**Target Specification**

| **Switch** | **Example** | **Description** |
| --- | --- | --- |
|  | nmap 192.168.1.1 | Scan a single IP |
|  | nmap 192.168.1.1 192.168.2.1 | Scan specific IPs |
|  | nmap 192.168.1.1-254 | Scan a range |
|  | nmap scanme.nmap.org | Scan a domain |
|  | nmap 192.168.1.0/24 | Scan using CIDR notation |
| -iL | nmap -iL targets.txt | Scan targets from a file |
| -iR | nmap -iR 100 | Scan 100 random hosts |
| --exclude | nmap --exclude 192.168.1.1 | Exclude listed hosts |

**Scan Techniques**

| **Switch** | **Example** | **Description** |
| --- | --- | --- |
| -sS | nmap 192.168.1.1 -sS | TCP SYN port scan (Default) |
| -sT | nmap 192.168.1.1 -sT | TCP connect port scan (Default without root privilege) |
| -sU | nmap 192.168.1.1 -sU | UDP port scan |
| -sA | nmap 192.168.1.1 -sA | TCP ACK port scan |
| -sW | nmap 192.168.1.1 -sW | TCP Window port scan |
| -sM | nmap 192.168.1.1 -sM | TCP Maimon port scan |

**Host Discovery**

| **Switch** | **Example** | **Description** |
| --- | --- | --- |
| -sL | nmap 192.168.1.1-3 -sL | No Scan. List targets only |
| -sn | nmap 192.168.1.1/24 -sn | Disable port scanning. Host discovery only. |
| -Pn | nmap 192.168.1.1-5 -Pn | Disable host discovery. Port scan only. |
| -PS | nmap 192.168.1.1-5 -PS22-25,80 | TCP SYN discovery on port x.  Port 80 by default |
| -PA | nmap 192.168.1.1-5 -PA22-25,80 | TCP ACK discovery on port x.  Port 80 by default |
| -PU | nmap 192.168.1.1-5 -PU53 | UDP discovery on port x.  Port 40125 by default |
| -PR | nmap 192.168.1.1-1/24 -PR | ARP discovery on local network |
| -n | nmap 192.168.1.1 -n | Never do DNS resolution |

**Port Specification**

| **Switch** | **Example** | **Description** |
| --- | --- | --- |
| -p | nmap 192.168.1.1 -p 21 | Port scan for port x |
| -p | nmap 192.168.1.1 -p 21-100 | Port range |
| -p | nmap 192.168.1.1 -p U:53,T:21-25,80 | Port scan multiple TCP and UDP ports |
| -p- | nmap 192.168.1.1 -p- | Port scan all ports |
| -p | nmap 192.168.1.1 -p http,https | Port scan from service name |
| -F | nmap 192.168.1.1 -F | Fast port scan (100 ports) |
| --top-ports | nmap 192.168.1.1 --top-ports 2000 | Port scan the top x ports |
| -p-65535 | nmap 192.168.1.1 -p-65535 | Leaving off initial port in range makes the scan start at port 1 |
| -p0- | nmap 192.168.1.1 -p0- | Leaving off end port in range  makes the scan go through to port 65535 |

**Service and Version Detection**

| **Switch** | **Example** | **Description** |
| --- | --- | --- |
| -sV | nmap 192.168.1.1 -sV | Attempts to determine the version of the service running on port |
| -sV --version-intensity | nmap 192.168.1.1 -sV --version-intensity 8 | Intensity level 0 to 9. Higher number increases possibility of correctness |
| -sV --version-light | nmap 192.168.1.1 -sV --version-light | Enable light mode. Lower possibility of correctness. Faster |
| -sV --version-all | nmap 192.168.1.1 -sV --version-all | Enable intensity level 9. Higher possibility of correctness. Slower |
| -A | nmap 192.168.1.1 -A | Enables OS detection, version detection, script scanning, and traceroute |

**OS Detection**

| **Switch** | **Example** | **Description** |
| --- | --- | --- |
| -O | nmap 192.168.1.1 -O | Remote OS detection using TCP/IP stack fingerprinting |
| -O --osscan-limit | nmap 192.168.1.1 -O --osscan-limit | If at least one open and one closed TCP port are not found it will not try OS detection against host |
| -O --osscan-guess | nmap 192.168.1.1 -O --osscan-guess | Makes Nmap guess more aggressively |
| -O --max-os-tries | nmap 192.168.1.1 -O --max-os-tries 1 | Set the maximum number x of OS detection tries against a target |
| -A | nmap 192.168.1.1 -A | Enables OS detection, version detection, script scanning, and traceroute |

