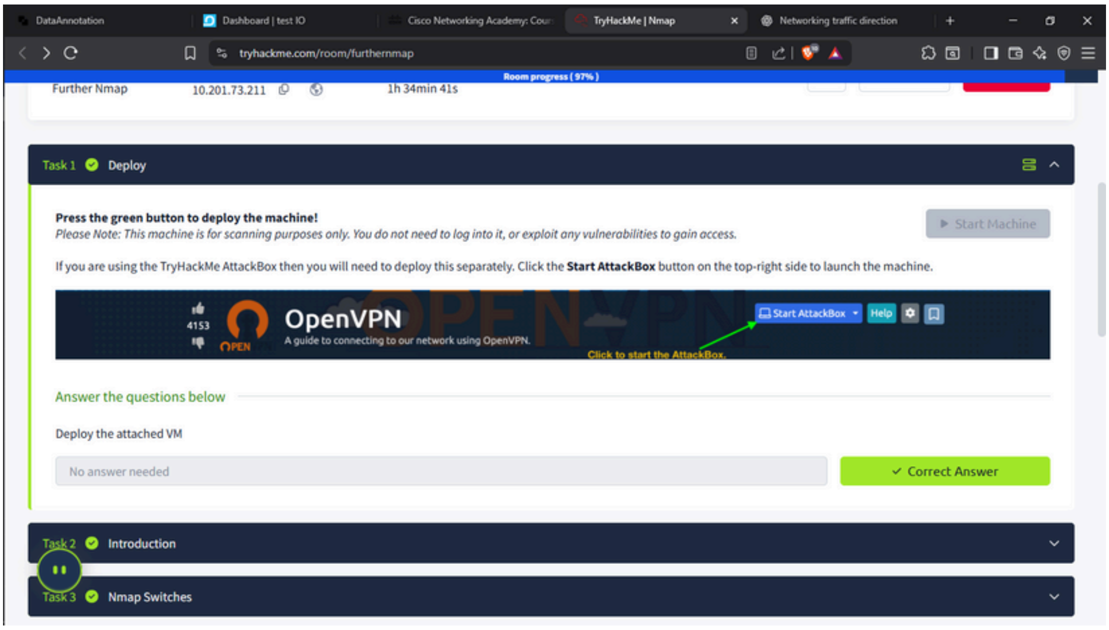
Cybersecurity Bootcamp – Network Scanning Report

Objective:

To analyze open ports and service behavior on a target system using various Nmap scanning techniques and Wireshark packet captures.

Images



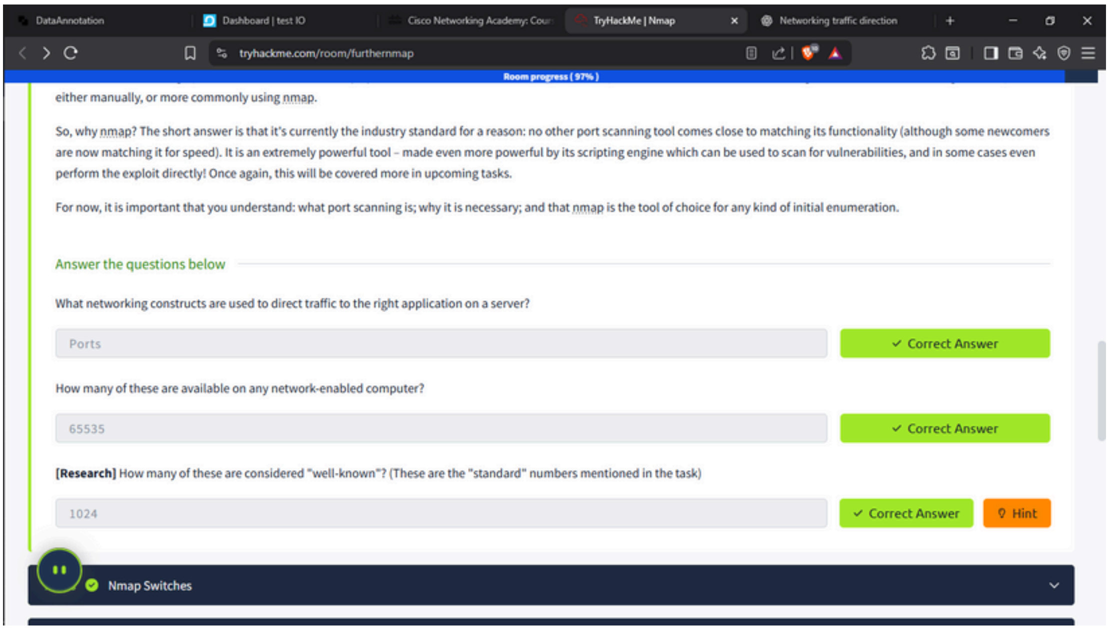
Press the green button to deploy the machine!
Please Note: This machine is for scanning purposes only. You do not need to log into it, or exploit any vulnerabilities to gain access.

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.

Answer the questions below

Deploy the attached VM

No answer needed



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

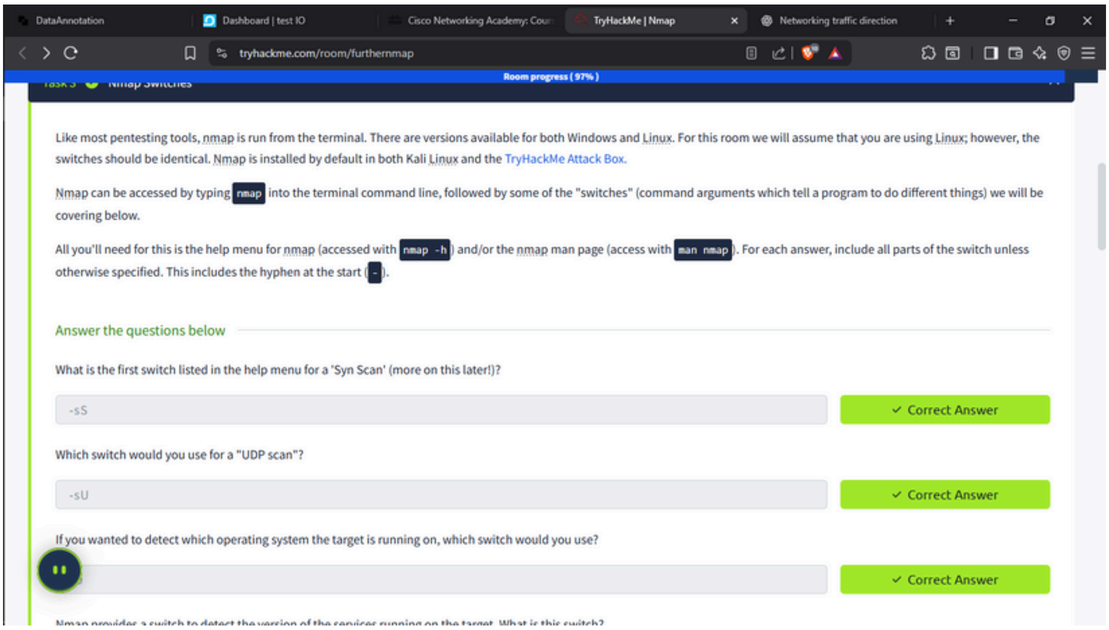
How many of these are available on any network-enabled computer?

