1.1 Nmap

1. Enumeration/1.1 Nmap/1.1 Nmap.md

1. Enumeration

TCP:

```
nmap -iL <IP list> -p- -sT -oN Normal_Scan
```

Aggressive Scan:

```
nmap -iL <IP LIST> -p<Ports found> -sC -sV -sT -oN Aggressive_Scan
```

UDP

```
nmap -sU -Pn --disable-arp-ping -iL <IP>
```

1. Enumeration/1.1 Nmap/1.1.0 Summary.md

[1.1 Nmap]

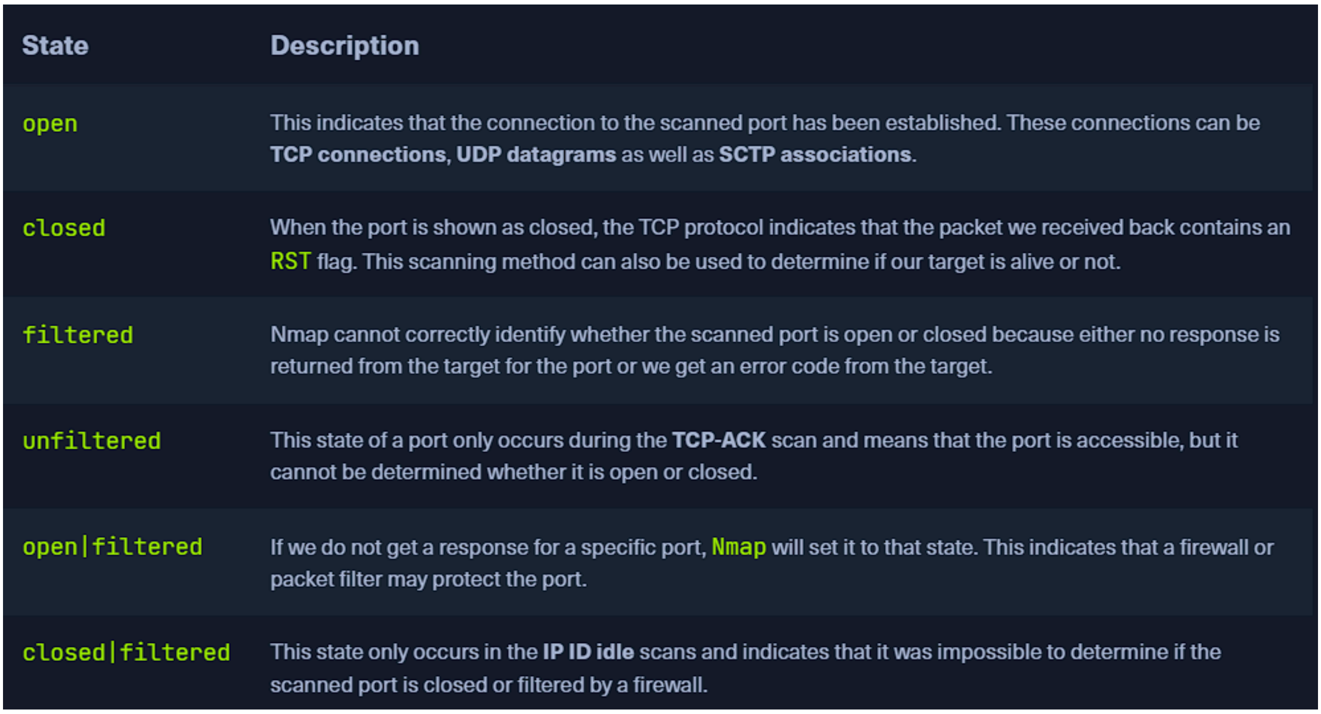
Host Discovery

Use `-sn` for no port scan, if we got host down check it by doing ICMP echo ping by adding `-PE` `--disable-arp-ping`
Linux uses TTL of 64, and windows uses TTL of 128.

Host Enumeration

We can add `--disable-arp-ping -Pn -n` .

6 different states.



State	Description
`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.

Flag	Description
-p	Specific port
--top-ports	Top ports in nmap database
-p-	All 65535 ports
-F	Fast scan (100 top ports)
-Pn	Disable ICMP echo request
-n	Disable DNS resolution
--diable-arp-ping	Disable Arp ping scan
--packet-trace	Show all packet sent and received

Full Scan

```
nmap -sS -p- -T4 --min-rate 300 -Pn -n --disable-arp-ping <IP> -oA nmap/full_scan
```

UDP Ports

```
sudo nmap -sU -Pn -n --disable-arp-ping -T4 --min-rate 80 --max-retries 3 <IP> -oA nmap/udp_full
```

Saving the Result

Flags	Description	File extension
-oN	Normal Output	.nmap
-oG	Greppable Output	.gnmap
-oX	XML output	.xml
-oA	In all format	N/A

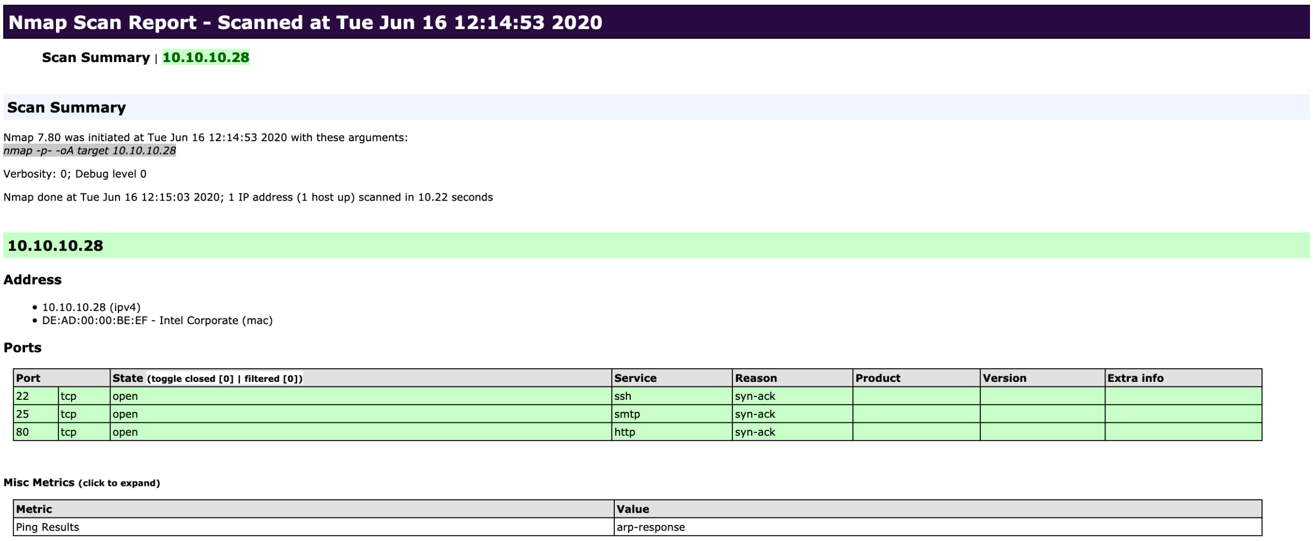
Stylesheets

With xml output we can easily crate HTML report that are easy to read.

Once you get the `.xml` file use:

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

Now we can open this .xml file in our browser.



