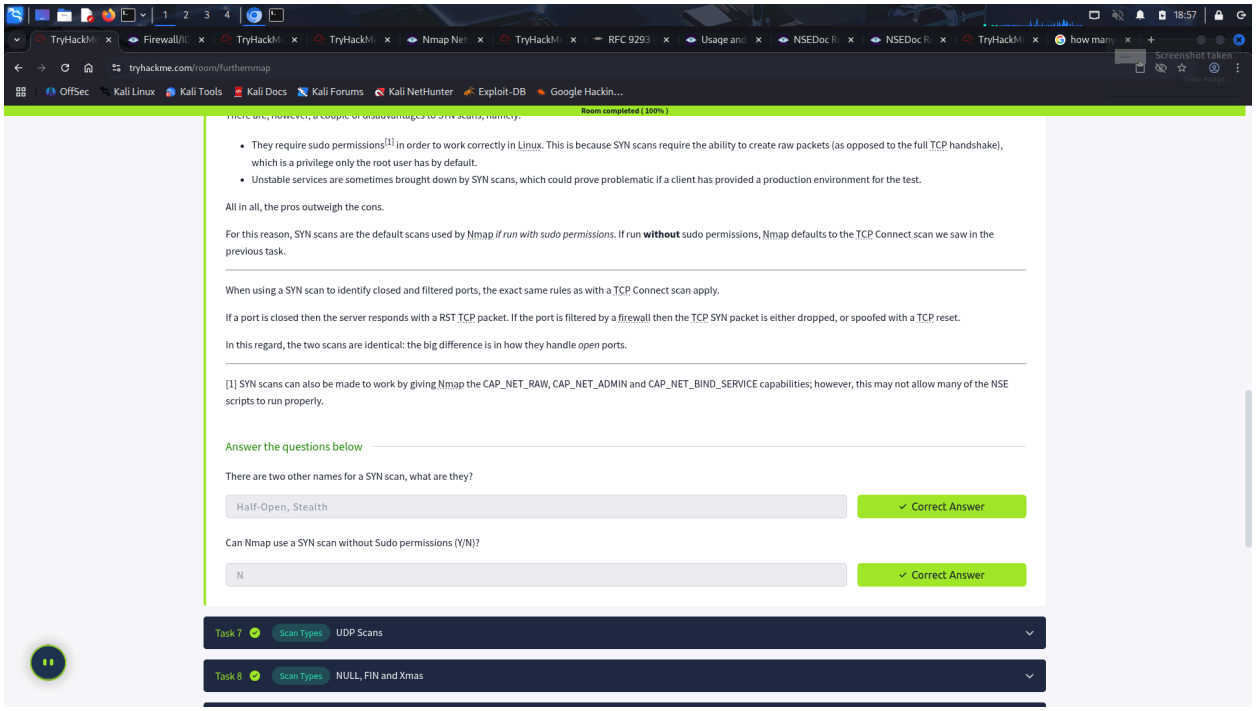
Task 4: Scan Types - Overview



Task 5: Scan Types - TCP Connect Scan



Task 6: Scan Types - Syn Scans



- They require sudo permissions[1] in order to work correctly in Linux. This is because SYN scans require the ability to create raw packets (as opposed to the full TCP handshake), which is a privilege only the root user has by default.
- Unstable services are sometimes brought down by SYN scans, which could prove problematic if a client has provided a production environment for the test.

All in all, the pros outweigh the cons.

For this reason, SYN scans are the default scans used by Nmap *if run with sudo permissions*. If run **without** sudo permissions, Nmap defaults to the TCP Connect scan we saw in the previous task.

When using a SYN scan to identify closed and filtered ports, the exact same rules as with a TCP Connect scan apply.

If a port is closed then the server responds with a RST TCP packet. If the port is filtered by a firewall then the TCP SYN packet is either dropped, or spoofed with a TCP reset.

In this regard, the two scans are identical: the big difference is in how they handle *open* ports.

---

[1] SYN scans can also be made to work by giving Nmap the CAP_NET_RAW, CAP_NET_ADMIN and CAP_NET_BIND_SERVICE capabilities; however, this may not allow many of the NSE scripts to run properly.

Answer the questions below

There are two other names for a SYN scan, what are they?

Half-Open, Stealth — ✓ Correct Answer

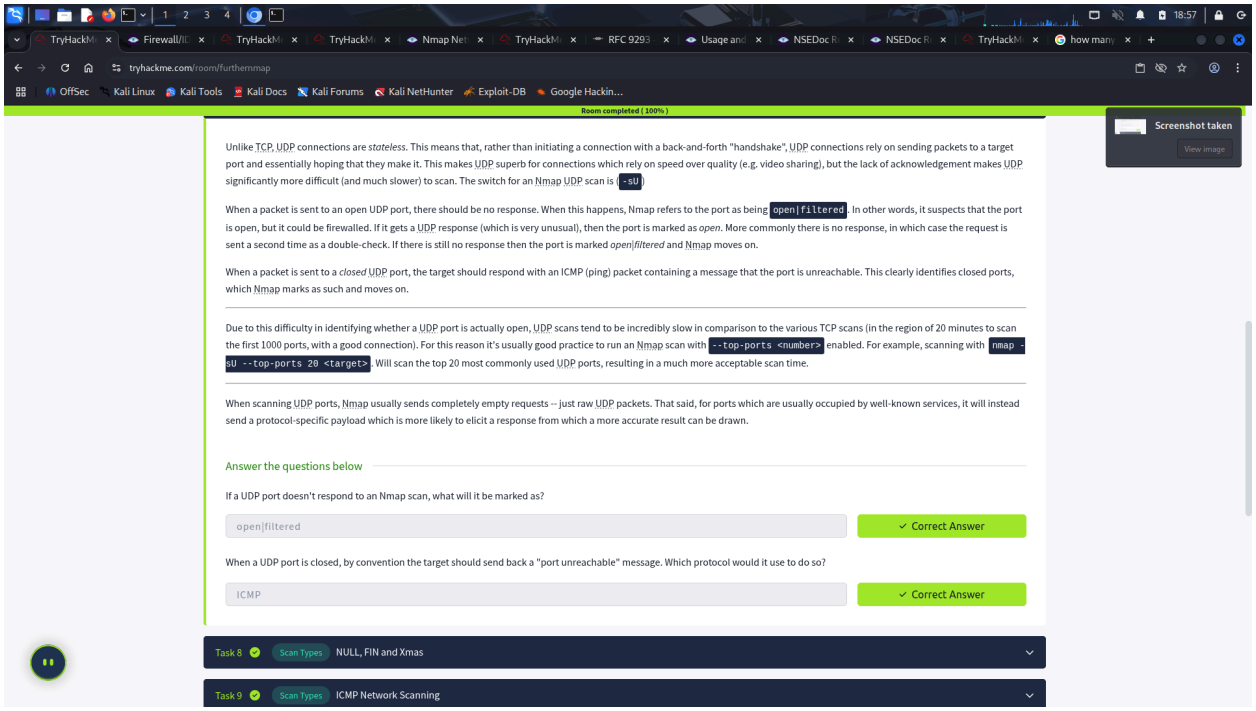
Can Nmap use a SYN scan without Sudo permissions (Y/N)?