65535

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

1024

Submitted By:
Edwin C George



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

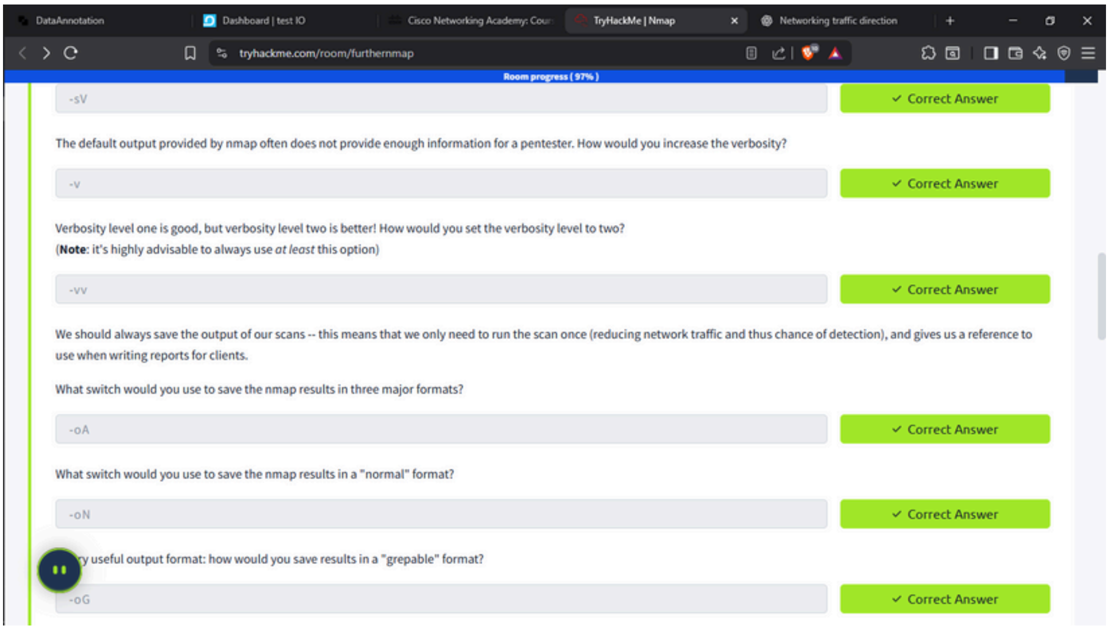
Which switch would you use for a "UDP scan"?

-sU ✓ Correct Answer

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

✓ Correct Answer

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


DataAnnotation
Dashboard | test IO
Cisco Networking Academy: Cour
TryHackMe | Nmap
Networking traffic direction
tryhackme.com/room/furthernmap
Room progress (97%)

-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?

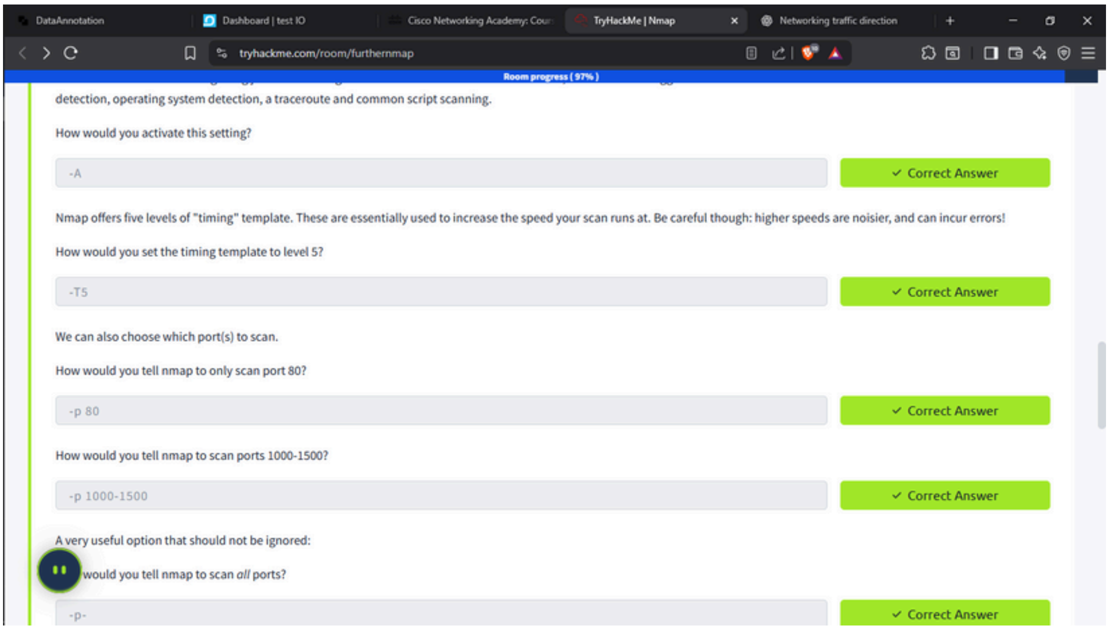
-oA ✓ Correct Answer

What switch would you use to save the nmap results in a "normal" format?

-oN ✓ Correct Answer

y useful output format: how would you save results in a "grepable" format?