Nmap Scan Report - Scanned at Tue Jun 16 12:14:53 2020

Scan Summary | **10.10.10.28**

Scan Summary

Nmap 7.80 was initiated at Tue Jun 16 12:14:53 2020 with these arguments:
nmap -p- -oA target 10.10.10.28

Verbosity: 0; Debug level 0

Nmap done at Tue Jun 16 12:15:03 2020; 1 IP address (1 host up) scanned in 10.22 seconds

10.10.10.28

Address

- 10.10.10.28 (ipv4)
- DE:AD:00:00:BE:EF - Intel Corporate (mac)

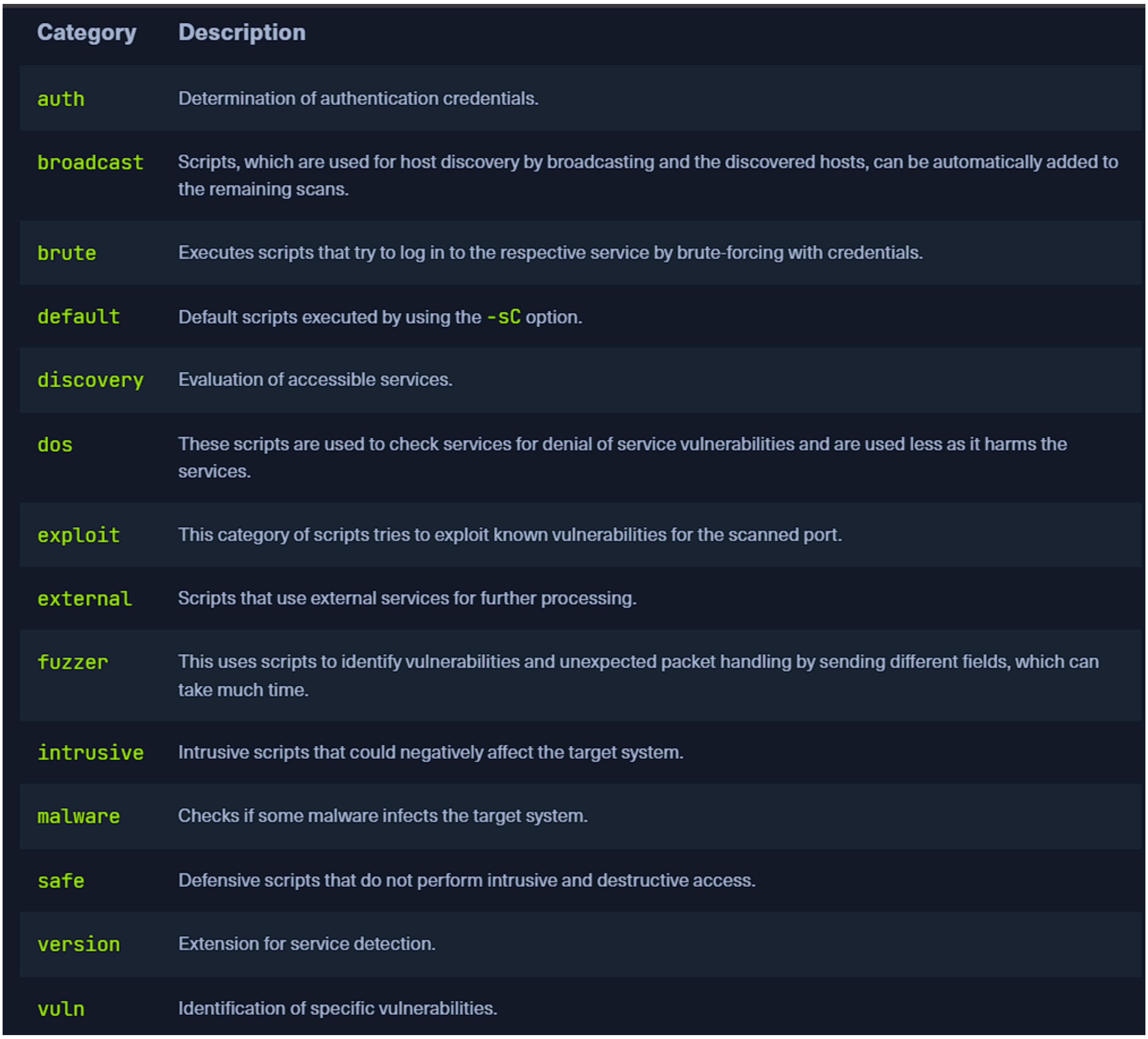
Ports

Port		State (toggle closed [0] | filtered [0])	Service	Reason	Product	Version	Extra info
22	tcp	open	ssh	syn-ack			
25	tcp	open	smtp	syn-ack			
80	tcp	open	http	syn-ack			

Misc Metrics (click to expand)

Metric	Value
Ping Results	arp-response

Scripting Engine



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.

use `--script <category>` or use `-sC` for default scan.

`-A` will be used instead of service scan , script scan and OS scan.

1. Enumeration/1.1 Nmap/1.1.1 Host Discovery.md

1.1 Nmap

Summary

Use `-sn` for no port scan. If we get host down for some reason, we can use `-PE --disable-arp-ping` scan to use ICMP ping instead of arp ping which is default.

Port ranges from 1 to 65535.
Well known ports are 1 to 1023.
Port 0 is considered as wild card port.

Syntax

```
nmap <scan types> <options> <target>
```

Host Discovery

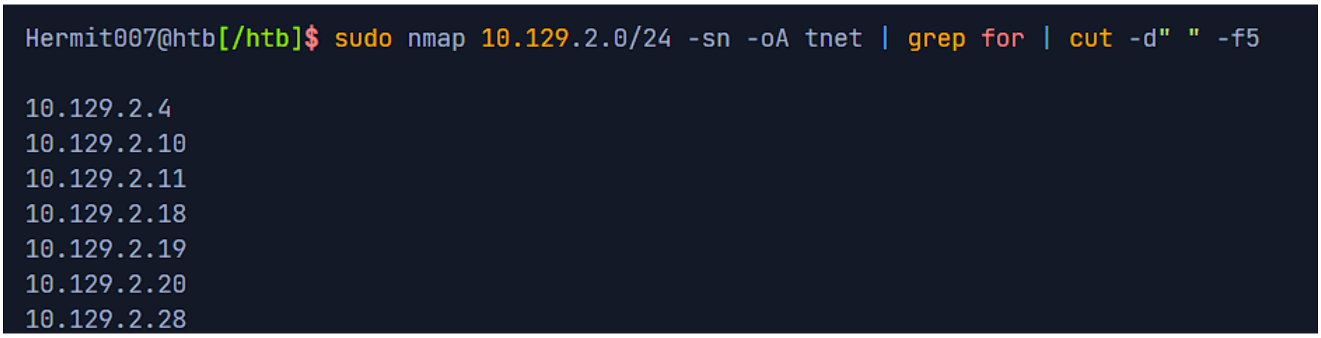
Flags used:

Flags	Description
-sn	No Port Scan
-iL	List of ips
--packet-trace	To trace the packet sent

Flags	Description
-PE --disable-arp-ping	Use ICMP echo rather than Arp ping (default)
--reason	Displays the reason for specific result
-oA	Save result in all format
-oN	Save result in human readable format

Subnet

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

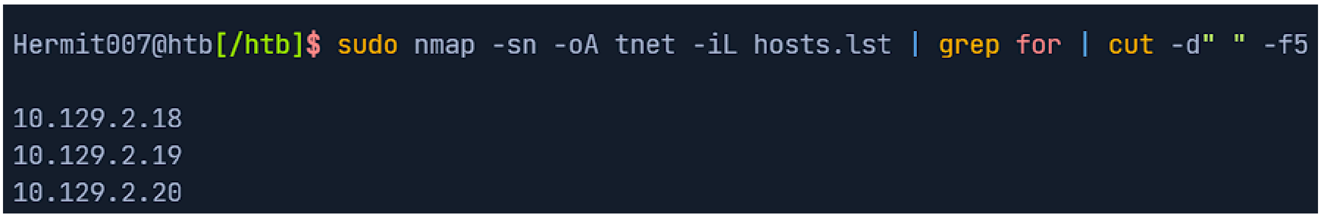


```
Hermit007@htb[/htb]$ sudo nmap 10.129.2.0/24 -sn -oA tnet | grep for | cut -d" " -f5

10.129.2.4
10.129.2.10
10.129.2.11
10.129.2.18
10.129.2.19
10.129.2.20
10.129.2.28
```

