Enumerating DNS Records of a domain name

With nslookup

```
nslookup -type=[type-of-dns-record] example.com
nslookup -type=mx google.com
```

With dig, example below querying A records

```
dig example.com A
```

With DnsDumpster
dnsdumpster.com

Enumerating subdomains, email addresses and hosts

```
root@kali: theharvester -d (target-domain) -b all -h
results.html
```

[subdomains]

```
root@kali:dnsrecon -d [domain]
```

[sudomains]

```
sublist3r.py -d [domain]
```

[sudomains]

```
ffuf -w /usr/share/wordlists/SecLists/Discovery/DNS/namelist.txt -H "Host: FUZZ.acmeitsupport.thm" -u http://MACHINE_IP -fs {size}
```

[subdomains using Google Dorks]

```
-site:www.domain.com site:*.domain.com
```

[subdomains using certificate logs]

```
https://crt.sh/
```

Enumerating shares and Netbios [Windows]

```
root@kali:smbclient -I TargetIP -L administrator -N -U ""
root@kali:sudo smbclient -L \\\ip\\\sharename -U admin]
```

Or

```
root@kali:smbclient.py admi4343n@172.31.1.21
root@kali:enum4linux.pl (options) targetip
```

Enumerating shares [Linux]

```
root@kali:Smbclient //(ip) -U (username)
```

Specifying a minimum version for SMBv1

Specifying a minimum version for SMBv1

```
root@kali:Smbclient //(ip) -U (username) -option=’client min protocol=NT1’
```

Logging in without username and specifying it later

```
root@kali:Smbclient //[ip]
smb: \> logon [username]
```

Enumerating NFS shares

Listing the shares

```
root@kali:Showmount -e [target-ip]
root@kali:Mount -t nfs [hostname or ip]:/path-to-share [path to local mount point]
root@kali:Mount -t nfs 192.168.1.1:/var/backups /mnt/backups
root@kali:Mount -t cifs -o username=admin , password=password //192.168.1.1/shares /mnt/shares
```

Nmap

Basic scan to reveal services with their version

```
nmap -sV 10.10.10.3
```

preforming a scan on a list of targets IPs/Domains

```
nmap -iL targets.txt
```

Scan services with OS detection

```
nmap -sV -O 10.10.10.3
```

Enabling fast mode

```
nmap -sV -F -O 10.10.10.3
```

Aggressive scan to reveal all details

```
nmap -A 10.10.10.3
```

Using scripting engine to scan for vulnerabilites

```
nmap --script vuln 10.10.10.4
```

peforming service detection scan with full scripting engine scan

```
nmap -sC -sV 10.10.10.5
```

Performing connect scan on all ports, specifying a min number of packets and output the results to a file

```
nmap -sT -p- --min-rate 10000 -oA scan-results.txt 10.10.10.5
```

Performing UDP scan with aggressive speed

```
nmap -sU -T4  10.10.10.5
```

Performing stealth and slow scan to try and evade firewalls

```
nmap -sS -T1 -f 10.10.10.5
```

Performing host discovery scan using APP Packets

```
nmap -PR -sn 10.10.210.6/24
```

Performing ICMP timestamp request to discover live hosts

```
nmap -PP -sn 10.10.210.6/24
-PE: ICMP mask request
-PE: ICMP echo request
```

Perfmroing TCP+ICMP scan to discover live hosts

```
nmap -PS -sn 10.10.210.6/24  S: SYN
nmap -PA -sn 10.10.210.6/24  A: Acknowledgement
nmap -PU -sn 10.10.68.220/24 U: UDP
```

You can go with [T0] but it would be very slow.
Acknowledgments scan are also useful for firewall evasion.

```
nmap -sA -T1 -f 10.10.10.5
```

Performing a null scan to bypass firewall and IDS

```
nmap -sN -T1 10.10.210.6
```

Performing FIN scan to bypass firewall and IDS

```
nmap -FN -T1 10.10.210.6
```

Performing Decoy scan to bypass firewall and IDS

```
nmap -D ip1,ip2,yourip 10.10.210.6
```

You can pickup any ip address and you can put as many as you want.

Scanning the target with hping3

```
root@kali:Hping3 -S [ip - domain ] -p [port] -c [number-of-packets-to-send]
### S: For TCP SYN scan
```

```
<root@kali:Hping3 -A [ip - domain ] -p [port] -c [number-of-packets-to-send]>
### A: For TCP ACK scan
```

```
root@kali:Hping3 -S [ip - domain ] -p ++[port] -c [number-of-packets-to-send]
### p ++port: scanning many ports starting from the given port and incrementing by one
```

Kereberos Enumeration

Enumerating usernames and Tickets on Kereberos

```
root@kali:./kerbrute_linux_amd64 userenum -d pentesting.local -dc [ip] [path-to-usernames-wordlist]
```

Check if a user among users in Active directory has a specified password in the input [Password Spray Attack]

```
root@kali:./kerbrute_linux_amd64 passwordspray -v -d pentesting.local -dc [ip] [users-list.txt] [the password]
```

Getting password hashes and TGTs for identified users in the previous Kerebros enumeration [ASREP ROASTING]

```
root@kali:python3 GetNPUsers.py -dc-ip [ip] pentesting.local/ -usersfile [list-of-found-users-from-command-above]
```