-oG ✓ Correct Answer


DataAnnotation
Dashboard | test IO
Cisco Networking Academy: Cour
TryHackMe | Nmap
Networking traffic direction
tryhackme.com/room/furthernmap
Room progress (97%)

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

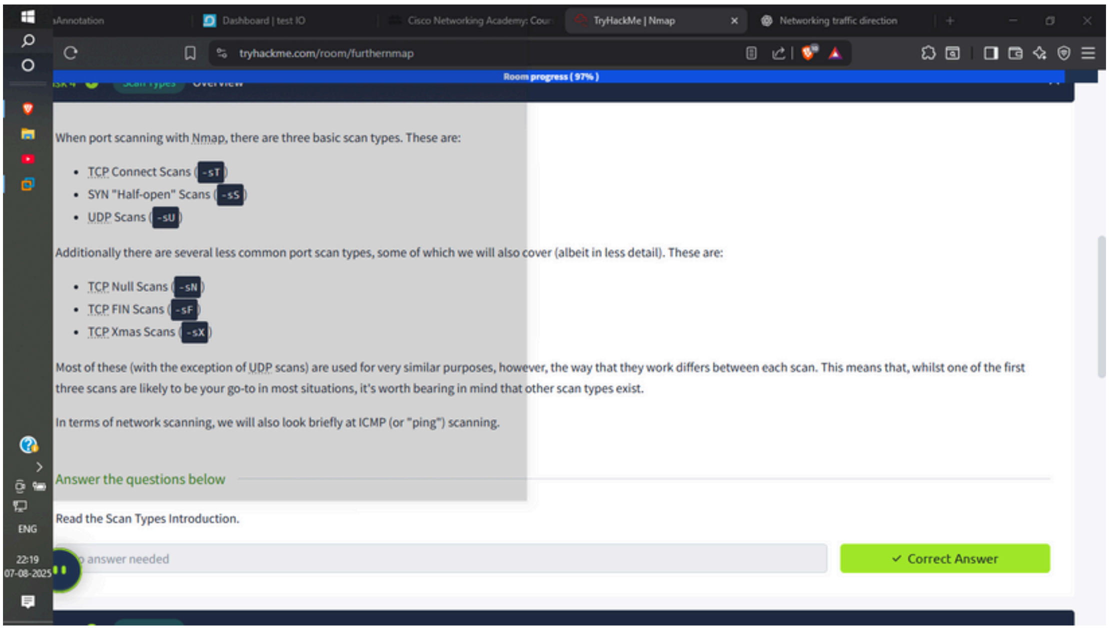
How would you tell nmap to scan ports 1000-1500?

-p 1000-1500 ✓ Correct Answer

A very useful option that should not be ignored:

would you tell nmap to scan *all* ports?

-p- ✓ Correct Answer


DataAnnotation
Dashboard | test IO
Cisco Networking Academy: Cour
TryHackMe | Nmap
Networking traffic direction
tryhackme.com/room/furthernmap
Room progress (97%)
ENG
22:19
07-08-2025

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`)

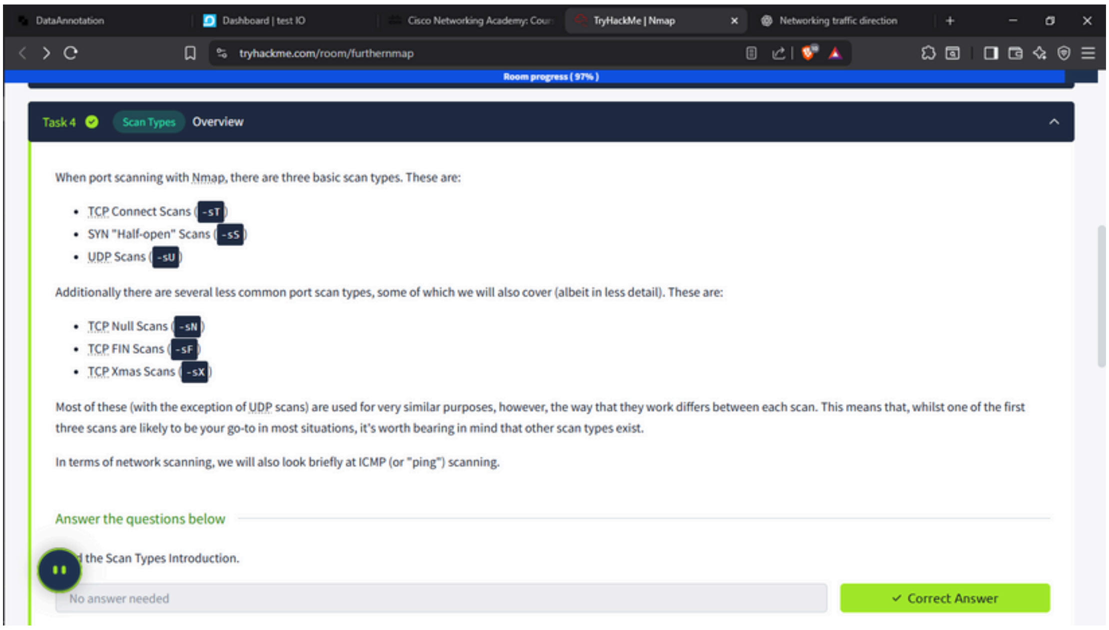
Most of these (with the exception of UDP scans) are used for very similar purposes, however, the way that they work differs between each scan. This means that, whilst one of the first three scans are likely to be your go-to in most situations, it's worth bearing in mind that other scan types exist.

