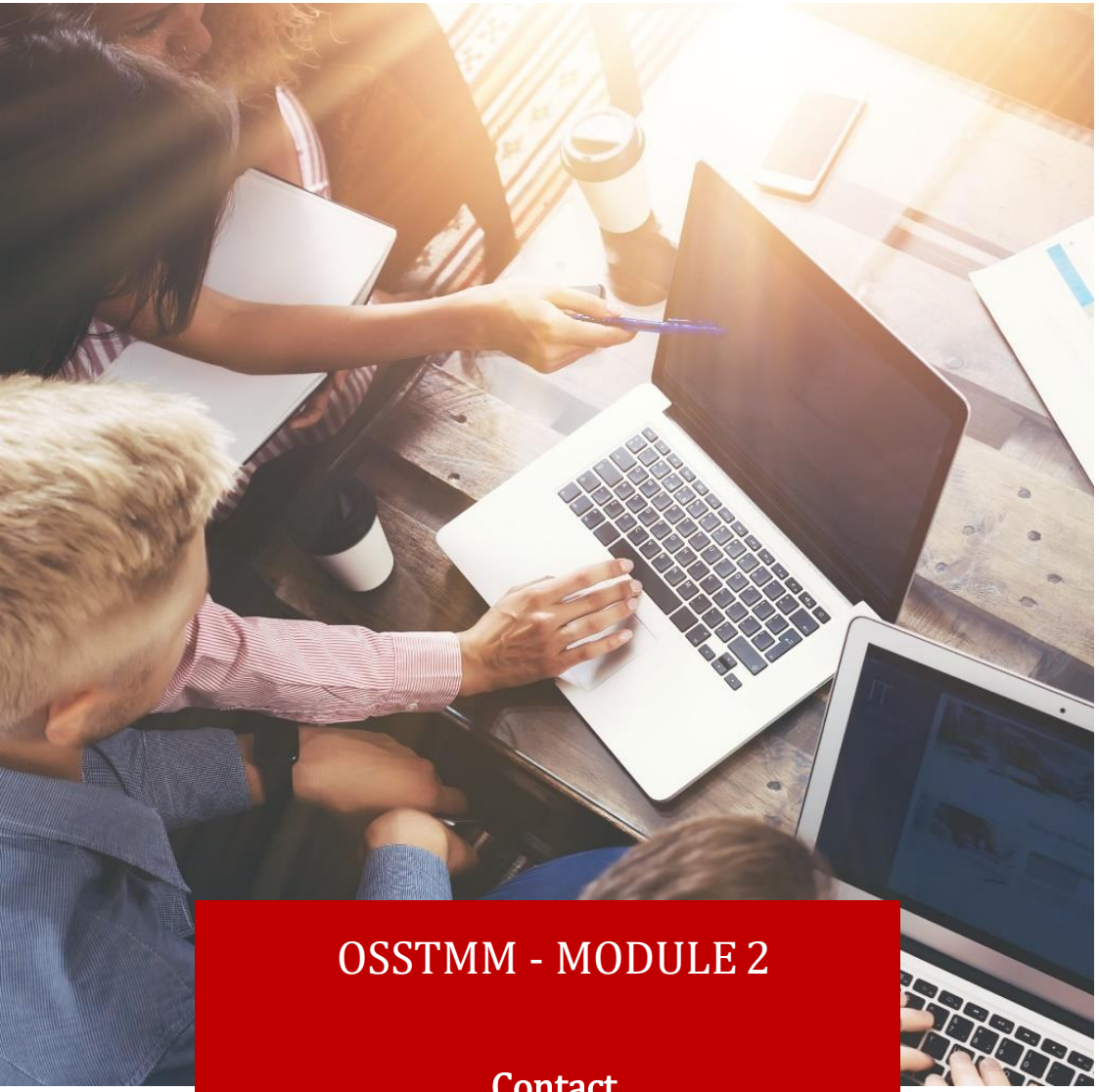


# OSSTMM - MODULE 2

## Contact

Created on 17 November 2021 | Our Reference: OSSTMM_Booklet_Module_2 | Version: 1.8

Sensitivity: PUBLIC

# Summary

# 1 Active information gathering

Direct interactions with the target takes place during the active information gathering phase. If the DNS is hosted by the target, DNS brute force becomes active OSINT. During this phase, pentesters will also fingerprint the operating systems and look for open ports along with running services on the machines they have discovered. Fingerprinting, banner grabbing and zone transfer are common tasks during this step.

## Nslookup / host / whois

Some common tools are useful to achieve the very first steps of information gathering and provide an attacker with the basic information needed to go further.

Tools like *nslookup*, *host* or *whois* search on the internet databases and query DNS to find information about a given IP address, a domain, etc.

*Nnlookup* is a program to query Internet domain name server. *Host* is an alternative to *Nslookup* and both are used to convert names to IP addresses and vice versa. *Whois* (command) is a client for the whois service, which provides information about a domain. Whois.net provides the same service.



**Hands on!**

Find information about Telindus Luxembourg (such as its IP range).

**Hands on!** ANSWERS

The first step is to query the DNS for *telindus.lu*:

```
root@kali:~# nslookup telindus.lu
…
Non-authoritative answer:
Name:   telindus.lu
Address: 31.204.90.51
```

Running a *whois* command on this IP address gives:

```
root@kali:~# whois 31.204.90.51
% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See http://www.ripe.net/db/support/db-terms-conditions.pdf
% Note: this output has been filtered.
%       To receive output for a database update, use the "-B" flag.
```

```
% Information related to '31.204.90.0 - 31.204.90.255'

% Abuse contact for '31.204.90.0 - 31.204.90.255' is
'abuse@proximus.lu'

inetnum:        31.204.90.0 - 31.204.90.255
descr:          Telindus Telecom internal assigned PA part 3
netname:        PA-TTL1
country:        LU
admin-c:        GM17277-RIPE
tech-c:         SG11179-RIPE
status:         ASSIGNED PA
mnt-by:         MNT-TTL
created:        2016-11-10T15:04:28Z
last-modified:  2017-05-12T18:06:38Z
source:         RIPE

person:         Gilles Mulheims
address:        Tango S.A.
address:        177 rue de Luxembourg
address:        L-8077 Bertrange
address:        Luxembourg
phone:          +352 27 777 101
nic-hdl:        GM17277-RIPE
mnt-by:         TANGO-MNT
created:        2013-10-31T09:52:59Z
last-modified:  2015-07-30T10:53:36Z
source:         RIPE

person:         Sebastien Grelot
remarks:        Telindus Telecom / Tango
address:        177 Rue de Luxembourg
address:        L-8077 Bertrange
address:        LUXEMBOURG
phone:          +352691777470
nic-hdl:        SG11179-RIPE
mnt-by:         MNT-TTL
created:        2013-03-22T09:13:45Z
last-modified:  2017-04-07T12:41:08Z
source:         RIPE

% Information related to '31.204.90.0/23AS56665'

route:          31.204.90.0/23
descr:          Telindus Telecom IPv4 allocation
origin:         AS56665
mnt-by:         MNT-TTL
created:        2011-06-22T15:28:21Z
last-modified:  2013-03-22T09:48:49Z
source:         RIPE

% This query was served by the RIPE Database Query Service version
1.91.2 (BLAARKOP)
```

The IP range of Telindus is *31.204.90.0/23*.

1- The dig command is also useful to query DNS about a domain:

```
root@kali:~# dig telindus.lu ANY

; <<>> DiG 9.17.19-1-Debian <<>> telindus.lu ANY
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60422
;; flags: qr rd ra; QUERY: 1, ANSWER: 11, AUTHORITY: 0, ADDITIONAL: 5

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4000
;; QUESTION SECTION:
;telindus.lu.                   IN      ANY

;; ANSWER SECTION:
telindus.lu.            375     IN      A       31.204.90.51
telindus.lu.            600     IN      NS
ns1.telindustelecom.lu.
telindus.lu.            600     IN      NS
ns2.telindustelecom.lu.
telindus.lu.            600     IN      SOA
ns1.telindustelecom.lu. ict.telindus.lu. 2012090645 3600 1800 864000
300
telindus.lu.            474     IN      MX      10 mx.proximus.lu.
telindus.lu.            600     IN      TXT     "v=spf1 mx
ip4:208.185.235.0/24 ip4:208.185.229.0/24 include:_spf.google.com
a:mail.clusil.lu a:smtp1.alarmtilt.net a:tangomail.tango.lu
a:mx.proximus.lu ~all"
telindus.lu.            600     IN      TXT     "MS=ms64573797"
telindus.lu.            600     IN      TXT     "google-site-
verification=Xif9HmcJ7MrJgn-ognF8ijAsJFcy75iHWt_2dC_lcCI"
telindus.lu.            600     IN      TXT     "MS=ms67386966"
telindus.lu.            600     IN      TXT     "cisco-ci-domain-
verification=42a452dccbaa4b4ade7bfc2c62bef0e282a4772ea7d7e0dfa43b562ca
69be552"
telindus.lu.            600     IN      TXT
"MS=E972B3C966E71E2772BEBFC52A81C054D3ED9418"

;; ADDITIONAL SECTION:
ns1.telindustelecom.lu. 16028   IN      A       31.204.88.225
ns2.telindustelecom.lu. 16028   IN      A       31.204.88.226
mx.proximus.lu.         3474    IN      A       31.204.93.82
mx.proximus.lu.         3474    IN      A       185.3.45.44

;; Query time: 8 msec
;; SERVER: 192.168.224.2#53(192.168.224.2) (TCP)
;; WHEN: Wed Nov 17 19:15:05 CET 2021
;; MSG SIZE  rcvd: 705
```

2- This information is also available on *ripe.net*.