N — ✓ Correct Answer

Task 7 Scan Types UDP Scans

Task 8 Scan Types NULL, FIN and Xmas

Task 7: Scan Types - UDP scans



Unlike TCP, UDP connections are *stateless*. This means that, rather than initiating a connection with a back-and-forth "handshake", UDP connections rely on sending packets to a target port and essentially hoping that they make it. This makes UDP superb for connections which rely on speed over quality (e.g. video sharing), but the lack of acknowledgement makes UDP significantly more difficult (and much slower) to scan. The switch for an Nmap UDP scan is (`-sU`).

When a packet is sent to an open UDP port, there should be no response. When this happens, Nmap refers to the port as being `open|filtered`. In other words, it suspects that the port is open, but it could be firewalled. If it gets a UDP response (which is very unusual), then the port is marked as *open*. More commonly there is no response, in which case the request is sent a second time as a double-check. If there is still no response then the port is marked *open|filtered* and Nmap moves on.

When a packet is sent to a *closed* UDP port, the target should respond with an ICMP (ping) packet containing a message that the port is unreachable. This clearly identifies closed ports, which Nmap marks as such and moves on.

Due to this difficulty in identifying whether a UDP port is actually open, UDP scans tend to be incredibly slow in comparison to the various TCP scans (in the region of 20 minutes to scan the first 1000 ports, with a good connection). For this reason it's usually good practice to run an Nmap scan with `--top-ports <number>` enabled. For example, scanning with `nmap -sU --top-ports 20 <target>`. Will scan the top 20 most commonly used UDP ports, resulting in a much more acceptable scan time.

When scanning UDP ports, Nmap usually sends completely empty requests -- just raw UDP packets. That said, for ports which are usually occupied by well-known services, it will instead send a protocol-specific payload which is more likely to elicit a response from which a more accurate result can be drawn.

Answer the questions below

If a UDP port doesn't respond to an Nmap scan, what will it be marked as?

open|filtered — ✓ Correct Answer

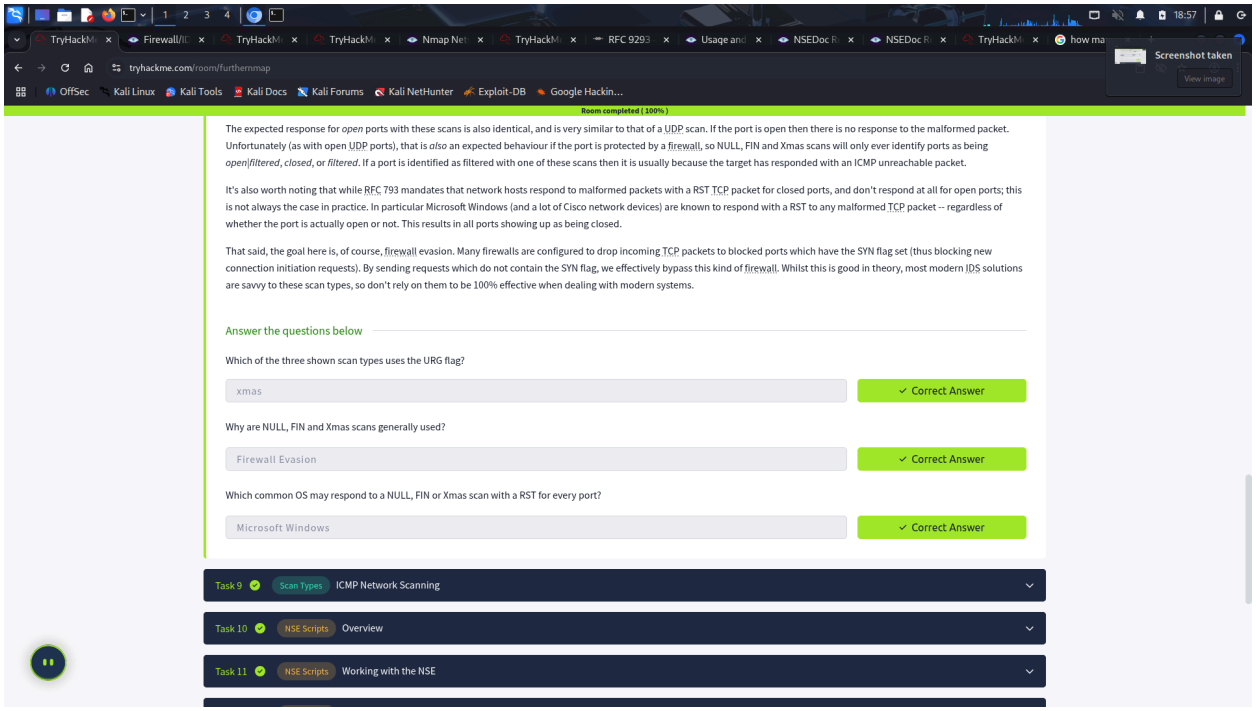
When a UDP port is closed, by convention the target should send back a "port unreachable" message. Which protocol would it use to do so?