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

Answer the questions below

Read the Scan Types Introduction.

o answer needed



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`)

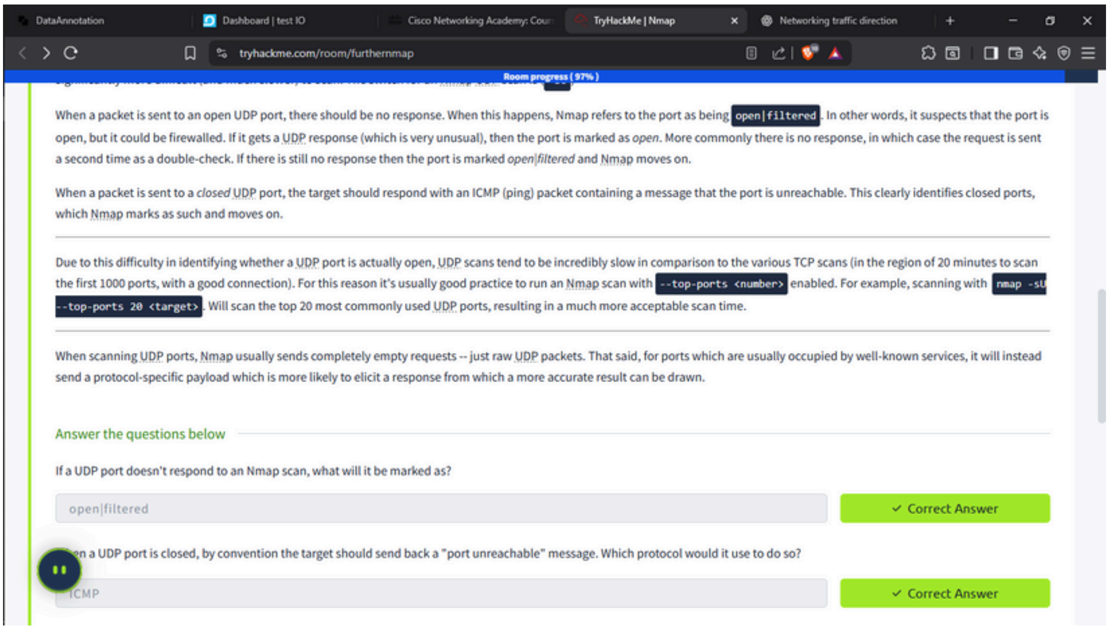
Most of these (with the exception of UDP scans) are used for very similar purposes, however, the way that they work differs between each scan. This means that, whilst one of the first three scans are likely to be your go-to in most situations, it's worth bearing in mind that other scan types exist.

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

Answer the questions below

d the Scan Types Introduction.

No answer needed


DataAnnotation
Dashboard | test IO
Cisco Networking Academy: Cour
TryHackMe | Nmap
Networking traffic direction
tryhackme.com/room/furthernmap
Room progress (97%)

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.

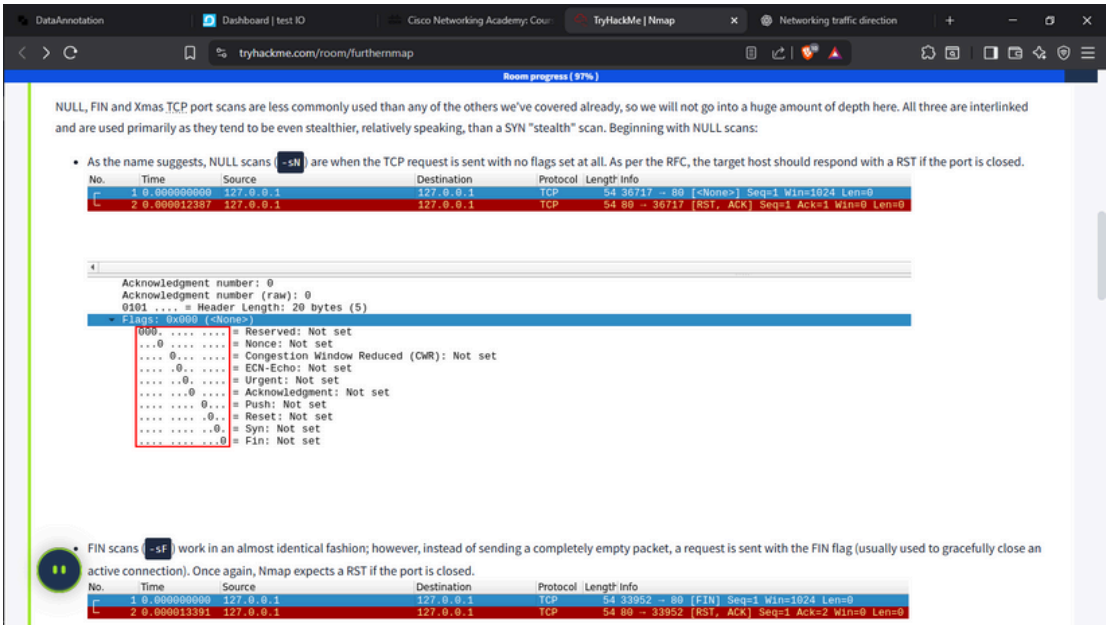
Answer the questions below

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

open|filtered

en 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


DataAnnotation
Dashboard | test IO
Cisco Networking Academy: Cour
TryHackMe | Nmap
Networking traffic direction
tryhackme.com/room/furthernmap
Room progress (97%)

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:

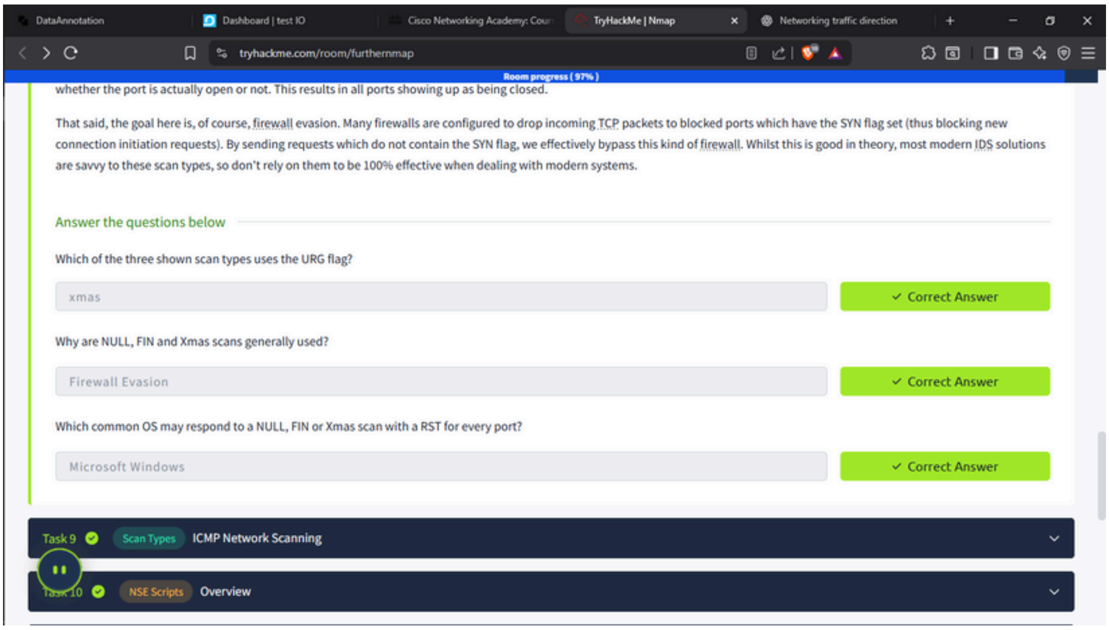
- As the name suggests, NULL scans (`-sN`) are when the TCP request is sent with no flags set at all. As per the RFC, the target host should respond with a RST if the port is closed.

```
No.   Time          Source       Destination   Protocol  Length Info
    1 0.000000000   127.0.0.1    127.0.0.1     TCP       54 36717 → 80 [<None>] Seq=1 Win=1024 Len=0
    2 0.000012387   127.0.0.1    127.0.0.1     TCP       54 80 → 36717 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
```

```
    Acknowledgment number: 0
    Acknowledgment number (raw): 0
    0101 .... = Header Length: 20 bytes (5)
  ▾ Flags: 0x000 (<None>)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...0 .... = Acknowledgment: Not set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
      .... .... ..0. = Syn: Not set
      .... .... ...0 = Fin: Not set
```