# DNS Reconnaissance

While *NSLookup* uses the DNS, it does in a gentle way, which should not be suspicious for the target. In this section, more active information gathering will be performed on DNS, such as brute force or zone transfer.

*DNSRecon* is an already integrated tool to Kali Linux which checks NS Records for Zone Transfer, enumerate general DNS Records for a given domain, brute force subdomain, etc.

Useful links:

- github.com/darkoperator/dnsrecon

## Hands on!

Run DNSRecon on the domain "sags.lu".

## Hands on! ANSWERS

```
root@kali:~# dnsrecon -d sags.lu -D /usr/share/dnsenum/dns.txt -t std
[*] Performing General Enumeration of Domain:sags.lu
[-] DNSSEC is not configured for sags.lu
[*]      SOA lan-w2k16adc01.sags.lu 192.168.4.253
[*]      NS lan-w2k16adc01.sags.lu 192.168.4.253
[*]      NS lan-w2k16adc02.sags.lu 192.168.4.254
[-] Could not Resolve MX Records for sags.lu
[*]      A sags.lu 192.168.4.253
[*]      A sags.lu 192.168.4.254
[*] Enumerating SRV Records
[*]      SRV _ldap._tcp.sags.lu LAN-W2K16ADC01.sags.lu 192.168.4.253
389 100
[*]      SRV _ldap._tcp.sags.lu LAN-W2K16ADC02.sags.lu 192.168.4.254
389 100
[*]      SRV _kerberos._tcp.sags.lu LAN-W2K16ADC02.sags.lu
192.168.4.254 88 100
[*]      SRV _kerberos._tcp.sags.lu LAN-W2K16ADC01.sags.lu
192.168.4.253 88 100
[*]      SRV _kerberos._udp.sags.lu LAN-W2K16ADC01.sags.lu
192.168.4.253 88 100
[*]      SRV _kerberos._udp.sags.lu LAN-W2K16ADC02.sags.lu
192.168.4.254 88 100
[*]      SRV _gc._tcp.sags.lu DZY-W2K16ADC01.sagsdmz.sags.lu
192.168.12.11 3268 100
[*]      SRV _gc._tcp.sags.lu LAN-W2K16ADC02.sags.lu 192.168.4.254
3268 100
[*]      SRV  gc. tcp.sags.lu LAN-W2K16ADC01.sags.lu 192.168.4.253
3268 100
[*]      SRV _ldap._tcp.pdc._msdcs.sags.lu LAN-W2K16ADC01.sags.lu
192.168.4.253 389 100
```

```
[*]      SRV _kerberos._tcp.dc._msdcs.sags.lu LAN-W2K16ADC01.sags.lu
192.168.4.253 88 100
[*]      SRV _kerberos._tcp.dc._msdcs.sags.lu LAN-W2K16ADC02.sags.lu
192.168.4.254 88 100
[*]      SRV _ldap._tcp.dc._msdcs.sags.lu LAN-W2K16ADC02.sags.lu
192.168.4.254 389 100
[*]      SRV _ldap._tcp.dc._msdcs.sags.lu LAN-W2K16ADC01.sags.lu
192.168.4.253 389 100
[*]      SRV _kpasswd._udp.sags.lu LAN-W2K16ADC01.sags.lu
192.168.4.253 464 100
[*]      SRV _kpasswd._udp.sags.lu LAN-W2K16ADC02.sags.lu
192.168.4.254 464 100
[*]      SRV _kpasswd._tcp.sags.lu LAN-W2K16ADC01.sags.lu
192.168.4.253 464 100
[*]      SRV _kpasswd._tcp.sags.lu LAN-W2K16ADC02.sags.lu
192.168.4.254 464 100
[*]      SRV _ldap._tcp.ForestDNSZones.sags.lu DZY-
W2K16ADC01.sagsdmz.sags.lu 192.168.12.11 389 100
[*]      SRV _ldap._tcp.ForestDNSZones.sags.lu LAN-W2K16ADC02.sags.lu
192.168.4.254 389 100
[*]      SRV _ldap._tcp.ForestDNSZones.sags.lu LAN-W2K16ADC01.sags.lu
192.168.4.253 389 100
[*]      SRV _ldap._tcp.gc._msdcs.sags.lu DZY-
W2K16ADC01.sagsdmz.sags.lu 192.168.12.11 3268 100
[*]      SRV _ldap._tcp.gc._msdcs.sags.lu LAN-W2K16ADC02.sags.lu
192.168.4.254 3268 100
[*]      SRV _ldap._tcp.gc._msdcs.sags.lu LAN-W2K16ADC01.sags.lu
192.168.4.253 3268 100
[*]      SRV _ldap._tcp.gc._msdcs.sags.lu DZY-
W2K16ADC02.sagsdmz.sags.lu 192.168.12.12 3289 100
[+] 25 Records Found
```



**Hands on!** MORE INFO

*Dnsenum* is another useful tool to bruteforce subdomains available on a top-level domain..

A quick demo to show how powerful gathering information about subdomains is powerful:

```
└─$ dnsenum telindus.lu
1 ×
dnsenum VERSION:1.2.6

-----   telindus.lu   -----


Host's addresses:
__________________

telindus.lu.                             5        IN    A
31.204.90.51
```

```
Name Servers:


ns1.telindustelecom.lu.                    5        IN    A
31.204.88.225
ns2.telindustelecom.lu.                    5        IN    A
31.204.88.226


Mail (MX) Servers:
___________________

mx.proximus.lu.                            5        IN    A
185.3.45.44
mx.proximus.lu.                            5        IN    A
31.204.93.82


Trying Zone Transfers and getting Bind Versions:
________________________________________________


Trying Zone Transfer for telindus.lu on ns2.telindustelecom.lu ...
AXFR record query failed: REFUSED

Trying Zone Transfer for telindus.lu on ns1.telindustelecom.lu ...
AXFR record query failed: REFUSED


Brute forcing with /usr/share/dnsenum/dns.txt:
_____________________________________________

enquete.telindus.lu.                       5        IN    A
80.92.66.135
lab.telindus.lu.                           5        IN    A
213.135.244.22
marketing.telindus.lu.                     5        IN    A
31.204.90.74
register.telindus.lu.                      5        IN    CNAME
formation.telindus.lu.
formation.telindus.lu.                     5        IN    A
31.204.90.85
training.telindus.lu.                      5        IN    CNAME
formation.telindus.lu.
formation.telindus.lu.                     5        IN    A
31.204.90.85
vsp.telindus.lu.                           5        IN    A
31.204.90.58
www.telindus.lu.                           5        IN    A
31.204.90.51


telindus.lu class C netranges:
______________________________

 31.204.90.0/24
```

```
 80.92.66.0/24
 213.135.244.0/24


Performing reverse lookup on 768 ip addresses:
_____________________________________________

 35.90.204.31.in-addr.arpa.               10800    IN    PTR
mcis-external.telindus.lu.
35.90.204.31.in-addr.arpa.               10800    IN    PTR      trfs-
external.telindus.lu.
39.90.204.31.in-addr.arpa.               10800    IN    PTR
call.telindus.lu.
39.90.204.31.in-addr.arpa.               10800    IN    PTR
mraedge1.telindus.lu.
39.90.204.31.in-addr.arpa.               10800    IN    PTR
b2bedge1.telindus.lu.
50.90.204.31.in-addr.arpa.               10800    IN    PTR
tlu1dme02.telindus.lu.
51.90.204.31.in-addr.arpa.               10800    IN    PTR
sentryap.telindus.lu.
51.90.204.31.in-addr.arpa.               10800    IN    PTR
sentryasuat.telindus.lu.
51.90.204.31.in-addr.arpa.               10800    IN    PTR
sentryatuat.telindus.lu.
51.90.204.31.in-addr.arpa.               10800    IN    PTR      actu-
mob.telindus.lu.
51.90.204.31.in-addr.arpa.               10800    IN    PTR
it2be.telindus.lu.
51.90.204.31.in-addr.arpa.               10800    IN    PTR
pxlshare-10.telindus.lu.
51.90.204.31.in-addr.arpa.               10800    IN    PTR
sts.telindus.lu.
51.90.204.31.in-addr.arpa.               10800    IN    PTR
sentryaspxl.telindus.lu.
51.90.204.31.in-addr.arpa.               10800    IN    PTR      u-
share.telindus.lu.
```

# Fingerprinting

Fingerprinting is a vital step as it allows pentesters to learn more about the devices that live on their target's IT Infrastructure. Using information collected through the network or thanks to banners, pentesters will even know which services are running on which machines.

