

# **TryHackMe Write-up**

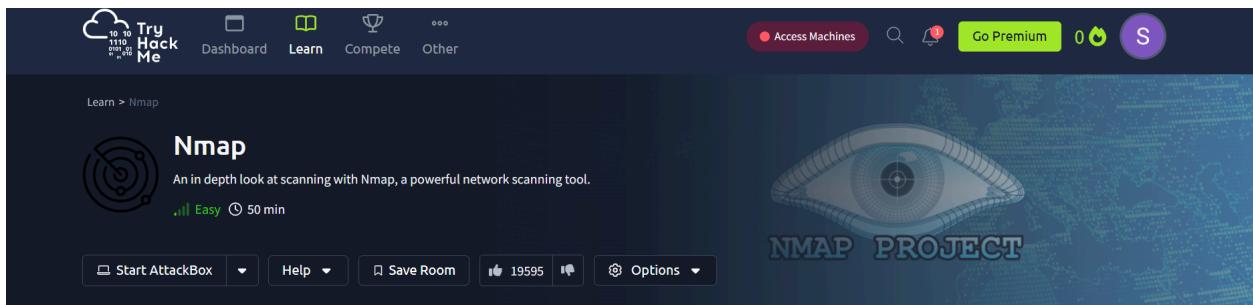
Room: Nmap

Prepared by: Shifna N.

Date: July 30, 2025

## •Introduction

This report summarizes the completion of the "Nmap" room on TryHackMe. This room covered its different scan types, options, and capabilities, including how to use Nmap effectively in real-world penetration testing scenarios. Each task in the room included hands-on exercises and questions, which have been answered using screenshots for proof and are explained in this report.



## •Task 1: Deploy

A screenshot of the THM-AttackBox interface. On the left, a sidebar shows 'Target Machine Information' with a title 'Further Nmap', target IP '10.10.139.45', and an expiration time of '47min 36s'. A red button labeled 'Deploy' is highlighted. Below this, a message says 'Press the green button to deploy the machine!' and 'Start Machine'. A note states: 'Person Note: This machine is for scanning purposes only. You do not need to log into it, or exploit any vulnerabilities to gain access.' Another note says: 'If you are using the TryHackMe AttackBox then you will need to deploy this separately. Click the Start AttackBox button on the top-right side to launch the machine.' At the bottom, there's a progress bar for 'Answer the questions below' and a note: 'Deploy the attached VM'. On the right, a desktop environment is shown with icons for root's Home, Terminal, Tools, Additional Tools, and NetworkConfigs. A status bar at the bottom right shows '47min 44s'.

In this task, we deployed the target machine provided by TryHackMe. This step is essential as all the practical exercises depend on having an active target system. A green message indicated that the machine had been successfully deployed. After deployment, we waited for the IP address to be assigned and confirmed connectivity before moving on to the scanning tasks.

## •Task 2: Introduction

This task gave a simple introduction to Nmap and how it is used in ethical hacking. It showed that Nmap helps scan open ports, find running services, and collect basic details about systems on a network. The task also introduced the room's structure, which includes multiple scan types and techniques that Nmap supports.

A screenshot of the THM-AttackBox interface. On the left, a sidebar shows a progress bar at '50%' with the text 'enumerating which services are running on each port - either manually, or more commonly using nmap.' A question asks: '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.' Below this, a section titled 'Answer the questions below' asks: 'What networking constructs are used to direct traffic to the right application on a server?' with a dropdown menu showing 'Ports'. A 'Correct Answer' button is shown. Another question asks: 'How many of these are available on any network-enabled computer?' with a dropdown menu showing '65535'. A 'Correct Answer' button is shown. A research note asks: '[Research] How many of these are considered "well-known"? (These are the "standard" numbers mentioned in the task)' with a dropdown menu showing '1024'. A 'Correct Answer' button and a 'Hint' button are shown. The status bar at the bottom right shows '45min 38s'.

