Network Enumeration with Nmap

Here is the link of my completion of the module

https://academy.hackthebox.com/achievement/1917469/19

Introduction

Enumeration is critical and foundation phase in penetration testing and its goal isn't direct system access but discovering how access could be achieved by identifying vulnerable points and gathering detailed information about services and protocols.

Tools are helpful but true effectiveness lies in understanding the information they reveal and knowing how to interact with target services. Success depends on deep knowledge of technologies,protocols,service behavior and communication syntax.

Main targets during enumeration are:

(i) Functions and/or resources that allows us to interact with the target and/or provide additional information.

(ii) Information that provides us with even more important information to access our target.

Mis-configurations are common causes of vulnerabilities often due to over reliance on firewalls,GPOs or updates by the administrators. While tools help,manual efforts are crucial especially when tools miss open or filtered ports due to timeouts or default configurations. Many people get stuck not because of missing tools but because they lack understanding of how the target services works I.e most people understand that they haven't tried all the tools to get the information they need but most of the time, however, it's not the tools we haven't tried, but rather the fact that we don't know how to interact with the service and what's relevant.

Introduction to Nmap

Nmap is an open source network scanning and security auditing tool written in C,C++,Python and Lua and is widely used to discover hosts and services on a network,identify application versions and OS ,analyze firewalls,IDS and packet filter configurations.

Nmap is commonly used for:

>Network security auditing.

>Penetration testing simulations

>Firewall and IDS configuration checks

>Network mapping and response analysis

>Identifying open ports

>Vulnerability assessment

Nmap uses various scanning techniques including:

(i) Host discovery using -sn flag

(ii) Port scanning for open TCP/UDP ports you can use -sS flag for TCP SYN scan, -sT for TCP scan connect,sU for UDP port scan and -p for specific port range scan.

(iii) Services enumeration and detection using -sV flag.

(iv) OS detection using -O flag but requires root privileges.

(v) Scriptable interaction with the target services

Command	Description
nmap --script=default <target>	Run default scripts
nmap --script=vuln <target>	Run vulnerability detection scripts
nmap --script=http-* <target>	Run all HTTP-related scripts
nmap --script-help <script>	Get help on a specific script

(vi) Firewall and evasion

Command	Description
nmap -D RND:10 <target>	Use decoys to mask source
nmap -f <target>	Fragment packets
nmap --source-port 53 <target>	Spoof source port (e.g., DNS)
nmap -Pn <target>	Skip host discovery (assume host is up)

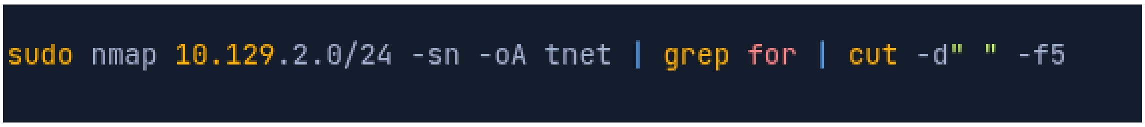
The syntax for Nmap look like this:

nmap <scan types> <options> <target>

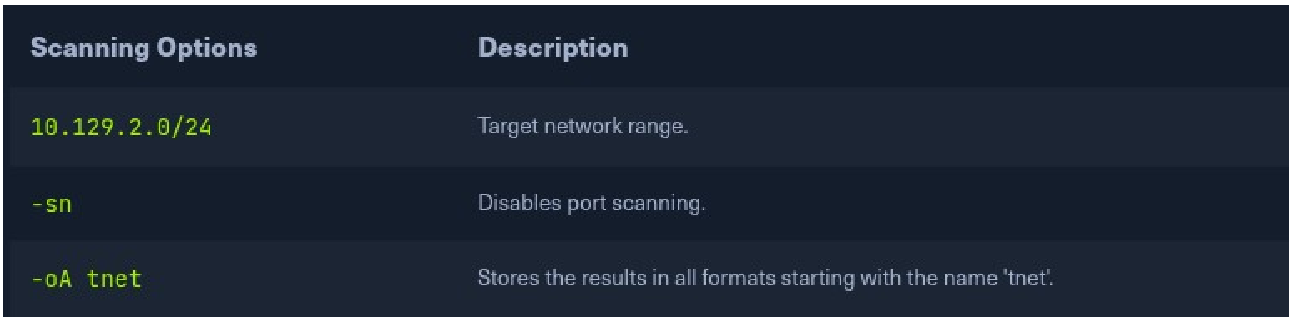
Host Discovery

Before starting an internal penetration test it is important to find out which systems are online and Nmap is a useful tool for this. It can check if devices are active using different methods and one of the best way is by sending ICMP echo requests(pings). It is also important to save all scan results as this will help with documentation,comparing results from different tools and tracking changes over time.

(i) Scan Network Range



```
sudo nmap 10.129.2.0/24 -sn -oA tnet | grep for | cut -d" " -f5
```

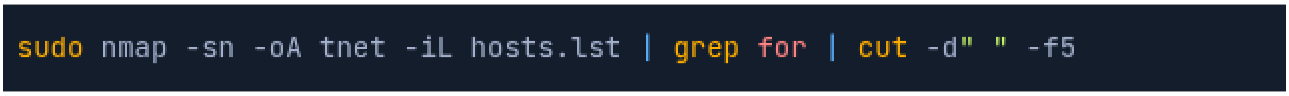


Scanning Options	Description
10.129.2.0/24	Target network range.
-sn	Disables port scanning.
-oA tnet	Stores the results in all formats starting with the name 'tnet'.

This scanning method will work only if the firewalls of the hosts allow it.

(ii) Scan IP List

Nmap gives the option of working with lists and reading the hosts from the list instead of manually defining or typing them in but during an internal penetration testing it is uncommon to be provided with an IP List with the host we need to test.