- FIN scans (`-sF`) work in an almost identical fashion; however, instead of sending a completely empty packet, a request is sent with the FIN flag (usually used to gracefully close an active connection). Once again, Nmap expects a RST if the port is closed.

```
No.   Time          Source       Destination   Protocol  Length Info
    1 0.000000000   127.0.0.1    127.0.0.1     TCP       54 33952 → 80 [FIN] Seq=1 Win=1024 Len=0
    2 0.000013391   127.0.0.1    127.0.0.1     TCP       54 80 → 33952 [RST, ACK] Seq=1 Ack=2 Win=0 Len=0
```


DataAnnotation
Dashboard | test IO
Cisco Networking Academy: Cour
TryHackMe | Nmap
Networking traffic direction
tryhackme.com/room/furthernmap
Room progress (97%)

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. Whilst 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?

xmas — ✓ Correct Answer

Why are NULL, FIN and Xmas scans generally used?

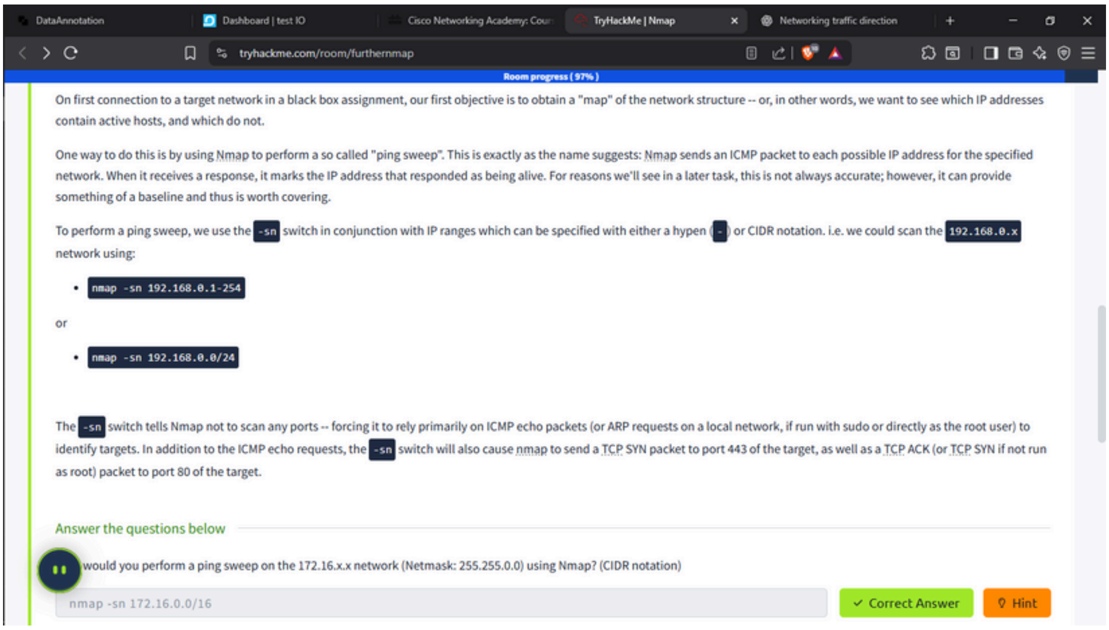
Firewall Evasion — ✓ Correct Answer

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

Microsoft Windows — ✓ Correct Answer

Task 9 ✓ Scan Types ICMP Network Scanning

Task 10 ✓ NSE Scripts Overview


DataAnnotation
Dashboard | test IO
Cisco Networking Academy: Cour
TryHackMe | Nmap
Networking traffic direction
tryhackme.com/room/furthernmap
Room progress (97%)