## •Task 3: Nmap Switches

The task window contains the following text:

Like most testing 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?  
-os

Here, we explored the different switches or command-line options that Nmap provides. We learned about commonly used switches like `-sS` for SYN scan, `-sT` for TCP Connect, and `-sU` for UDP scans. Other useful switches such as `-p` for specifying ports, `-A` for aggressive scan, and `-T` for timing templates were also covered. These switches help customize scans depending on the situation.

## •Task 4: Scan Types Overview

This task offered a classification of various scan types available in Nmap. This task explained the difference between TCP scans, UDP scans, and special types like NULL, FIN, and Xmas scans. It helped us understand which scans are more hidden and how they can avoid firewalls or detection systems.

The task window contains the following text:

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`
- TCPXmas 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 Scans

The task window contains the following text:

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.

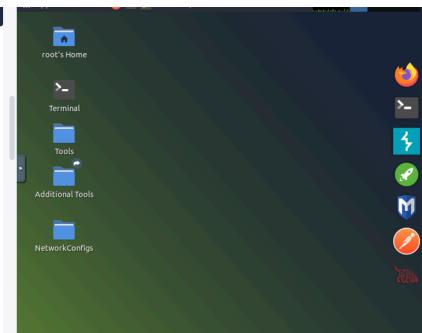
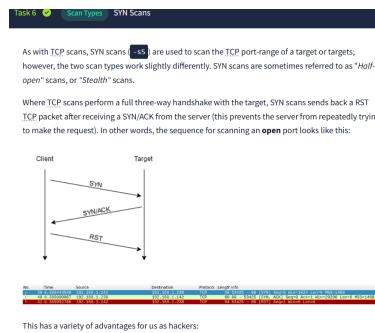
This is one of the fundamental principles of TCP/IP networking, but how does it relate to Nmap?

Well, as the name suggests, a TCP Connect scan works by performing the three-way handshake with

We learned that a TCP Connect scan uses the system's own networking stack to establish a full TCP connection. This scan is easy to spot because it finishes the full three-way handshake. It's mostly used when SYN scans can't be done due to limitations or lack of permission. Even though it's not hidden, it's reliable and helpful in some situations.

## •Task 6: SYN Scans

This task was about SYN scans, also known as "half-open scans." They send SYN packets and wait for SYN-ACK replies but don't finish the connection. This makes the scan quicker and harder to detect than TCP Connect scans, so it's commonly used by penetration testers.. SYN scans require root privileges to operate.



## •Task 7: UDP Scans

marks as such and moves on.

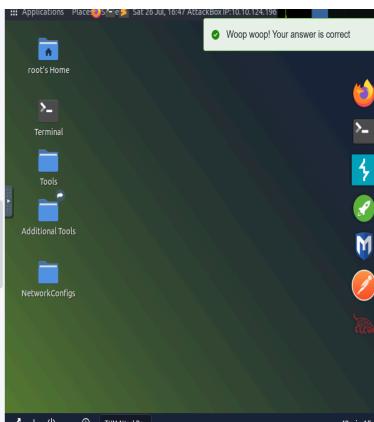
Due to this difficulty in identifying whether a UDP port is actually open, UDP scans tend to be incredibly slow in comparison to the various TCP scans (in the region of 20 minutes to scan the first 1000 ports, with a good connection). For this reason it's usually good practice to run an Nmap scan with `--top-ports <numbers>` enabled. For example, scanning with `nmap -sU --top-ports 26 [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?

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?



We learned about UDP scans, which help find services running on UDP ports. These scans are slower and less reliable because UDP doesn't send acknowledgments. Still, Nmap checks for ICMP responses or no replies to find open ports, making it a useful method even with its drawbacks.

## •Task 8: NULL, FIN, and Xmas Scans

In this task, we examined three advanced scan types: NULL, FIN, and Xmas scans. These scans change TCP flags to try and trick firewalls and avoid being detected. For example, NULL scans send packets with no flags, FIN scans send just a FIN flag, and Xmas scans use a mix of FIN, PSH, and URG flags . These are stealth scans often used when firewalls block more traditional scan types.