`-sn` : Disables port scan

List of IP

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

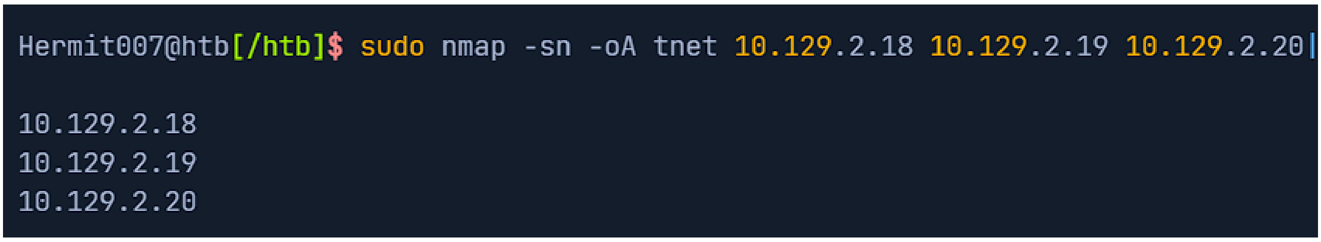


```
Hermit007@htb[/htb]$ sudo nmap -sn -oA tnet -iL hosts.lst | grep for | cut -d" " -f5

10.129.2.18
10.129.2.19
10.129.2.20
```

`-iL` Perform scan against targets provided in hosts.txt

Multiple IP

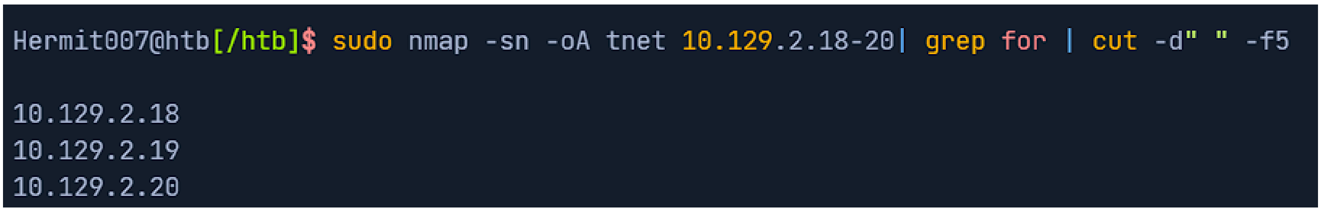
```
sudo nmap -sn -oA tnet 10.129.2.18 10.129.2.19 10.129.2.20| grep for | cut -d" " -f5
```



```
Hermit007@htb[/htb]$ sudo nmap -sn -oA tnet 10.129.2.18 10.129.2.19 10.129.2.20|

10.129.2.18
10.129.2.19
10.129.2.20
```

Adjacent IPs

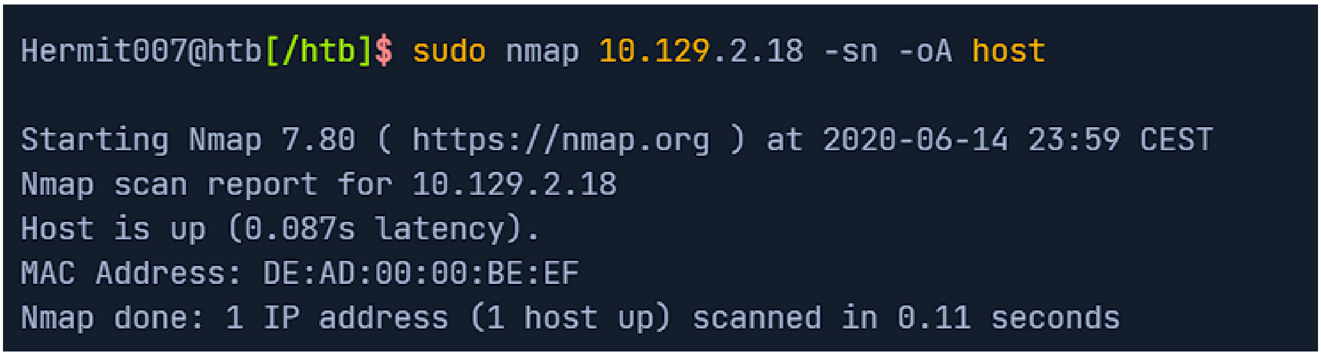
```
sudo nmap -sn -oA tnet 10.129.2.18-20| grep for | cut -d" " -f5
```



```
Hermit007@htb[/htb]$ sudo nmap -sn -oA tnet 10.129.2.18-20| grep for | cut -d" " -f5

10.129.2.18
10.129.2.19
10.129.2.20
```

Single IP

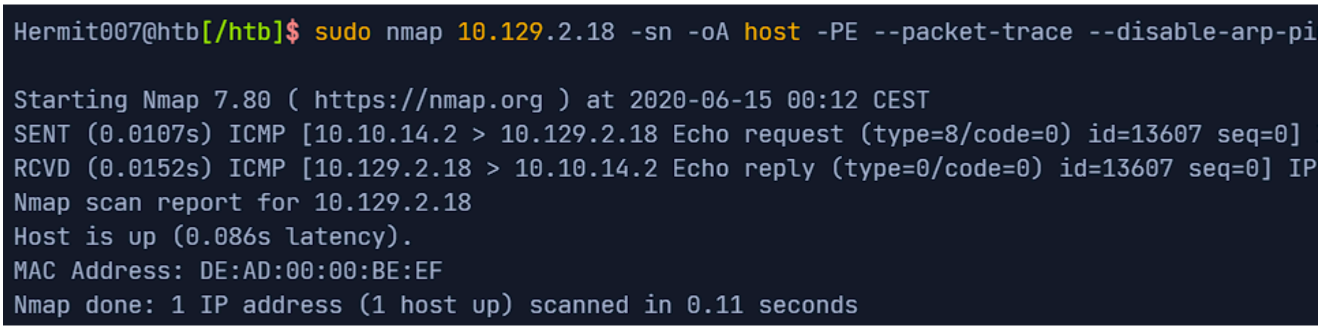
```
sudo nmap 10.129.2.18 -sn -oA host
```



```
Hermit007@htb[/htb]$ sudo nmap 10.129.2.18 -sn -oA host

Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-14 23:59 CEST
Nmap scan report for 10.129.2.18
Host is up (0.087s latency).
MAC Address: DE:AD:00:00:BE:EF
Nmap done: 1 IP address (1 host up) scanned in 0.11 seconds
```

To get info about why nmap reach to a specific outcome we can use `--reason`
To use ICMP echo ping instead of Arp ping (default), we can use `-PE --disable-arp-ping`

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



```
Hermit007@htb[/htb]$ sudo nmap 10.129.2.18 -sn -oA host -PE --packet-trace --disable-arp-pi

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]
RCVD (0.0152s) ICMP [10.129.2.18 > 10.10.14.2 Echo reply (type=0/code=0) id=13607 seq=0] IP
Nmap scan report for 10.129.2.18
Host is up (0.086s latency).
MAC Address: DE:AD:00:00:BE:EF
Nmap done: 1 IP address (1 host up) scanned in 0.11 seconds
```

TTL for OS discovery

Linux uses TTL of 64 , vs windows use TTL of 128.

