# Nmap Room Report

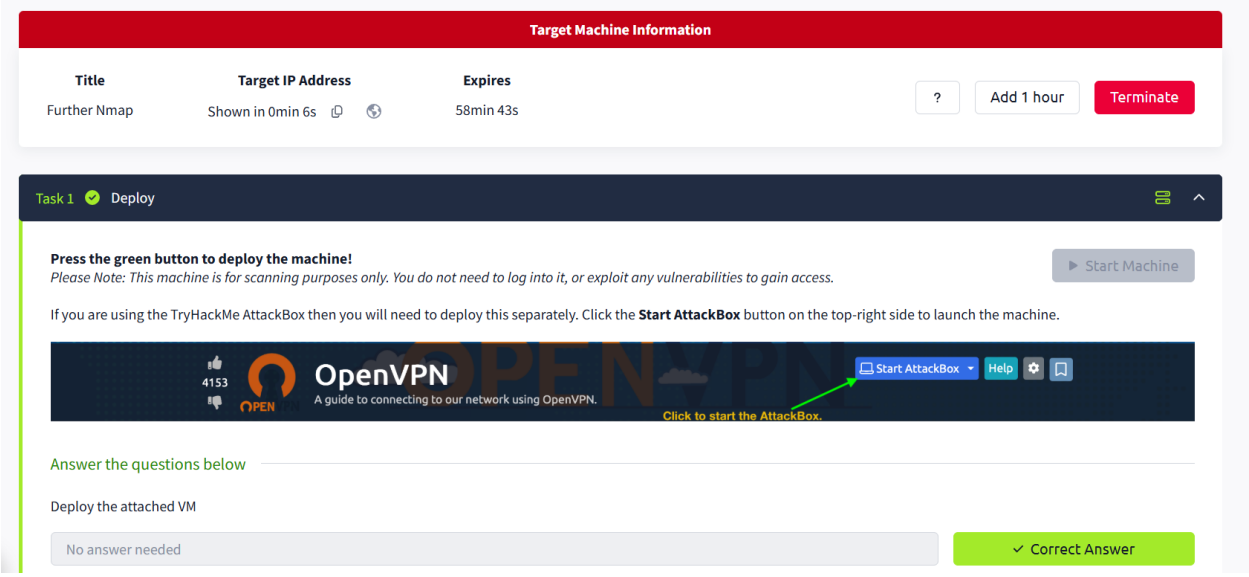
Report by: Ann Maria Joseph

**Introduction**

This report documents the completion of the TryHackMe “Further Nmap” room, focusing on advanced port scanning and enumeration techniques using Nmap. The room covered different scan types, Nmap switches, NSE scripting, and firewall evasion techniques. Practical tasks included performing SYN, TCP Connect, UDP, NULL, FIN, and Xmas scans, identifying open ports and services, using NSE scripts, and interpreting scan results. The report summarises the key commands, concepts, and findings from each task, with screenshots added as evidence of the steps performed.

## Task 1 – Deploy

- Deploying a virtual machine to start attack.



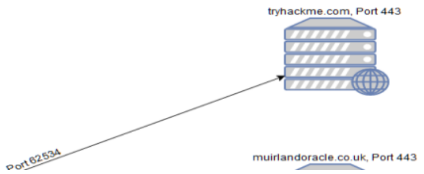

## Task 2 – Introduction

- Port scanning is used to identify open ports and services on a target machine.
- Nmap is the industry-standard tool for port scanning and enumeration.

Task 2 Introduction

When it comes to hacking, knowledge is power. The more knowledge you have about a target system or network, the more options you have available. This makes it imperative that proper enumeration is carried out before any exploitation attempts are made.

Say we have been given an IP (or multiple IP addresses) to perform a security audit on. Before we do anything else, we need to get an idea of the "landscape" we are attacking. What this means is that we need to establish which services are running on the targets. For example, perhaps one of them is running a webserver, and another is acting as a Windows Active Directory Domain Controller. The first stage in establishing this "map" of the landscape is something called port scanning. When a computer runs a network service, it opens a networking construct called a "port" to receive the connection. Ports are necessary for making multiple network requests or having multiple services available. For example, when you load several webpages at once in a web browser, the program must have some way of determining which tab is loading which web page. This is done by establishing connections to the remote webservers using different ports on your local machine. Equally, if you want a server to be able to run more than one service (for example, perhaps you want your webserver to run both HTTP and HTTPS versions of the site), then you need some way to direct the traffic to the appropriate service. Once again, ports are the solution to this. Network connections are made between two ports – an open port listening on the server and a randomly selected port on your own computer. For example, when you connect to a web page, your computer may open port 49534 to connect to the server's port 443.