It's also worth noting that while RFC 7231 mandates that network hosts respond to malformed packets with a RST TCP packet for closed ports, and don't respond at all for open ports; this is not always the case in practice. In particular Microsoft Windows (and a lot of Cisco network devices) are known to respond with a RST to any malformed TCP packet – regardless of whether the port is actually open or not. This results in all ports showing up as being closed.

That said, the goal here is, of course, Firewall evasion. Many firewalls are configured to drop incoming TCP packets to blocked ports which have the SYN flag set (thus blocking new connection initiation requests). By sending requests which do not contain the SYN flag, we effectively bypass this kind of firewall. While this is good in theory, most modern IDS solutions are savvy to these scan types, so don't rely on them to be 100% effective when dealing with modern systems.

Answer the questions below

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

Why are NULL, FIN and Xmas scans generally used?

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



- Task 9: ICMP Network Scanning

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 hyphen (`-s-`) or CIDR notation. i.e. we could scan the `192.168.0.0` 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.

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 172.16.0.0/26      Correct Answer      9 Hint

File Edit View Search Terminal Help  
01

This machine can access other machines you deploy on TryHackMe.

Please keep in mind the following:

1. Penetrating any target that is not deployed by you on TryHackMe is prohibited.
2. You are solely responsible for your actions.
3. The TryHackMe team reserves the right to terminate you still have time left.
4. Once this machine is terminated, all data will be lost.
5. If you are a subscriber, this machine is exposed to the Internet and maybe an easy target. Please make sure the machine is secure, do not store sensitive files, usage instructions, etc.

Usage Instructions:

1. Tools are located in `/root/Desktop/tools/` & `/opt/`
2. Web shells are located in `/usr/share/webshells/`
3. Wordlists are located in `/usr/share/wordlists/`
4. Exploit development tools are located in `/usr/share/exploitdev/`
5. To use Empire & Stagerkiller, read the following file: `/root/Instructions/empire-starkerkiller.txt`

Do you think something's missing? Let us know! support@tryhackme.com

Press ENTER key to close. ■

ICMP scanning is used for host discovery, determining which hosts are up without performing port scans. It sends echo requests (ping), timestamp requests, or address mask requests. This task explained how to use ICMP to quickly map live hosts in a network, especially when port scanning isn't possible.

- Task 10: NSE Scripts Overview

In this task, we learned about the Nmap Scripting Engine (NSE), which helps with automating network tasks. NSE scripts are written in Lua and can check software versions, find security issues, and try password guessing. This showed us that Nmap can do more than just simple scanning.

The Nmap Scripting Engine (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 reconnaissance, 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.

```
Terminal
File Edit View Search Terminal Help
          61

This machine can access other machines you deploy on TryHackMe.

Please keep in mind the following:
1. Penetrating any target that is not deployed by you on TryHackMe is prohibited.
2. You are solely responsible for your actions.
3. You must have permission from the owner to ensure you still have time left.
4. Once this machine is terminated, all data will be lost.
5. If you are a subscriber, this machine is exposed to the internet and maybe an
   automatically scanned. While the machine is secure, do not store sensitive files.

Usage Instructions:
1. Tools are located in /root/Desktop/Tools & /opt/
2. Webshells are located in /usr/share/webshells
3. Wordlists are located in /usr/share/wordlists
4. Instructions are located in /root/Instructions
5. To use Empire & Stageriller, read the following file: /root/Instructions/empire
   & stageriller.txt

Do you think something's missing? Let us know! support@tryhackme.com

Press ENTER key to close. █
```

- Task 11: Working with the NSE

Multiple scripts can be run simultaneously in this fashion by separating them by a comma. For example: `script1.sh,script2.sh,script3.sh`