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

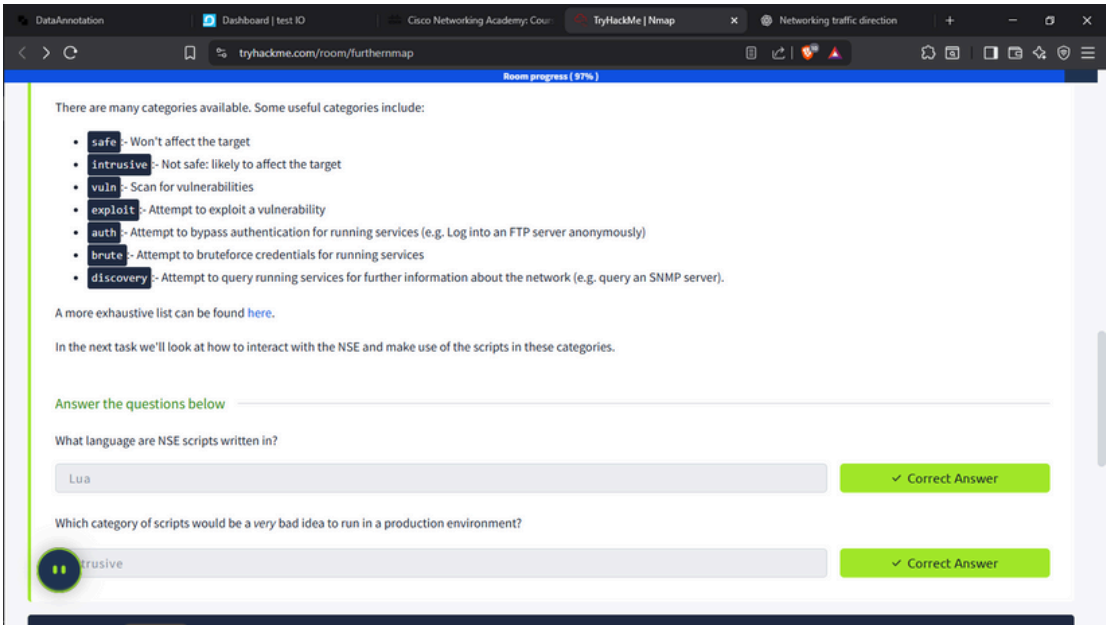
- `nmap -sn 192.168.0.0/24`

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.

Answer the questions below

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


DataAnnotation
Dashboard | test IO
Cisco Networking Academy: Cour
TryHackMe | Nmap
Networking traffic direction
tryhackme.com/room/furthernmap
Room progress (97%)

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?

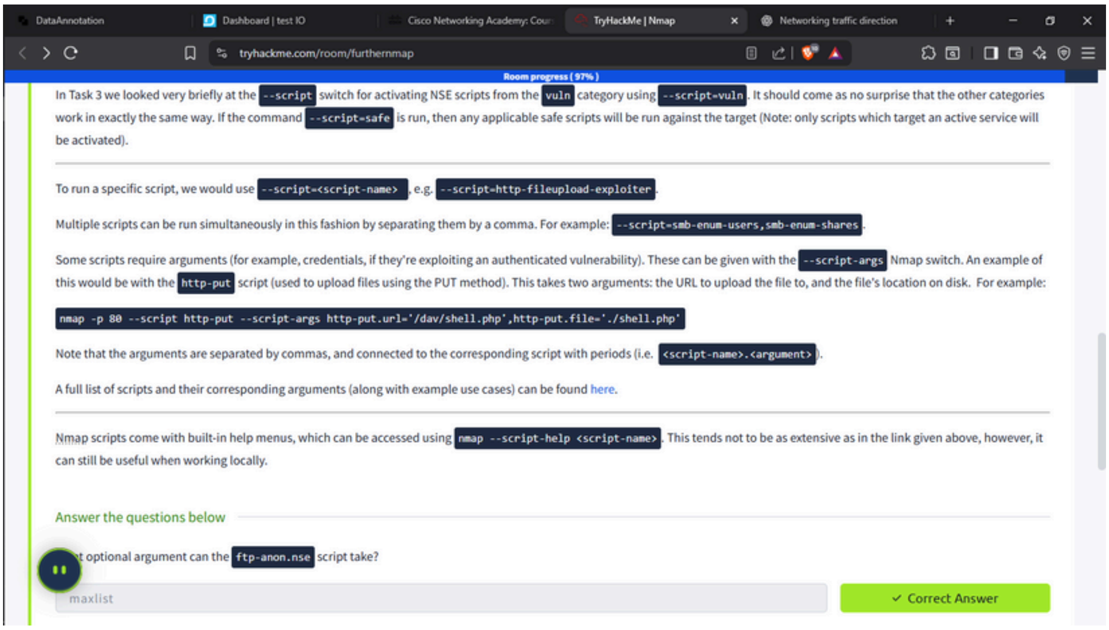
Lua

✓ Correct Answer

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

trusive

✓ Correct Answer


DataAnnotation
Dashboard | test IO
Cisco Networking Academy: Cour
TryHackMe | Nmap
Networking traffic direction
tryhackme.com/room/furthernmap
Room progress (97%)

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.

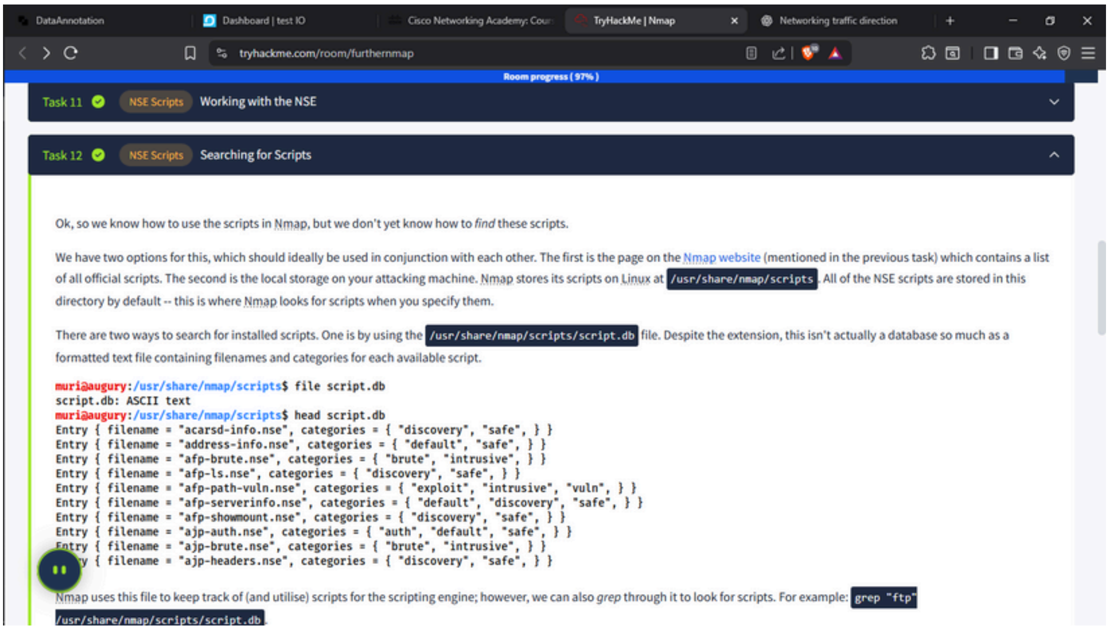
Nmap scripts come with built-in help menus, which can be accessed using `nmap --script-help <script-name>`. This tends not to be as extensive as in the link given above, however, it can still be useful when working locally.