**Timing and Performance**

| **Switch** | | **Example** | **Description** |
| --- | --- | --- | --- |
| -T0 | | nmap 192.168.1.1 -T0 | Paranoid (0) Intrusion Detection System evasion |
| -T1 | | nmap 192.168.1.1 -T1 | Sneaky (1) Intrusion Detection System evasion |
| -T2 | | nmap 192.168.1.1 -T2 | Polite (2) slows down the scan to use less bandwidth and use less target machine resources |
| -T3 | | nmap 192.168.1.1 -T3 | Normal (3) which is default speed |
| -T4 | | nmap 192.168.1.1 -T4 | Aggressive (4) speeds scans; assumes you are on a reasonably fast and reliable network |
| -T5 | | nmap 192.168.1.1 -T5 | Insane (5) speeds scan; assumes you are on an extraordinarily fast network |
|  | |  |  |
| **Switch** | | **Example input** | **Description** |
| --host-timeout <time> | | 1s; 4m; 2h | Give up on target after this long |
| --min-rtt-timeout/max-rtt-timeout/initial-rtt-timeout <time> | | 1s; 4m; 2h | Specifies probe round trip time |
| --min-hostgroup/max-hostgroup <size<size> | | 50; 1024 | Parallel host scan group sizes |
| --min-parallelism/max-parallelism <numprobes> | | 10; 1 | Probe parallelization |
| --scan-delay/--max-scan-delay <time> | | 20ms; 2s; 4m; 5h | Adjust delay between probes |
| --max-retries <tries> | | 3 | Specify the maximum number of port scan probe retransmissions |
| --min-rate <number> | | 100 | Send packets no slower than <numberr> per second |
| --max-rate <number> | | 100 | Send packets no faster than <number> per second |

**NSE Scripts**

| **Switch** | **Example** | **Description** |
| --- | --- | --- |
| -sC | nmap 192.168.1.1 -sC | Scan with default NSE scripts. Considered useful for discovery and safe |
| --script default | nmap 192.168.1.1 --script default | Scan with default NSE scripts. Considered useful for discovery and safe |
| --script | nmap 192.168.1.1 --script=banner | Scan with a single script. Example banner |
| --script | nmap 192.168.1.1 --script=http\* | Scan with a wildcard. Example http |
| --script | nmap 192.168.1.1 --script=http,banner | Scan with two scripts. Example http and banner |
| --script | nmap 192.168.1.1 --script "not intrusive" | Scan default, but remove intrusive scripts |
| --script-args | nmap --script snmp-sysdescr --script-args snmpcommunity=admin 192.168.1.1 | NSE script with arguments |

**Useful NSE Script Examples**

| **Command** | **Description** |
| --- | --- |
| nmap -Pn --script=http-sitemap-generator scanme.nmap.org | http site map generator |
| nmap -n -Pn -p 80 --open -sV -vvv --script banner,http-title -iR 1000 | Fast search for random web servers |
| nmap -Pn --script=dns-brute domain.com | Brute forces DNS hostnames guessing subdomains |
| nmap -n -Pn -vv -O -sV --script smb-enum\*,smb-ls,smb-mbenum,smb-os-discovery,smb-s\*,smb-vuln\*,smbv2\* -vv 192.168.1.1 | Safe SMB scripts to run |
| nmap --script whois\* domain.com | Whois query |
| nmap -p80 --script http-unsafe-output-escaping scanme.nmap.org | Detect cross site scripting vulnerabilities |
| nmap -p80 --script http-sql-injection scanme.nmap.org | Check for SQL injections |

**Firewall / IDS Evasion and Spoofing**

| **Switch** | **Example** | **Description** |
| --- | --- | --- |
| -f | nmap 192.168.1.1 -f | Requested scan (including ping scans) use tiny fragmented IP packets. Harder for packet filters |
| --mtu | nmap 192.168.1.1 --mtu 32 | Set your own offset size |
| -D | nmap -D 192.168.1.101,192.168.1.102, 192.168.1.103,192.168.1.23 192.168.1.1 | Send scans from spoofed IPs |
| -D | nmap -D decoy-ip1,decoy-ip2,your-own-ip,decoy-ip3,decoy-ip4 remote-host-ip | Above example explained |
| -S | nmap -S www.microsoft.com www.facebook.com | Scan Facebook from Microsoft (-e eth0 -Pn may be required) |
| -g | nmap -g 53 192.168.1.1 | Use given source port number |
| --proxies | nmap --proxies http://192.168.1.1:8080, http://192.168.1.2:8080 192.168.1.1 | Relay connections through HTTP/SOCKS4 proxies |
| --data-length | nmap --data-length 200 192.168.1.1 | Appends random data to sent packets |