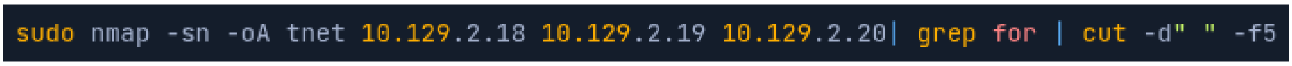
```
sudo nmap -sn -oA tnet -iL hosts.lst | grep for | cut -d" " -f5
```

The -iL option performs defined scan against targets in provided “hosts.lst” list.

Scans only host in the list and checks their availability.

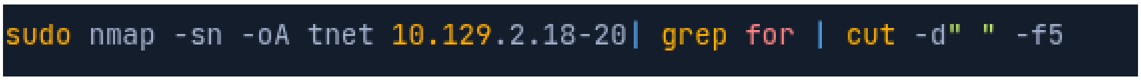
(iii) Scan multiple IPs

Lists of IPs: sudo nmap -sn -oA tnet 10.129.2.18 10.129.2.19 10.129.2.20 | grep for | cut -d" " -f5



```
sudo nmap -sn -oA tnet 10.129.2.18 10.129.2.19 10.129.2.20| grep for | cut -d" " -f5
```

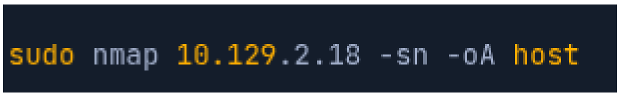
IP range: sudo nmap -sn -oA tnet 10.129.2.18-20 | grep for | cut -d" " -f5



```
sudo nmap -sn -oA tnet 10.129.2.18-20| grep for | cut -d" " -f5
```

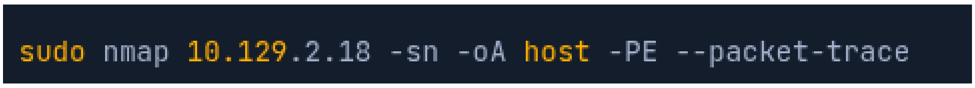
(iv) Scan single IP

Before doing this we need to determine if it is alive or not. The below command save the results in all format starting with the name “host” as show using -oA.



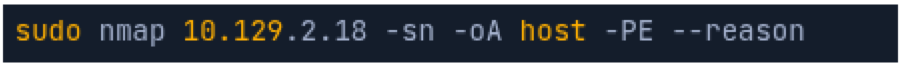
```
sudo nmap 10.129.2.18 -sn -oA host
```

The -PE option below forces ICMP Echo requests and --packet-trace shows packet sent and received.



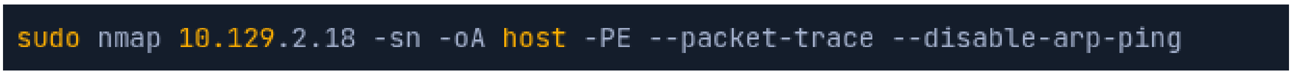
```
sudo nmap 10.129.2.18 -sn -oA host -PE --packet-trace
```

To display why host is marked alive we can you use the --reason option to show detection reasons.



```
sudo nmap 10.129.2.18 -sn -oA host -PE --reason
```

To disable ARP requests and scan our target with desired ICMP echo requests we can disable ARP pings by setting the “--disable-arp-ping” option. This prevents ARP discovery helpful for learning or bypassing behavior and ensure only ICMP is used for host detection.

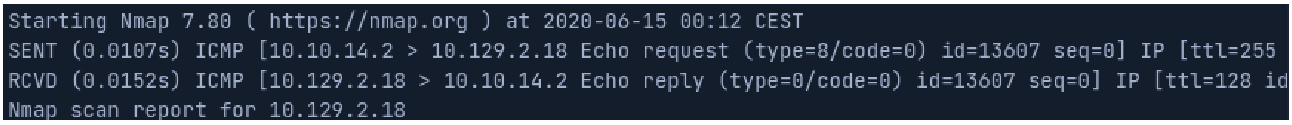


```
sudo nmap 10.129.2.18 -sn -oA host -PE --packet-trace --disable-arp-ping
```

Question:

Based on the last result, find out which operating system it belongs to. Submit the name of the operating system as result.

The answer is window as ttl response received was 128 which matches window’s default.



```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-15 00:12 CEST
SENT (0.0107s) ICMP [10.10.14.2 > 10.129.2.18 Echo request (type=8/code=0) id=13607 seq=0] IP [ttl=255
RCVD (0.0152s) ICMP [10.129.2.18 > 10.10.14.2 Echo reply (type=0/code=0) id=13607 seq=0] IP [ttl=128 id
Nmap scan report for 10.129.2.18
```

OS	Default TTL
Windows	**128**
Linux	64
Cisco	255
BSD/Unix	64–255
macOS	64

Host and Port Scanning

To use any scanning tool effectively it is important to understand how it works and how it processes information. This is after identifying the target is alive and the information we need is:

