rootsh3ll Labs

Network Reconnaissance

Information gathering is the first and most important phase of penetration testing. Also known as the reconnaissance phase. Penetration testing without information on the target host/network is like attacking in the blind. Gaining maximum information about the target host should be your purpose before diving into the exploitation phase of a pentest.

Reconnaissance is performed in 2 ways:

1. Active Reconnaissance
2. Passive Reconnaissance

Active recon is performed by directly probing the client and analysing the response, whereas Passive recon is rather simply sit and wait for the packets to arrive to you, either via MITM or simple sniffing promiscuously. In this lab, you'll learn both, Active and Passive Reconnaissance

Objective:

1. Identify servers running on non-TCP ports
2. Detect servers behind firewall
3. Identify wireless router manufacturer
4. Gather probing client's information

TABLE OF CONTENTS

1. Identify the IP address on your wired LAN running a server on port 347

Running *ifconfig* show a list of connected interfaces to your lab. We have 2 main interface. One is wired, the ethernet, interface connected to a network with host range of /24, as the subnet mask suggests. Another is a wireless interface that you can use to perform wireless scanning activities.

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.1.3.37  netmask 255.255.255.0  broadcast 10.1.3.255
        ether 02:42:0a:01:03:25  txqueuelen 0  (Ethernet)
        RX packets 3172  bytes 215263 (210.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 7483  bytes 1573045 (1.5 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=803<UP,BROADCAST,NOTRAILERS,PROMISC,ALLMULTI>  mtu 1500
        ether 02:00:00:00:00:00  txqueuelen 1000  (Ethernet)
        RX packets 88471  bytes 10623809 (10.1 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Since the flag asks us to scan the wired lab for port 347, we use nmap on our subnet /24 targeted for port 347 and show results only for --open ports

```
nmap -T5 -p 347 10.1.3.1/24 --open
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-08 19:04 UTC
Nmap done: 256 IP addresses (5 hosts up) scanned in 14.71 seconds
```

and, that doesn't show us any response. why? Because by default nmap performs a TCP scan on all ports. But, inside a LAN a serve might be running over UDP ports as well. Or other possibility could be that the server is behind a firewall.

To run a UDP scan with nmap we need to pass the **-sU** flag along our previous command. Which results in the following server. Running on port 347

```
nmap -T5 -sU -p 347 10.1.3.1/24 --open
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-08 19:04 UTC
Nmap scan report for netcat_udp.lab_9 (10.1.3.145)
Host is up (0.000032s latency).

PORT    STATE         SERVICE
347/udp open|filtered fatserv
MAC Address: 02:42:0A:01:03:91 (Unknown)
```

2. Bypass the network firewall and identify the server running on port 8090 of your wired LAN.

On a simple TCP SYN scan for open port number 8090 on our subnet /24, nmap show no interesting results.

```
nmap -T5 10.1.3.1/24 -v -p 8090 --open
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-08 19:06 UTC
Initiating ARP Ping Scan at 19:06
Scanning 255 hosts [1 port/host]
Completed ARP Ping Scan at 19:06, 1.50s elapsed (255 total hosts)
Initiating Parallel DNS resolution of 255 hosts. at 19:06
Completed Parallel DNS resolution of 255 hosts. at 19:06, 13.00s elapsed
Initiating SYN Stealth Scan at 19:06
Scanning 4 hosts [1 port/host]
Completed SYN Stealth Scan at 19:06, 0.04s elapsed (4 total ports)
Initiating SYN Stealth Scan at 19:06
Scanning rootsh3llLabs (10.1.3.37) [1 port]
Completed SYN Stealth Scan at 19:06, 0.03s elapsed (1 total ports)
Read data files from: /usr/bin/../share/nmap
Nmap done: 256 IP addresses (5 hosts up) scanned in 14.71 seconds
           Raw packets sent: 512 (14.416KB) | Rcvd: 11 (384B)
```

The above result shows that we are possibly behind a firewall that is blocking our TCP SYN stealth scan packets to reach the target server.

Nmap allows us to run a variety of scans that helps us bypass the firewall as usually firewalls just block SYN packets which are used to start a TCP handshake. But nmap can perform null scan, a FIN or XMAS scan.