Answer the questions below

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

maxlist

✓ Correct Answer


DataAnnotation
Dashboard | test IO
Cisco Networking Academy: Cour
TryHackMe | Nmap
Networking traffic direction
tryhackme.com/room/furthernmap
Room progress (97%)

Task 11 NSE Scripts Working with the NSE

Task 12 NSE Scripts Searching for Scripts

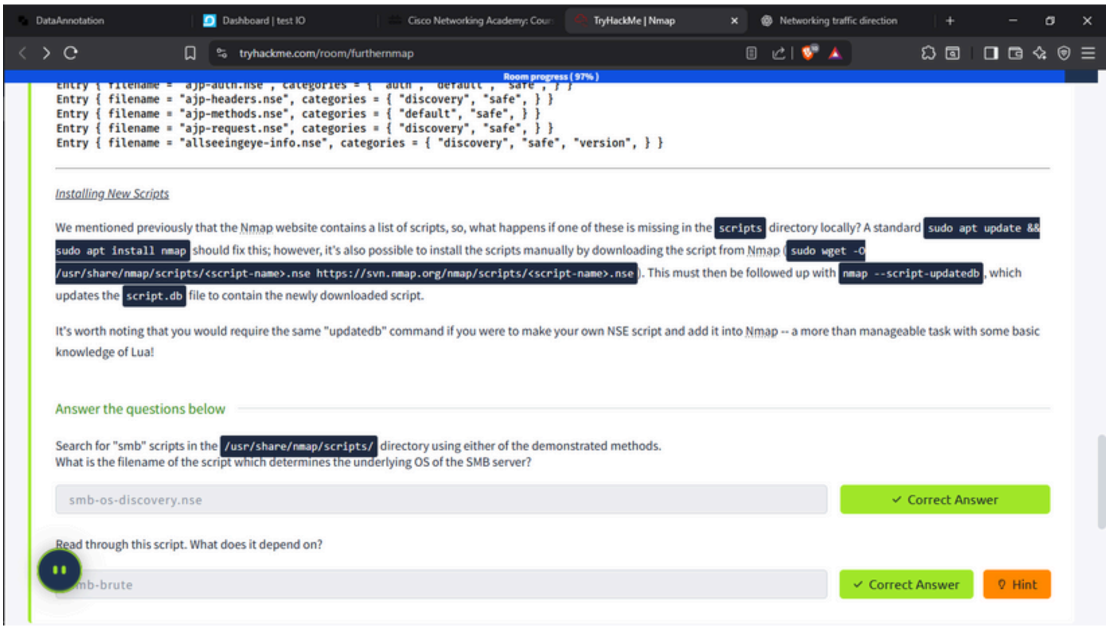
Ok, so we know how to use the scripts in Nmap, but we don't yet know how to *find* these scripts.

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 contains a list 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 in this 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 as 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", } }
Entry { filename = "ajp-auth.nse", categories = { "auth", "default", "safe", } }
Entry { filename = "ajp-brute.nse", categories = { "brute", "intrusive", } }
    y { 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" /usr/share/nmap/scripts/script.db`.