Brute forcing usernames and passwords with Kereberos [Kerebroasting]

```
root@kali:python kerbrute.py -domain pentesting.local -users users.txt -passwords passwords.txt -outputfile passwords-found.txt
```

Enumerating web application directories

Regular scan with wordlist

```
root@kali:dirb http://10.5.5.25:8080/ -w
 ### -w: to continue enumerating past the warning
messages
root@kali:gobuster dir -u ‘url’ -w [path-to-wordlist]
```

Directory enumeratio with file extensions specified

```
root@kali:dirb http://10.5.5.25:8080/ -w -e
php,html,txt,js
```

Filtering output

Lets say we want to filter out 403 responses

```
root@kali:dirb http://10.5.5.25:8080/ -w -e
php,html,txt,js  -x 403
```

Increasing the scan speed

```
root@kali:dirb http://10.5.5.25:8080/ -w -e
php,html,txt,js  -x 403 -t 50
```

Enumerating samba shares

```
root@kali:smbclient -N -L \\\\ip
root@kali:smbclient  \\\\ip\\[sharename]
get [filename]
```

Enumerating and interacting with svnserve

Usually runs on port 3690

Connect and display info about the server

```
root@kali:Svn info svn://domain.com
```

Display files on the current directory

```
root@kali:Svn list svn://domain.com
```

Export specific file from the server

```
root@kali:Svn export svn://domain.com/file.txt
```

Checking out revisions

```
root@kali:Svn checkout -r 1 svn://domain.com
root@kali:Svn checkout -r 2 svn://domain.com
```

Enumerating and interacting with RPC clients

Usually run on port 111

Logging in

```
root@kali:Rpcclient [ip-or dns name] -U ‘username’
```

Logging in with hash

```
root@kali:Rpcclient --pw-nt-hash -U [username] [ip-or-domain]
```

Querying and displaying info after logging in

```
rpcclient $>querydispinfo
```

Display users

```
rpcclient $> enumdomusers
```

Display privileges

```
rpcclient $> enumprivs
```

Display Printers

```
rpcclient $> enumprinters
```

Enumerating and interacting with MSRPC TCP 135

Listing Current RCP mappings and interfaces [requires impacket]

```
root@kali:python rpcmap.py 'ncacn_ip_tcp:10.10.10.213'
```

Identifying hosts and other endpoints

```
root@kali:python IOXIDResolver.py -t 10.10.10.21
```

Finding if its vulnerable to PrintNightMare or print spooler service vulnerability CVE-2021-1675 / CVE-2021-34527

```
rpcdump.py @192.168.1.10 | egrep 'MS-RPRN|MS-PAR'
```

rpcdump.py is part of impacket tools.

Enumerating Rsync

Rsync is a linux tool for remote and local file and directory synchronization.

Connecting to a remote rsync server

```
rsync rsync://rsync-connect@ip-address/
```

Listing synced files

```
rsync rsync://rsync-connect@ip-address/Conf
Conf is an example
```

Downloading a file

```
rsync -v rsync://rsync-connect@ip-address/Conf/filename
[~/Desktop/file]
```

Uploading a file back to the server

```
rsync -v [~/Desktop/file] rsync://rsync-connect@ip-
address/Conf/filename
```

Enumerating Drupal CMS

```
root@kali: /opt/droopescan/droopescan scan drupal -u
http://ip]
```

Enumerating Directories, Files, Parameters and Brute Forcing passwords with ffuf

Enumerating Extensions

```
ffuf -u http://MACHINE_IP/indexFUZZ -w
/usr/share/seclists/Discovery/Web-Content/web-
```

```
extensions.txt
```

Enumerating Directories

```
ffuf -u http://MACHINE_IP/FUZZ -w /usr/share/seclists/Discovery/Web-Content/big.txt
```

Enumerating Files

```
ffuf -u http://MACHINE_IP/FUZZ -w /usr/share/seclists/Discovery/Web-Content/raft-medium-words-lowercase.txt -e .php,.txt
```

Filtering for 403 status codes

```
ffuf -u http://MACHINE_IP/FUZZ -w /usr/share/seclists/Discovery/Web-Content/raft-medium-files-lowercase.txt -fc 403
```

Showing only 200 status codes

```
ffuf -u http://MACHINE_IP/FUZZ -w /usr/share/seclists/Discovery/Web-Content/raft-medium-files-lowercase.txt -mc 200
```

Fuzzing parameters

```
 ffuf -u 'http://MACHINE_IP/sqli-labs/Less-1/?FUZZ=1' -c -w /usr/share/seclists/Discovery/Web-Content/burp-parameter-names.txt -fw 39
```

Numeric wordlist as STDOUT

```
$ for i in {0..255}; do echo $i; done | ffuf -u 'http://MACHINE_IP/sqli-labs/Less-1/?id=FUZZ' -c -w - -fw 33
```

Brute Forcing passwords in login forms

```
ffuf -u http://MACHINE_IP/sqli-labs/Less-11/ -c -w /usr/share/seclists/Passwords/Leaked-Databases/hak5.txt -X POST -d 'uname=Dummy&passwd=FUZZ&submit=Submit' -fs 1435 -H 'Content-Type: application/x-www-form-urlencoded'
```

Port scan simple script

This script can be used in the absence of nmap and other scanning tools

```
#!/bin/bash
host=[ip-address]
for port in {1..65535}; do
timeout .1 bash -c "echo >/dev/tcp/$host/$port" &&
echo "port $port is open"
done
echo "Done"
```