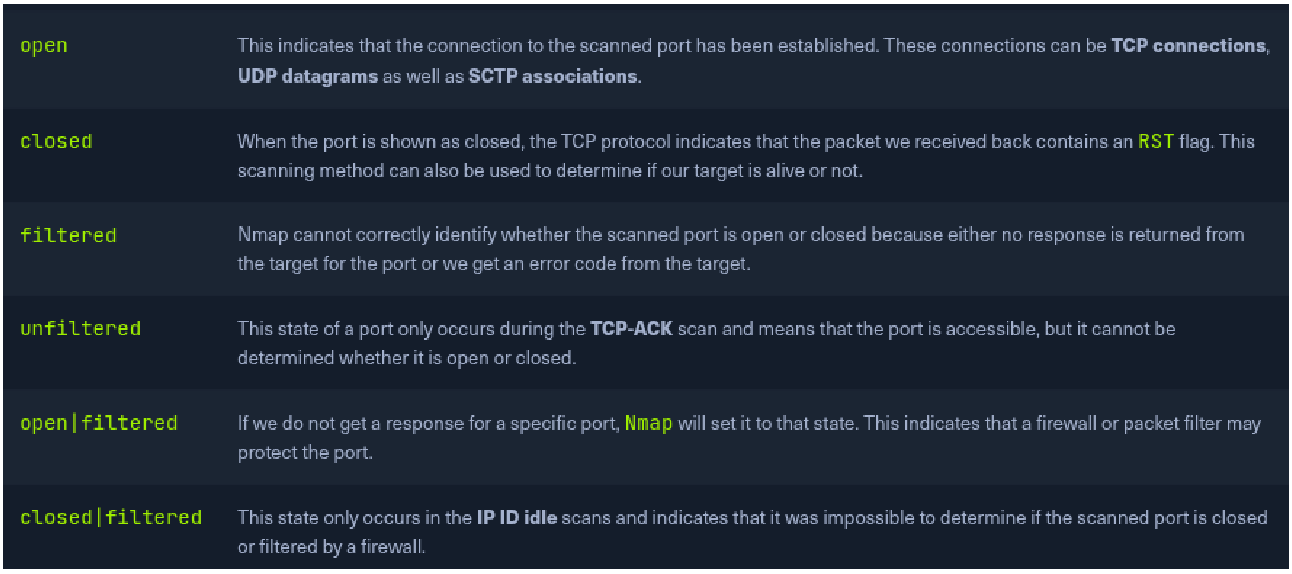
Open ports and its services

Service versions

Information that the services provided

Operating system

There are a total of 6 different states for a scanned port we can obtain:



`open`	This indicates that the connection to the scanned port has been established. These connections can be **TCP connections**, **UDP datagrams** as well as **SCTP associations**.
`closed`	When the port is shown as closed, the TCP protocol indicates that the packet we received back contains an `RST` flag. This scanning method can also be used to determine if our target is alive or not.
`filtered`	Nmap cannot correctly identify whether the scanned port is open or closed because either no response is returned from the target for the port or we get an error code from the target.
`unfiltered`	This state of a port only occurs during the **TCP-ACK** scan and means that the port is accessible, but it cannot be determined whether it is open or closed.
`open|filtered`	If we do not get a response for a specific port, `Nmap` will set it to that state. This indicates that a firewall or packet filter may protect the port.
`closed|filtered`	This state only occurs in the **IP ID idle** scans and indicates that it was impossible to determine if the scanned port is closed or filtered by a firewall.

The Nmap TCP connect scan (-sT) uses the TCP three-way handshake to determine if a specific port on a target host is open or closed. The scan sends an SYN packet to the target port and waits for a response. If it responds with SYN-ACK it is considered open and RST it is closed.

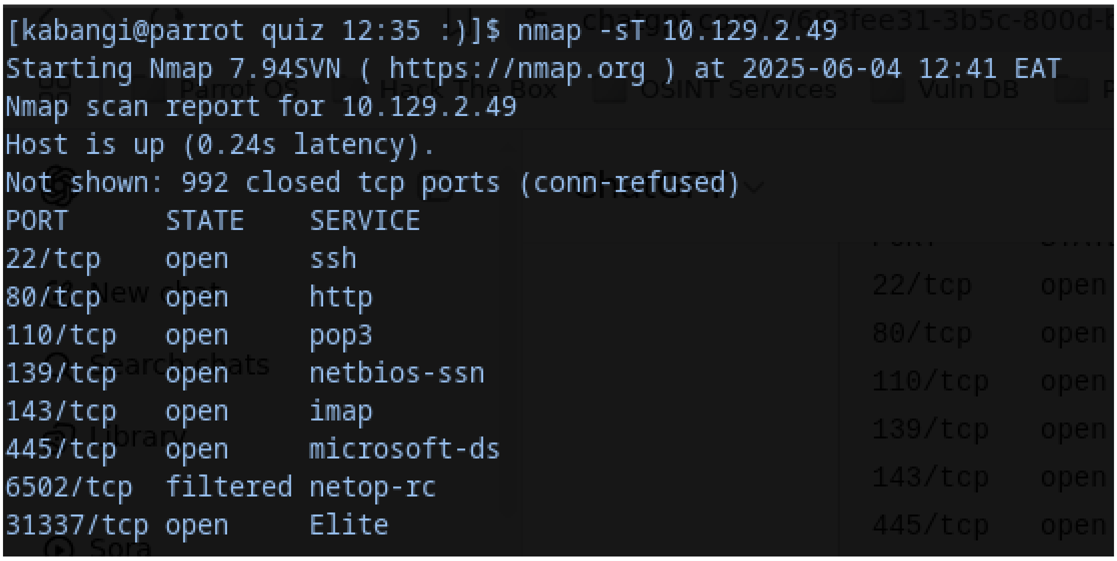
The benefit of using TCP Connect scan is that it is very accurate,good for testing services and bypasses some personal firewalls. Its downside is that it is easy to be detected by firewalls and logging systems and it is slower as it requires the scanner to wait for a response from the target after each packet it sends.

Questions:

Find all TCP ports on your target. Submit the total number of found TCP ports as the answer.

I counted only the open ports as they are accessible and open for communication.