We'll start off with a FIN scan. A FIN scan sends the packet only set with a FIN flag, so it is not required to complete the TCP handshaking, hence we get a response as **open|filtered** if the server is indeed active.

```
nmap -T5 -sF 10.1.3.1/24 -v  -p 8090 --open
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-08 19:05 UTC
Initiating ARP Ping Scan at 19:05
Scanning 255 hosts [1 port/host]
Completed ARP Ping Scan at 19:05, 1.50s elapsed (255 total hosts)
Initiating Parallel DNS resolution of 255 hosts. at 19:05
Completed Parallel DNS resolution of 255 hosts. at 19:06, 13.00s elapsed
Initiating FIN Scan at 19:06
Scanning 4 hosts [1 port/host]
Completed FIN Scan at 19:06, 0.13s elapsed (4 total ports)
Nmap scan report for server.lab_9 (10.1.3.13)
Host is up (0.000015s latency).

PORT     STATE         SERVICE
8090/tcp open|filtered opsmessaging
MAC Address: 02:42:0A:01:03:0D (Unknown)
```

3. Which access point is manufactured by `Cisco-Linksys, LLC`?

Every MAC address is unique and have the first 3 bits reserved for its vendor. This helps us to identify the manufacturer of a network device.

This unique identifier is called Organizationally Unique Identifier (OUI). Wireshark uses it behind the scenes to resolve MAC to corresponding manufactures, airodump-ng can resolve MAC to OUI in realtime with the **`--manufacturer`** flag.

For that first put your wireless card in monitor mode and run airodump-ng to sniff the air.

```
ifconfig wlan0 down             # Brings the wireless interface down
iwconfig wlan0 mode monitor     # Changes mode from managed to monitor for sniffing
ifconfig wlan0 up               # Brings up the interface
```

Now, run airodump-ng with the **`--manufacturer`** flag and you'd notice the MAC addresses are being resolved to their manufacturers in realtime, without internet.

```
airodump-ng wlan0 --manufacturer
```

```
BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID   MANUFACTURER

00:0C:41:82:B2:55  -49  261         227   22   1   54   WPA2 CCMP   PSK  Coherer Cisco-Linksys, LLC

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

65:78:F7:B7:60:A9  11:5A:08:13:2C:86  -49    0 - 2    956        3
(not associated)   00:0F:66:16:94:73  -49    0 - 1     56        4  linksys
FF:FF:FF:FF:FF:3F  40:04:94:70:85:FD  -49    0 - 2      0        1
```

> NOTE: BSSID, OR BASIC SERVICE SET IDENTIFIER IS ALTERNATIVELY KNOWN AS MAC ADDRESS OF THE ACCESS POINT'S MAC ADDRESS.

