Task 1: Scanning Local IP and Ports

BY: KUSH THAKER. (kthaker442@gmail.com)

IP discovery and Scanning IP range

For this task, I first needed to determine the local network range. My personal Booted kali Linux already having a static IP Address assigned so I already knew the IP address (192.168.31.50) and its range (192.168.31.0/24 as subnet of 255.255.255.0). However, for the sake of the documentation and completeness, I still ran the command to confirm the IP address.

1) ifconfig: (images attached in the images folder)

After knowing the IP and subnet. We know the IP range is 192.168.31.0/24. Now using NMAP to scan the range and to discover the open ports and running services over that port and finding vulnerabilities.

2) nmap -sS -sV -T4 -O -F 192.168.31.0/24

(its image is attached “nmaptask1” and “nmapscanend” in the images folder)

(the nmap scan is in the main directory of repository as “task1.html”)

The Running services are mentioned as below:

PTO

IP Address	Device Name	Port Numbers	Services
192.168.31.1 (Gateway)	Jio Air Fiber Router	53, 80, 443, 8080,8443	DNS, HTTP, HTTPS, HTTP proxy, HTTPS Alternate.
192.168.31.30	UNKNOWN DEVICE	none	none
192.168.31.106, 192.168.31.143	Android Device (RAT Infected)	49152,8888	tcpwrapped
192.168.31.139	Set-Top-Box JIO ROUTER	none	none
192.168.31.170	UNKNOWN DEVICE	none	none
192.168.31.50	KALI LINUX HOST	none	none
-	-	-	-

Analysing the detected open Ports and services.

Port 53 – DNS:: Used for Resolving Domain names. (If Misconfigured then Vulnerable with attacks like DNS Spoofing.)

Port 80 – HTTP: Standard web interface for devices but not encrypted. (If Credentials are leaked then Vulnerable.)

Port 433 – HTTPS: Encrypted and secure web interface. Not vulnerable until misconfigured.
Port 8080 – HTTP Proxy: Often used to Locally host a WEB server. (Same Risks like port 80)

Port 8443 – HTTPS Alternate: An alternate port to be used when the 433 is unavailable/not in use. (Same Risks like port 433)

Port 4915,8888 – tcpwrapped: A reverse TCP port configured in Android device by a RAT. (Highly risky and VULNERABLE).

Wireshark Packet capture and Sniffing. (Optional)

I didn’t go through a direct approach over the router. Instead of that I will take an approach using aircrack-ng suite.

so, I used below commands. (Screenshots are In the images folder)

3) airmon-ng check kill && airmon-ng start wlan0:

    Here “check kill” commands terminate the all wifi card’s processed and start wlan0 will start that particular “wlan0” wifi-card’s monitor mode. Now the wifi card’s name will be “wlan0mon”.

4) airodump-ng wlan0mon:

    This will scan all nearby access points and router.

5) airodump-ng –bssid BE:0A:F3:77:E4:BD -c 1 -w task1

    This will particularly scan the router(gateway) and collect the packets in

    “task1-01.cap” file.

    --bssid ; is used to senter that particular wifi’s mac address.

    -c; is used to enter the channel in which the router is running.
    -w; is used to write all that captured packets in a **.cap** file.

Now click on that **.cap** file and open with Wireshark. You will see the capture packets. But will be encrypted in WPA2 encryption.

CONCLUSION:

I began by confirming my host details: although my Kali Linux machine is assigned a **static IP** (192.168.31.50), I executed the standard interface check (ip a / ifconfig) and saved a screenshot for documentation. From the subnet mask 255.255.255.0 I derived the network range 192.168.31.0/24 and targeted that range for a focused scan.

Using Nmap I discovered seven live hosts. The router at 192.168.31.1 (JIO AIR FIBER) exposed multiple management ports (53, 80, 443, 8080, 8443). Two Android devices under my control — 192.168.31.106 and 192.168.31.143 — presented high/ephemeral ports (49152, 8888) that correspond to implanted RAT channels. Other hosts on the subnet either presented no open TCP ports or had standard closed/filtered states.

I captured live traffic with Wireshark (and used wireless monitor mode tools where appropriate) to observe and analyse the traffic packets.

REGARDS,

KUSH THAKER.