## Using ping

The way machines are communicating on a network leaks enough data to determine which Operating System is running on a given host. For instance, not all Operating Systems have the same TTL, and an OS could be fingerprinted with a simple ping command (cf. hands on).

| Operating System | IP Initial TTL | TCP Window size |
|---|---|---|
| Linux (kernel 2.4 and 2.6) | 64 | 5840 |
| Google's customized Linux | 64 | 5720 |
| FreeBSD | 64 | 65535 |
| Windows XP | 128 | 65535 |
| Windows 7, Vista and server 2008 | 128 | 8192 |
| Cisco Router (IOS 12.4) | 255 | 4128 |

See the full table on http://www.kellyodonnell.com/content/determining-os-type-ping.

## Hands on!

Write a Windows / Linux command, which ping all 192.168.22.0/24 range.

## Hands on! ANSWERS

**Warning**: This technique does not work when using a Virtual Machine with NAT: use bridge instead. Indeed, trying this method in a Kali Linux running in VMWare results in all machine being detected as Windows (TTL is always 128). That is because the NAT alters the IP packet and changes its TTL.

On Windows:

```
FOR /L %i IN (1,1,254) DO ping -n 1 192.168.22.%i | FIND /i
"Reply">>C:\ipaddresses.txt
```

On Linux:

Here is a small bash script to scan the 192.168.22.0/24 network:

```
# /bin/bash
RANGE="192.168.22.X";
```

```
echo "Starting scan for range $RANGE";

for i in `seq 1 254`; do
        IP="${RANGE/X/$i}"
        echo "[*] Pinging $IP...";
        ping_res="$(ping -W 1 -c1 $IP | grep from | cut -d' ' -f6 |
cut -d'=' -f2)"
        if [ ! -z $ping_res ]; then
                echo "    Host is up, guessing OS...";
                if [ "$ping_res" -lt 64 ]; then
                        echo "    OS might be Linux.";
                        continue;
                fi
                if [ "$ping_res" -lt 128 ];  then
                        echo "    OS might be Windows.";
                        continue;
                fi
                if [ "$ping_res" -lt 255 ];  then
                        echo "    OS might be Cisco.";
                        continue;
                fi
        fi
done
```

Running it on the lab gives:

```
root@kali:~# ./ping-scan.sh
Starting scan for range 192.168.22.X
[*] Pinging 192.168.22.2...
    Host is up, guessing OS...
    OS might be Linux.
[*] Pinging 192.168.22.2...
[*] Pinging 192.168.22.3...
[*] Pinging 192.168.22.4...
[*] Pinging 192.168.22.5...
[*] Pinging 192.168.22.6...
[*] Pinging 192.168.22.7...
[*] Pinging 192.168.22.8...
[*] Pinging 192.168.22.9...
[*] Pinging 192.168.22.20...
    Host is up, guessing OS...
    OS might be Linux.
[*] Pinging 192.168.22.21...

        [SKIPPED]

[*] Pinging 192.168.22.20...
[*] Pinging 192.168.22.10...
    Host is up, guessing OS...
    OS might be Linux.

        [SKIPPED]

[*] Pinging 192.168.22.40...
    Host is up, guessing OS...
    OS might be Windows.
```

```
        [SKIPPED]

[*] Pinging 192.168.22.50...
    Host is up, guessing OS...
    OS might be Windows.
[*] Pinging 192.168.22.42...

        [SKIPPED]
```

## Banner grabbing with netcat

Banners are messages received from a host that usually contain information about a service such as the name or the version number. Banner grabbing consists in collecting this data to learn more about services running on a target host. It can be achieved using direct connection to the host or using online tools.

Here is a banner example:

```
Date: Wed, 20 Jun 2018 13:39:10 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2 mod_fastcgi/2.4.6 PHP/5.2.4-
2ubuntu5 with Suhosin-Patch mod_ssl/2.2.8 OpenSSL/0.9.8g
Last-Modified: Sun, 02 Nov 2014 18:20:24 GMT
ETag: "ccb16-24c-506e4489b4a00"
Accept-Ranges: bytes
Content-Length: 588
Content-Type: text/html
X-Cache: MISS from localhost
X-Cache-Lookup: MISS from localhost:3128
Connection: keep-alive
```

*Netcat* is a network utility whose role is to read and writes data across network connection using UDP and TCP protocols. It can act as both server (listening for incoming connections) and client (initiating connections to a given host).

Banners collection can also be achieved with this tool.

### Hands on!

What is the webserver running on 192.168.22.2 ?

What is the SMTP server running on mail.sags.lu ?

### Hands on! ANSWERS

An Apache Server seems to be running on 192.168.22.2:

```
root@kali:~# nc 192.168.22.2 80
GET / HTTP/1.1

HTTP/1.1 200 OK
Date: Wed, 13 Jun 2018 11:56:10 GMT
Server: Apache/2.4.18 (Ubuntu)
Last-Modified: Tue, 24 Oct 2017 10:04:41 GMT
ETag: "2c39-55c481153905e"
Accept-Ranges: bytes
Content-Length: 11321
Vary: Accept-Encoding
Content-Type: text/html
X-Cache: MISS from localhost
X-Cache-Lookup: MISS from localhost:3128
Connection: keep-alive
```

A Microsoft Exchange server is running on mail.sags.lu:

```
root@kali:~/Documents/Tools# nc mail.sags.lu 25
220 LAN-W2K16EXG01.sags.lu Microsoft ESMTP MAIL Service ready at Wed,
13 Jun 2018 13:56:23 +0200
```

## Getting to know the OS using *nmap*

*Nmap* (Network Mapper) is a security scanner used to discover hosts on a network. *Nmap* has dozens of options available and can deal with many tasks from discovery scan to vulnerability scan but also banner grabbing etc. It provides functionalities to discover services running on these hosts and can guess OS type. *Nmap* has many other functionalities and will be covered deeper in the next sections.

### Hands on!

What is the OS running on 192.168.22.40?

### Hands on! ANSWERS

Using the –O option to enable OS Detection:

```
root@kali:~# nmap -O 192.168.22.40
Running: Microsoft Windows XP|7|2012
OS CPE: cpe:/o:microsoft:windows_xp::sp3 cpe:/o:microsoft:windows_7
cpe:/o:microsoft:windows_server_2012
OS details: Microsoft Windows XP SP3, Microsoft Windows XP SP3 or
Windows 7 or Windows Server 2012
```

Adding some service detection (-sV option) gives results that are more accurate:

```
root@kali:~# nmap -O -sV 192.168.22.40
. . .
Service Info: Host: tst-wxp-build26; OSs: Windows, Windows XP; CPE:
cpe:/o:microsoft:windows, cpe:/o:microsoft:windows_xp
. . .
```

## Port scanning

Port scanning consists in sending requests to a range of port addresses on a host in order to find active ports. These probes allow pentesters to determine available services on a remote machine. The standard tool used to perform port scanning is also *Nmap*.

Many scan types exist, such as TCP scan, SYN scan, UDP. Each scan allows an attacker to discover more information on a network or a host. ARP scans are useful to get a list of active host on the network, XMAS scans might give listening port, etc. However, some scans make more noise than other does so an attacker might be careful by using them.

Tips: to get a list of services per port, use this command along with *grep*/*more*:

```
root@kali:~# sort -r -k3 /usr/share/nmap/nmap-services | sed '/^#/ d'
| cut -d$'\t' -f1,2 | column -t | more
```



### Hands on!

1. Basic questions
    a. How to save results in *Nmap*?
    b. How to import targets?

2. Scanning
    a. Scan 192.168.22.0/24 with a protocol scan.
    b. Scan 192.168.22.0/24 with a TCP full port scan.
    c. Scan 192.168.22.0/24 with an UDP top port 20 scan.

3. Scanning options

a. Scan 192.168.22.0/24 with a TCP top port 1000 and add the following options:

i. - - reason: Did you notice any change?

ii. – O: Did you notice any change?

iii. - - osscan-guess: Did you notice any change?

iv. – Pn: What is the goal of this option?

| Hands on! | ANSWERS |
|---|---|

1. Basic questions

a. How to save results? In the three major formats at once.

```
root@kali:~# nmap -oA filename
```

b. How to import targets? One target per line in *filename*.

```
root@kali:~# nmap -iL filename
```

2. Scanning

a. Scan 192.168.22.0/24 with a protocol scan.

```
root@kali:~# nmap -sO 192.168.22.0/24
```

b. Scan 192.168.22.0/24 with a TCP full port scan.

```
root@kali:~# nmap -sS -p- 192.168.22.0/24
```

c. Scan 192.168.22.0/24 with an UDP top port 20 scan.

```
root@kali:~# nmap -sU --top-port 20 192.168.22.0/24
```

3. Scanning options

a. Scan 192.168.22.0/24 with a TCP top port 1000 and add the following options:

i. - - reason: Did you notice any change?

```
Explain why a port is filtered or closed.
```

ii. – O: Did you notice any change?

```
Enable OS detection.
```

iii. - - osscan-guess: Did you notice any change?

```
Try to guess OS more aggressively.
```

iv. – Pn: What is the goal of this option?

```
Treat all hosts as online (scan a machine even if it does not reply to
ping).
```

Hands on! MORE INFO

Scripts could also help to determine the operating system or even the computer name, its workgroup, etc. For instance, one can use the *smb-os-discovery* script with *nmap*:

```
root@kali:~/Documents/Tools/Scripts# nmap --script smb-os-
discovery.nse -p445 192.168.22.40
Starting Nmap 7.70 ( https://nmap.org ) at 2018-06-13 15:51 CEST
Nmap scan report for 192.168.22.40
Host is up (0.00074s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds

Host script results:
| smb-os-discovery:
|   OS: Windows XP (Windows 2000 LAN Manager)
|   OS CPE: cpe:/o:microsoft:windows_xp::-
|   Computer name: tst-wxp-build26
|   NetBIOS computer name: TST-WXP-BUILD26\x00
|   Workgroup: WORKGROUP\x00
|_  System time: 2018-06-13T15:52:00+02:00

Nmap done: 1 IP address (1 host up) scanned in 1.08 seconds
```



NB: nmap has also a grepable output option (-oG). The line is then split into fields separated with a tabulation. This output could be useful if you plan to script things.