ICMP — ✓ Correct Answer

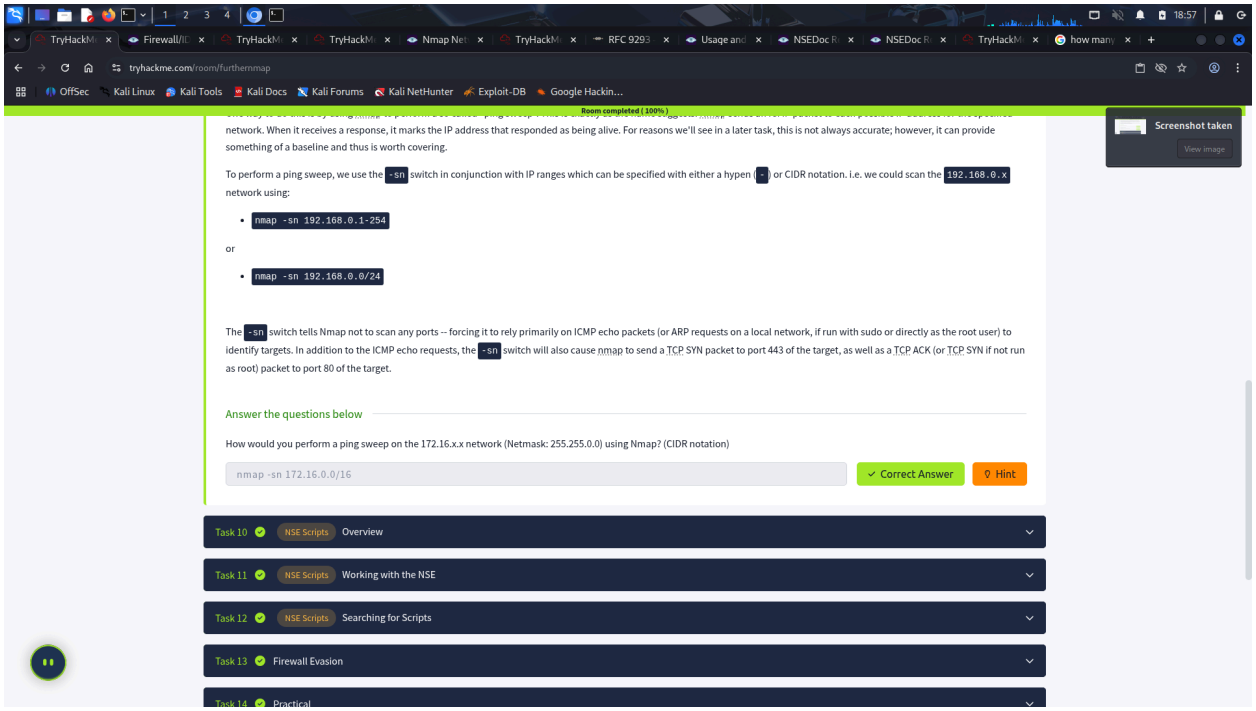
Task 8 Scan Types NULL, FIN and Xmas

Task 9 Scan Types ICMP Network Scanning

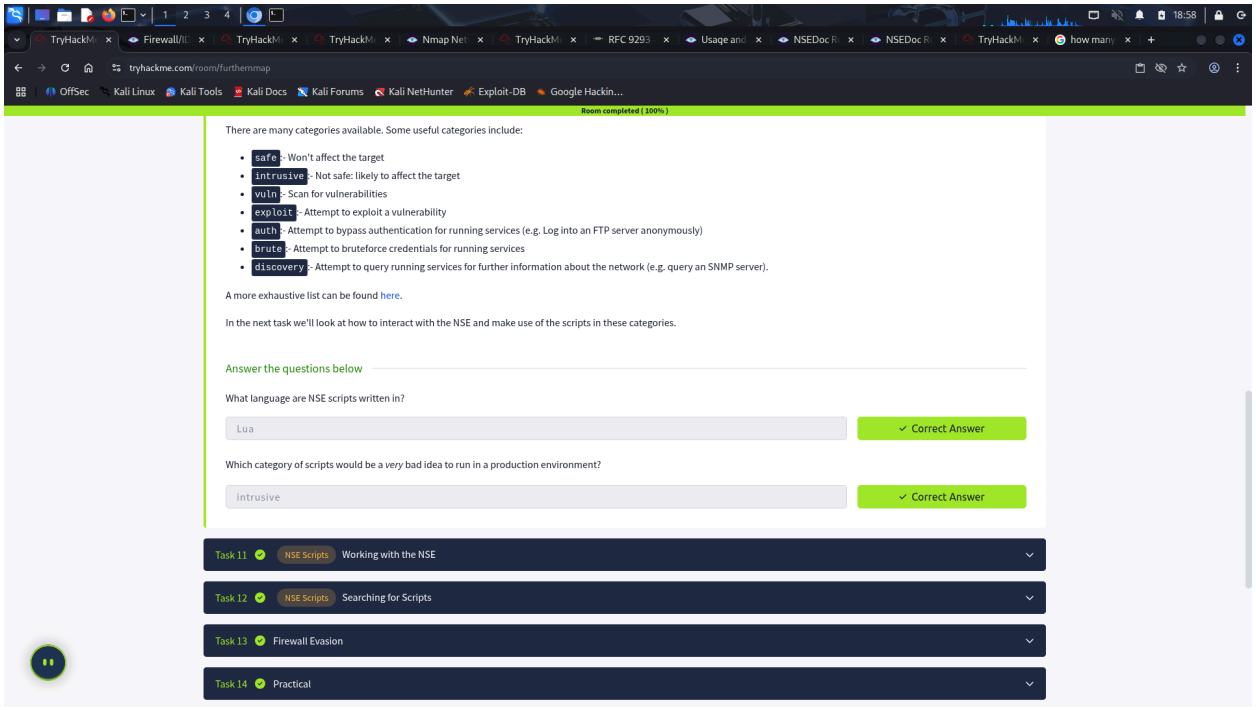
Task 8: Scan Types - NULL, Fin and Xmas



Task 9: Scan Types - ICMP Network Scanning



Task 10: NSE Scripts - Overview



There are many categories available. Some useful categories include:

- `safe` - Won't affect the target
- `intrusive` - Not safe: likely to affect the target
- `vuln` - Scan for vulnerabilities
- `exploit` - Attempt to exploit a vulnerability
- `auth` - Attempt to bypass authentication for running services (e.g. Log into an FTP server anonymously)
- `brute` - Attempt to bruteforce credentials for running services
- `discovery` - Attempt to query running services for further information about the network (e.g. query an SNMP server).

A more exhaustive list can be found here.

In the next task we'll look at how to interact with the NSE and make use of the scripts in these categories.

Answer the questions below

What language are NSE scripts written in?

Lua — ✓ Correct Answer

Which category of scripts would be a *very* bad idea to run in a production environment?