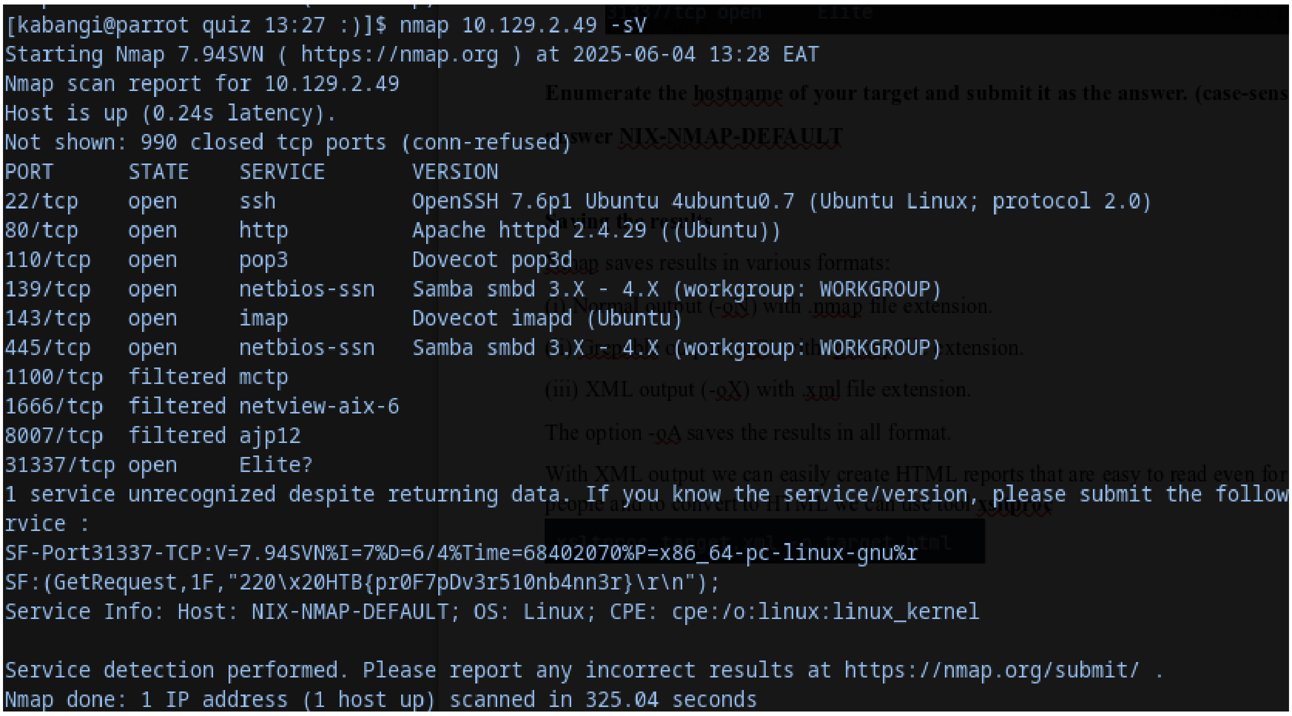
Answer 7



```
[kabangi@parrot quiz 12:35 :)]$ nmap -sT 10.129.2.49
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-04 12:41 EAT
Nmap scan report for 10.129.2.49
Host is up (0.24s latency).
Not shown: 992 closed tcp ports (conn-refused)
PORT      STATE    SERVICE
22/tcp    open     ssh
80/tcp    open     http
110/tcp   open     pop3
139/tcp   open     netbios-ssn
143/tcp   open     imap
445/tcp   open     microsoft-ds
6502/tcp  filtered netop-rc
31337/tcp open     Elite
```

Enumerate the hostname of your target and submit it as the answer. (case-sensitive)

answer NIX-NMAP-DEFAULT shown on the service info part.



```
[kabangi@parrot quiz 13:27 :)]$ nmap 10.129.2.49 -sV
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-04 13:28 EAT
Nmap scan report for 10.129.2.49
Host is up (0.24s latency).
Not shown: 990 closed tcp ports (conn-refused)
PORT      STATE    SERVICE       VERSION
22/tcp    open     ssh           OpenSSH 7.6p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux; protocol 2.0)
80/tcp    open     http          Apache httpd 2.4.29 ((Ubuntu))
110/tcp   open     pop3          Dovecot pop3d
139/tcp   open     netbios-ssn   Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
143/tcp   open     imap          Dovecot imapd (Ubuntu)
445/tcp   open     netbios-ssn   Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
1100/tcp  filtered mctp
1666/tcp  filtered netview-aix-6
8007/tcp  filtered ajp12
31337/tcp open     Elite?
1 service unrecognized despite returning data. If you know the service/version, please submit the follow
rvice :
SF-Port31337-TCP:V=7.94SVN%I=7%D=6/4%Time=68402070%P=x86_64-pc-linux-gnu%r
SF:(GetRequest,1F,"220\x20HTB{pr0F7pDv3r510nb4nn3r}\r\n");
Service Info: Host: NIX-NMAP-DEFAULT; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 325.04 seconds
```

Saving the results

Nmap saves results in various formats:

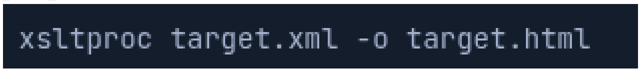
(i) Normal output (-oN) with .nmap file extension.

(ii) Grepable output (-oG) with .gnmap file extension.

(iii) XML output (-oX) with .xml file extension.

The option -oA saves the results in all format.

With XML output we can easily create HTML reports that are easy to read even for non technical people and to convert to HTML we can use tool **xsltproc**