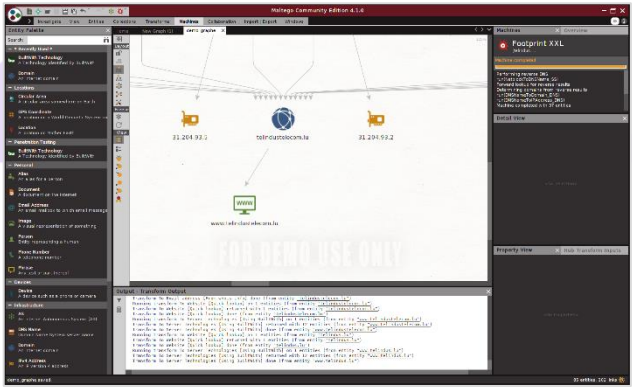
Output example: **`Host: 64.13.134.52 (scanme.nmap.org)`**

# 2 Frameworks

## Maltego

Maltego is a proprietary software used for information gathering. It is available in both commercial and community edition. It focuses on data representation, mining and analysis. It also has a transforms library which is a real asset regarding data collection and discovery.

| HANDS ON | DEMO |
|---|---|




Starting from the domain *telindus.lu* and performing a "Footprinting XXL" (Machines > Run machine > Footprinting XXL) a new domain is obtained and can now be explored (*telindustelecom.lu* ).

# Metasploit

Metasploit (or msf) is a penetration testing framework that enables pentesters (and hackers) to find, exploit and validate vulnerabilities. Both commercial and community editions of this framework exist. This courses use the community edition. Metasploit is capable of automating a lot of the pentester job: discovery and vulnerability scans, exploit, etc.

Moreover, it works great with *nmap*, Nessus and many others tools.

## Port scanning with MSF

*msf* has a command to launch *nmap* within the framework and to automatically store results in msf database. Its name is "*db_nmap*". For instance, the following command will perform a scan on the 192.168.22.0/24 network and store the results in the msf database.

```
msf > db_nmap -sn 192.168.22.0/24
```

If an nmap scan is likely to be imported in multiple frameworks, a db_import command exists.

```
root@kali:~/ nmap 192.168.22.2 -oX scan.xml
root@kali:~/ msfconsole

                SKIPPED

msf > db_import /path/to/scan.xml
```

The "*hosts*" command displays the hosts in the msf database.

The "*services*" command displays the detected services in the msf database.

If for some reasons, *nmap* is not installed on the system, *msf* has also many modules that do portscan:

```
msf > search portscan

Matching Modules
================

   Name                                                Disclosure Date  Rank    Description
   ----                                                ---------------  ----    -----------
   auxiliary/scanner/http/wordpress_pingback_access                     normal  Wordpress Pingback Locator
   auxiliary/scanner/natpmp/natpmp_portscan                             normal  NAT-PMP External Port Scanner
   auxiliary/scanner/portscan/ack                                       normal  TCP ACK Firewall Scanner
   auxiliary/scanner/portscan/ftpbounce                                 normal  FTP Bounce Port Scanner
   auxiliary/scanner/portscan/syn                                       normal  TCP SYN Port Scanner
   auxiliary/scanner/portscan/tcp                                       normal  TCP Port Scanner
   auxiliary/scanner/portscan/xmas                                      normal  TCP "XMas" Port Scanner
   auxiliary/scanner/sap/sap_router_portscanner                         normal  SAPRouter Port Scanner
```

**Hands on!**

Use the *db_nmap* command to obtain up hosts on the 192.168.22.0/24 network.

Give a try to the "*hosts*" command to visualize the results.

Detect services and OS running on the 192.168.22.2 host. Visualize results with the "*hosts*" and the "*services*" command.

| Hands on! | ANSWERS |
|---|---|

First, perform a *db_nmap* to get up hosts:

```
msf > db_nmap -sn 192.168.22.0/24
[*] Nmap: Starting Nmap 7.70 ( https://nmap.org ) at 2018-06-18 14:23
CEST
[*] Nmap: Nmap scan report for 192.168.22.2
[*] Nmap: Host is up (0.0013s latency).
[*] Nmap: Nmap scan report for 192.168.22.2
[*] Nmap: Host is up (0.00073s latency).
              SKIPPED
[*] Nmap: Host is up (0.0013s latency).
[*] Nmap: Nmap scan report for 192.168.22.254
[*] Nmap: Host is up (0.00079s latency).
[*] Nmap: Nmap done: 256 IP addresses (13 hosts up) scanned in 4.56
seconds
```

"*Hosts*" command shows hosts that are stored in the msf database with information that have been gathered on them (such as MAC Address, OS, etc.):

```
msf > hosts

Hosts
=====

address         mac  name  os_name  os_flavor  os_sp  purpose  info
-------         ---  ----  -------  ---------  -----  -------  ----
192.168.22.2
192.168.22.2
192.168.22.3
192.168.22.20
192.168.22.10
192.168.22.40
192.168.22.50
192.168.22.53
192.168.22.203
192.168.22.204
192.168.22.252
192.168.22.253
192.168.22.254
```

As the information gathering progresses, information is automatically stored in the database. For instance, when performing an OS and services detection on 192.168.22.2:

```
msf > db_nmap -A 192.168.22.2
```

```
[*] Nmap: Starting Nmap 7.70 ( https://nmap.org ) at 2018-06-18 14:27
CEST
[*] Nmap: Nmap scan report for 192.168.22.2
[*] Nmap: Host is up (0.0010s latency).
[*] Nmap: Not shown: 997 filtered ports
[*] Nmap: PORT    STATE  SERVICE VERSION
[*] Nmap: 22/tcp  open   ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.2
(Ubuntu Linux; protocol 2.0)
[*] Nmap: | ssh-hostkey:
[*] Nmap: |   2048 ea:98:1d:1a:ba:3f:d3:6b:a2:53:a5:63:5d:e9:8e:54
(RSA)
[*] Nmap: |   256 ef:59:b6:68:31:96:8b:09:9a:9e:65:04:3f:e8:c9:78
(ECDSA)
[*] Nmap: |_  256 69:18:17:40:d2:c0:56:db:7e:35:68:a0:c8:af:05:e3
(ED25519)
[*] Nmap: 80/tcp  open   http    Apache httpd 2.4.18 ((Ubuntu))
[*] Nmap: |_http-server-header: Apache/2.4.18 (Ubuntu)
[*] Nmap: |_http-title: Apache2 Ubuntu Default Page: It works
[*] Nmap: 443/tcp closed https
[*] Nmap: Device type: general purpose
[*] Nmap: Running: Linux 3.X|4.X
[*] Nmap: OS CPE: cpe:/o:linux:linux_kernel:3
cpe:/o:linux:linux_kernel:4
[*] Nmap: OS details: Linux 3.11 - 4.1
[*] Nmap: Network Distance: 1 hop
[*] Nmap: Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
[*] Nmap: TRACEROUTE (using port 80/tcp)
[*] Nmap: HOP RTT     ADDRESS
[*] Nmap: 1   0.76 ms 192.168.22.2
[*] Nmap: OS and Service detection performed. Please report any
incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 16.21 seconds

msf > hosts

Hosts
=====

address         mac  name  os_name  os_flavor  os_sp  purpose  info
-------         ---  ----  -------  ---------  -----  -------  ----
192.168.22.2              Linux              3.X    server
192.168.22.2
192.168.22.3
192.168.22.20
192.168.22.10
192.168.22.40
192.168.22.50
192.168.22.53
192.168.22.203
192.168.22.204
192.168.22.252
192.168.22.253
192.168.22.254
```

Using the "*services*" command also shows the collected services for each host:

```
msf > services
Services
========

host           port  proto  name   state   info
----           ----  -----  ----   -----   ----
192.168.22.2   22    tcp    ssh    open    OpenSSH 7.2p2 Ubuntu
4ubuntu2.2 Ubuntu Linux; protocol 2.0
192.168.22.2   80    tcp    http   open    Apache httpd 2.4.18 (Ubuntu)
192.168.22.2   443   tcp    https  closed
```