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

- **Syn Scan switch:** `-sS`
- **UDP Scan switch:** `-sU`
- **OS Detection:** `-O`
- **Service Version Detection:** `-sV`
- **Increase Verbosity:** `-v`
- **Verbosity Level 2:** `-vv`
- **Save Output (3 formats):** `-oA <filename>`
- **Normal Output:** `-oN <filename>`
- **Grepable Output:** `-oG <filename>`
- **Aggressive Scan Mode:** `-A`
- **Timing Template 5:** `-T5`
- **Scan Only Port 80:** `-p 80`
- **Scan All Ports:** `-p-`
- **Run a Script:** `--script <script-name>`

- **Run All Vuln Scripts:** `--script=vuln`

Task 3 Nmap Switches

Like most pentesting tools, nmap is run from the terminal. There are versions available for both Windows and Linux. For this room we will assume that you are using Linux; however, the switches should be identical. Nmap is installed by default in both Kali Linux and the TryHackMe Attack Box.

Nmap can be accessed by typing `nmap` into the terminal command line, followed by some of the "switches" (command arguments which tell a program to do different things) we will be covering below.

All you'll need for this is the help menu for nmap (accessed with `nmap -h`) and/or the nmap man page (access with `man nmap`). For each answer, include all parts of the switch unless otherwise specified. This includes the hyphen at the start (`-`).

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

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

-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 – TCP Connect Scans

- TCP Connect scans (`-sT`) perform a full three-way handshake.
- **RFC defining TCP protocol:** RFC 9293.

Task 4 Scan Types Overview

When port scanning with Nmap, there are three basic scan types. These are:

- TCP Connect Scans (`-sT`)
- SYN "Half-open" Scans (`-sS`)
- UDP Scans (`-sU`)

Additionally there are several less common port scan types, some of which we will also cover (albeit in less detail). These are:

- TCP Null Scans (`-sN`)
- TCP FIN Scans (`-sF`)
- TCP Xmas Scans (`-sX`)

Most of these (with the exception of UDP scans) are used for very similar purposes, however, the way that they work differs between each scan. This means that, whilst one of the first three scans are likely to be your go-to in most situations, it's worth bearing in mind that other scan types exist.

In terms of network scanning, we will also look briefly at ICMP (or "ping") scanning.

Answer the questions below

Read the Scan Types Introduction.

No answer needed ✓ Correct Answer

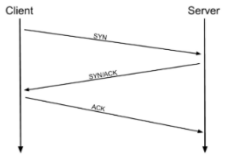
# Task 5 – TCP Connect Status

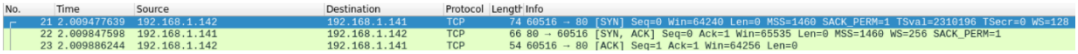
- Explaining three-way handshake
- RFC 9293 defining behaviour for TCP
- RST – reset flag used If port is close.

Task 5 Scan Types TCP Connect Scans

To understand TCP Connect scans (`-sT`), it's important that you're comfortable with the *TCP three-way handshake*. If this term is new to you then completing Introductory Networking before continuing would be advisable.

As a brief recap, the three-way handshake consists of three stages. First the connecting terminal (our attacking machine, in this instance) sends a TCP request to the target server with the SYN flag set. The server then acknowledges this packet with a TCP response containing the SYN flag, as well as the ACK flag. Finally, our terminal completes the handshake by sending a TCP request with the ACK flag set.




This is one of the fundamental principles of TCP/IP networking, but how does it relate to Nmap?

```
iptables -I INPUT -p tcp --dport <port> -j REJECT --reject-with tcp-reset
```

This can make it extremely difficult (if not impossible) to get an accurate reading of the target(s).

Answer the questions below

Which RFC defines the appropriate behaviour for the TCP protocol?