intrusive — ✓ Correct Answer

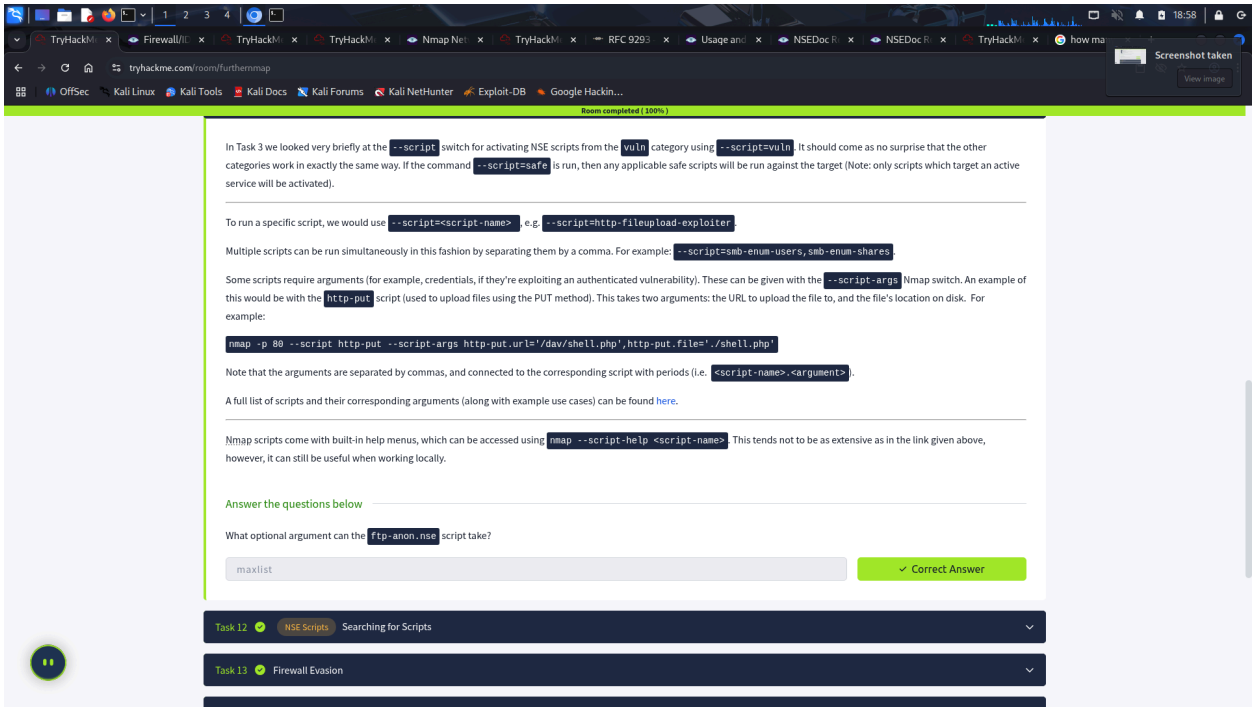
Task 11 NSE Scripts Working with the NSE

Task 12 NSE Scripts Searching for Scripts

Task 13 Firewall Evasion

Task 14 Practical

Task 11: NSE Scripts - Working with the NSE



In Task 3 we looked very briefly at the `--script` switch for activating NSE scripts from the `vuln` category using `--script=vuln`. It should come as no surprise that the other categories work in exactly the same way. If the command `--script=safe` is run, then any applicable safe scripts will be run against the target (Note: only scripts which target an active service will be activated).

To run a specific script, we would use `--script=<script-name>`, e.g. `--script=http-fileupload-exploiter`.

Multiple scripts can be run simultaneously in this fashion by separating them by a comma. For example: `--script=smb-enum-users,smb-enum-shares`.

Some scripts require arguments (for example, credentials, if they're exploiting an authenticated vulnerability). These can be given with the `--script-args` Nmap switch. An example of this would be with the `http-put` script (used to upload files using the PUT method). This takes two arguments: the URL to upload the file to, and the file's location on disk. For example:

```
nmap -p 80 --script http-put --script-args http-put.url='/dav/shell.php',http-put.file='./shell.php'
```

Note that the arguments are separated by commas, and connected to the corresponding script with periods (i.e. `<script-name>.<argument>`).

A full list of scripts and their corresponding arguments (along with example use cases) can be found here.

Nmap scripts come with built-in help menus, which can be accessed using `nmap --script-help <script-name>`. This tends not to be as extensive as in the link given above, however, it can still be useful when working locally.

Answer the questions below

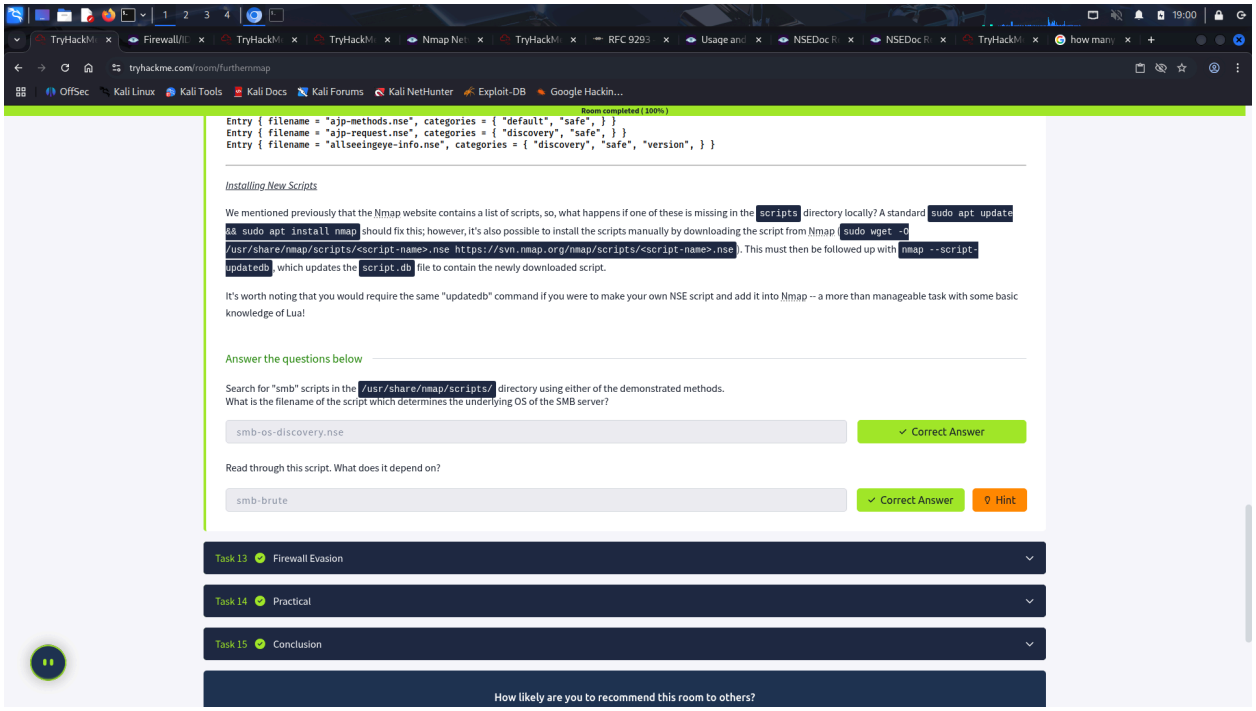
What optional argument can the `ftp-anon.nse` script take?

maxlist — ✓ Correct Answer

Task 12 NSE Scripts Searching for Scripts

Task 13 Firewall Evasion

Task 12: NSE Scripts - Searching for scripts



```
Entry { filename = "ajp-methods.nse", categories = { "default", "safe", } }
Entry { filename = "ajp-request.nse", categories = { "discovery", "safe", } }
Entry { filename = "allseeingeye-info.nse", categories = { "discovery", "safe", "version", } }
```

Installing New Scripts

We mentioned previously that the Nmap website contains a list of scripts, so, what happens if one of these is missing in the `scripts` directory locally? A standard `sudo apt update && sudo apt install nmap` should fix this; however, it's also possible to install the scripts manually by downloading the script from Nmap (`sudo wget -O /usr/share/nmap/scripts/<script-name>.nse https://svn.nmap.org/nmap/scripts/<script-name>.nse`). This must then be followed up with `nmap --script-updatedb`, which updates the `script.db` file to contain the newly downloaded script.