**Example IDS Evasion command**

nmap -f -t 0 -n -Pn –data-length 200 -D 192.168.1.101,192.168.1.102,192.168.1.103,192.168.1.23 192.168.1.1

**Output**

| **Switch** | **Example** | **Description** |
| --- | --- | --- |
| -oN | nmap 192.168.1.1 -oN normal.file | Normal output to the file normal.file |
| -oX | nmap 192.168.1.1 -oX xml.file | XML output to the file xml.file |
| -oG | nmap 192.168.1.1 -oG grep.file | Grepable output to the file grep.file |
| -oA | nmap 192.168.1.1 -oA results | Output in the three major formats at once |
| -oG - | nmap 192.168.1.1 -oG - | Grepable output to screen. -oN -, -oX - also usable |
| --append-output | nmap 192.168.1.1 -oN file.file --append-output | Append a scan to a previous scan file |
| -v | nmap 192.168.1.1 -v | Increase the verbosity level (use -vv or more for greater effect) |
| -d | nmap 192.168.1.1 -d | Increase debugging level (use -dd or more for greater effect) |
| --reason | nmap 192.168.1.1 --reason | Display the reason a port is in a particular state, same output as -vv |
| --open | nmap 192.168.1.1 --open | Only show open (or possibly open) ports |
| --packet-trace | nmap 192.168.1.1 -T4 --packet-trace | Show all packets sent and received |
| --iflist | nmap --iflist | Shows the host interfaces and routes |
| --resume | nmap --resume results.file | Resume a scan |

**Helpful Nmap Output examples**

| **Command** | **Description** |
| --- | --- |
| nmap -p80 -sV -oG - --open 192.168.1.1/24 | grep open | Scan for web servers and grep to show which IPs are running web servers |
| nmap -iR 10 -n -oX out.xml | grep "Nmap" | cut -d " " -f5 > live-hosts.txt | Generate a list of the IPs of live hosts |
| nmap -iR 10 -n -oX out2.xml | grep "Nmap" | cut -d " " -f5 >> live-hosts.txt | Append IP to the list of live hosts |
| ndiff scanl.xml scan2.xml | Compare output from nmap using the ndif |
| xsltproc nmap.xml -o nmap.html | Convert nmap xml files to html files |
| grep " open " results.nmap | sed -r 's/ +/ /g' | sort | uniq -c | sort -rn | less | Reverse sorted list of how often ports turn up |

**Miscellaneous Options**

| **Switch** | **Example** | **Description** |
| --- | --- | --- |
| -6 | nmap -6 2607:f0d0:1002:51::4 | Enable IPv6 scanning |
| -h | nmap -h | nmap help screen |

**Other Useful Nmap Commands**

| **Command** | **Description** |
| --- | --- |
| nmap -iR 10 -PS22-25,80,113,1050,35000 -v -sn | Discovery only on ports x, no port scan |
| nmap 192.168.1.1-1/24 -PR -sn -vv | Arp discovery only on local network, no port scan |
| nmap -iR 10 -sn -traceroute | Traceroute to random targets, no port scan |
| nmap 192.168.1.1-50 -sL --dns-server 192.168.1.1 | Query the Internal DNS for hosts, list targets only |

**Nmap Usage**

Nmap needs the following information port number, script name, any script arguments (optional), and the IP of the target.

**nmap -p <port> –script <script-same> –script-args <script arguemens> <target IP>**

**Listing HTTP MEthods**

**nmap -p 8585 -sV –script http-methods,http-trace –script-args http-methods.test-all=true,http-methods.url-path=’/uploads’ 192.168.2.66**

**A screenshot of a cell phone

Description automatically generated**Using the http-methods script we can see all the available methods. The http-methods script will not identify the TRACE method we have to add the http-trace script. These scrips will locate any risky methods that could hint to exploitation.  
GET,HEAD,POST,OPTIONS,DELETE,CONNECT,TRACE

**HTTP methods status codes**

**nmap -p 8585 -sV –script http-methods –script-args http-methods.retest 192.168.2.66**

A screenshot of a cell phone

Description automatically generated  
If we change the script argument to http-methods.retest the response will be the status response for the available http methods.

**HTTP methods url directory**

A screenshot of a cell phone

Description automatically generated  
Sometimes subdirectories on a web server can have different methods that could be vulnerable adding the http-methods.url-path script. In this example, I’ve added the “uploads” directory, and the results were different.

**Notes**  
TRACE method is susceptible to Cross-Site Tracing (XST) attack. CONNECT method might allow the web server to be used as a http proxy. The PUT and DELETE can enable changes to the folder contents.