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-pwd` 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:

```
map -p 80 --script http-pwd --script-args http.put.url="/day/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.

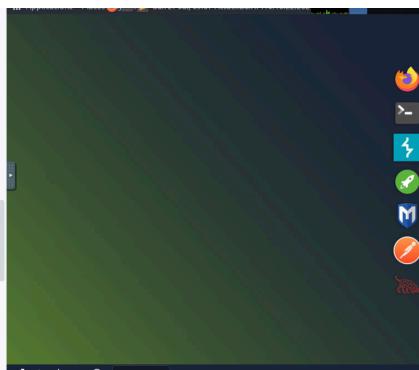
Nmap scripts come with built-in help menus, which can be accessed using `map script-help [script_name]`. This tends not to be as extensive as in the link given above, however, it can still be useful when working locally.

Answer the questions below

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

maxlist

✓ Correct Answer

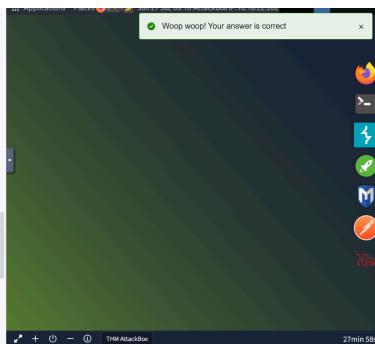


We explored how to use NSE scripts in scans by using the --script option. Examples included running default scripts with -sC, or specifying a category or individual script. We saw how NSE can add detailed information about services, vulnerabilities, and configurations when included in scans.

## •Task 12: Searching for Scripts

This task taught us how to find NSE scripts that match our needs. By navigating to the Nmap scripts directory or using command-line tools like locate or grep, we can search for specific keywords or categories. Understanding how to find and use the right script makes our scans more effective.

## •Task 13: Firewall Evasion

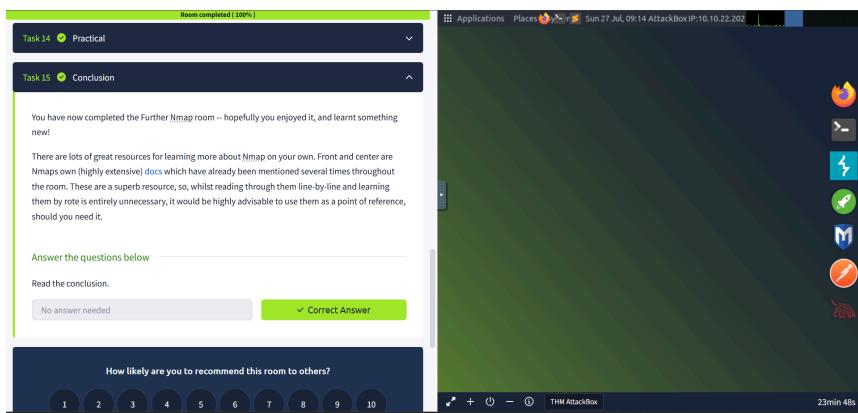


We learned techniques for evading firewalls and IDS during scans. Methods include fragmenting packets (-f), using decoys (--decoy), randomizing scan orders, and changing source ports. These techniques help avoid detection and gain access to more information when networks are actively monitored.

## •Task 14: Practical

In the practical task, we applied what we had learned in the previous tasks by conducting various Nmap scans on the deployed target. We used different scan types and options, including NSE scripts, and interpreted the results. This hands-on section reinforced our understanding and helped us gain confidence in using Nmap effectively.

## •Task 15: Conclusion



The final task summarized everything covered in the room. It showed how important Nmap is for finding and scanning targets during penetration testing. Learning about different scan types, NSE scripts, and ways to avoid detection is useful for both attacking and defending in cybersecurity.

## •Conclusion

The "Further Nmap" room helped me learn advanced Nmap scanning techniques. I now understand how TCP, SYN, and UDP scans work, and I can use different ways to bypass firewalls and security tools. I also learned to use Nmap scripts to find security issues and create my own scripts when needed. These skills make it easier for me to scan networks and check for security issues.