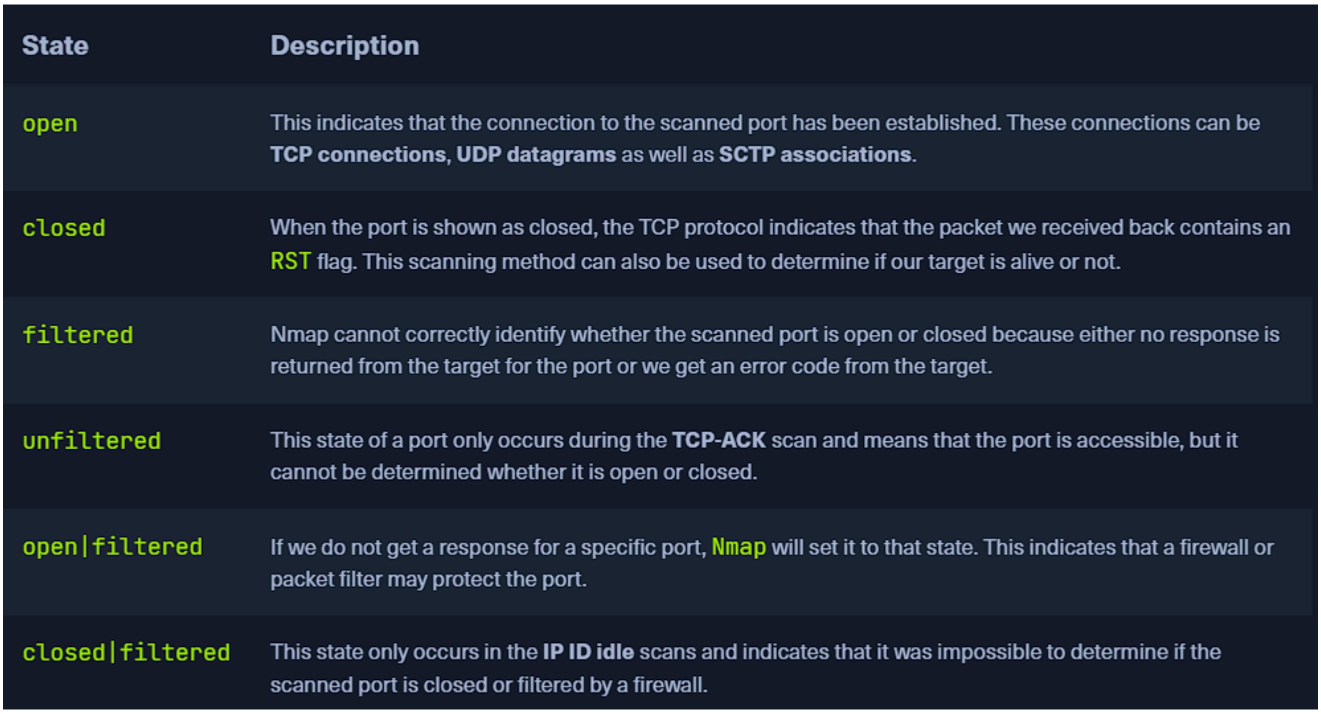
1. Enumeration/1.1 Nmap/1.1.2 Host and Port Scanning.md

1.1 Nmap

Summary

After we discover that the host is alive then we should use flags `--disable-arp-ping -Pn -n --reason` to make it more reliable and less congesting.

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



State	Description
`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.

TCP Ports

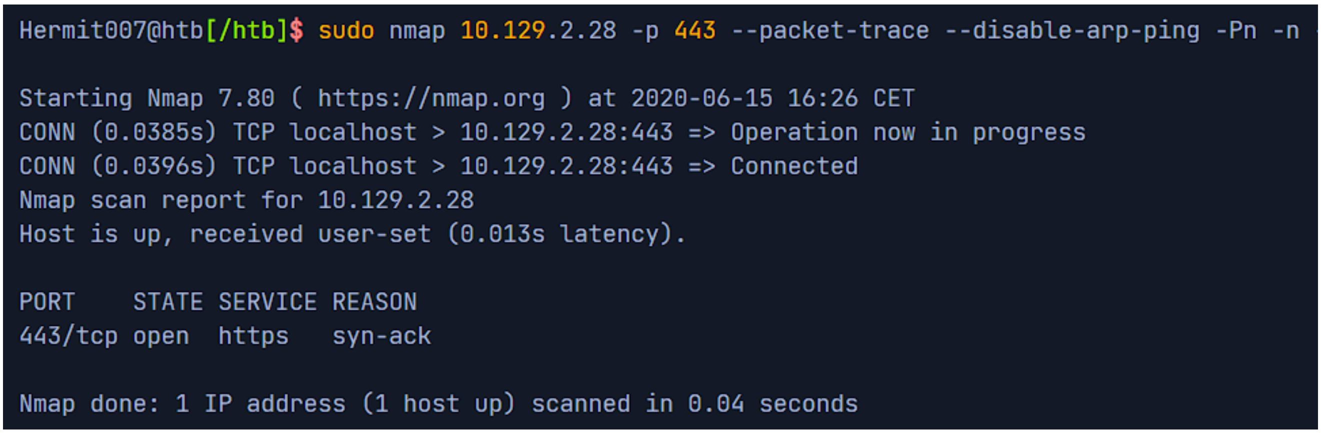
By default nmap scan for 1000 TCP port with SYN (-sS) scan if run as root and -sT if run as non root.

Flag	Description
-p	Specific port
--top-ports	Top ports in nmap database
-p-	All 65535 ports
-F	Fast scan (100 top ports)
-Pn	Disable ICMP echo request
-n	Disable DNS resolution
--diable-arp-ping	Disable Arp ping scan
--packet-trace	Show all packet sent and received

Note: To have clear view of SYN scan we disable arp ping, icmp echo and DNS resolution

Example of TCP Connect scan:

```
sudo nmap 10.129.2.28 -p 443 --packet-trace --disable-arp-ping -Pn -n --reason -sT
```



```
Hermit007@htb[/htb]$ sudo nmap 10.129.2.28 -p 443 --packet-trace --disable-arp-ping -Pn -n

Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-15 16:26 CET
CONN (0.0385s) TCP localhost > 10.129.2.28:443 => Operation now in progress
CONN (0.0396s) TCP localhost > 10.129.2.28:443 => Connected
Nmap scan report for 10.129.2.28
Host is up, received user-set (0.013s latency).

PORT    STATE SERVICE REASON
443/tcp open  https   syn-ack

Nmap done: 1 IP address (1 host up) scanned in 0.04 seconds
```

Full Scan

```
sudo nmap -sS -p- 192.168.109.98 -max-retries 3 -Pn --disable-arp-ping -n --max-parallelism 60 --min-rate 300 --initial-rtt-timeout 200ms --max-rtt-timeout 1000ms --stats-every 10s --host-timeout 15m -oA nmap/full_scan
```

UDP Ports

We will not receive acknowledgement in this port.

```
sudo nmap -sU 192.168.109.98 -Pn -n --max-retries 5 --initial-rtt-timeout 300ms --max-rtt-timeout 2000ms --min-rate 80 --max-parallelism 30 --host-timeout 60m --stats-every 15s -oA nmap/udp_full_reliable
```

Flag	Description
-sU (Mandatory)	UDP Scan
-F	Fast scab (top 100 ports)
-sV	Identify version and service name

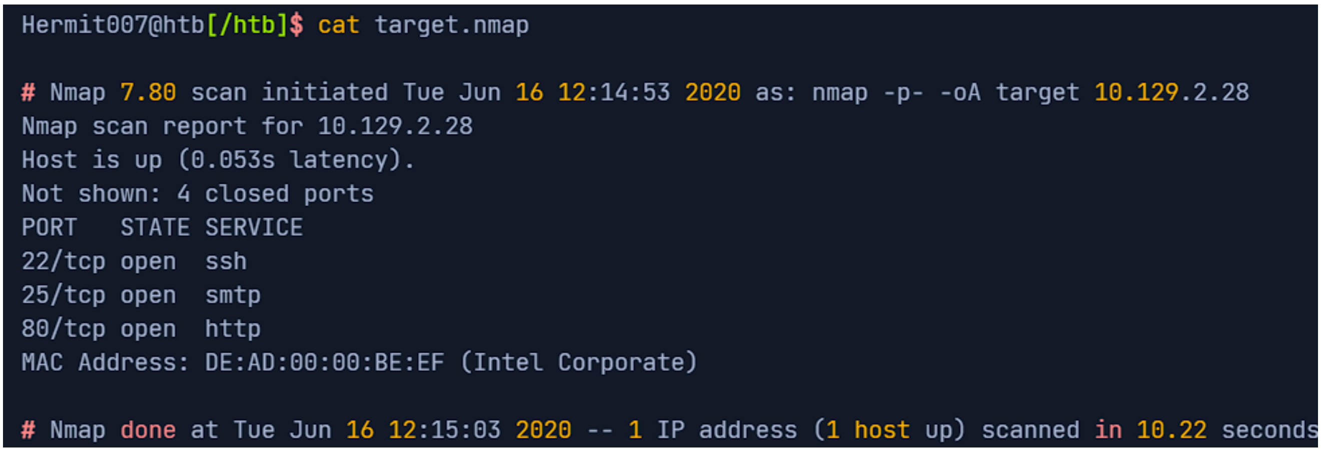
1. Enumeration/1.1 Nmap/1.1.3 Saving the Results.md