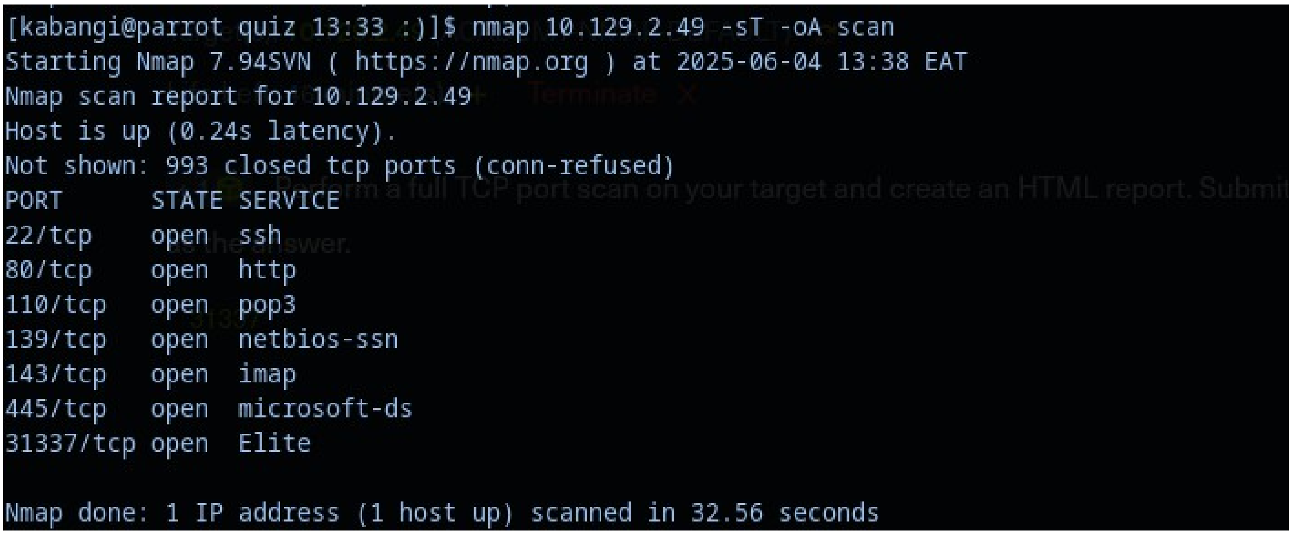


```
xsltproc target.xml -o target.html
```

Question:

Perform a full TCP port scan on your target and create an HTML report. Submit the number of the highest port as the answer.

31337 is the answer



```
[kabangi@parrot quiz 13:33 :)]$ nmap 10.129.2.49 -sT -oA scan
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-04 13:38 EAT
Nmap scan report for 10.129.2.49
Host is up (0.24s latency).
Not shown: 993 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
110/tcp   open  pop3
139/tcp   open  netbios-ssn
143/tcp   open  imap
445/tcp   open  microsoft-ds
31337/tcp open  Elite

Nmap done: 1 IP address (1 host up) scanned in 32.56 seconds
```

Service Enumeration

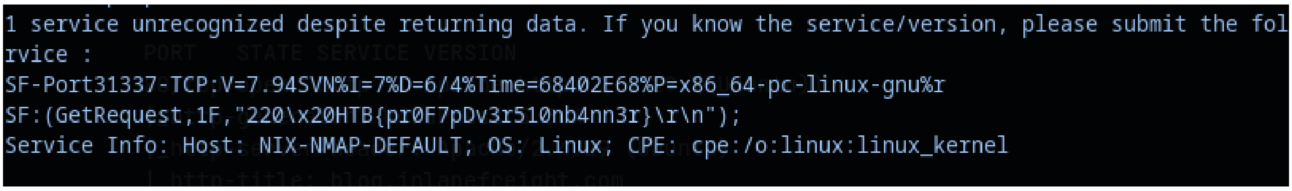
To show all open ports we use -p- and their versions -sV. A full port scan can take quite a long time so as to identify the progress we can use --stats-every=5s option.

As nmap shows the summary of the scan to increase its verbosity we can use the -v option which will show open ports directly when Nmap detects them.

Question:

Enumerate all ports and their services. One of the services contains the flag you have to submit as the answer.