RFC 9293 ✓ Correct Answer Hint

If a port is closed, which flag should the server send back to indicate this?

RST ✓ Correct Answer

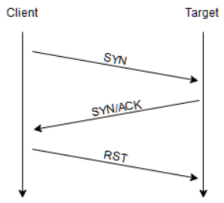
# Task 6 – SYN Scans

- SYN scans (`-sS`) are half-open/stealth scans.
- **Other Names:** Half-Open, Stealth.
- **Need sudo permissions:** Yes.

Task 6 ✓ Scan Types SYN Scans

As with TCP scans, SYN scans (`-sS`) are used to scan the TCP port-range of a target or targets; however, the two scan types work slightly differently. SYN scans are sometimes referred to as "*Half-open*" scans, or "*Stealth*" scans.

Where TCP scans perform a full three-way handshake with the target, SYN scans sends back a RST TCP packet after receiving a SYN/ACK from the server (this prevents the server from repeatedly trying to make the request). In other words, the sequence for scanning an **open** port looks like this:




| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 39 | 8.389443540 | 192.168.1.142 | 192.168.1.238 | TCP | 58 | 53425 → 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460 |
| 40 | 8.389900067 | 192.168.1.238 | 192.168.1.142 | TCP | 60 | 80 → 53425 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 |
| 41 | 8.389992786 | 192.168.1.142 | 192.168.1.238 | TCP | 54 | 53425 → 80 [RST] Seq=1 Win=0 Len=0 |

This has a variety of advantages for us as hackers:

- It can be used to bypass older Intrusion Detection systems as they are looking out for a full three way handshake. This is often no longer the case with modern IDS solutions; it is for this reason that SYN scans are still frequently referred to as "stealth" scans.
- SYN scans are often not logged by applications listening on open ports, as standard practice is to log a connection once it's been fully established. Again, this plays into the idea of

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?