1.1 Nmap

Flags	Description	File extension
-oN	Normal Output	.nmap

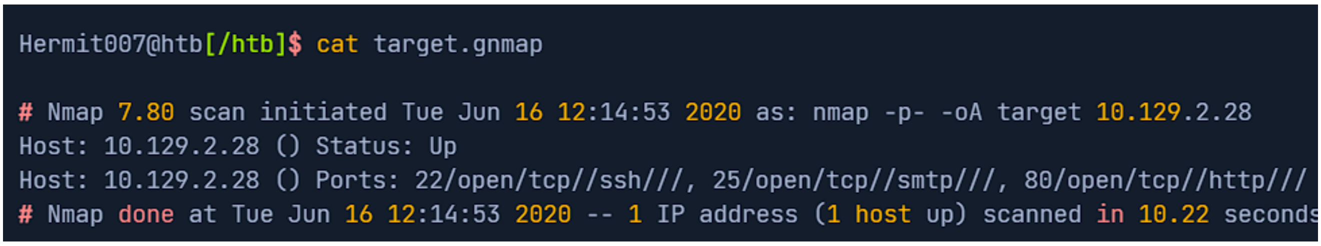
Flags	Description	File extension
-oG	Greppable Output	.gnmap
-oX	XML output	.xml
-oA	In all format	N/A

Normal



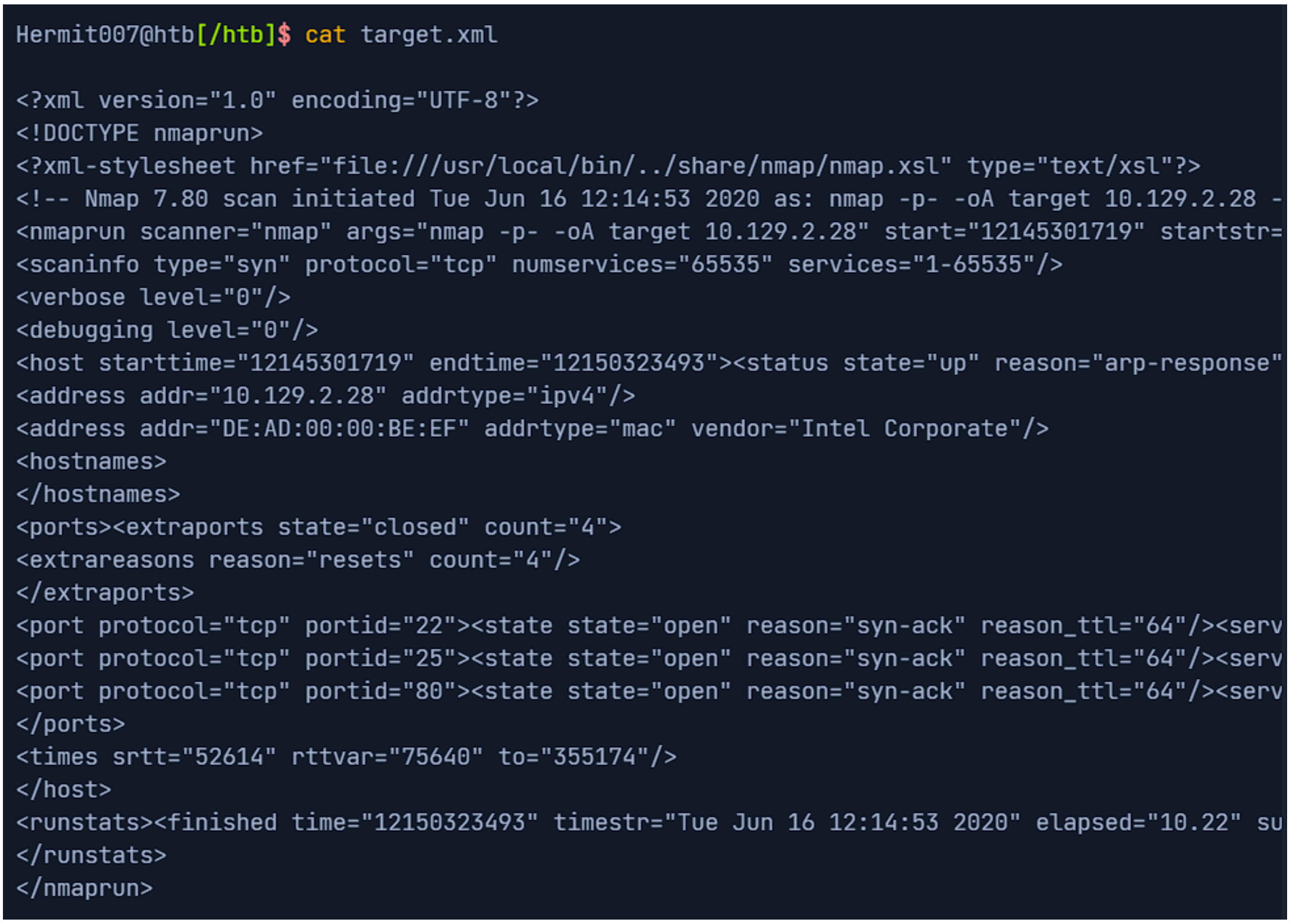
```
Hermit007@htb[/htb]$ cat target.nmap

# Nmap 7.80 scan initiated Tue Jun 16 12:14:53 2020 as: nmap -p- -oA target 10.129.2.28
Nmap scan report for 10.129.2.28
Host is up (0.053s latency).
Not shown: 4 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
25/tcp open  smtp
80/tcp open  http
MAC Address: DE:AD:00:00:BE:EF (Intel Corporate)

# Nmap done at Tue Jun 16 12:15:03 2020 -- 1 IP address (1 host up) scanned in 10.22 seconds
```

Greppable Output



```
Hermit007@htb[/htb]$ cat target.gnmap

# Nmap 7.80 scan initiated Tue Jun 16 12:14:53 2020 as: nmap -p- -oA target 10.129.2.28
Host: 10.129.2.28 ()	Status: Up
Host: 10.129.2.28 ()	Ports: 22/open/tcp//ssh///, 25/open/tcp//smtp///, 80/open/tcp//http///
# Nmap done at Tue Jun 16 12:14:53 2020 -- 1 IP address (1 host up) scanned in 10.22 seconds
```