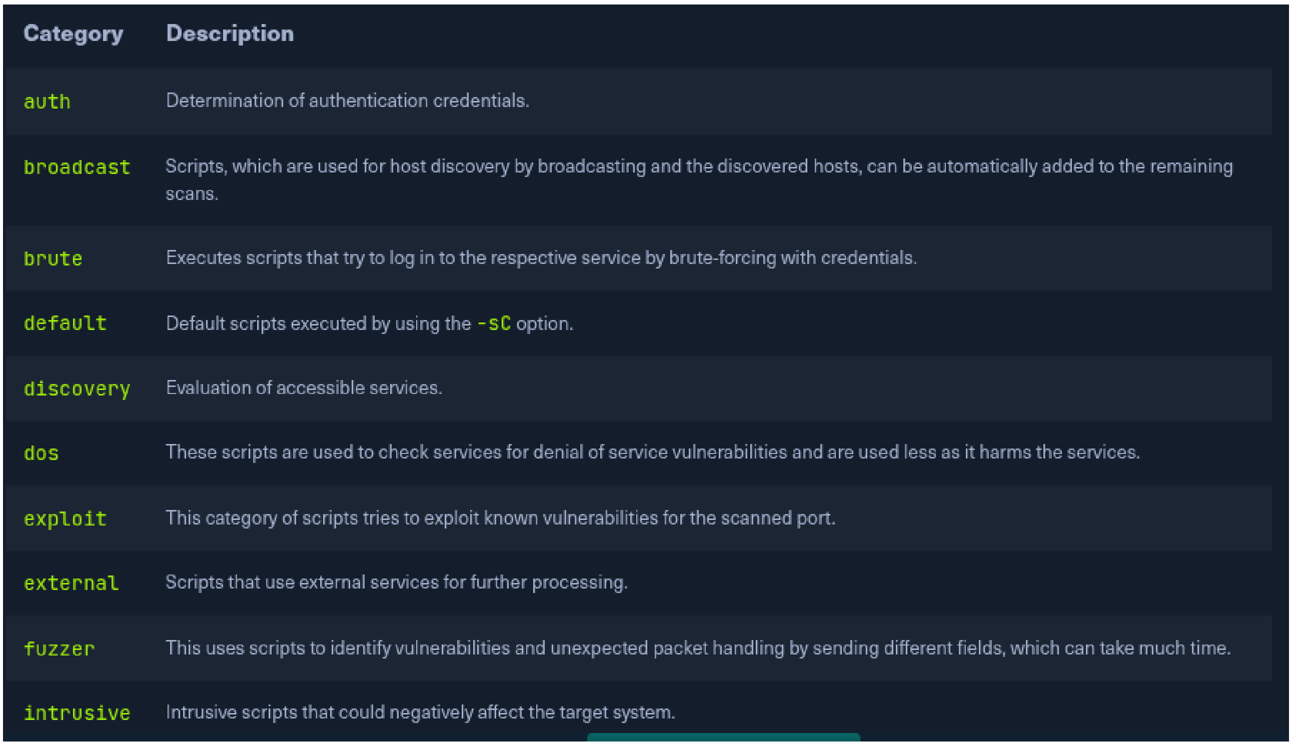
Answer HTB{pr0F7pDv3r510nb4nn3r}

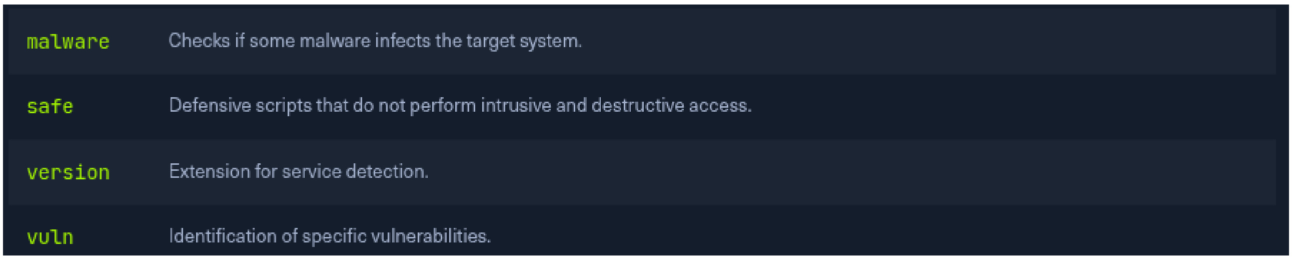


```
1 service unrecognized despite returning data. If you know the service/version, please submit the fol
rvice :
SF-Port31337-TCP:V=7.94SVN%I=7%D=6/4%Time=68402E68%P=x86_64-pc-linux-gnu%r
SF:(GetRequest,1F,"220\x20HTB{pr0F7pDv3r510nb4nn3r}\r\n");
Service Info: Host: NIX-NMAP-DEFAULT; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Nmap Scripting Engine(NSE)

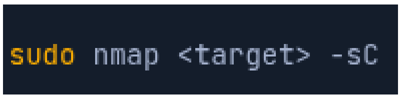
NSE provides with the possibility to create scripts in Lua for interaction with certain services and is divided into 14 categories.





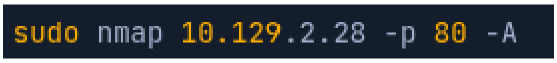
Category	Description
`auth`	Determination of authentication credentials.
`broadcast`	Scripts, which are used for host discovery by broadcasting and the discovered hosts, can be automatically added to the remaining scans.
`brute`	Executes scripts that try to log in to the respective service by brute-forcing with credentials.
`default`	Default scripts executed by using the `-sC` option.
`discovery`	Evaluation of accessible services.
`dos`	These scripts are used to check services for denial of service vulnerabilities and are used less as it harms the services.
`exploit`	This category of scripts tries to exploit known vulnerabilities for the scanned port.
`external`	Scripts that use external services for further processing.
`fuzzer`	This uses scripts to identify vulnerabilities and unexpected packet handling by sending different fields, which can take much time.
`intrusive`	Intrusive scripts that could negatively affect the target system.
`malware`	Checks if some malware infects the target system.
`safe`	Defensive scripts that do not perform intrusive and destructive access.
`version`	Extension for service detection.
`vuln`	Identification of specific vulnerabilities.

The default script commonly used is -sC option.



```
sudo nmap <target> -sC
```

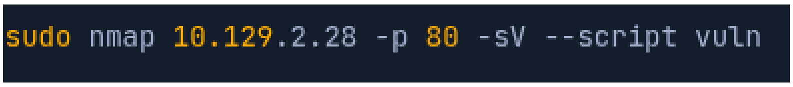
Nmap also gives us option to scan our target with the aggressive option(-A). This scans the target with multiple options as service detection(-sV),OS detection(-O),traceroute(--traceroute)and with the default NSE script(-sC).



```
sudo nmap 10.129.2.28 -p 80 -A
```

The option -p 80 specifies port 80 is the one being scanned.

To see what information and vulnerabilities are on port 80 we can use the **vuln** category from NSE.



```
sudo nmap 10.129.2.28 -p 80 -sV --script vuln
```

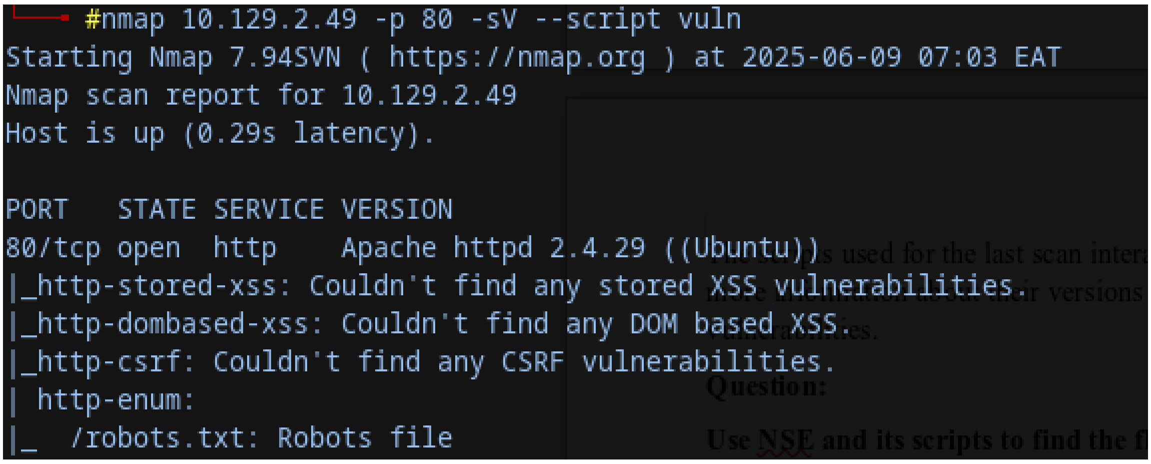
The scripts used for the last scan interact with the webserver and its web application to find out more information about their versions and check various databases to see if there are known vulnerabilities.

Question:

Use NSE and its scripts to find the flag that one of the services contain and submit it as the answer.

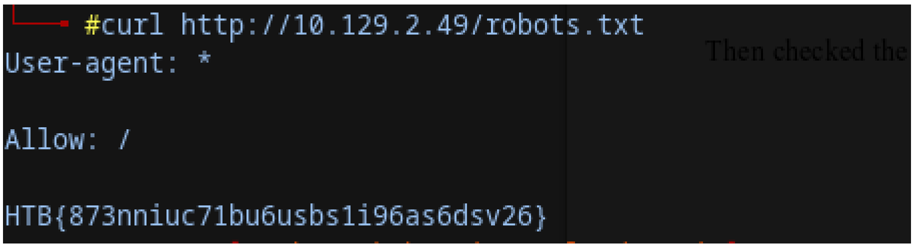
Answer HTB{873nniuc71bu6usbs1i96as6dsv26}

First ran the script vuln category on port 80 and came across robots.txt