## Fingerprinting OS

Metasploit has also modules to fingerprint the OS (such as the SMB version). The "search" command is useful to find a module.

```
msf > search portscan

Matching Modules
================

   Name                                              Rank    Description
   ----                                              ----    -----------
   auxiliary/scanner/http/wordpress_pingback_access  normal  Wordpress Pingback Locator
   auxiliary/scanner/natpmp/natpmp_portscan          normal  NAT-PMP External Port Scanner
   auxiliary/scanner/portscan/ack                    normal  TCP ACK Firewall Scanner
   auxiliary/scanner/portscan/ftpbounce              normal  FTP Bounce Port Scanner
   auxiliary/scanner/portscan/syn                    normal  TCP SYN Port Scanner
   auxiliary/scanner/portscan/tcp                    normal  TCP Port Scanner
   auxiliary/scanner/portscan/xmas                   normal  TCP "XMas" Port Scanner
   auxiliary/scanner/sap/sap_router_portscanner      normal  SAPRouter Port Scanner

msf > use auxiliary/scanner/portscan/tcp
```

Modules often require a bit of configuration before being able to run. To see which options are required, there is the "*show options*" command. Then setting an option is achieved with *"**set OPTION VALUE**"*.



### Hands on!

Find the module that does a SMB Version Detection and use it on 192.168.22.50.

### Hands on! ANSWERS

The first step is searching for the right module to use with the "search" command:

```
msf > search type:auxiliary smb

Matching Modules
================

   Name                                                Disclosure Date  Rank    Description
   ----                                                ---------------  ----    -----------
   auxiliary/scanner/smb/impacket/dcomexec             2018-03-19       normal  DCOM Exec
   auxiliary/scanner/smb/pipe auditor                                   normal  SMB Session Pipe
Auditor
   auxiliary/scanner/smb/pipe_dcerpc_auditor                            normal  SMB Session Pipe
DCERPC Auditor
```

```
   auxiliary/scanner/smb/psexec loggedin users                        normal  Microsoft Windows
Authenticated Logged In Users Enumeration
   auxiliary/scanner/smb/smb1                                         normal  SMBv1 Protocol
Detection
   auxiliary/scanner/smb/smb2                                         normal  SMB 2.0 Protocol
Detection
   auxiliary/scanner/smb/smb_enum_gpp                                 normal  SMB Group Policy
Preference Saved Passwords Enumeration
   auxiliary/scanner/smb/smb enumshares                               normal  SMB Share
Enumeration
   auxiliary/scanner/smb/smb enumusers                                normal  SMB User Enumeration
(SAM EnumUsers)
   auxiliary/scanner/smb/smb enumusers domain                         normal  SMB Domain User
Enumeration
   auxiliary/scanner/smb/smb_login                                    normal  SMB Login Check
Scanner
   auxiliary/scanner/smb/smb lookupsid                                normal  SMB SID User
Enumeration (LookupSid)
   auxiliary/scanner/smb/smb ms17 010                                 normal  MS17-010 SMB RCE
Detection
   auxiliary/scanner/smb/smb uninit cred                              normal  Samba
 netr ServerPasswordSet Uninitialized Credential State
   auxiliary/scanner/smb/smb_version                                  normal  SMB Version
Detection
```

Next step is using "*smb_version*", showing the options and setting the required one.

```
msf > use auxiliary/scanner/smb/smb_version

msf auxiliary(scanner/smb/smb_version) > show options

Module options (auxiliary/scanner/smb/smb_version):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   RHOSTS                      yes       The target address range or CIDR
identifier
   SMBDomain  .                no        The Windows domain to use for
authentication
   SMBPass                     no        The password for the specified username
   SMBUser                     no        The username to authenticate as
   THREADS    1                yes       The number of concurrent threads

setting (not globally) the target

msf auxiliary(scanner/smb/smb_version) > set RHOSTS 192.168.22.50
RHOSTS => 192.168.22.50
```

Last step is running the module with the "*run*" command:

```
msf auxiliary(scanner/smb/smb_version) > run

[+] 192.168.22.50:445     - Host is running Windows 2003 SP2 (build:3790)
(name:SAGS-FXWV1C5WK5) (workgroup:WORKGROUP )
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed

msf auxiliary(scanner/smb/smb_version) > hosts

Hosts
=====

address         mac  name             os_name       os_flavor  os_sp  purpose
-------         ---  ----             -------       ---------  -----  -------
192.168.22.2                          Linux                    3.X    server
```

```
192.168.17.2
192.168.17.3
192.168.22.20
192.168.22.10
192.168.22.40
192.168.22.50        SAGS-FXWV1C5WK5  Windows 2003             SP2    server
192.168.22.60
192.168.17.203
192.168.17.204
192.168.17.252
192.168.17.253
192.168.17.254

msf auxiliary(scanner/smb/smb_version) > services

Services
========

host            port  proto  name   state   info
----            ----  -----  ----   -----   ----
192.168.22.2    22    tcp    ssh    open    OpenSSH 7.2p2 Ubuntu 4ubuntu2.2
Ubuntu Linux; protocol 2.0
192.168.22.2    80    tcp    http   open    Apache httpd 2.4.18 (Ubuntu)
192.168.22.2    443   tcp    https  closed
192.168.22.50   445   tcp    smb    open    Windows 2003 SP2 (build:3790)
(name:SAGS-FXWV1C5WK5) (workgroup:WORKGROUP )
```

## Service and version detection

Once an attacker completed services detection on a host, he may have information on opened ports and on which service might be running on a given port. However, during a pentest it is important to collect as much information as possible. That is why it is relevant to try detecting versions.

### Hands on!

Perform a TCP scan on 192.168.22.10:1-100 using a msf module (and not db_nmap).

Find a way to obtain ssh and ftp version running on 192.168.22.10 using msf.

### Hands on! ANSWERS

Using the TCP scan module:

```
msf > use auxiliary/scanner/portscan/tcp
msf auxiliary(scanner/portscan/tcp) > set RHOSTS 192.168.22.10
RHOSTS => 192.168.22.10
msf auxiliary(scanner/portscan/tcp) > set PORTS 1-100
PORTS => 1-100
msf auxiliary(scanner/portscan/tcp) > run

[+] 192.168.22.10:        - 192.168.22.10:25 - TCP OPEN
```

```
[+] 192.168.22.10:        - 192.168.22.10:23 - TCP OPEN
[+] 192.168.22.10:        - 192.168.22.10:22 - TCP OPEN
[+] 192.168.22.10:        - 192.168.22.10:21 - TCP OPEN
[+] 192.168.22.10:        - 192.168.22.10:53 - TCP OPEN
[+] 192.168.22.10:        - 192.168.22.10:80 - TCP OPEN
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed


Show services status

msf > services
Services
========

host            port  proto  name   state   info
----            ----  -----  ----   -----   ----
                SKIPPED
192.168.22.10   21    tcp            open
192.168.22.10   22    tcp            open
192.168.22.10   23    tcp            open
192.168.22.10   25    tcp            open
192.168.22.10   53    tcp            open
192.168.22.10   80    tcp            open
                SKIPPED
```

## Password sniffing

Once on a local network, it can be useful to listen to the traffic and to try sniffing interesting data such as passwords, emails, etc. Tools like *dsniff*, *filesnarf* or *urlsnarf* accomplish this job. However, in Metasploit, a module exists to perfom password sniffing on a network: *psnuffle*.

It can sniff live traffic or load one from a pcap file.

### Hands on!

The file capture.gz.pcap contains captured traffic on the 192.168.22.0/24 network. Retrieve interesting information in it.

What is/are the involved protocol(s)? What are the involved machines?

### Hands on! ANSWERS

We load *psnuffle* and set the PCAPFILE options:

```
msf > use auxiliary/sniffer/psnuffle
msf auxiliary(sniffer/psnuffle) > show options

Module options (auxiliary/sniffer/psnuffle):
```

```
   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   FILTER                      no        The filter string for capturing traffic
   INTERFACE                   no        The name of the interface
   PCAPFILE                    no        The name of the PCAP capture file to
process
   PROTOCOLS  all              yes       A comma-delimited list of protocols to
sniff or "all".
   SNAPLEN    65535            yes       The number of bytes to capture
   TIMEOUT    500              yes       The number of seconds to wait for new
data


Auxiliary action:

   Name     Description
   ----     -----------
   Sniffer


msf auxiliary(sniffer/psnuffle) > set PCAPFILE capture.gz.pcap
PCAPFILE => capture.gz.pcap
```

Running the module:

```
msf auxiliary(sniffer/psnuffle) > run
[*] Auxiliary module running as background job 0.
msf auxiliary(sniffer/psnuffle) >
[*] Loaded protocol FTP from /usr/share/metasploit-
framework/data/exploits/psnuffle/ftp.rb...
[*] Loaded protocol IMAP from /usr/share/metasploit-
framework/data/exploits/psnuffle/imap.rb...
[*] Loaded protocol POP3 from /usr/share/metasploit-
framework/data/exploits/psnuffle/pop3.rb...
[*] Loaded protocol SMB from /usr/share/metasploit-
framework/data/exploits/psnuffle/smb.rb...
[*] Loaded protocol URL from /usr/share/metasploit-
framework/data/exploits/psnuffle/url.rb...
[*] Sniffing traffic.....
[*] Successful FTP Login: 192.168.1.244:54490-192.168.22.10:21 >>
msfadmin / msfadmin
[*] Finished sniffing
```

The "*creds*" command displays the gathered credentials during a pentest.

```
msf > creds
Credentials
===========

host           origin         service       public    private   realm  private_type
----           ------         -------       ------    -------   -----  ------------
192.168.22.10  192.168.22.10  21/tcp (ftp)  msfadmin  msfadmin         Password
```

## SNMP Sweeping / Enum

SNMP (Simple Network Management Protocol) is a wonderful source of information about a specific system. Its aim is to help collecting and organizing information about managed devices on IP networks.