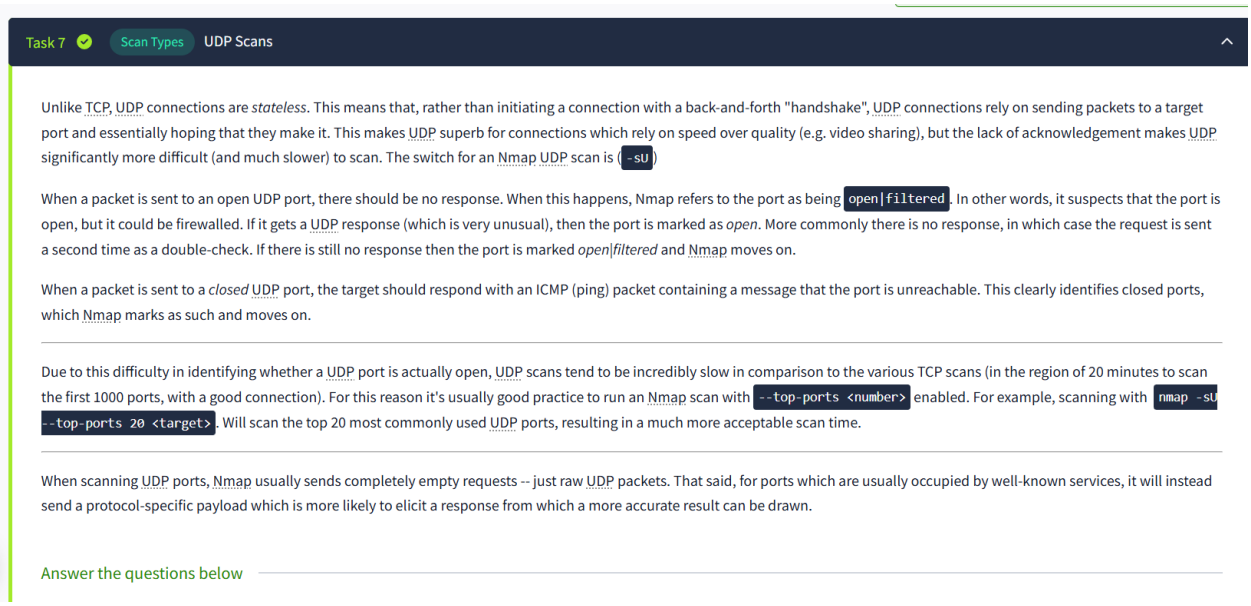
Half-Open, Stealth ✓ Correct Answer

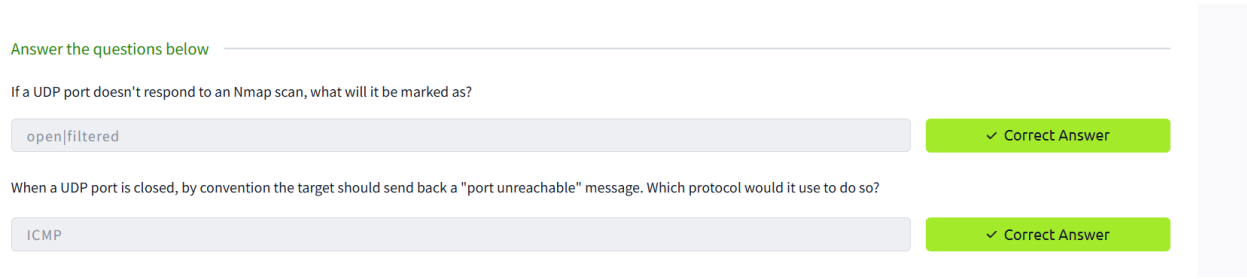
Can Nmap use a SYN scan without Sudo permissions (Y/N)?

N ✓ Correct Answer

# Task 7 – UDP Scans

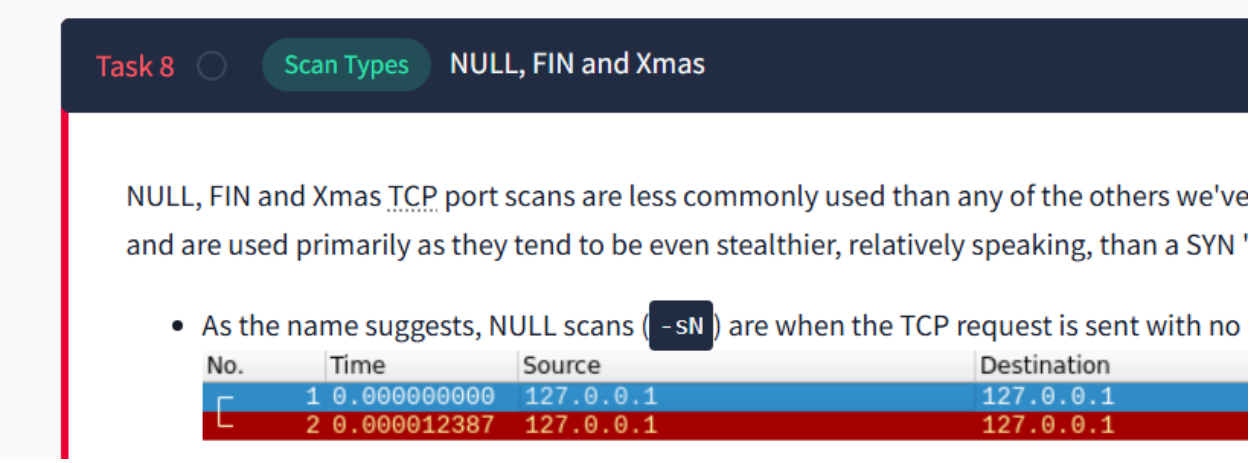
- UDP is stateless and slow to scan.
- **If no response:** marked as open|`filtered`.
- **Closed ports reply with:** ICMP port unreachable.




# Task 8 – NULL, FIN & Xmas Scans

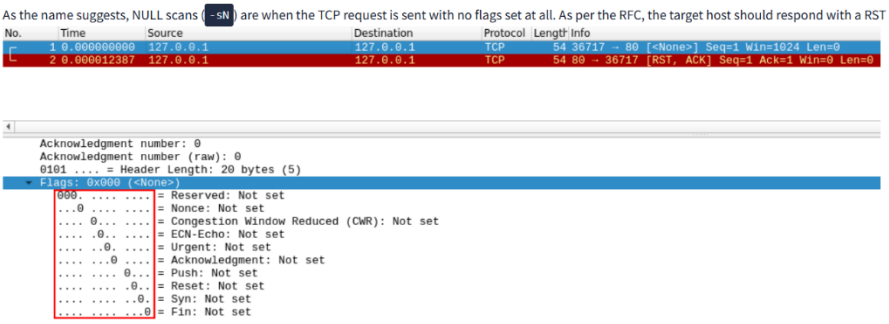
- **Xmas scan flag:** URG (among PSH & FIN).
- **Purpose:** Firewall evasion.
- **Common OS responding RST always:** Microsoft Windows.