```
└──╼ #nmap 10.129.2.49 -p 80 -sV --script vuln
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-09 07:03 EAT
Nmap scan report for 10.129.2.49
Host is up (0.29s latency).

PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-csrf: Couldn't find any CSRF vulnerabilities.
| http-enum:
|_  /robots.txt: Robots file
```

Then checked the content in the robots.txt and found the flag.



```
└──╼ #curl http://10.129.2.49/robots.txt
User-agent: *

Allow: /

HTB{873nniuc71bu6usbs1i96as6dsv26}
```

Firewall and IDS/IPS Evasion

Nmap provides various techniques to evade firewalls and intrusion detection/prevention systems such as packet fragmentation and decoy scans.

Firewalls monitor and control incoming and outgoing network traffic based on pre-defined rules. They can block, ignore, or allow packets to protect systems from unauthorized access.

IDS detects and reports suspicious activity, while IPS goes further by actively blocking such threats. Both rely on signature and pattern matching.

Filtered Ports in Nmap often indicate firewall rules at play. Firewalls may drop packets (no response) or reject them (RST flag or ICMP errors like Host Unreachable or Port Prohibited).

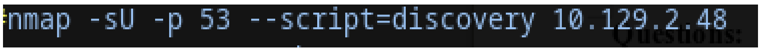
TCP ACK Scans (-sA) are harder for firewalls and IDS/IPS to detect since they only send ACK flags, mimicking part of an existing connection. These are more likely to pass through firewalls than SYN or Connect scans, which look like new connection attempts.

Questions:

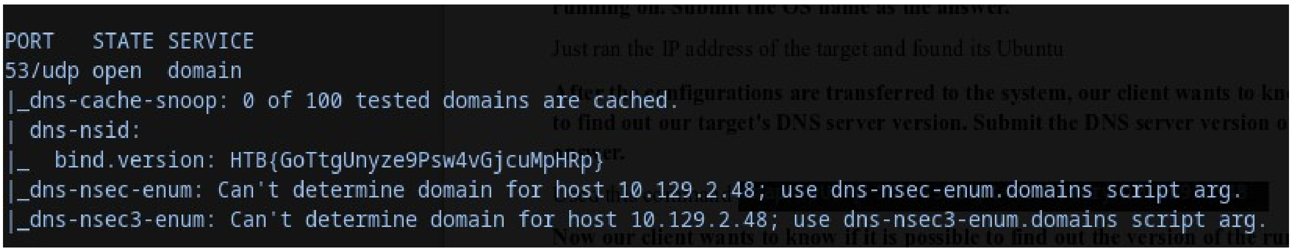
Our client wants to know if we can identify which operating system their provided machine is running on. Submit the OS name as the answer.

Just ran the IP address of the target and found its Ubuntu

After the configurations are transferred to the system, our client wants to know if it is possible to find out our target's DNS server version. Submit the DNS server version of the target as the answer.



Used this command `nmap -sU -p 53 --script=discovery 10.129.2.48` to perform scanning on UDP(-sU) on port 53 default port for DNS services using NSE script discovery and found the flag .

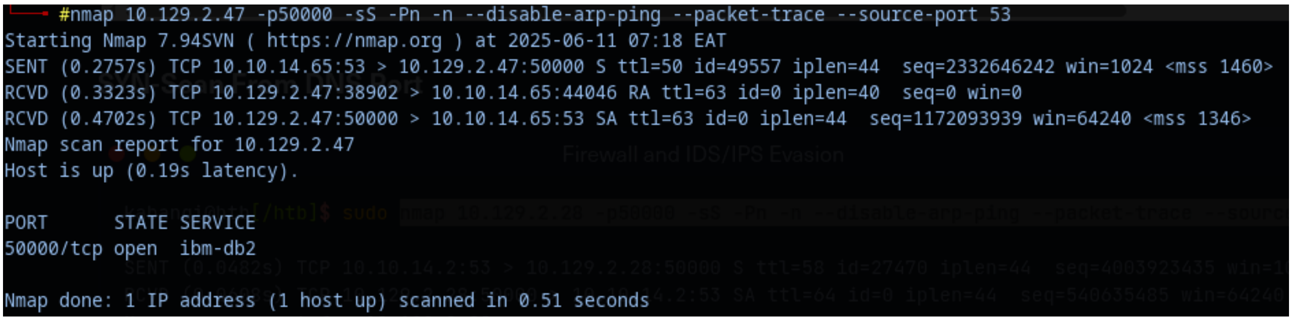