XML output



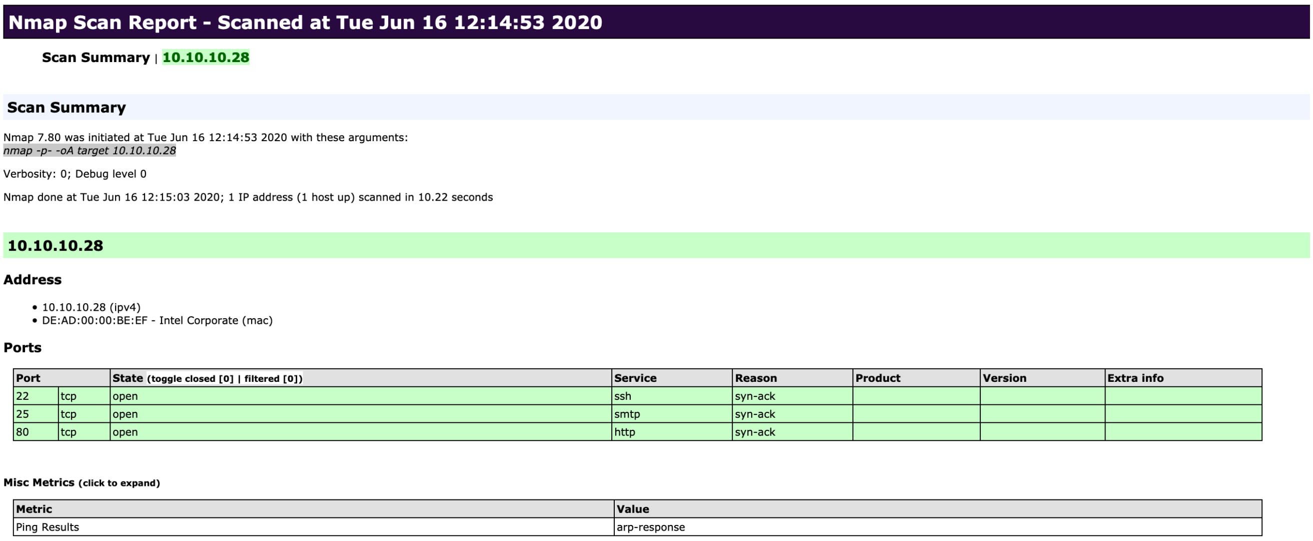
```
Hermit007@htb[/htb]$ cat target.xml

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE nmaprun>
<?xml-stylesheet href="file:///usr/local/bin/../share/nmap/nmap.xsl" type="text/xsl"?>
<!-- Nmap 7.80 scan initiated Tue Jun 16 12:14:53 2020 as: nmap -p- -oA target 10.129.2.28 -
<nmaprun scanner="nmap" args="nmap -p- -oA target 10.129.2.28" start="12145301719" startstr=
<scaninfo type="syn" protocol="tcp" numservices="65535" services="1-65535"/>
<verbose level="0"/>
<debugging level="0"/>
<host starttime="12145301719" endtime="12150323493"><status state="up" reason="arp-response"
<address addr="10.129.2.28" addrtype="ipv4"/>
<address addr="DE:AD:00:00:BE:EF" addrtype="mac" vendor="Intel Corporate"/>
<hostnames>
</hostnames>
<ports><extraports state="closed" count="4">
<extrareasons reason="resets" count="4"/>
</extraports>
<port protocol="tcp" portid="22"><state state="open" reason="syn-ack" reason_ttl="64"/><serv
<port protocol="tcp" portid="25"><state state="open" reason="syn-ack" reason_ttl="64"/><serv
<port protocol="tcp" portid="80"><state state="open" reason="syn-ack" reason_ttl="64"/><serv
</ports>
<times srtt="52614" rttvar="75640" to="355174"/>
</host>
<runstats><finished time="12150323493" timestr="Tue Jun 16 12:14:53 2020" elapsed="10.22" su
</runstats>
</nmaprun>
```

Stylesheets

With xml output we can easily crate HTML report that are easy to read.
Once you get the `.xml` file use:

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

Now we can open this .xml file in our browser.



Nmap Scan Report - Scanned at Tue Jun 16 12:14:53 2020

Scan Summary | **10.10.10.28**

Scan Summary

Nmap 7.80 was initiated at Tue Jun 16 12:14:53 2020 with these arguments:
nmap -p- -oA target 10.10.10.28

Verbosity: 0; Debug level 0

Nmap done at Tue Jun 16 12:15:03 2020; 1 IP address (1 host up) scanned in 10.22 seconds

10.10.10.28

Address

- 10.10.10.28 (ipv4)
- DE:AD:00:00:BE:EF - Intel Corporate (mac)

Ports

Port		State (toggle closed [0] | filtered [0])	Service	Reason	Product	Version	Extra info
22	tcp	open	ssh	syn-ack			
25	tcp	open	smtp	syn-ack			
80	tcp	open	http	syn-ack			

Misc Metrics (click to expand)

Metric	Value
Ping Results	arp-response

To combined multiple scan

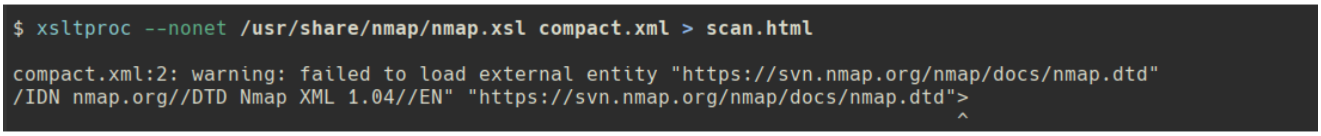
First combined multiple .xml file in one using:

```
python2 gnxmerge.py -sources=default_service_scan,full_scan,full_aggressive_scan > compact.xml
```

Note: gnxmerge is located in ~/Desktop/Tools_Scripts/utility in kali linux

Then convert this in html file:

```
xsltproc --nonet /usr/share/nmap/nmap.xsl compact.xml > scan.html
```



```
$ xsltproc --nonet /usr/share/nmap/nmap.xsl compact.xml > scan.html

compact.xml:2: warning: failed to load external entity "https://svn.nmap.org/nmap/docs/nmap.dtd"
/IDN nmap.org//DTD Nmap XML 1.04//EN" "https://svn.nmap.org/nmap/docs/nmap.dtd">
                                                                                 ^
```

It gives a warning

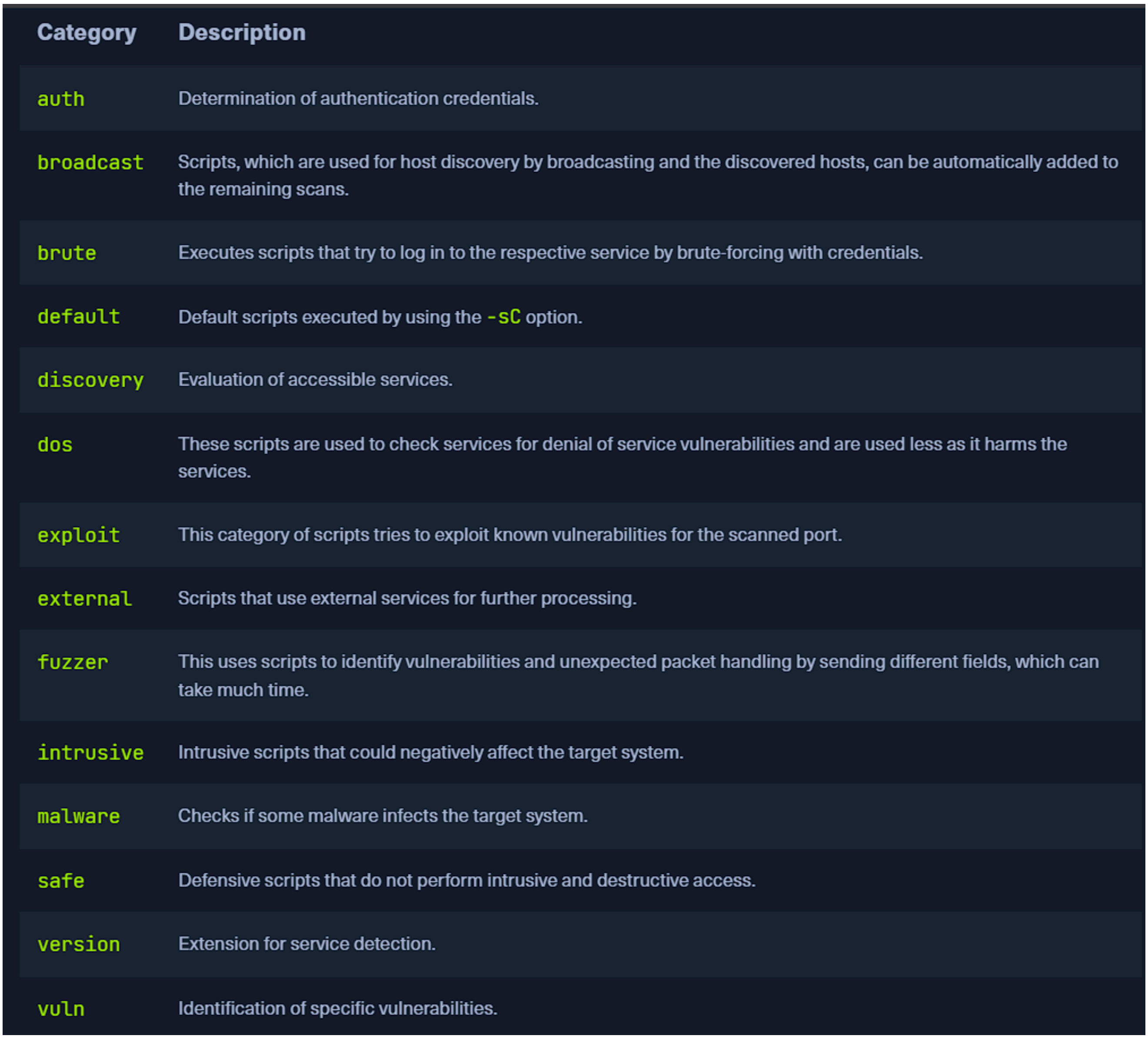
Now to open it in browser:

```
xdg-open scan.html
```