It's worth noting that you would require the same "updatedb" command if you were to make your own NSE script and add it into Nmap -- a more than manageable task with some basic knowledge of Lua!

Answer the questions below

Search for "smb" scripts in the `/usr/share/nmap/scripts/` directory using either of the demonstrated methods.
What is the filename of the script which determines the underlying OS of the SMB server?

smb-os-discovery.nse — ✓ Correct Answer

Read through this script. What does it depend on?

smb-brute — ✓ Correct Answer — Hint

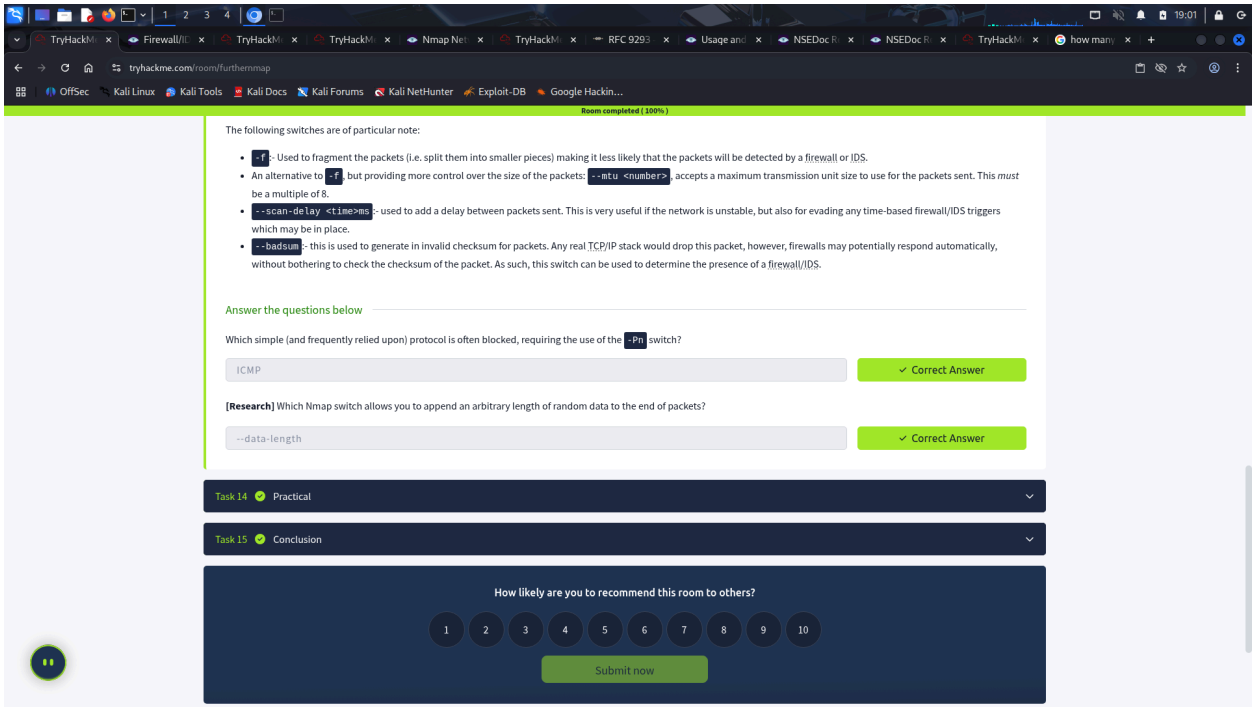
Task 13 Firewall Evasion

Task 14 Practical

Task 15 Conclusion

How likely are you to recommend this room to others?

Task 13: Firewall evasion



The following switches are of particular note:

- `-f` :- Used to fragment the packets (i.e. split them into smaller pieces) making it less likely that the packets will be detected by a firewall or IDS.
- An alternative to `-f`, but providing more control over the size of the packets: `--mtu <number>`, accepts a maximum transmission unit size to use for the packets sent. This *must* be a multiple of 8.
- `--scan-delay <time>ms` :- used to add a delay between packets sent. This is very useful if the network is unstable, but also for evading any time-based firewall/IDS triggers which may be in place.
- `--badsum` :- this is used to generate in invalid checksum for packets. Any real TCP/IP stack would drop this packet, however, firewalls may potentially respond automatically, without bothering to check the checksum of the packet. As such, this switch can be used to determine the presence of a firewall/IDS.

Answer the questions below

Which simple (and frequently relied upon) protocol is often blocked, requiring the use of the `-Pn` switch?

ICMP — ✓ Correct Answer

[Research] Which Nmap switch allows you to append an arbitrary length of random data to the end of packets?