Note: according to the documentation and depending on the kali version, SNMP service only listens to localhost by default and needs some configuration. Open "/etc/default/snmpd" with your favorite text editor and change it as follow:

```
root@kali:~/ vi /etc/default/snmpd
```

Change the line:

```
SNMPDOPTS='-Lsd -Lf /dev/null -u snmp -I -smux -p /var/run/snmpd.pid
127.0.0.1'
```

With:

```
SNMPDOPTS='-Lsd -Lf /dev/null -u snmp -I -smux -p /var/run/snmpd.pid
0.0.0.0'
```

Finally, restart the service:

```
root@kali:~/ service snmpd restart
```

### Hands on!

Use the *snmp-enum* module to obtain information about 192.168.22.60.

### Hands on! ANSWERS

```
msf auxiliary(scanner/snmp/snmp_enum) > show options

Module options (auxiliary/scanner/snmp/snmp_enum):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   COMMUNITY  public           yes       SNMP Community String
   RETRIES    1                yes       SNMP Retries
   RHOSTS                      yes       The target address range or
CIDR identifier
   RPORT      161              yes       The target port (UDP)
   THREADS    1                yes       The number of concurrent
threads
   TIMEOUT    1                yes       SNMP Timeout
   VERSION    1                yes       SNMP Version <1/2c>

msf auxiliary(scanner/snmp/snmp_enum) > set RHOSTS 192.168.22.60
RHOSTS => 192.168.22.60
msf auxiliary(scanner/snmp/snmp_enum) > run

[+] 192.168.22.60, Connected.
```

```
[*] System information:

Host IP                         : 192.168.22.60
Hostname                        : bee-box
Description                     : Linux bee-box 2.6.24-16-generic #1 SMP
Thu Apr 10 13:23:42 UTC 2008 i686
Contact                         : Your master bee
Location                        : Every bee needs a home!
Uptime snmp                     : 4 days, 18:00:33.84
Uptime system                   : 4 days, 17:59:01.97
System date                     : 2018-6-19 08:46:11.0

[*] Network information:

IP forwarding enabled           : no
Default TTL                     : 64
TCP segments received           : 12398
TCP segments sent               : 11109
TCP segments retrans            : 161
Input datagrams                 : 84167
Delivered datagrams             : 84163
Output datagrams                : 13115

[*] Network interfaces:

Interface                       : [ up ] lo
Id                              : 1
Mac Address                     : :::::
Type                            : softwareLoopback
Speed                           : 10 Mbps
MTU                             : 16436
In octets                       : 188354
Out octets                      : 188354

Interface                       : [ up ] eth0
Id                              : 2
Mac Address                     : 00:50:56:b5:1a:ad
Type                            : ethernet-csmacd
Speed                           : 10 Mbps
MTU                             : 1500
In octets                       : 12226061
Out octets                      : 2358838

               SKIPPED
```

As you can see on this output, a lot of information are available from misconfigured SNMP.

# 3 Vulnerability scan

A vulnerability scan run against computers, networks or applications to detect weakness and known vulnerabilities. Modern scanners allow both unauthenticated and authenticated scans. They can detect many vulnerabilities and are constantly updated (even if some professional scanners are some months ahead the community editions).

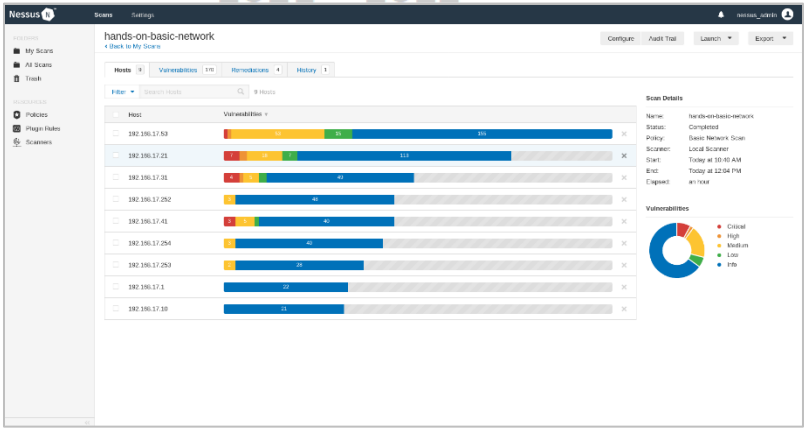
## Nessus

Nessus is a proprietary vulnerability scanner developed by Tenable. Both commercial and community edition exist, allowing many people to use it to identify vulnerabilities, misconfiguration and prevent attackers to penetrate their network.

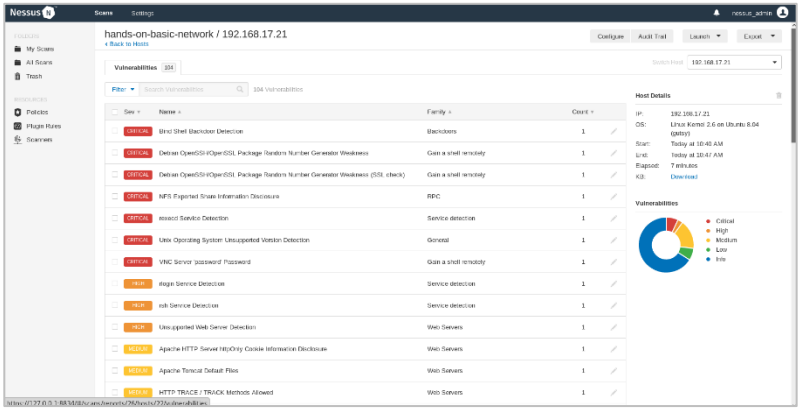
| Hands on! | DEMO |
|---|---|

The goal of this demo is to scan the 192.168.22.0/24 network and to demonstrate how powerful Nessus is regarding vulnerability analysis.

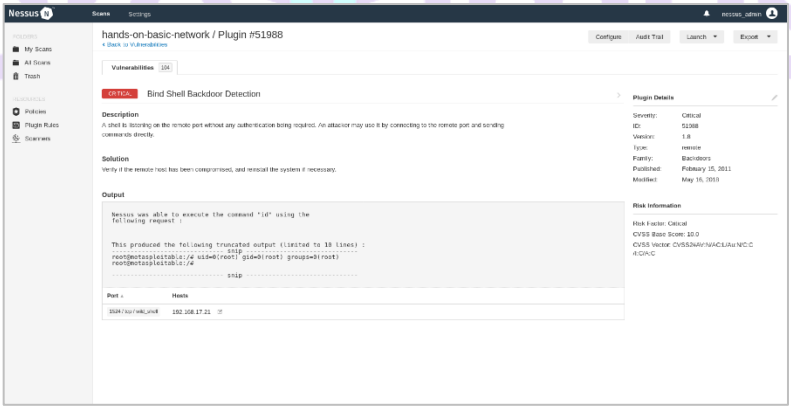
A basic scan on the network returns vulnerabilities found on each host:



A detailed view is available for each host and gives us details about which vulnerabilities have been found:




Detailed information on vulnerabilities are also given:




At the end of this module, we will work with Nessus export functionality to use the scan results with the Metasploit Framework.

## Is your system vulnerable to CVE-2014-0160?

As for the OS detection, scripts exist to detect a specific vulnerability. For instance, the OpenSSL Heartbleed vulnerability can be detected using such a script with *nmap*.

Heartbleed is a serious vulnerability in OpenSSL cryptographic software library. It allows memory reading to anyone on the internet and thus compromise the secret key used to encrypt the traffic. Names and passwords can be eavesdrop and services can be impersonated.

Useful links:

- nmap.org/nsedoc/scripts/ssl-heartbleed.html

## Hands on!

Find the machine that has the OpenSSL Heartbleed vulnerability (CVE-2014-0160) on the lab network (192.168.22.0/24).

Additional information: port is **8443** and not 443.