```
PORT   STATE SERVICE
53/udp open  domain
|_dns-cache-snoop: 0 of 100 tested domains are cached.
| dns-nsid:
|_  bind.version: HTB{GoTtgUnyze9Psw4vGjcuMpHRp}
|_dns-nsec-enum: Can't determine domain for host 10.129.2.48; use dns-nsec-enum.domains script arg.
|_dns-nsec3-enum: Can't determine domain for host 10.129.2.48; use dns-nsec3-enum.domains script arg.
```

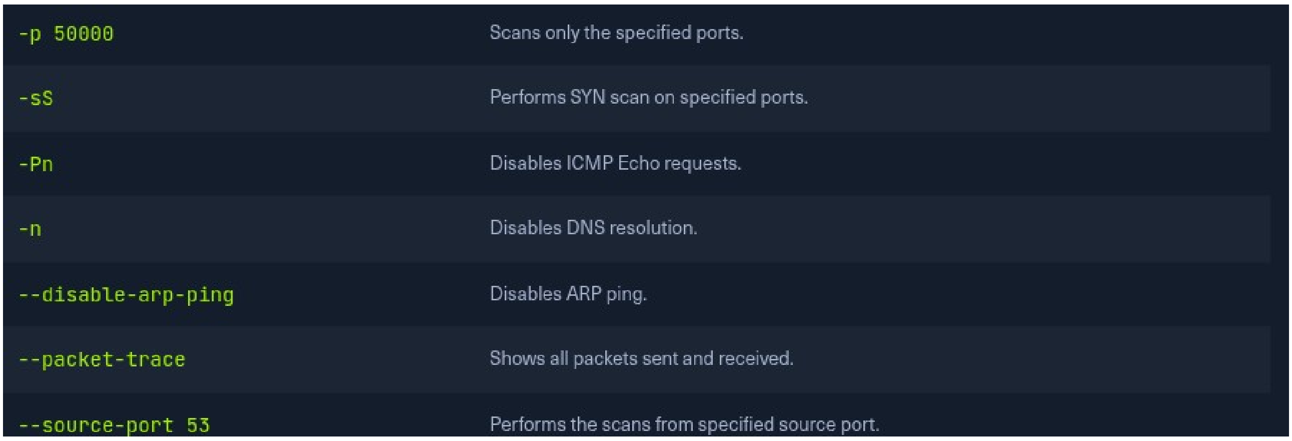
Now our client wants to know if it is possible to find out the version of the running services. Identify the version of service our client was talking about and submit the flag as the answer.

Answer is HTB{kjnsdf2n982n1827eh76238s98di1w6}

To solve this I will scan SYN from DNS port

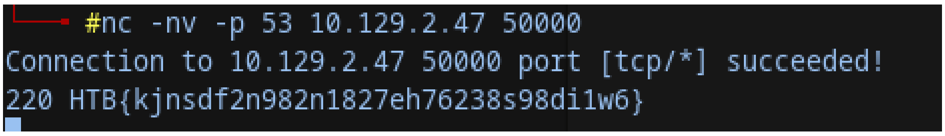


```
└──╼ #nmap 10.129.2.47 -p50000 -sS -Pn -n --disable-arp-ping --packet-trace --source-port 53
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-11 07:18 EAT
SENT (0.2757s) TCP 10.10.14.65:53 > 10.129.2.47:50000 S ttl=50 id=49557 iplen=44  seq=2332646242 win=1024 <mss 1460>
RCVD (0.3323s) TCP 10.129.2.47:38902 > 10.10.14.65:44046 RA ttl=63 id=0 iplen=40  seq=0 win=0
RCVD (0.4702s) TCP 10.129.2.47:50000 > 10.10.14.65:53 SA ttl=63 id=0 iplen=44  seq=1172093939 win=64240 <mss 1346>
Nmap scan report for 10.129.2.47
Host is up (0.19s latency).

PORT      STATE SERVICE
50000/tcp open  ibm-db2

Nmap done: 1 IP address (1 host up) scanned in 0.51 seconds
```



`-p 50000`	Scans only the specified ports.
`-sS`	Performs SYN scan on specified ports.
`-Pn`	Disables ICMP Echo requests.
`-n`	Disables DNS resolution.
`--disable-arp-ping`	Disables ARP ping.
`--packet-trace`	Shows all packets sent and received.
`--source-port 53`	Performs the scans from specified source port.

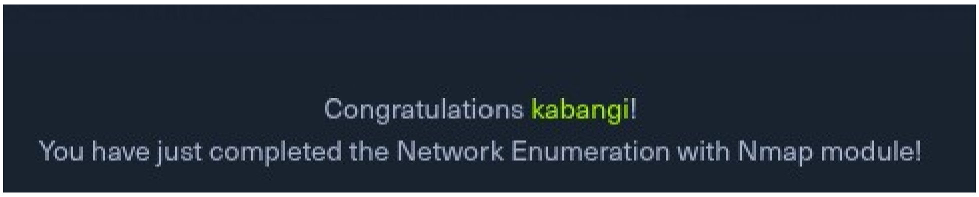
Now that we have found out that the firewall accepts TCP port 53, it is very likely that IDS/IPS filters might also be configured much weaker than others. We can test this by trying to connect to this port by using Netcat.



```
└──╼ #nc -nv -p 53 10.129.2.47 50000
Connection to 10.129.2.47 50000 port [tcp/*] succeeded!
220 HTB{kjnsdf2n982n1827eh76238s98di1w6}
```

Conclusion

That was all that had to be done in this module on all about Nmap. This module highlighted how Nmap can be used to effectively discover hosts, detect open ports, identify running services and their versions, and determine the operating system of a target. It also covered techniques to bypass firewalls using source port manipulation. Overall, it demonstrated the importance of detailed network enumeration in penetration testing.



Congratulations kabangi!
You have just completed the Network Enumeration with Nmap module!