Task 8 Scan Types NULL, FIN and Xmas

NULL, FIN and Xmas TCP port scans are less commonly used than any of the others we've covered already, so we will not go into a huge amount of depth here. All three are interlinked and are used primarily as they tend to be even stealthier, relatively speaking, than a SYN "stealth" scan. Beginning with NULL scans:

- As the name suggests, NULL scans ( `-sN` ) are when the TCP request is sent with no flags set at all. As per the RFC, the target host should respond with a RST if the port is closed.




Answer the questions below

Which of the three shown scan types uses the URG flag?

xmas ✓ Correct Answer

Why are NULL, FIN and Xmas scans generally used?

Firewall Evasion ✓ Correct Answer

Which common OS may respond to a NULL, FIN or Xmas scan with a RST for every port?

Microsoft Windows ✓ Correct Answer

# Task 9 – Ping Sweeps

- **Ping sweep on 172.16.x.x/16:**

```
nmap -sn 172.16.0.0/16
```

Task 9 Scan Types ICMP Network Scanning

On first connection to a target network in a black box assignment, our first objective is to obtain a "map" of the network structure -- or, in other words, we want to see which IP addresses contain active hosts, and which do not.

One way to do this is by using Nmap to perform a so called "ping sweep". This is exactly as the name suggests: Nmap sends an ICMP packet to each possible IP address for the specified network. When it receives a response, it marks the IP address that responded as being alive. For reasons we'll see in a later task, this is not always accurate; however, it can provide something of a baseline and thus is worth covering.

To perform a ping sweep, we use the `-sn` switch in conjunction with IP ranges which can be specified with either a hypen ( `-` ) or CIDR notation. i.e. we could scan the `192.168.0.x` network using:

- `nmap -sn 192.168.0.1-254`

or

- `nmap -sn 192.168.0.0/24`

The `-sn` switch tells Nmap not to scan any ports -- forcing it to rely primarily on ICMP echo packets (or ARP requests on a local network, if run with sudo or directly as the root user) to identify targets. In addition to the ICMP echo requests, the `-sn` switch will also cause nmap to send a TCP SYN packet to port 443 of the target, as well as a TCP ACK (or TCP SYN if not run as root) packet to port 80 of the target.

- `nmap -sn 192.168.0.0/24`

Woop woop! Your answer is

The `-sn` switch tells Nmap not to scan any ports -- forcing it to rely primarily on ICMP echo packets (or ARP requests on a local network, if run with sudo or directly as the root user) to identify targets. In addition to the ICMP echo requests, the `-sn` switch will also cause nmap to send a TCP SYN packet to port 443 of the target, as well as a TCP ACK (or TCP SYN if not run as root) packet to port 80 of the target.

Answer the questions below

How would you perform a ping sweep on the 172.16.x.x network (Netmask: 255.255.0.0) using Nmap? (CIDR notation)

nmap -sn 172.16.0.0/16 ✓ Correct Answer Hint

# Task 10 – NSE Scripts

- **Language:** Lua
- **Dangerous category:** intrusive

Task 10 NSE Scripts Overview

The **N**map **S**cripting **E**ngine (NSE) is an incredibly powerful addition to Nmap, extending its functionality quite considerably. NSE Scripts are written in the *Lua* programming language, and can be used to do a variety of things: from scanning for vulnerabilities, to automating exploits for them. The NSE is particularly useful for reconnaisance, however, it is well worth bearing in mind how extensive the script library is.

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

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 – Running Scripts

- **Optional argument for ftp-anon:** maxlist

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 – Searching Scripts

- **SMB OS Discovery Script:** smb-os-discovery.nse
- **Depends on:** smb

Task 12 NSE Scripts Searching for Scripts

Ok, so we know how to use the scripts in Nmap, but we don't yet know how to *find* these scripts.