--data-length — ✓ Correct Answer

Task 14 Practical

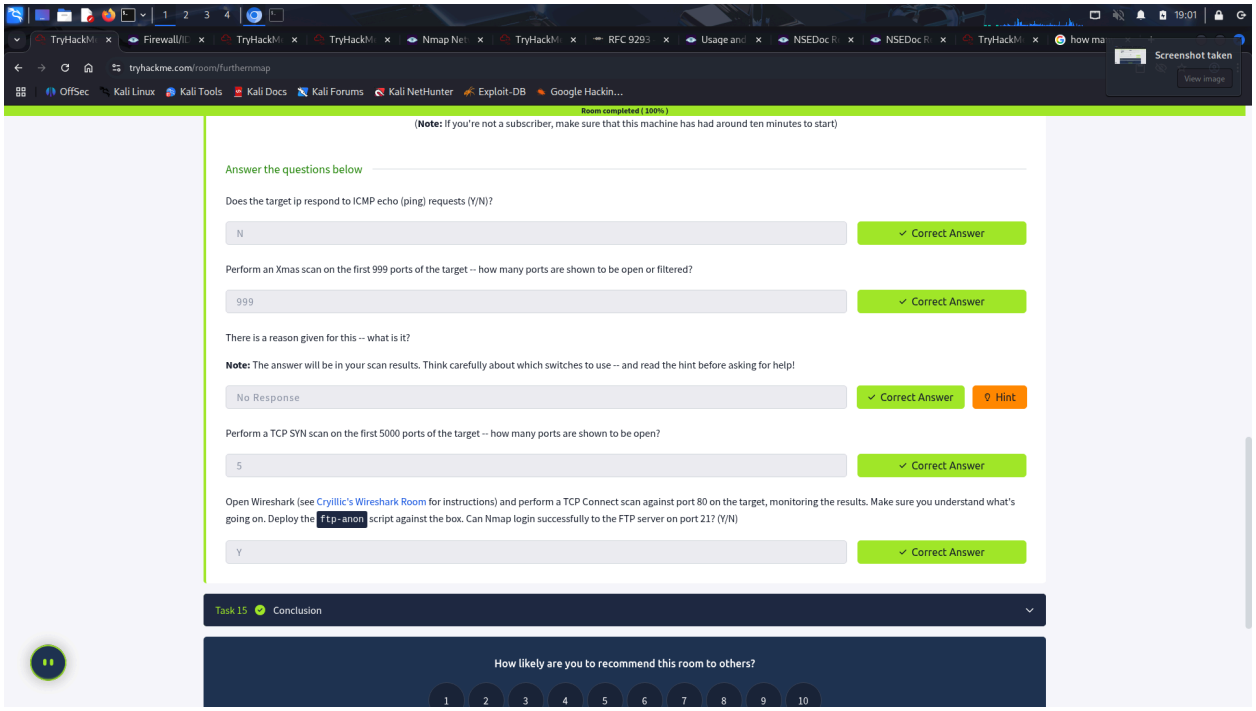
Task 15 Conclusion

How likely are you to recommend this room to others?

