Network Discovery

Summary

Nmap

- Ping sweep (No port scan, No DNS resolution)

```
nmap -sn -n --disable-arp-ping 192.168.1.1-254 | grep -v "host down"
-sn : Disable port scanning. Host discovery only.
-n : Never do DNS resolution
```

- Basic NMAP

```
sudo nmap -sSV -p- 192.168.0.1 -oA OUTPUTFILE -T4
sudo nmap -sSV -oA OUTPUTFILE -T4 -iL INPUTFILE.csv

• the flag -sSV defines the type of packet to send to the server and tells Nmap to
try and determine any service on open ports
• the -p- tells Nmap to check all 65,535 ports (by default it will only check the
most popular 1,000)
• 192.168.0.1 is the IP address to scan
• -oA OUTPUTFILE tells Nmap to output the findings in its three major formats at once
using the filename "OUTPUTFILE"
• -iL INPUTFILE tells Nmap to use the provided file as inputs
```

- CTF NMAP

This configuration is enough to do a basic check for a CTF VM

```
nmap -sV -sC -oA ~/nmap-initial 192.168.1.1

-sV : Probe open ports to determine service/version info
-sC : to enable the script
-oA : to save the results
```

```
After this quick command you can add "-p-" to run a full scan while you work with the
previous result
```

- Aggressive NMAP

```
nmap -A -T4 scanme.nmap.org
• -A: Enable OS detection, version detection, script scanning, and traceroute
• -T4: Defines the timing for the task (options are 0-5 and higher is faster)
```

- Using searchsploit to detect vulnerable services

```
nmap -p- -sV -oX a.xml IP_ADDRESS; searchsploit --nmap a.xml
```

- Generating nice scan report

```
nmap -sV IP_ADDRESS -oX scan.xml && xsltproc scan.xml -o "`date +%m%d%y`_report.html"
```

- NMAP Scripts

```
nmap -sC : equivalent to --script=default

nmap --script 'http-enum' -v web.xxxx.com -p80 -oN http-enum.nmap
PORT   STATE SERVICE
80/tcp open  http
| http-enum:
|   /phpmyadmin/: phpMyAdmin
|   /.git/HEAD: Git folder
|   /css/: Potentially interesting directory w/ listing on 'apache/2.4.10 (debian)'
|_  /image/: Potentially interesting directory w/ listing on 'apache/2.4.10 (debian)'

nmap --script smb-enum-users.nse -p 445 [target host]
Host script results:
| smb-enum-users:
|   METASPLOITABLE\backup (RID: 1068)
|     Full name:   backup
|     Flags:       Account disabled, Normal user account
|   METASPLOITABLE\bin (RID: 1004)
|     Full name:   bin
|     Flags:       Account disabled, Normal user account
|   METASPLOITABLE\msfadmin (RID: 3000)
|     Full name:   msfadmin,,,
|     Flags:       Normal user account

List Nmap scripts : ls /usr/share/nmap/scripts/
```

Spyse

- Spyse API - for detailed info is better to check Spyse

- Spyse Wrapper

Searching for subdomains

```
spyse -target xbox.com --subdomains
```

Reverse IP Lookup

```
spyse -target 52.14.144.171 --domains-on-ip
```

Searching for SSL certificates

```
spyse -target hotmail.com --ssl-certificates
```

```
spyse -target "org: Microsoft" --ssl-certificates
```

Getting all DNS records

```
spyse -target xbox.com --dns-all
```

Masscan

```
masscan -iL ips-online.txt --rate 10000 -p1-65535 --only-open -oL masscan.out
masscan -e tun0 -p1-65535,U:1-65535 10.10.10.97 --rate 1000

# find machines on the network
sudo masscan --rate 500 --interface tap0 --router-ip $ROUTER_IP --top-ports 100 $NETWORK -oL masscan_machines.tmp
cat masscan_machines.tmp | grep open | cut -d " " -f4 | sort -u > masscan_machines.lst

# find open ports for one machine
sudo masscan --rate 1000 --interface tap0 --router-ip $ROUTER_IP -p1-65535,U:1-65535 $MACHINE_IP --banners -oL $MACHINE_IP/scans/masscan-ports.lst


# TCP grab banners and services information
TCP_PORTS=$(cat $MACHINE_IP/scans/masscan-ports.lst| grep open | grep tcp | cut -d " " -f3 | tr '\n' ',' | head -c -1)
[ "$TCP_PORTS" ] && sudo nmap -sT -sC -sV -v -Pn -n -T4 -p$TCP_PORTS --reason --version-intensity=5 -oA $MACHINE_IP/scans/nmap_tcp $MACHINE_IP

# UDP grab banners and services information
UDP_PORTS=$(cat $MACHINE_IP/scans/masscan-ports.lst| grep open | grep udp | cut -d "
" -f3 | tr '\n' ',' | head -c -1)
[ "$UDP_PORTS" ] && sudo nmap -sU -sC -sV -v -Pn -n -T4 -p$UDP_PORTS --reason --
version-intensity=5 -oA $MACHINE_IP/scans/nmap_udp $MACHINE_IP
```

Reconnoitre

Dependencies:

- nbtscan
- nmap

```
python2.7 ./reconnoitre.py -t 192.168.1.2-252 -o ./results/ --pingsweep --hostnames -
-services --quick
```

If you have a segfault with nbtscan, read the following quote.

> Permission is denied on the broadcast address (.0) and it segfaults on the gateway (.1) - all other addresses seem fine here.So to mitigate the problem: nbtscan 192.168.0.2-255

Netdiscover

```
netdiscover -i eth0 -r 192.168.1.0/24
Currently scanning: Finished!   |   Screen View: Unique Hosts

20 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 876
─────────────────────────────────────────────────────────────────────────────
IP              At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.1.AA    68:AA:AA:AA:AA:AA     15     630  Sagemcom
192.168.1.XX    52:XX:XX:XX:XX:XX      1      60  Unknown vendor
192.168.1.YY    24:YY:YY:YY:YY:YY      1      60  QNAP Systems, Inc.
192.168.1.ZZ    b8:ZZ:ZZ:ZZ:ZZ:ZZ      3     126  HUAWEI TECHNOLOGIES CO.,LTD
```

Responder

```
responder -I eth0 -A # see NBT-NS, BROWSER, LLMNR requests without responding.
responder.py -I eth0 -wrf
```

Alternatively you can use the Windows version

Bettercap

```
bettercap -X --proxy --proxy-https -T <target IP>
# better cap in spoofing, discovery, sniffer
# intercepting http and https requests,
# targetting specific IP only
```

References

- TODO