We have two options for this, which should ideally be used in conjunction with each other. The first is the page on the Nmap website (mentioned in the previous task) which con of all official scripts. The second is the local storage on your attacking machine. Nmap stores its scripts on Linux at `/usr/share/nmap/scripts`. All of the NSE scripts are stored directory by default -- this is where Nmap looks for scripts when you specify them.

There are two ways to search for installed scripts. One is by using the `/usr/share/nmap/scripts/script.db` file. Despite the extension, this isn't actually a database so much a formatted text file containing filenames and categories for each available script.

```
muri@augury:/usr/share/nmap/scripts$ file script.db
script.db: ASCII text
muri@augury:/usr/share/nmap/scripts$ head script.db
Entry { filename = "acarsd-info.nse", categories = { "discovery", "safe", } }
Entry { filename = "address-info.nse", categories = { "default", "safe", } }
Entry { filename = "afp-brute.nse", categories = { "brute", "intrusive", } }
Entry { filename = "afp-ls.nse", categories = { "discovery", "safe", } }
Entry { filename = "afp-path-vuln.nse", categories = { "exploit", "intrusive", "vuln", } }
Entry { filename = "afp-serverinfo.nse", categories = { "default", "discovery", "safe", } }
Entry { filename = "afp-showmount.nse", categories = { "discovery", "safe", } }
```

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

- **Protocol often blocked:** ICMP
- **Append random data:** `--data-length <num>`

Task 13 — Firewall Evasion

We have already seen some techniques for bypassing firewalls (think stealth scans, along with NULL, FIN and Xmas scans); however, there is another very common firewall configuration which it's imperative we know how to bypass.

Your typical Windows host will, with its default firewall, block all ICMP packets. This presents a problem: not only do we often use *ping* to manually establish the activity of a target, Nmap does the same thing by default. This means that Nmap will register a host with this firewall configuration as dead and not bother scanning it at all.

So, we need a way to get around this configuration. Fortunately Nmap provides an option for this: `-Pn`, which tells Nmap to not bother pinging the host before scanning it. This means that Nmap will always treat the target host(s) as being alive, effectively bypassing the ICMP block; however, it comes at the price of potentially taking a very long time to complete the scan (if the host really is dead then Nmap will still be checking and double checking every specified port).

It's worth noting that if you're already directly on the local network, Nmap can also use ARP requests to determine host activity.

---

There are a variety of other switches which Nmap considers useful for firewall evasion. We will not go through these in detail, however, they can be found here.

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 Scanning

## 1 Does the target IP respond to ICMP echo requests?

- Answer: N

## 2 Xmas scan on first 999 ports

- **Open|Filtered Ports:** 999
- **Reason:** All ports are open|filtered

## 3 TCP SYN scan on first 5000 ports

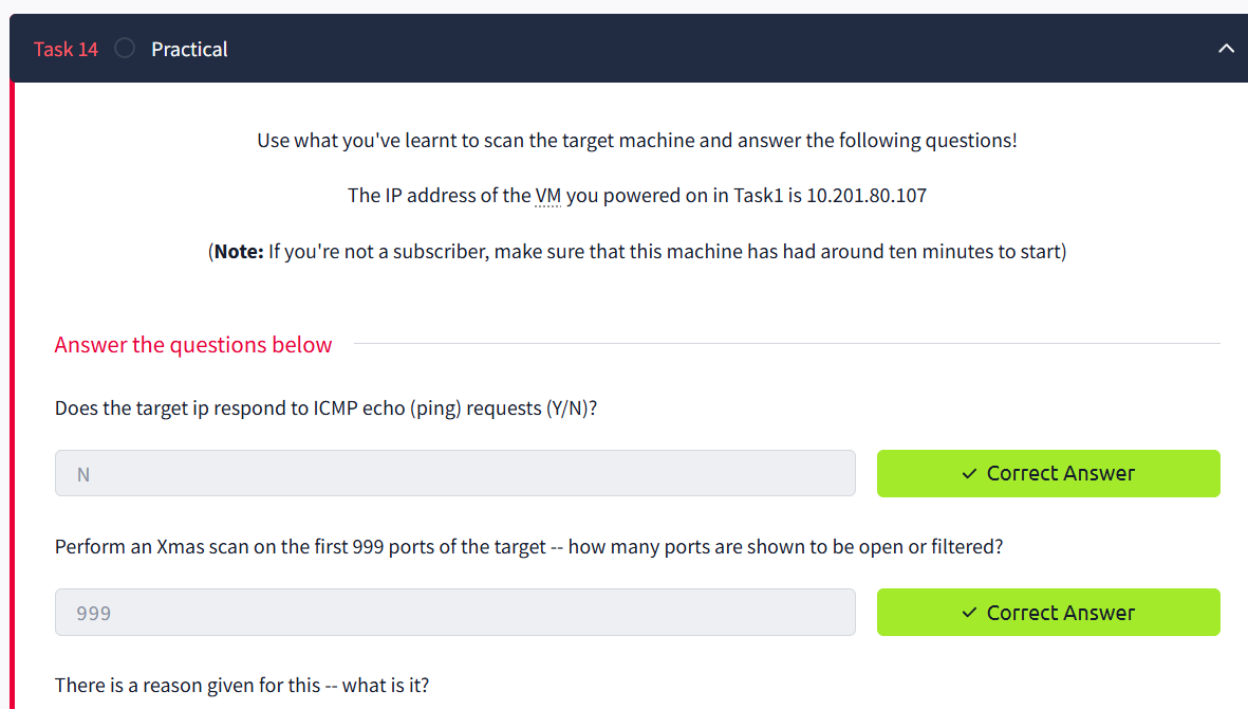
- **Open Ports:** (Add after scanning)