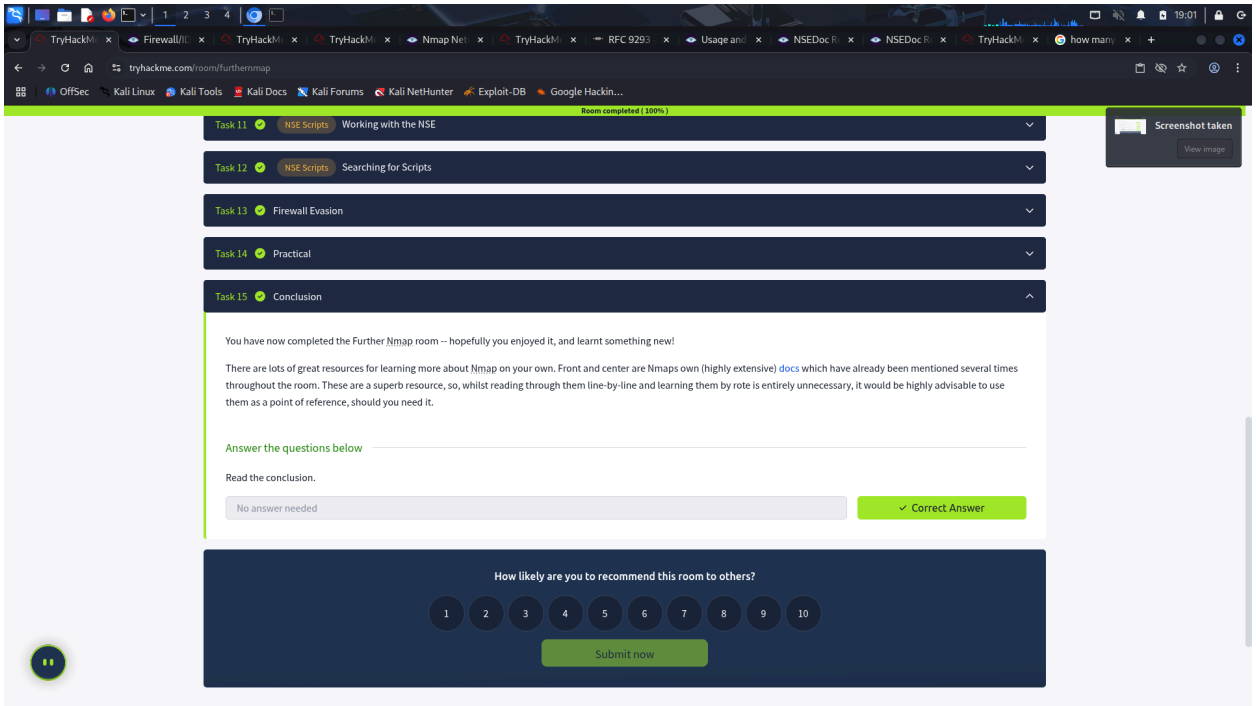
1 2 3 4 5 6 7 8 9 10

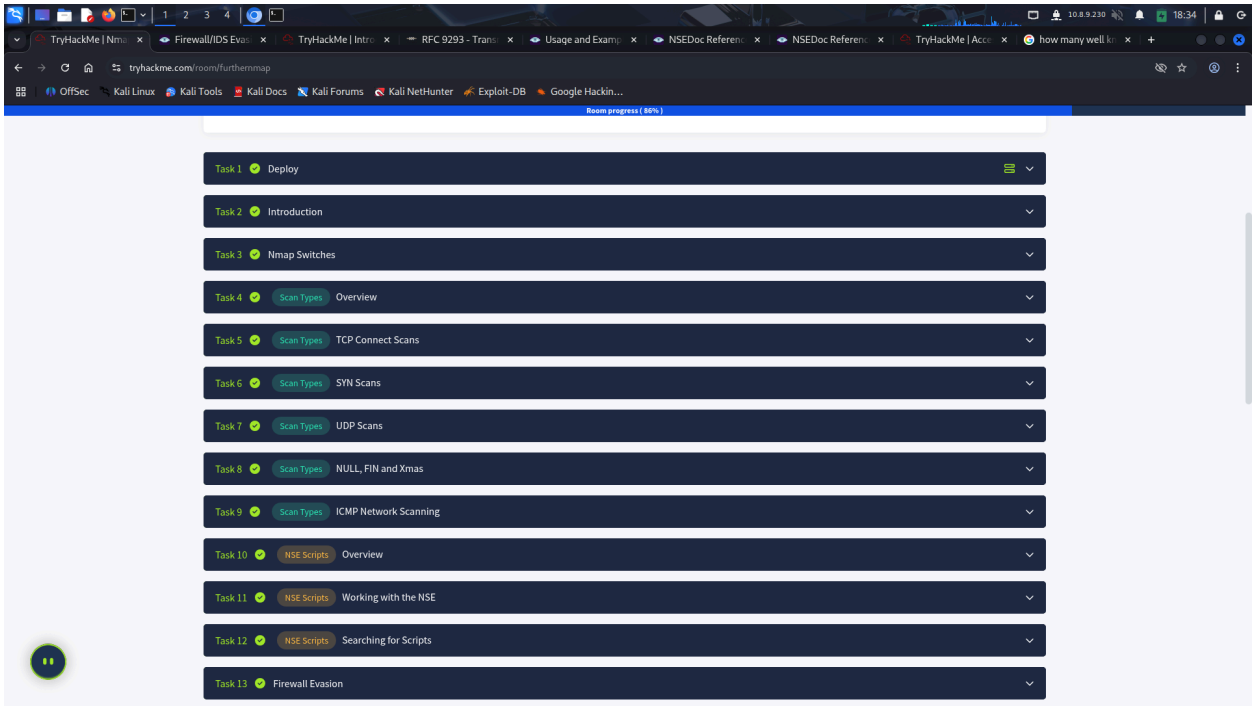
Submit now

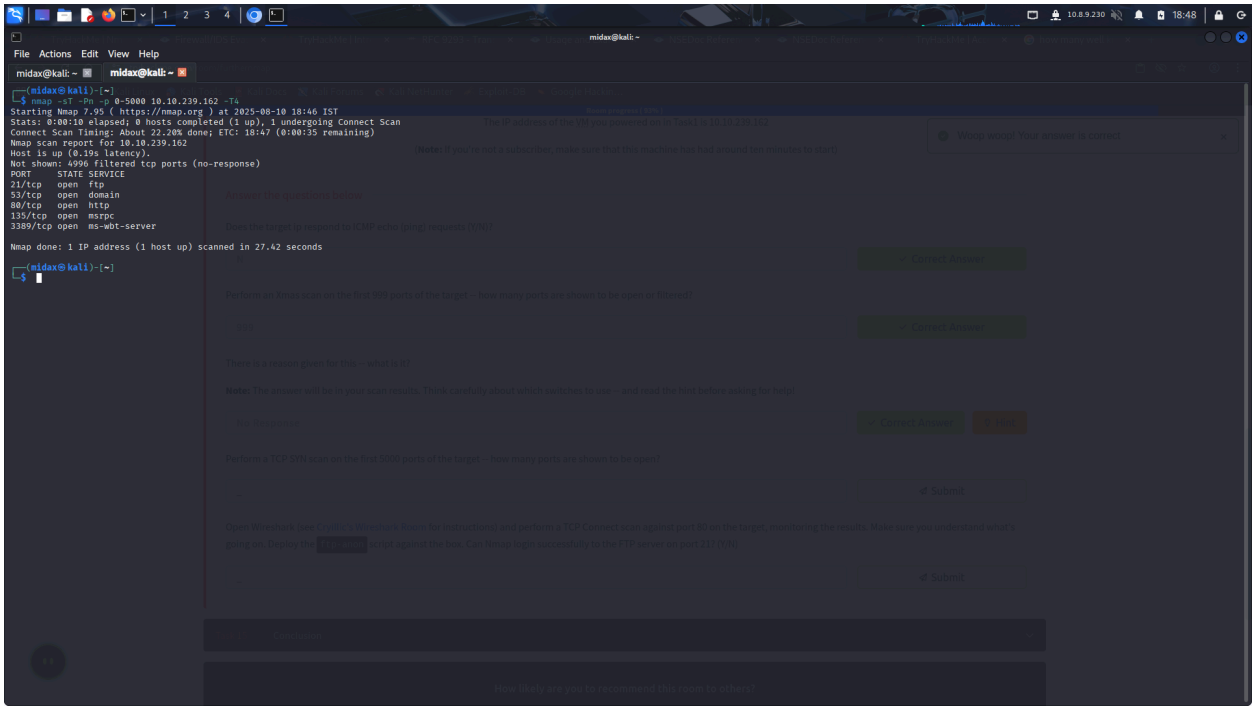
Task 14: Practical

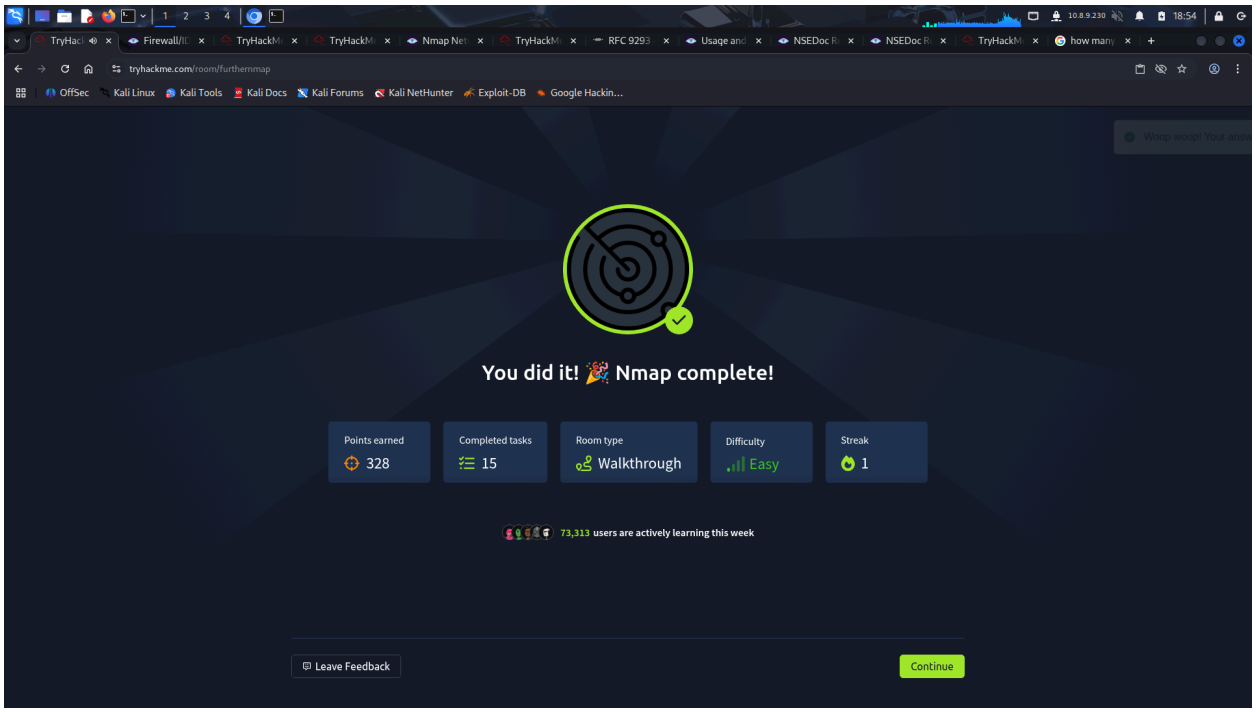


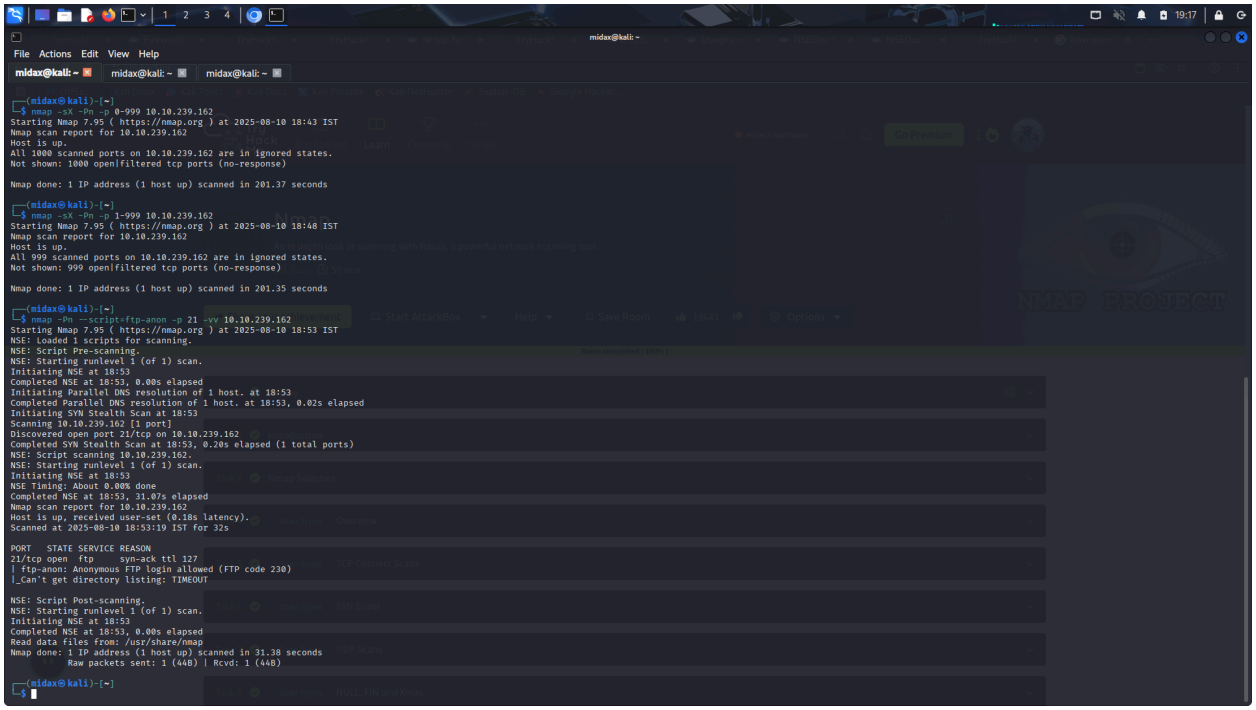
Task 15: Conclusion




Room progress (86%)
Task 1 Deploy
Task 2 Introduction
Task 3 Nmap Switches
Task 4 Scan Types Overview
Task 5 Scan Types TCP Connect Scans
Task 6 Scan Types SYN Scans
Task 7 Scan Types UDP Scans
Task 8 Scan Types NULL, FIN and Xmas
Task 9 Scan Types ICMP Network Scanning
Task 10 NSE Scripts Overview
Task 11 NSE Scripts Working with the NSE
Task 12 NSE Scripts Searching for Scripts
Task 13 Firewall Evasion


File Actions Edit View Help
midax@kali: ~
midax@kali: ~
┌──(midax㉿kali)-[~]
└─$ nmap -sT -Pn -p 0-5000 10.10.239.162 -T4
Starting Nmap 7.95 (https://nmap.org) at 2025-08-10 18:46 IST
Stats: 0:00:10 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 22.20% done; ETC: 18:47 (0:00:35 remaining)
Nmap scan report for 10.10.239.162
Host is up (0.19s latency).
Not shown: 4996 filtered tcp ports (no-response)
PORT STATE SERVICE
21/tcp open ftp
53/tcp open domain
80/tcp open http
135/tcp open msrpc
3389/tcp open ms-wbt-server
Nmap done: 1 IP address (1 host up) scanned in 27.42 seconds
┌──(midax㉿kali)-[~]
└─$


tryhackme.com/room/furthernmap
You did it! 🎉 Nmap complete!
Points earned
328
Completed tasks
15
Room type
Walkthrough
Difficulty
Easy
Streak
1
73,313 users are actively learning this week
Leave Feedback
Continue


midax@kali: ~
File Actions Edit View Help
midax@kali: ~
midax@kali: ~
midax@kali: ~
┌──(midax㉿kali)-[~]
└─$ nmap -sX -Pn -p 0-999 10.10.239.162
Starting Nmap 7.95 (https://nmap.org) at 2025-08-10 18:43 IST
Nmap scan report for 10.10.239.162
Host is up.