## Hands on! ANSWERS

```
root@kali:~# nmap -p 8443 --script ssl-heartbleed.nse 192.168.22.0/24

Nmap scan report for 192.168.22.60
Host is up (0.00089s latency).

PORT     STATE SERVICE
8443/tcp open  https-alt
| ssl-heartbleed:
|   VULNERABLE:
|   The Heartbleed Bug is a serious vulnerability in the popular
OpenSSL cryptographic software library. It allows for stealing
information intended to be protected by SSL/TLS encryption.
|     State: VULNERABLE
|     Risk factor: High
|       OpenSSL versions 1.0.1 and 1.0.2-beta releases (including
1.0.1f and 1.0.2-beta1) of OpenSSL are affected by the Heartbleed bug.
The bug allows for reading memory of systems protected by the
vulnerable OpenSSL versions and could allow for disclosure of
otherwise encrypted confidential information as well as the encryption
keys themselves.
|
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0160
|       http://cvedetails.com/cve/2014-0160/
|_      http://www.openssl.org/news/secadv_20140407.txt Frameworks
```

The vulnerable machine is the one with the IP address: 192.168.22.60.

# Metasploit scanning features

Metasploit has module to search for known weakness on hosts. There are many vulnerability scans that can be performed and only three methods will be explained here: scanning for VNC weak passwords, looking for web vulnerabilities with *wmap* and working with Nessus.

## VNC Login

An attacker can check for a given password for VNC using VNC Login scanner.

**Hands on!** DEMO

The goal of this demo is to illustrate how Metasploit detects vulnerabilities and stores them for future exploitation.

```
msf > use auxiliary/scanner/vnc/vnc_login
msf auxiliary(scanner/vnc/vnc_login) > set RHOSTS 192.168.22.10
RHOSTS => 192.168.22.10
msf auxiliary(scanner/vnc/vnc_login) > run

[*] 192.168.22.10:5900    - 192.168.22.10:5900 - Starting VNC login
sweep
[+] 192.168.22.10:5900    - 192.168.22.10:5900 - Login Successful:
:password
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

The credentials are stored in the database:

```
msf auxiliary(scanner/vnc/vnc_login) > creds

Credentials
===========

host            origin          service          private   private_type
----            ------          -------          -------   ------------
192.168.22.10   192.168.22.10   5900/tcp (vnc)   password  Password
```

**Hands on!**

Find the machine that has OpenSSL Heartbleed vulnerability (CVE-2014-0160) on the lab network (192.168.22.0/24) **without using nmap**.

Additional information: port is **8443** and not 443.

Once the right module is selected (*openssl_heartbleed*) and the RHOSTS, RPORT and THREADS parameters set, the command is run:

```
msf auxiliary(scanner/ssl/openssl_heartbleed) > set RPORT 8443
RPORT => 8443
msf auxiliary(scanner/ssl/openssl_heartbleed) > set THREADS 256
THREADS => 256
msf auxiliary(scanner/ssl/openssl heartbleed) > set RHOSTS
192.168.22.0/24
RHOSTS => 192.168.22.0/24
msf auxiliary(scanner/ssl/openssl_heartbleed) > run


[+] 192.168.22.60:8443    - Heartbeat response with leak
[*] Scanned  58 of 256 hosts (22% complete)
[*] Scanned  68 of 256 hosts (26% complete)
[*] Scanned 118 of 256 hosts (46% complete)
[*] Scanned 146 of 256 hosts (57% complete)
[*] Scanned 154 of 256 hosts (60% complete)
[*] Scanned 192 of 256 hosts (75% complete)
[*] Scanned 231 of 256 hosts (90% complete)
[*] Scanned 235 of 256 hosts (91% complete)
[*] Scanned 238 of 256 hosts (92% complete)
[*] Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
```

## WMAP Web Scanner



There is a web application vulnerability scanner within Metasploit that allows us to conduct web application scanning. This tool is based on SQLMap, which performs automatic SQL injections.

Hands on! DEMO

Using *wmap*, find "vulnerabilities" on mutillidae (192.168.22.10 with the following url: /mutillidae/index.php).

```
msf > load wmap

.-.-.-..-.-.-..---..---.
| | | || | | || | || |-'
`-----'`-'-'-'`-^-'`-'
[WMAP 1.5.1] ===  et [  ] metasploit.com 2012
[*] Successfully loaded plugin: wmap
```

First step is to add a site to the *wmap* database:

```
msf > wmap_sites -a http://192.168.22.10
[*] Site created.
msf > wmap_sites -l

[*] Available sites
===============

     Id  Host           Vhost          Port  Proto  # Pages  # Forms
     --  ----           -----          ----  -----  -------  -------
     0   192.168.22.10  192.168.22.10  80    http   0        0
```

Next step is to set the targeted site:

```
msf > wmap_targets -t http://192.168.22.10/mutillidae/index.php
msf > wmap_targets -l

[*] Defined targets
===============

 Id  Vhost          Host           Port  SSL    Path
 --  -----          ----           ----  ---    ----
 0   192.168.22.10  192.168.22.10  80    false  /mutillidae/index.php
```

Finally, running the scan:

```
msf > wmap_run -e
[*] Using ALL wmap enabled modules.
[-] NO WMAP NODES DEFINED. Executing local modules
[*] Testing target:
[*]     Site: 192.168.22.10 (192.168.22.10)
[*]     Port: 80 SSL: false
============================================================
[*] Testing started. 2018-06-19 14:51:14 +0200
[*] Loading wmap modules...
[*] 39 wmap enabled modules loaded.
[*]
=[ SSL testing ]=
============================================================
[*] Target is not SSL. SSL modules disabled.
[*]
=[ Web Server testing ]=
============================================================
[*] Module auxiliary/scanner/http/http_version

[+] 192.168.22.10:80 Apache/2.2.8 (Ubuntu) DAV/2 ( Powered by
PHP/5.2.4-2ubuntu5.10 )
[*] Module auxiliary/scanner/http/open_proxy
[*] Module auxiliary/admin/http/tomcat_administration
[*] Module auxiliary/admin/http/tomcat_utf8_traversal
[*] Attempting to connect to 192.168.22.10:80
[+] No File(s) found
[*] Module auxiliary/scanner/http/drupal_views_user_enum
```

```
[-] 192.168.22.10 does not appear to be vulnerable, will not continue
[*] Module auxiliary/scanner/http/frontpage login
[*] 192.168.22.10:80      - http://192.168.22.10/ may not support
FrontPage Server Extensions
[*] Module auxiliary/scanner/http/host_header_injection
[*] Module auxiliary/scanner/http/options
[*] Module auxiliary/scanner/http/robots_txt
[*] Module auxiliary/scanner/http/scraper
[+] [192.168.22.10] / [Metasploitable2 - Linux]
[*] Module auxiliary/scanner/http/svn_scanner
[*] Using code '404' as not found.
[*] Module auxiliary/scanner/http/trace
[+] 192.168.22.10:80 is vulnerable to Cross-Site Tracing

               SKIPPED

msf > wmap_vulns -l
[*] + [192.168.22.10] (192.168.22.10): scraper /
[*]     scraper Scraper
[*]     GET Metasploitable2 - Linux
[*] + [192.168.22.10] (192.168.22.10): directory /dav/
[*]     directory Directory found.
[*]     GET Res code: 200
[*] + [192.168.22.10] (192.168.22.10): directory /cgi-bin/
[*]     directory Directoy found.
[*]     GET Res code: 403
[*] + [192.168.22.10] (192.168.22.10): directory /doc/
[*]     directory Directoy found.
[*]     GET Res code: 200

               SKIPPED
```

The pentester can now investigate things in details with the help of gathered information.



## Working with Nessus

Metasploit and Nessus works great together: you can import a Nessus scan in Metasploit using the "*db_import*" command and work on it.

**Hands on!**

Use the Nessus scan to visualize hosts, services and vulnerabilities on the 192.168.22.0/24 network.

**Hands on!** ANSWERS

Check that hosts database is empty:

```
msf > hosts
```

```
Hosts
=====

address  mac  name  os_name  os_flavor  os_sp  purpose  info  comments
-------  ---  ----  -------  ---------  -----  -------  ----  --------
```

Import the Nessus Scan and check that the hosts database is not empty anymore:

```
msf > db_import hands-on-basic-network_lstidg.nessus
[*] Importing 'Nessus XML (v2)' data
[*] Importing host 192.168.22.254
[*] Importing host 192.168.22.253
[*] Importing host 192.168.22.252
[*] Importing host 192.168.22.53
[*] Importing host 192.168.22.50
[*] Importing host 192.168.22.40
[*] Importing host 192.168.22.10
[*] Importing host 192.168.22.20
[*] Importing host 192.168.22.2
[*] Successfully imported hands-on-basic-network_lstidg.nessus
```

Visualize hosts database:

