rootsh3ll Labs

Nmap Essentials 101

Objective

Nmap ("Network Mapper") is a network reconnaissance tool that helps you discover live hosts and open ports on a network.

It is one of the most widely used tools by System Administrators and Penetrations Testers alike. Where SysAdmins use it for managing devices on a network, penetration testers use it to gather information about a network to further use it for possible exploitation into the network.

Being an active scanner, Nmap is often detected and blocked by firewalls. But to begin with the network scanner tool, we will be using a real-world scenario without a firewall blocking our scans.

Complete the following challenges to acquire the general Nmap recon skills:

1. Which service is running on port 22 for IP 10.1.3.2
2. Identify server running Samba on port 445
3. What is the version of Redis running on server 10.1.3.9
4. Redis server is running on which port number?
5. Identify the server running on port 8090 and Retrieve Flag using curl from server's /flag location

TABLE OF CONTENTS

0. OBJECTIVE
1. IDENTIFY SERVER SSH VERSION
2. IDENTIFY SAMBA SERVER IP
3. IDENTIFY REDIS SERVICE VERSION
4. IDENTIFY REDIS PORT NUMBER
5. IDENTIFY REMOTE SERVER AND FETCH FLAG

Skip to Flag 1 - Identify Server SSH Version >>

Identify the server's SSH version running on port 22 of your subnet

Nmap always Runs scans against and IP address or a domain name (which resolves to an IP). We are not provided with any IP in this flag. Upon running ifconfig we discover 2 interfaces:

1. `eth0` (`10.1.3.37`) - default ethernet (wired) Lab interface
2. `lo` (`127.0.0.1`): System's loopback interface a.k.a localhost. Used for inter-process communication

Since loopback (lo) doesn't play a role in inter-network communication, we are only left with `eth0`. Which has a class-A IP address (reserved range): `10.1.3.37`

```
ifconfig
```

```
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 134  bytes 11265 (11.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 96  bytes 534585 (522.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Scan Entire Subnet

Class A IP address reserved-range starts from `10.0.0.1` to `10.0.0.255`. Which makes up to a total of `16,777,216` possible hosts for nmap to scan. Read more about reserved IP range here.

That'll take us hours, if not days to actually completely scan every single host for just one target port i.e 22 in this case. So, we look at the subnet mask to get an estimate of our target network size. Note the Netmask value in the ifconfig's output, it says `255.255.255.0`

255 means that specific part of the IP range is static, and it'll not change. .0 means this part of the IP range would be variable. So, if we map the subnet mask onto our assigned IP, we get a range of IP address `10.0.0.1 - 10.0.0.254`

Read more about subnetting on Cisco blog

Note that we only changed the last bit of the IP, based on our subnet mask. And since total hosts in any category can be from 1 to 254 we get a scan range of `10.0.0.1-10.0.0.254`

You can pass this range as it is or use the shortcut `/24` over your own IP address to scan the entire range of `10.0.0.1 - 10.0.0.255`

An IP address is a combination of 4 bytes of data. Let's say `x.y.z.a` is our IP, x, y, z, and a can be a number from 0-255 which consumes 8 bytes of data. So, in the data form we can write:

```
(8 bit) + (8 bit) + (8 bit) + (8 bit) = 32 bit
```

According to our subnet mast the first 3 bytes are static and never changes in the network. Put the mask on the above code and we get `8+8+8 =24 bits` of the IP to be static and only remaining 8 bit is changeable. Nmap understands this and automatically keeps the first 3 bytes of your IP address static, resulting in a range of `10.0.0.1 - 10.0.0.255`

If you kept it `/16`, your scan range would be: `10.0.0.1 - 10.0.255.255`

```
nmap -T5 -v 10.1.3.37/24 -p 22 --open
```

`-T5`	Runs nmap in multi threaded mode.
`-v`	Gives you verbose (realtime) output of scan
`--open`	Filters the output to show only results for open ports

Upon successful scan you'll see an output like this:

```
Nmap scan report for victim_1_openssh.lab_nmap (10.1.3.2)
Host is up (0.000023s latency).

PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 02:42:0A:01:03:02 (Unknown)
```

The default scan helped us identify the server running SSH service, but we do not have visibility into its exact version number. This is done by nmap to prevent wasting too much of time scanning a host or set of hosts. We can tell nmap to enable service scan and service version scan in nmap during scan.

Run Nmap Service Version Scan

Use the service version scanner option carefully. It is not ideal to run Service version scan on a larger subnet, since nmap runs a lot scans against the service to identify its service version. It may either lead you to DDoS the service or get detected by the IDS/IPS.

You do not want either of them. It is advised to always recon your target and then perform host specific scans and attacks to keep the detection minimal. Since we now have the target IP address, tell nmap to perform a service version scan on port 22 against our target host

```
nmap -T5 -v 10.1.3.2 -p 22 -sS -sV --open
```

`-sS`	Enable Service Scan
`-sV`	Enable Service Version Scan

This will reveal the SSH server's approximate software version in use. Enter the information retrieved from **`VERSION`** column into the **`Verify Flag`** section of your lab description page

Identify server running Samba on port 445

Like we did for port 22, just tell nmap to scan the entire subnet (`/24`) for a open port (`--open`) 445