All 1000 scanned ports on 10.10.239.162 are in ignored states.
Not shown: 1000 open|filtered tcp ports (no-response)
Nmap done: 1 IP address (1 host up) scanned in 201.37 seconds
┌──(midax㉿kali)-[~]
└─$ nmap -sX -Pn -p 1-999 10.10.239.162
Starting Nmap 7.95 (https://nmap.org) at 2025-08-10 18:48 IST
Nmap scan report for 10.10.239.162
Host is up.
All 999 scanned ports on 10.10.239.162 are in ignored states.
Not shown: 999 open|filtered tcp ports (no-response)
Nmap done: 1 IP address (1 host up) scanned in 201.35 seconds
┌──(midax㉿kali)-[~]
└─$ nmap -Pn --script=ftp-anon -p 21 -vv 10.10.239.162
Starting Nmap 7.95 (https://nmap.org) at 2025-08-10 18:53 IST
NSE: Loaded 1 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 1) scan.
Initiating NSE at 18:53
Completed NSE at 18:53, 0.00s elapsed
Initiating Parallel DNS resolution of 1 host. at 18:53
Completed Parallel DNS resolution of 1 host. at 18:53, 0.02s elapsed
Initiating SYN Stealth Scan at 18:53
Scanning 10.10.239.162 [1 port]
Discovered open port 21/tcp on 10.10.239.162
Completed SYN Stealth Scan at 18:53, 0.20s elapsed (1 total ports)
NSE: Script scanning 10.10.239.162.
NSE: Starting runlevel 1 (of 1) scan.
Initiating NSE at 18:53
NSE Timing: About 0.00% done
Completed NSE at 18:53, 31.07s elapsed
Nmap scan report for 10.10.239.162
Host is up, received user-set (0.18s latency).
Scanned at 2025-08-10 18:53:19 IST for 32s
PORT STATE SERVICE REASON
21/tcp open ftp syn-ack ttl 127
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: TIMEOUT
NSE: Script Post-scanning.
NSE: Starting runlevel 1 (of 1) scan.
Initiating NSE at 18:53
Completed NSE at 18:53, 0.00s elapsed
Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 31.38 seconds
Raw packets sent: 1 (44B) | Rcvd: 1 (44B)
┌──(midax㉿kali)-[~]
└─$