DataAnnotation
Dashboard | test IO
Cisco Networking Academy: Cour
TryHackMe | Nmap
Networking traffic direction
tryhackme.com/room/furthernmap
Room progress (97%)

```
Entry { filename = "ajp-auth.nse", categories = { "auth", "default", "safe", } }
Entry { filename = "ajp-headers.nse", categories = { "discovery", "safe", } }
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

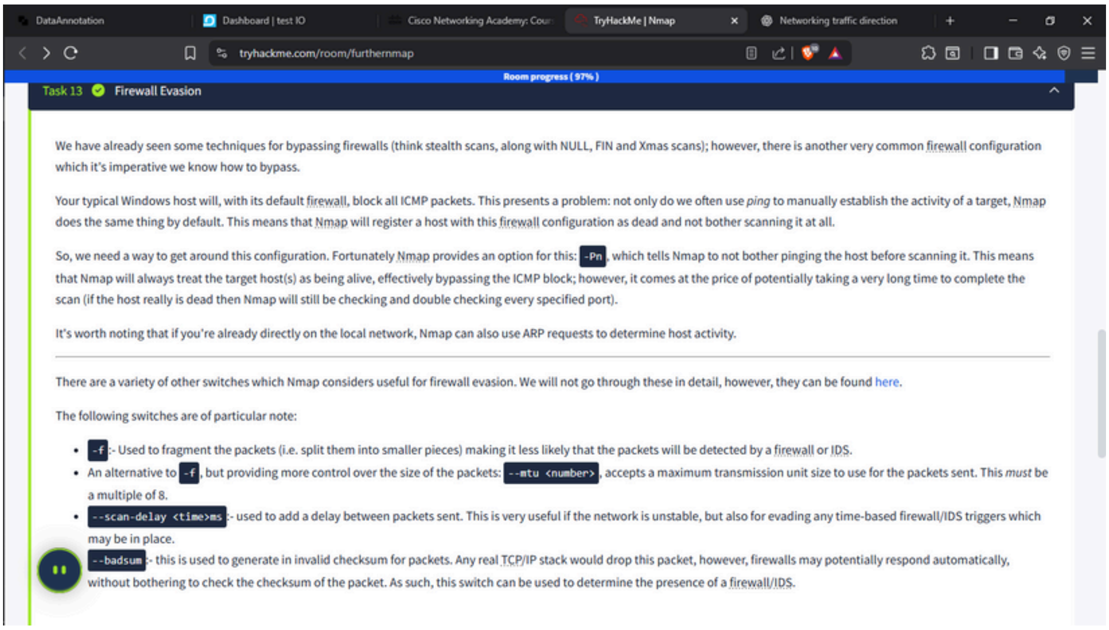
✓ Correct Answer

Read through this script. What does it depend on?

smb-brute

✓ Correct Answer

Hint


DataAnnotation
Dashboard | test IO
Cisco Networking Academy: Cour
TryHackMe | Nmap
Networking traffic direction
tryhackme.com/room/furthernmap
Room progress (97%)

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.

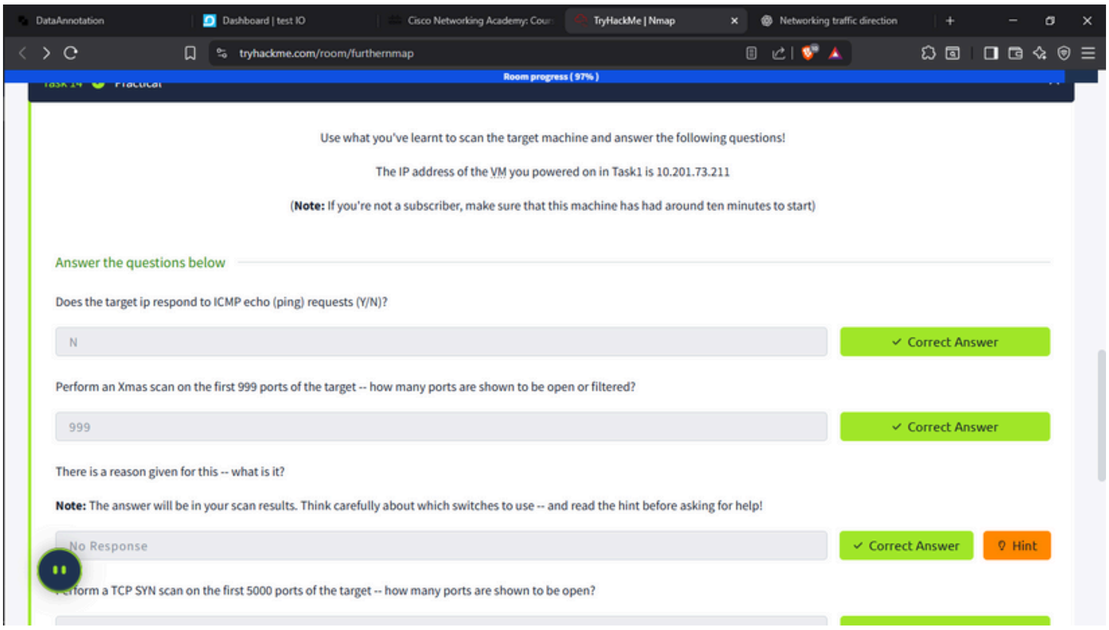
So, we need a way to get around this configuration. Fortunately Nmap provides an option for this: `-Pn`, which tells Nmap to not bother pinging the host before scanning it. This means that Nmap will always treat the target host(s) as being alive, effectively bypassing the ICMP block; however, it comes 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.


DataAnnotation
Dashboard | test IO
Cisco Networking Academy: Cour
TryHackMe | Nmap
Networking traffic direction
tryhackme.com/room/furthernmap
Room progress (97%)

Task 14 ✓ Practical

Use what you've learnt to scan the target machine and answer the following questions!

The IP address of the VM you powered on in Task1 is 10.201.73.211

(**Note:** If you're not a subscriber, make sure that this machine has had around ten minutes to start)

Answer the questions below

Does the target ip respond to ICMP echo (ping) requests (Y/N)?

N

✓ Correct Answer

Perform an Xmas scan on the first 999 ports of the target -- how many ports are shown to be open or filtered?

999

✓ Correct Answer

There is a reason given for this -- what is it?

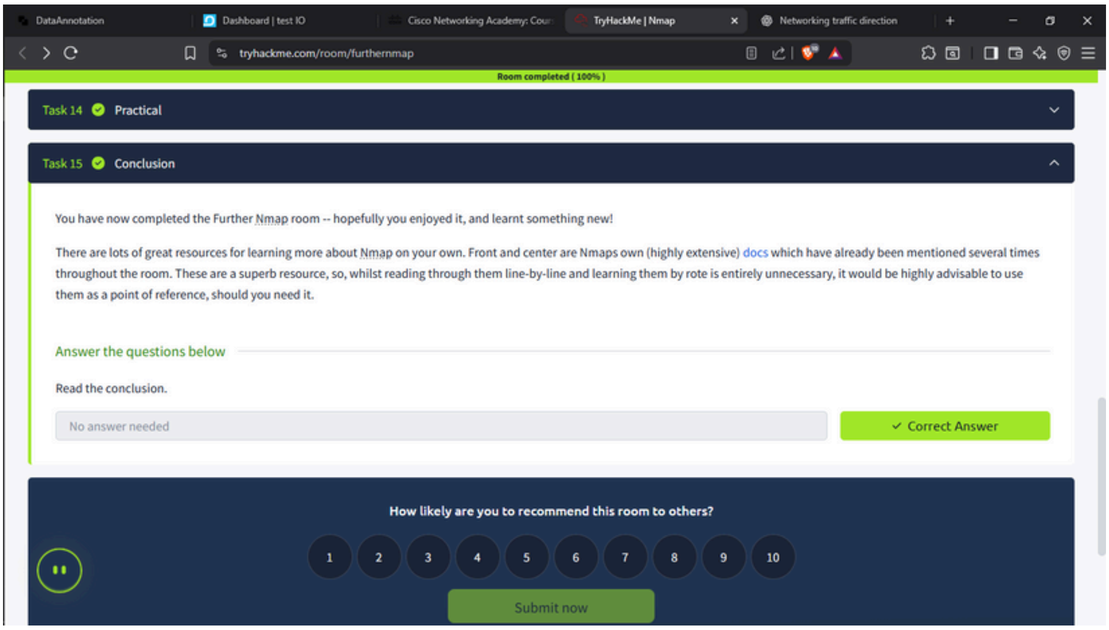
Note: The answer will be in your scan results. Think carefully about which switches to use -- and read the hint before asking for help!

No Response

✓ Correct Answer

Hint

Perform a TCP SYN scan on the first 5000 ports of the target -- how many ports are shown to be open?



You have now completed the Further Nmap room -- hopefully you enjoyed it, and learnt something new!

There are lots of great resources for learning more about Nmap on your own. Front and center are Nmaps own (highly extensive) docs which have already been mentioned several times throughout the room. These are a superb resource, so, whilst reading through them line-by-line and learning them by rote is entirely unnecessary, it would be highly advisable to use them as a point of reference, should you need it.

Answer the questions below

Read the conclusion.

No answer needed

Conclusion:

This lab demonstrated the use of Nmap and Wireshark to perform detailed network enumeration. Understanding scan results and live packet capture is essential in penetration testing and vulnerability assessments.