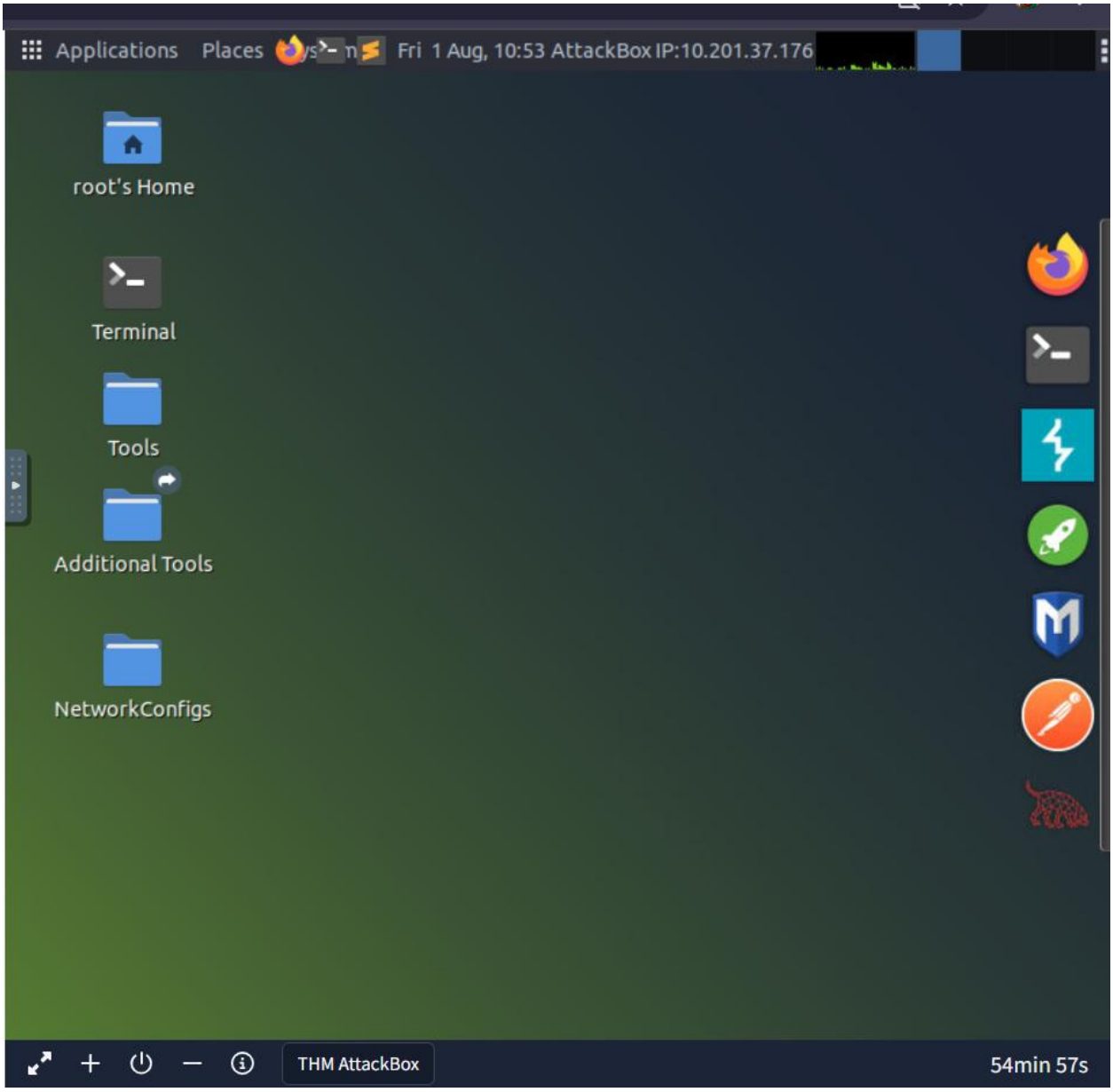
## 4 TCP Connect scan on port 80 (Wireshark)