1.1.4 Scripting Engine

1.1 Nmap

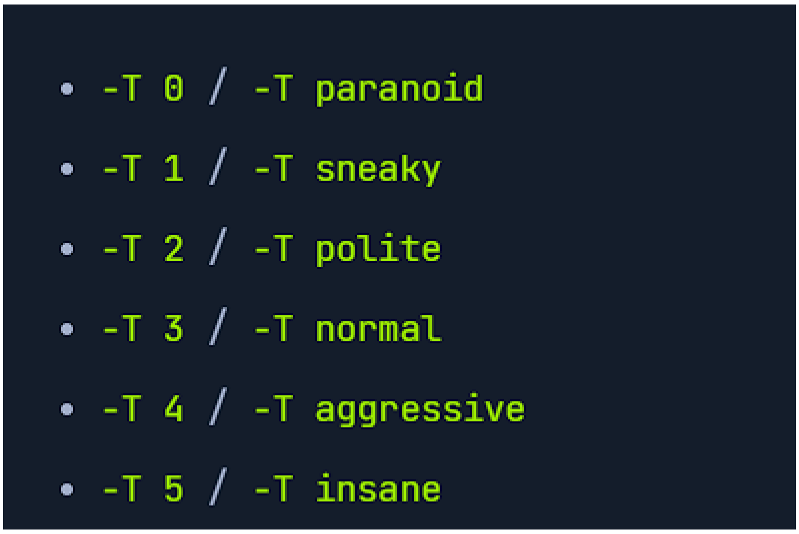
Categories:



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.

1.1.5 Performance

1.1 Nmap



- `-T 0` / `-T paranoid`
- `-T 1` / `-T sneaky`
- `-T 2` / `-T polite`
- `-T 3` / `-T normal`
- `-T 4` / `-T aggressive`
- `-T 5` / `-T insane`

Uses -T4 with --min-rate 300 for host port scan.
Note: -T4 only handles:

```
--initial-rtt-timeout
--max-rtt-timeout
--max-parallelism
--max-retries
```

We need to manually set --min-rate , ideal is 300.
Recomended:

```
nmap -p- -T4 --min-rate 300 -Pn -n <IP>
-oA full
```

1.1.6 Firewall and IDS IPS Evasion

1.1 Nmap

When firewall is set against a specific port it can either be dropped or rejected.
It will show filtered when it is dropped, but when it rejected TCP packets are returned with an RST flag, while ICMP can contain different types of

error codes:
Such errors include:

- Net Unreachable
- Net Prohibited
- Host Unreachable
- Host Prohibited
- Port Unreachable
- Proto Unreachable

ICMP for scanning

In the above paragraph we said if the packet is rejected then ICMP can contain errors, it is to be noted that ICMP is not used here for port scanning, but when the Target OS received a packet on FIREWALL guarded port it send a ICMP Type 3 Code 3 packet which contains error.

If we received filtered or gets a ICMP error we can use `-sA` method for scanning which contains a `ACK` flag. It confuses the firewall as it cannot understand from the connection is initiated.

Firewall vs IDS vs IPS

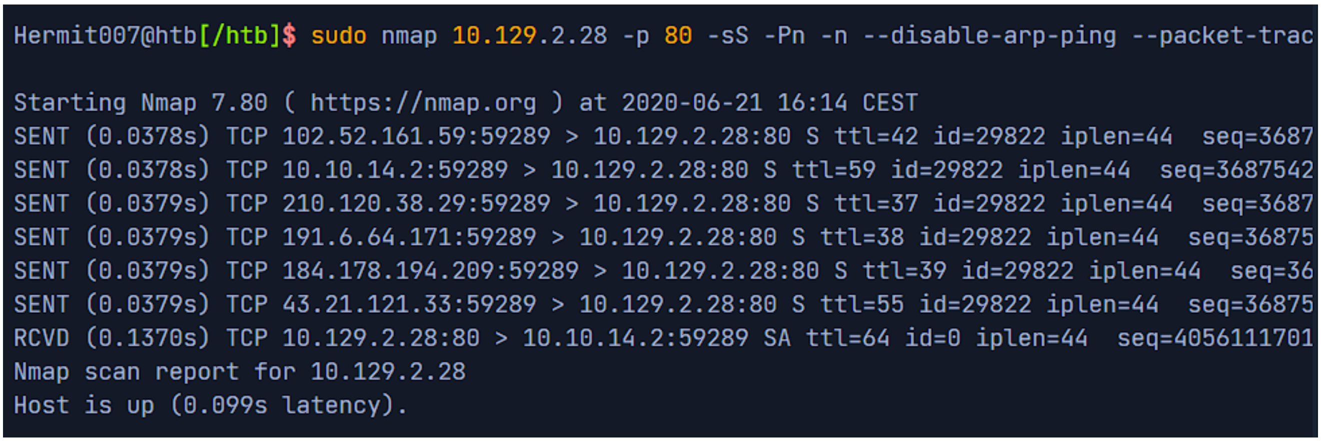
Firewall blocks the packet where as IDS is a passive system it blocks the malicious packet and inform the relevant authorities but it does not block. IPS detects malicious pattern and blocks it. The key difference between firewall and IPS is firewall blocks packet based on rules vs IPS blocks based on malicious pattern. IPS servers as supplement to IDS.

To detect an IPS, the author recommends using a VPS (Virtual Private Server). A VPS is a virtual machine created on a physical server, and many such virtual servers can be hosted on the same machine. When performing aggressive scanning, we use the public IP address of our VPS. If the target network suddenly blocks all traffic from that VPS after some time, we can assume that an IPS is active. At that point, we can switch to another VPS and continue the penetration test.

