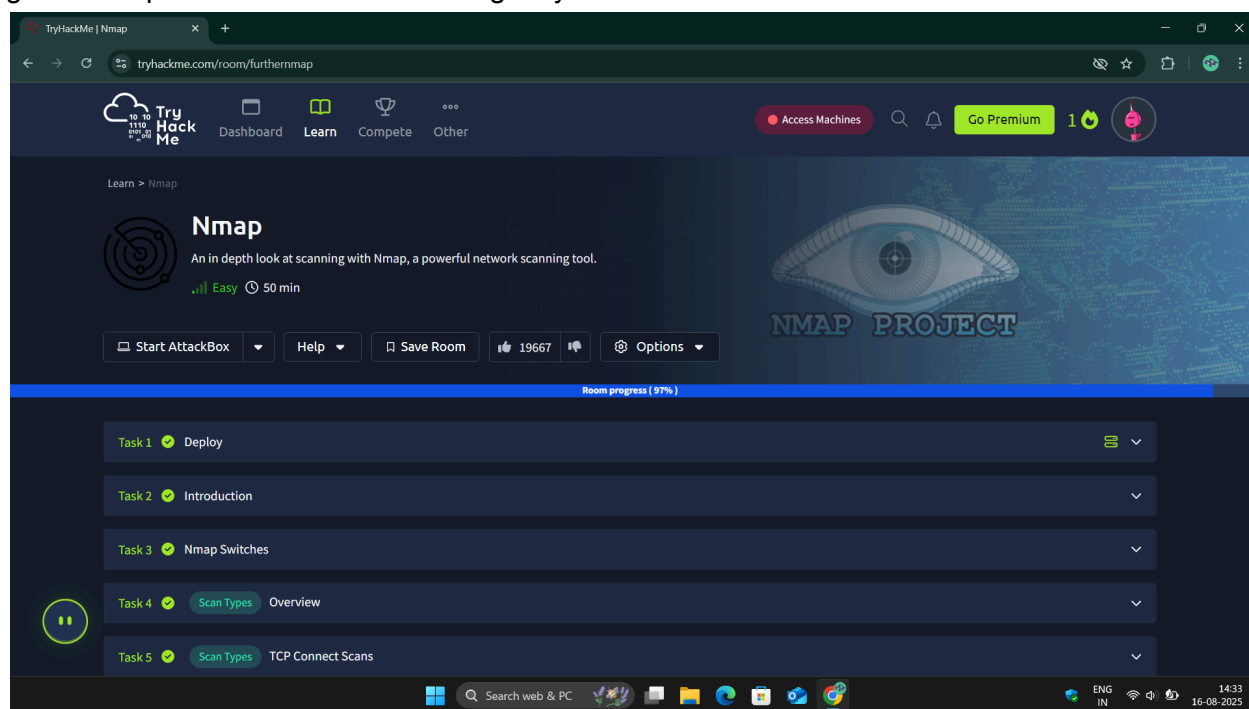


Report: TryHackMe Room – *Further Nmap*

Introduction

This room helped me go beyond the basics of Nmap and understand how it can be used as a powerful reconnaissance tool in cybersecurity. I already had an idea of Nmap's basic usage for scanning ports, but through this room, I learned how to apply it in more advanced ways to gather deeper information about a target system.



Key Learnings & Understanding

1. Basic Nmap Usage

- I now clearly understand how to run a simple scan on a target to discover open ports.
- I learned that the default scan checks the top 1000 most common ports and provides a quick overview of a system.

2. Different Scanning Techniques

- I understood the difference between a normal TCP connect scan and a **SYN (stealth) scan**.
- I learned how SYN scans are faster and less noisy, making them useful when avoiding detection is important.

3. Service and Version Detection

- I discovered that Nmap can not only show open ports but also identify the exact **services and versions** running on them, which is essential for vulnerability assessment.

4. Aggressive Mode

- I understood how the aggressive mode combines multiple features like OS detection, version scanning, and traceroute into one command.
- I realized that while this gives very detailed information, it is also very noisy and easily detectable.

5. Nmap Scripting Engine (NSE)

- I learned about using scripts to detect vulnerabilities or gather extra details about services.
- Running scripts such as those in the “vuln” category can quickly highlight possible weaknesses in a system.

6. Saving Scan Results

- I now know how to save results in different formats (normal, grepable, XML), which is very useful for documentation and automation.

7. Firewall Evasion Concepts

- I got introduced to how Nmap can bypass filters and firewalls using options like skipping host discovery, fragmenting packets, or changing source ports.

TryHackMe | Nmap

Room progress (75%)

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 as a formatted text file containing filenames and categories for each available script.

```
muriaugury:/usr/share/nmap/scripts$ file script.db
script.db: ASCII text
muriaugury:/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", } }
Entry { filename = "ajp-auth.nse", categories = { "auth", "default", "safe", } }
Entry { filename = "ajp-brute.nse", categories = { "brute", "intrusive", } }
Entry { filename = "ajp-headers.nse", categories = { "discovery", "safe", } }
```

Nmap uses this file to keep track of (and utilise) scripts for the scripting engine; however, we can also `grep` through it to look for scripts. For example: `grep "ftp"`

```
muriaugury:/usr/share/nmap/scripts$ grep "ftp" /usr/share/nmap/scripts/script.db
Entry { filename = "ftp-anon.nse", categories = { "auth", "default", "safe", } }
Entry { filename = "ftp-bounce.nse", categories = { "default", "safe", } }
Entry { filename = "ftp-brute.nse", categories = { "brute", "intrusive", } }
Entry { filename = "ftp-libopie.nse", categories = { "intrusive", "vuln", } }
Entry { filename = "ftp-proftpd-backdoor.nse", categories = { "exploit", "intrusive", "malware", "vuln", } }
Entry { filename = "ftp-syst.nse", categories = { "default", "discovery", "safe", } }
Entry { filename = "ftp-vsftpd-backdoor.nse", categories = { "exploit", "intrusive", "malware", "vuln", } }
Entry { filename = "ftp-vuln-cve2010-4221.nse", categories = { "intrusive", "vuln", } }
Entry { filename = "tftp-enum.nse", categories = { "discovery", "intrusive", } }
```

The second way to search for scripts is quite simply to use the `ls` command. For example, we could get the same results as in the previous screenshot by using `ls -l`

```
muriaugury:/usr/share/nmap/scripts$ ls -l /usr/share/nmap/scripts/ftp*
-rw-r--r-- 1 root root 4530 Oct 12 14:29 /usr/share/nmap/scripts/ftp-anon.nse
```

Conclusion

Before this room, I only understood Nmap as a tool to scan ports. After completing it, I now understand:

- The **different scan types** and when to use them.
- How to gather **detailed information** about services and operating systems.
- The importance of **NSE scripts** for vulnerability scanning.
- How to **save and document results** for reporting.
- Basic techniques to **evade firewalls and IDS**.