- 3-way handshake observed (SYN → SYN/ACK → ACK)

## 5 ftp-anon script on port 21

- **Anonymous Login:** (Y/N based on scan output)






```
root@ip-10-201-37-176:~# ping -c 4 10.201.83.29
PING 10.201.83.29 (10.201.83.29) 56(84) bytes of data.
From 10.201.37.176 icmp_seq=1 Destination Host Unreachable
From 10.201.37.176 icmp_seq=2 Destination Host Unreachable
From 10.201.37.176 icmp_seq=3 Destination Host Unreachable
From 10.201.37.176 icmp_seq=4 Destination Host Unreachable

--- 10.201.83.29 ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3073ms
pipe 3
root@ip-10-201-37-176:~#
```

```
root@ip-10-201-24-50:~# sudo nmap -sX -p 1-999 -Pn 10.201.80.107
Starting Nmap 7.80 ( https://nmap.org ) at 2025-08-01 11:13 BST
Nmap scan report for ip-10-201-80-107.ec2.internal (10.201.80.107)
Host is up (0.00020s latency).
All 999 scanned ports on ip-10-201-80-107.ec2.internal (10.201.80.107) are open|
filtered
MAC Address: 16:FF:FF:B5:A1:3D (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 21.29 seconds
```

```
root@ip-10-201-24-50:~# sudo nmap -sS -p 1-5000 -Pn  10.201.80.107
Starting Nmap 7.80 ( https://nmap.org ) at 2025-08-01 11:23 BST
Nmap scan report for ip-10-201-80-107.ec2.internal (10.201.80.107)
Host is up (0.00031s latency).
Not shown: 4995 filtered ports
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
MAC Address: 16:FF:FF:B5:A1:3D (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 14.56 seconds
root@ip-10-201-24-50:~#
```

```
root@ip-10-201-24-50: ~
File  Edit  View  Search  Terminal  Help
root@ip-10-201-24-50:~# sudo nmap -sT -p 80 10.201.80.107
Starting Nmap 7.80 ( https://nmap.org ) at 2025-08-01 11:25 BST
Nmap scan report for ip-10-201-80-107.ec2.internal (10.201.80.107)
Host is up (0.00011s latency).

PORT   STATE SERVICE
80/tcp open  http
MAC Address: 16:FF:FF:B5:A1:3D (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.13 seconds
root@ip-10-201-24-50:~# sudo nmap -p 21 --script=ftp-anon 10.201.80.107
Starting Nmap 7.80 ( https://nmap.org ) at 2025-08-01 11:27 BST
Nmap scan report for ip-10-201-80-107.ec2.internal (10.201.80.107)
Host is up (0.00012s latency).

PORT   STATE SERVICE
21/tcp open  ftp
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: TIMEOUT
MAC Address: 16:FF:FF:B5:A1:3D (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 30.90 seconds
root@ip-10-201-24-50:~#
```

## Task 15 - Conclusion

- Learned various Nmap scanning techniques, NSE scripting, firewall evasion, and practical enumeration methods.
- Nmap is a powerful tool for reconnaissance and vulnerability detection in penetration testing.