```
nmap -T5 -v 10.1.3.37/24 -p 445 --open
```

```
Nmap scan report for victim_2_samba.lab_nmap (10.1.3.5)
Host is up (0.000013s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds
MAC Address: 02:42:0A:01:03:05 (Unknown)
```

What is the version of Redis running on server 10.1.3.9

Run a service scan against your target host for all open ports.

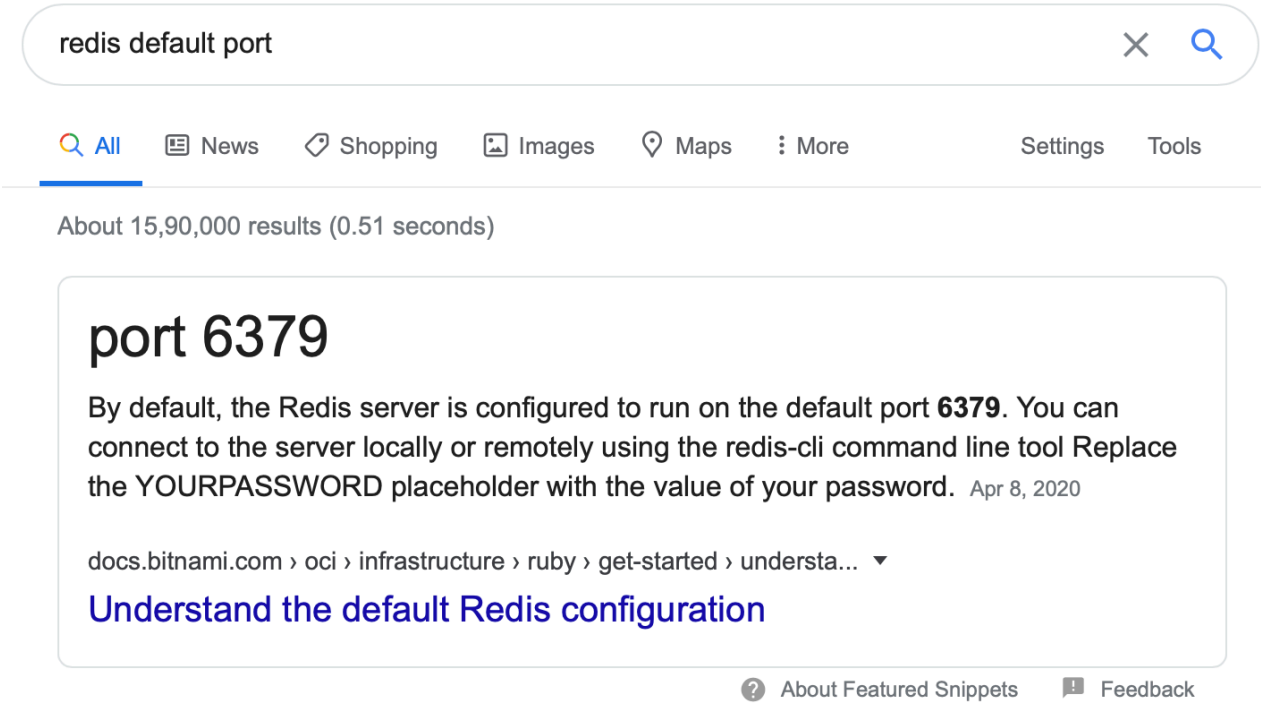
```
nmap -T5 10.1.3.9 -sS –open
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-09 14:46 UTC
Nmap scan report for victim_3_redis.lab_nmap (10.1.3.9)
Host is up (0.000010s latency).
All 1000 scanned ports on victim_3_redis.lab_nmap (10.1.3.9) are closed
MAC Address: 02:42:0A:01:03:09 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
```

You'll notice that nmap doesn't result in any open port or running service. This is because by default nmap runs a scan on top 1000 ports of the total 65,535 ports.

Also, a quick google search on "Redis default port" shows us that Redis runs on port 6379 by default. Which lies outside of nmap's default port scan range.



Since we did not get a desired response in the default service port range, we increase the port scan range from 1000 to 10,000. We can also specify port to be `6379`. But we are running a wider range scan just to be sure if Redis is not running on a pseudo port.

```
nmap -T5 10.1.3.9 -p 1-10000
```

```
PORT     STATE SERVICE
6379/tcp open  redis
MAC Address: 02:42:0A:01:03:09 (Unknown)
```

Now, run a service version scan on this IP to reveal the exact version of Redis running.

Redis server is running on which port number?

6379, default port as revealed by our scan results.

Identify the server running on port 8090 and Retrieve Flag using curl from server's /flag location

Scan the entire subnet for open port (--open) 8090

```
nmap -T5 -p 8090 10.1.3.0/24 -v --open
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-09 14:49 UTC
Initiating ARP Ping Scan at 14:49
Scanning 255 hosts [1 port/host]
Completed ARP Ping Scan at 14:49, 1.40s elapsed (255 total hosts)
Initiating Parallel DNS resolution of 255 hosts. at 14:49
Completed Parallel DNS resolution of 255 hosts. at 14:49, 13.00s elapsed
Initiating SYN Stealth Scan at 14:49
Scanning 5 hosts [1 port/host]
Discovered open port 8090/tcp on 10.1.3.13
Completed SYN Stealth Scan at 14:49, 0.03s elapsed (5 total ports)
Nmap scan report for victim_4_golang.lab_nmap (10.1.3.13)
Host is up (0.000021s latency).

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

We discover a server having an open port 8090. Let's try connecting to it using curl on port 8090 and directory /flag.

Connect to remote server (:8090)

```
curl 10.1.3.13:8090/flag
```

On successful execution you'll receive the required flag in the Terminal console.
Copy the flag and enter into the appropriate `Verify Flag` section