```
msf > hosts

Hosts
=====

address         mac                name            os name                        os sp               purpose
-------         ---                ----            -------                        -----               -------
192.168.22.2                       192.168.22.2    Linux                          4.4                 server
192.168.22.20                      192.168.22.20   Linux                          4.4                 server
192.168.22.10                      192.168.22.10   Linux                          2.6                 server
192.168.22.40   00:50:56:b5:47:fc  192.168.22.40   Windows XP                     SP2                 client
192.168.22.50   00:50:56:b5:02:00  192.168.22.50   Windows 2003                   SP2                 server
192.168.22.60   00:50:56:b5:1a:ad  192.168.22.60   Linux                          2.6.24-16-generic   server
192.168.17.252                     192.168.17.252  FreeBSD 11.1-RELEASE-p7 (amd64)                    device
192.168.17.253                     192.168.17.253  pfSense                                            device
192.168.17.254                     192.168.17.254  FreeBSD 11.1-RELEASE-p7 (amd64)                    device
```

Last but not least, vulnerabilities have also been imported:

```
msf > vulns

                    SKIPPED

[*] Time: 2018-06-19 13:57:27 UTC Vuln: host=192.168.22.10 name=Nessus Scan Information
refs=NSS-19506
[*] Time: 2018-06-19 13:57:27 UTC Vuln: host=192.168.22.10 name=Patch Report refs=NSS-66334
[*] Time: 2018-06-19 13:57:27 UTC Vuln: host=192.168.22.10 name=Unknown Service Detection:
Banner Retrieval refs=NSS-11154
[*] Time: 2018-06-19 13:57:27 UTC Vuln: host=192.168.22.10 name=Backported Security Patch
Detection (SSH) refs=NSS-39520
[*] Time: 2018-06-19 13:57:27 UTC Vuln: host=192.168.22.10 name=Backported Security Patch
Detection (WWW) refs=NSS-39521
[*] Time: 2018-06-19 13:57:27 UTC Vuln: host=192.168.22.10 name=SSL/TLS Diffie-Hellman
Modulus <= 1024 Bits (Logjam) refs=CVE-2015-4000,BID-74733,OSVDB-122331,NSS-83875,BID-
74733,OSVDB-122331,NSS-83738
[*] Time: 2018-06-19 13:57:27 UTC Vuln: host=192.168.22.10 name=SSL Version 2 and 3 Protocol
Detection refs=NSS-20007
```

```
[*] Time: 2018-06-19 13:57:27 UTC Vuln: host=192.168.22.10 name=SSLv3 Padding Oracle On
Downgraded Legacy Encryption Vulnerability (POODLE) refs=CVE-2014-3566,BID-70574,OSVDB-
113251,CERT-577193,NSS-78479
[*] Time: 2018-06-19 13:57:27 UTC Vuln: host=192.168.22.10 name=SSL Anonymous Cipher Suites
Supported refs=CVE-2007-1858,BID-28482,OSVDB-34882,NSS-31705

                    SKIPPED
```

# 4 How to protect yourself

## Detecting scans

Detecting and blocking scans is the first step to protect against attacks. However, port scan can be legitimate (an administrator that scan its machine for e.g.) but is not legal in most countries.

One should know that attackers perform two types of scan: network scan to detect active hosts and port scan to detect running services and vulnerabilities. Many methods exist to detect those scans, from network monitoring (to detect patterns) to probabilistic models based on expected network behavior.

During a scan, a Firewall can answer with three different ways: open, closed or choose not to answer. Those last one, along with IDS are often configured to detect scans but scanners are able to cover their tracks by changing their scanning rate or by randomizing port scanning order.

*Fail2ban* scans log files and bans IP that conducted too many failed login attempts (for instance too many failed ssh login) and could be useful to block attackers that are brute forcing login. Combined with *iptables*, it is a simple way to ban IP that tries to scan ports that are not open.

TCP Wrappers is a way to control and filter access to ports. Nevertheless, be careful, as they are a potential attack vector if misconfigured.



## System Hardening

OS and application in general are rarely designed with security as focus, leading to security risks if the system is not hardened.

Hardening is the act of configuring the OS securely, with updates and policies that help governing the system in a secure manner. Unnecessary applications and services are removed to reduce exposed perimeter and mitigate risk.

This section cannot be an exhaustive checklist of best hardening practices, but here are some advice to harden a system:

1. **Remove unnecessary programs**: every program could be a potential entrance point for an attacker.
2. **Install the latest version**: even if this does not protect against zero-day attacks, it reduces the risk while being easy to follow.
3. **Have group policies**: users are often an entry point and reducing their rights to the minimum is always a good practice. Moreover, having a

strong password policy is also advised (see Module 3 for more information about password cracking).

4. Allow **remote access** only through **Virtual Private Network**.

Tools exist to automate security auditing, compliance testing and vulnerability detection. *Lynis* is one of them. This open-source software, developed by Cisofy assists the user in:

- Configuration and asset management
- Software patch management
- System hardening
- Penetration testing (privilege escalation)
- Intrusion detection

Source: *Lynis* github repository.

## Banner modification

Some countermeasures exist to prevent hackers from learning sensitive information such as operating system or applications version from banners.

Administrators can display false banners to cloud the issue or even disable them. Turning off unnecessary services is a way of limiting information exposure.

On Apache 2.X for instance, the *mod_headers* module enable changing banners information.

Disabling the server signature can also reduce risks.

Finally, as file extensions can sometime provide information about the technology used, hiding them is a good practice (*mod_negociation* on Apache).

## Active defense

Active defense is defined by the SANS Institute as “any measures originated by the defender against the attacker”. These defenses can be split into categories, from counterattack to active deception. Some legal considerations must be checked before implementing any of these measures.

A honeypot is a trap, which aims to detect unauthorized use of an information system. A honeypot should detect and learn from attacks and this information might be used to improve the security of a system.

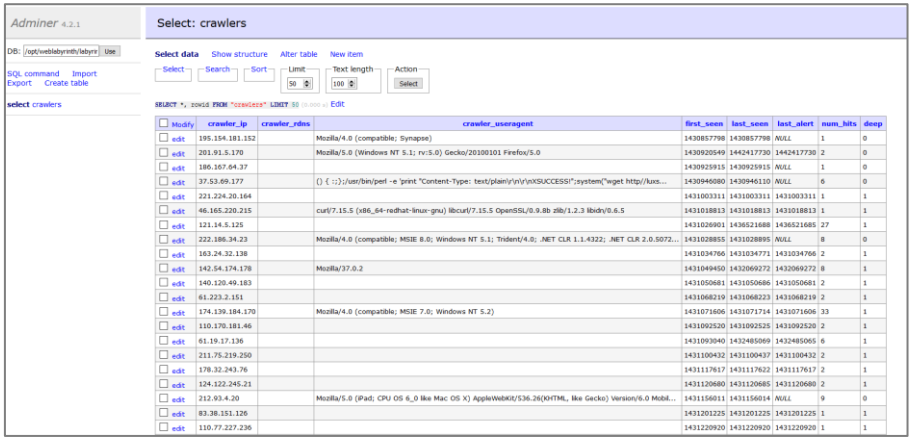
The honeypot should be a “copy” of the actual system in production (same technologies for example), but with fake information. It must not be connected to any production systems neither being registered to productions systems. This ensure that every device that connects to the honeypot is either misconfigured or a potential attacker. Thus, every packet is considered as suspicious on a honeypot.

This way, attacks are easily detected and it is a real asset to detect zero-day vulnerability.

A simpler “honeypot” that can be implemented on a website consists in having fake webpage with hidden form. This form should only be submitted by scanner or by a hacker. Logging IP address, which submitted the form, and banning them can reduce attack risk.

See Active Defense Harbinger Distribution for a Linux distribution with multiple installed and preconfigured active defense systems.





Here is an example of web labyrinth logs. Web crawlers IP's have been saved in database.

## Port spoofing

To protect a network against ARP spoofing, static ARP tables can be used but not all systems respect this static mapping. A switch hardening can also be a good way to protect against ARP attacks. Some switch have a port security feature that assign only one MAC address to each physical port. This avoid attacker to change their MAC address on the fly.

More information about how to protect against ARP attacks on SANS website:

- https://www.sans.org/reading-room/whitepapers/threats/address-resolution-protocol-spoofing-man-in-the-middle-attacks-474

# 5 Conclusion

This concludes the second phase of an OSSTMM test. The information gathering and the enumeration are two essential steps during a pentest and this is why two modules are devoted to those last ones. The more exhaustive is the enumeration the better is the pentest.

Moreover, plenty tools exist to assist an attacker gathering information about his target and most of them are open-source and trivial to use: that is also, why script kiddies could be a real danger for a company.

Administrators and developers should also be aware that those tools exist and that they can use them to audit their system for hardening purpose for instance. Many defense systems can be implemented to prevent attackers from gaining access to the system. Some of them require having a honeypot but the vast majority are only best practices, such the ones that prevent most of information gathering.

# Contact information


.....

cybersecurity@telindus.lu

Cybersecurity Department

Telindus Luxembourg

Twitter: @S_Team_Approved

.....

Proximus House

Z.A. Bourmicht - 18, rue du Puits Romain

L-8070 Bertrange

T +352 27 777 00

.....

**Damien GITTER**

Technology Leader Ethical Hacking

Cybersecurity Department

GIAC Certified (GSEC, GCIA, GCIH, GPEN, GWAPT, GMOB, GXPN,GMON)

Certified OSSTMM (OPST & OPSA)

T +352 23 28 20 7784

M +352 691 777 784

damien.gitter@telindus.lu