Decoys

Using decoy nmap generates many ip address and insert them into packets to disguise the original ip address.
We can either generate random decoy ip address or we can give specific ip addresses distinguish them using `:`

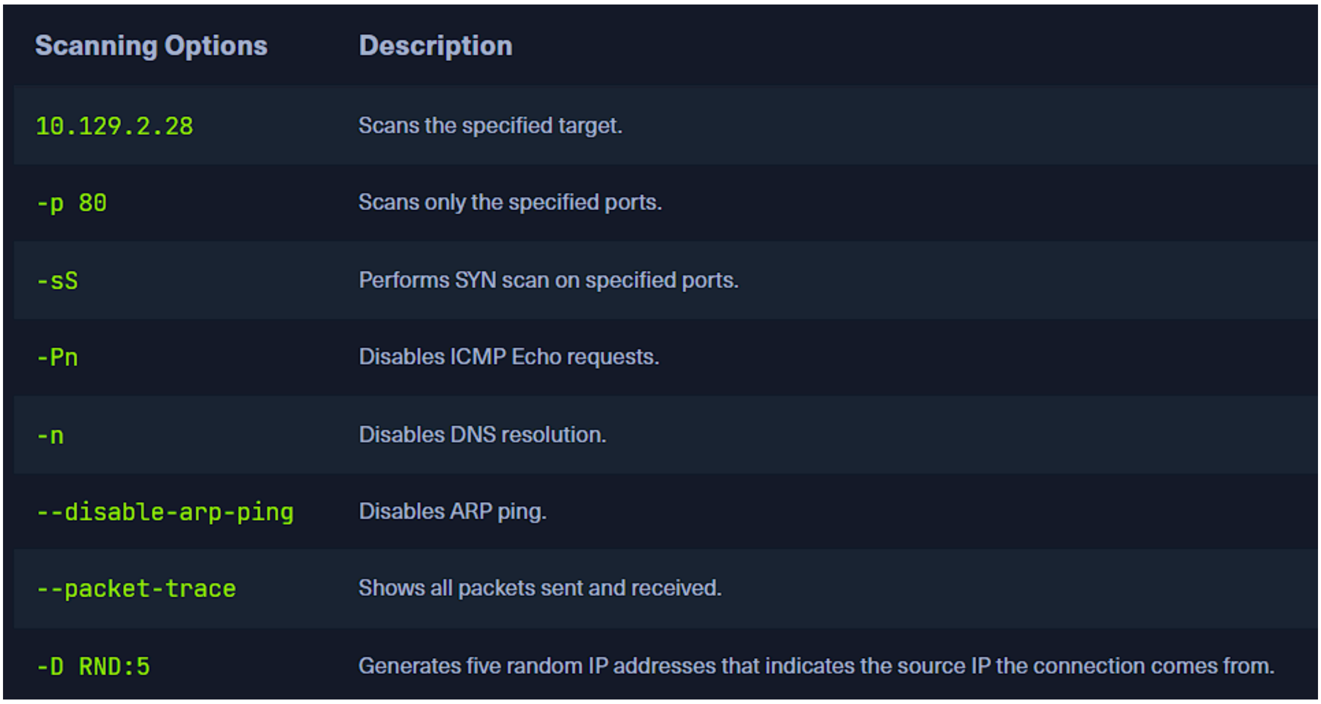
```
sudo nmap 10.129.2.28 -p 80 -sS -Pn -n --disable-arp-ping --packet-trace -D RND:5
```



```
Hermit007@htb[/htb]$ sudo nmap 10.129.2.28 -p 80 -sS -Pn -n --disable-arp-ping --packet-trac

Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-21 16:14 CEST
SENT (0.0378s) TCP 102.52.161.59:59289 > 10.129.2.28:80 S ttl=42 id=29822 iplen=44  seq=3687
SENT (0.0378s) TCP 10.10.14.2:59289 > 10.129.2.28:80 S ttl=59 id=29822 iplen=44  seq=3687542
SENT (0.0379s) TCP 210.120.38.29:59289 > 10.129.2.28:80 S ttl=37 id=29822 iplen=44  seq=3687
SENT (0.0379s) TCP 191.6.64.171:59289 > 10.129.2.28:80 S ttl=38 id=29822 iplen=44  seq=36875
SENT (0.0379s) TCP 184.178.194.209:59289 > 10.129.2.28:80 S ttl=39 id=29822 iplen=44  seq=36
SENT (0.0379s) TCP 43.21.121.33:59289 > 10.129.2.28:80 S ttl=55 id=29822 iplen=44  seq=36875
RCVD (0.1370s) TCP 10.129.2.28:80 > 10.10.14.2:59289 SA ttl=64 id=0 iplen=44  seq=4056111701
Nmap scan report for 10.129.2.28
Host is up (0.099s latency).
```

We can see on the left hand side many random

IP are generated.



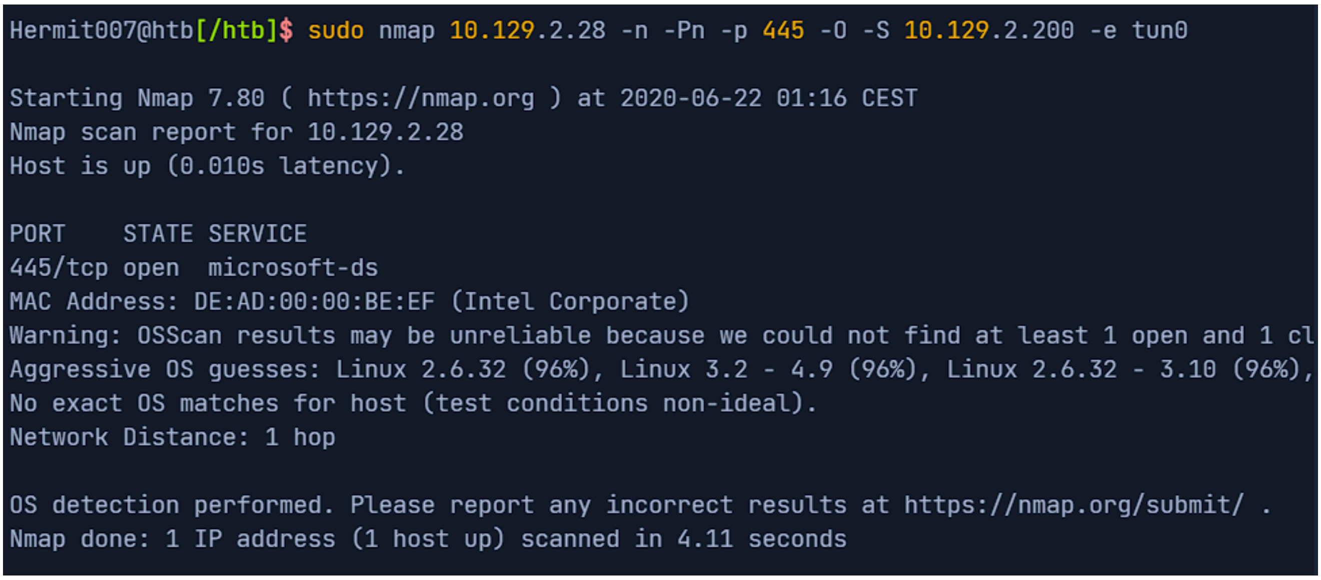
Scanning Options	Description
`10.129.2.28`	Scans the specified target.
`-p 80`	Scans only the specified ports.
`-sS`	Performs SYN scan on specified ports.
`-Pn`	Disables ICMP Echo requests.
`-n`	Disables DNS resolution.
`--disable-arp-ping`	Disables ARP ping.
`--packet-trace`	Shows all packets sent and received.
`-D RND:5`	Generates five random IP addresses that indicates the source IP the connection comes from.

Remember only 1 IP that is our original ip address is real ip, so it will only receive result and other will be revolve in the infinite internet

We can change our source ip without using decoys using `-S` flag, It is used to change the source ip address with the given ip address.

```
sudo nmap 10.129.2.28 -n -Pn -p 445 -O -S 10.129.2.200 -e tun0
```

Here we have given the interface tun0, so mostly the source ip is overwritten by the VPN ip *address, making this attempt useless*



```
Hermit007@htb[/htb]$ sudo nmap 10.129.2.28 -n -Pn -p 445 -O -S 10.129.2.200 -e tun0

Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-22 01:16 CEST
Nmap scan report for 10.129.2.28
Host is up (0.010s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds
MAC Address: DE:AD:00:00:BE:EF (Intel Corporate)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 cl
Aggressive OS guesses: Linux 2.6.32 (96%), Linux 3.2 - 4.9 (96%), Linux 2.6.32 - 3.10 (96%),
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 4.11 seconds
```

DNS Proxying

By default nmap perform reverse DNS resolution for finding more information. We can specify the DNS server to use by flag `--dns-server <ns>,<ns>` . It is usually used in DMZ where we want to use company dns server. We can also specify port number by which we want our nmap to send request from using `--source-port` .

```
sudo nmap 10.129.2.28 -p50000 -sS -Pn -n --disable-arp-ping --packet-trace --source-port 53
```