4. Which client is probing for an SSID: `linksys`?

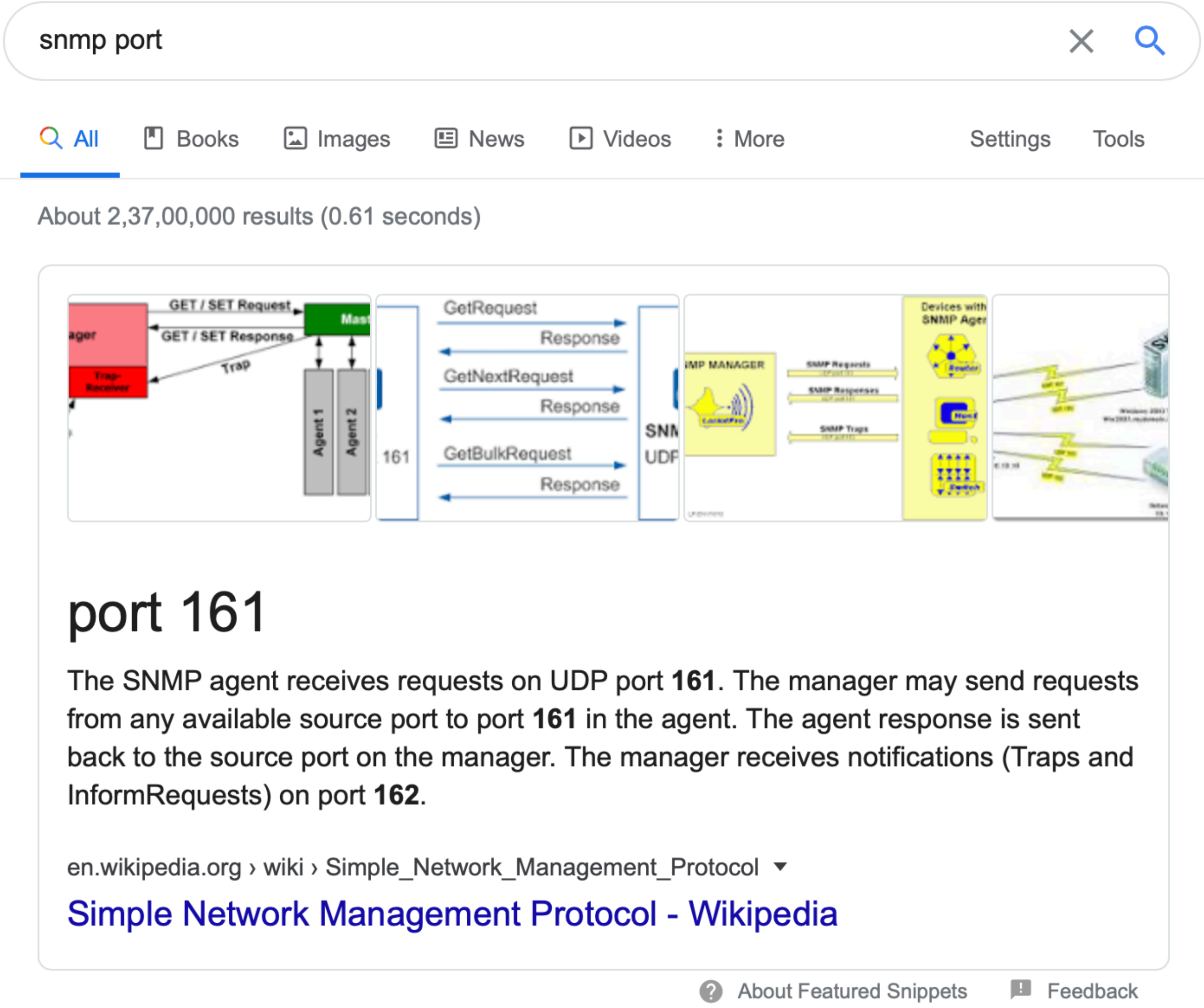
Kill the previous scan window by hitting `ctrl+c` and look in to second horizontal section for the column "Probe". This shows you client devices probing for a certain AP or APs to connect to. Look for the information and identify which. Device is probing for the SSID: **`linksys`** in the response

```
airodump-ng wlan0
```

```
BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

65:78:F7:B7:60:A9   -1        0        0    0  -1  -1                   <length:  0>
32:4F:B0:32:14:9A  -29      111        0    0   1  11   WPA2 CCMP   PSK  Airport WiFi
00:0C:41:82:B2:55  -49      214      201   20   1  54   WPA2 CCMP   PSK  Coherer

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

65:78:F7:B7:60:A9  11:5A:08:13:2C:86  -49    0 - 2    956        3
(not associated)   00:0F:66:16:94:73  -49    0 - 1     56        4  linksys
00:0C:41:82:B2:55  00:0D:93:82:36:3A  -49   48 -54     21      147  Coherer
```

5. Find the SNMP server running on your wired LAN.

A simple google search tells us that the default port of **snmp** is 161. This helps us narrow down our scan range and save us the scan time, which will by default be spent on 1000. Ports per host for our /24 range.



About 2,37,00,000 results (0.61 seconds)

port 161

The SNMP agent receives requests on UDP port **161**. The manager may send requests from any available source port to port **161** in the agent. The agent response is sent back to the source port on the manager. The manager receives notifications (Traps and InformRequests) on port **162**.

en.wikipedia.org › wiki › Simple_Network_Management_Protocol ▾

Simple Network Management Protocol - Wikipedia

Running a simple scan for port 161 doesn't return fruitful results.

```
nmap -T5 --open 10.1.3.1/24 -p 161
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-08 19:10 UTC
Nmap done: 256 IP addresses (5 hosts up) scanned in 14.69 seconds
```

As we learned in the. First and 3rd flag, we must scan the target for UDP ports and/or filtered ports via FIN, NULL or XMAS scan using nmap.

If you perform a FIN, or XMAS scan on this port you'd discover that we still do not see any result in our entire subnet. No traces for an **snmp** server running.

Let's see if the UDP ports tells us the same story…

```
nmap -sU -T5 --open 10.1.3.1/24 -p 161
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-08 19:09 UTC
Nmap scan report for snmp.lab_9 (10.1.3.213)
Host is up (0.000031s latency).

PORT    STATE SERVICE
161/udp open  snmp
MAC Address: 02:42:0A:01:03:D5 (Unknown)
```

And, we successfully identified the server running SNMP service on port 161 without wasting time on 1000 ports per host.