**Http Open Proxy**

**nmap -p 8080 -sV –script http-open-proxy 192.168.2.66**

A close up of text on a black background

Description automatically generated  
Using the http-open-proxy, we can detect web servers that are configured to act as a proxy. This can allow an attacker to hide their real ip for other attacks.

**Http folder and file discovery**

**nmap –script http-enum -p 8585 192.168.2.66**

A screenshot of a cell phone

Description automatically generated  
Using the http-enum script we can find any possible misconfiguration that could lead to exposing sensitive data or even directory traversal and expose system files.

**Http enumeration with Nikto database**

**nmap -sV –script http-enum –script-args http-enum.nikto-db-path=/usr/share/nikto/db\_dictinary -p 8585 192.168.2.66**

A screenshot of a cell phone

Description automatically generated  
Nikto is a powerful web server enumeration tool. With the http-enum.nikto-db  we can use Nikto databases for identifiying directories on a web server.

FTP-Brute

**nmap -p 21 –script=ftp-brute –script-args userdb=list.txt,passdb=pass.txt,brute.threads=4 192.168.2.62**

**A screenshot of a cell phone

Description automatically generated**As you can see I ran the scan using user list list.txt, password file rockyou.txt, time out of 4 sec. I was able to brute force the username “msfadmin” and password of “msfadmin.”

**SSH-BRUTE**  
With the ssh-brute script, we can control various inputs such as usernames, passwords, timeout, and threads. In this example I will be using lists for both usernames and passwords, as well setting a timeout and number of concurrent threads.

**nmap -p 22 –script ssh-brute –script-args userdb=users.lst,passdb=pass.lst,ssh-brute.timeout=4s,brute.threads=6 192.168.1.1**

A screenshot of a cell phone

Description automatically generated  
As you can see I ran the scan using user list list.txt, password file rockyou.txt, time out of 4 sec, and six threads. I was able to brute force the username “msfadmin” and password of “msfadmin.”

**http-default-accounts**With the http-default-accounts script, we can find any web application using the default credentials.

**nmap -p 8180 –script=http-default-accounts 192.168.2.62**

A screenshot of a cell phone

Description automatically generated  
Using the http-default-accounts script I was able to find that tomcat service on port 8180 is using the default credentials username tomcat and password tomcat.

**Email Scraping**  
Using the http-grep script we can search the http pages for any email address located on the page.

**nmap -p80 –script http-grep –script-args http-grep.builtins=e-mail 192.168.2.62**

A screenshot of a cell phone

Description automatically generated  
Using the http-grep script I was able to find the email address “mutillidae-development @ gmail.com.”

Nmap needs the following information port number, script name, any script arguments (optional), and the IP of the target.

**nmap -p <port> –script <script-same> –script-args <script arguemens> <target IP>**

**SMB OS discovery**

**nmap -p 139,445 –script smb-os-discovery 192.168.2.66**

A picture containing screenshot

Description automatically generated  
Using the smb-os-discovery script we can collect information about the operating system from the SMB service.

**SMB signing check**

**nmap -p137,139,445 –script smb-security-mode 192.168.2.66**

A close up of text on a black background

Description automatically generated  
Using the smb-security-mode script we can see that the message\_signing is disabled.

**IIS web server name disclosure**

**nmap -p 80 –script http-iis-short-name-brute 192.168.2.66**

**MS08-067 (netapi) vulnerability check**

**nmap -p 445 –script smb-vuln-ms08-067 192.168.2.66**

**Checking all smb**vulnerability**scripts**

**nmap -p 445 –script smb-vuln-\* 192.168.2.66**

A screenshot of text

Description automatically generated  
Using a wildcard character we can call all the scripts that start with “smb-vuln-.”

**Netbios and MAC address lookup**

**nmap -sU -p137 –script nbstat 192.168.2.66**

A picture containing device

Description automatically generated  
The nbstatscript reveals the NetBIOS name and the mac address.

**Enumerating user accounts**

**nmap -p 139,445 –script smb-enum-users 192.168.2.66**

**Enumerating window shares**

**nmap -p 139,445 –script smb-enum-shares –script-args smbusername=vagrant,smbpassword=vagrant 192.168.2.66**

A screenshot of a cell phone

Description automatically generated  
Using the smb-enum-shares with valid credentials we can see what smb shares are open, the directory the shares points to and our permissions for the shares.

**SMB**loging**brute force**

**nmap -p 445 –script smb-brute –script-args userdb=unamelist.txt,passdb=testlist.txt 192.168.2.66**

**Finding domain controllers**

**nmap -p 389 -sV <target>**

**Detecting shadow brokers double pulsar smb**

**nmap -p 445 –script smb-double-pulsar-backdoor**

Resources: