Assignment 1

Analysis of Network Reconnaissance

Sami

Jan 14, 2024

Table of Contents

Executive Summary

The primary objective of this network reconnaissance task was to conduct a security evaluation of network environments. This involved mapping the network to identify active devices and their open ports to find their running services, detecting their operating systems, and assessing potential vulnerabilities.

A variety of advanced tools were employed to achieve these objectives. Zenmap was used for network mapping and port scanning, providing details of active devices, open ports, and services on the network. Maltego was used for gathering open-source intelligence to understand digital footprints and network connections. theHarvester was utilized to collect publicly available information about domains and associated entities. Wafw00f focused on identifying Web Application Firewalls to find web servers' security level. WhatWeb identified various web technologies, such as server types, and frameworks utilized within the network. whois provided information about domain registration and ownership to understand the network's structure of our target domain. Dmitry uses whois and other tools to offer a deeper insight into network-related information such as administrative contacts.

Our passive and active reconnaissance revealed several findings. Zenmap's quick scan inside the BCIT' indicated 29 active devices, including an analysis of the 100 most common ports (SSH, Kerberos-sec, VNC) and the Mac address of each device which helped to highlight potential security risks associated with these services. A comprehensive slow scan identified various operating systems, such as Linux, FreeBSD, and Windows, and services like SSH, RPC, and syslog. It provided insights into network configurations that could be used to either strengthen or weaken the security of those devices. Wafw00f's analysis suggested a potential absence of Web Application Firewalls on the ptnmed company's domain that indicated a vulnerability against web-based attacks. On the other hand, it provided information about the web application firewall that example.com is using, helping to identify points of attack related to that firewall service WhatWeb's results provided a deeper understanding of the technologies and configurations used on our domain such as server types, frameworks, and scripting languages to understand our attacking surface. whois and offered a comprehensive view of the network's external structure and the domain's security measures.

The findings from this task highlight the need for enhanced security measures for ports and services. Common open ports like SSH, Kerberos-sec, and VNC require robust password

policies and configurations to prevent unauthorized access. The detailed results from our comprehensive scan underscore the importance of regular network scanning to identify vulnerabilities. Moreover, the absence of Web Application Firewalls points to the area of web-based attacks and vulnerabilities. Also, the domain registration and configuration information emphasize the need to prevent issues like unauthorized transfers or hijacking.

Introduction

The objectives of this network reconnaissance endeavour were multifaceted. It involves Zenmap for detailed network mapping and port scanning, Maltego for analyzing digital footprints, and other tools like theHarvester, Wafw00f, WhatWeb, whois, and Dmitry for gathering comprehensive information about the domains and their networks. This task aimed to assess the network security strength of the specified network environment such as the Datacomm lab by identifying active devices, evaluating open ports, and detecting operating systems to uncover potential vulnerabilities.

In the Hacking perspective, reconnaissance is a critical initial phase. It gathers information about the target to identify vulnerabilities that could be exploited in future phases to make the hacking process and use of to. Reconnaissance can be either passive or active. Passive reconnaissance, which includes Online research, social media analysis, Domain and Network whois lookups, public document analysis, Open-source Intelligence tools, and monitoring the target's digital footprints, allows attackers to gather information without direct interaction with the target. On the other hand, active reconnaissance involves direct interaction with the system to gather information, which includes network scanning, DNS record enumeration, vulnerability testing, brute force attacks, subdomain analysis, physical reconnaissance, and metadata analysis. Passive and Active Reconnaissance are crucial forms for collecting information as much as possible about our target before launching an attack.

Adherence to ethical guidelines and standards was paramount throughout this reconnaissance task in the assignment. The analysis was conducted with respect for privacy and legality. For instance, in the Zenmap scan, I excluded the restricted devices that were listed and performed without interacting with them. Maltego was used to only analyze digital footprints of myself and ethically focus on publicly available data without infringing with anyone's privacy.

TheHarvester, Wafw00f, WhatWeb, whois, and Dmitry were utilized responsibly to test only the ptnmed.com which the website of our company and I have the rights to test which helped to ensure that the reconnaissance did not extend beyond the approved scope. We should be aware that the assignment was conducted for educational and security enhancement purposes, avoiding any harm or unauthorized access to devices or individuals' data. I strictly followed ethical guidelines and maintained a professional and legal approach, focusing solely on scanning and analyzing the devices or networks that I have the authority to perform the tasks.

Body

- **Zenmap:** Zenmap was used for scanning the 192.168.0.1/24 subnet, employing a quick scan strategy and detailed scanning of the 5 IP addresses 192.168.0.3 192.168.0.11 192.168.0.15 192.168.0.24 192.168.0.27 to identify active devices and open ports, and so forth while excluding the restricted IP addresses.
- **Maltego:** Maltego was employed to analyze my digital footprints using publicly available data. The focus was on my personal data points, integrating additional entities like email addresses, and social media accounts to the map.
- **theHarvester:** theHarvester was used to gather public data from the domain ptnmed.com, utilizing search engines like Bing and Yahoo.
- **Wafw00f:** Wafw00fwas deployed to identify the presence of Web Application Firewalls on the domains ptnmed.com, and example.com.
- **WhatWeb:** WhatWeb was utilized to identify web technologies used on the ptnmed.com website, analyzing elements like HTTP headers and HTML code.
- **whois:** whois was used to obtain domain registration and ownership information for ptnmed.com.
- **Dmitry:** Dmitry was used to gather comprehensive information about the domain ptnmed.com, focusing on IP ranges and administrative contacts.

Zenmap

Network Mapper (Nmap) is a powerful tool that creates a map of a network by scanning IP addresses, identifying open ports, and detecting the services and applications running on a network. Zenmap is the graphical interface that takes the core functionality of Nmap Zenmap offers the same depth of network scanning that Nmap is known for. They send packets to specified IP addresses and listen for responses to find which hosts are active and what services they're running. They can also perform OS detection to figure out what operating system a device is running with the probability of its version.

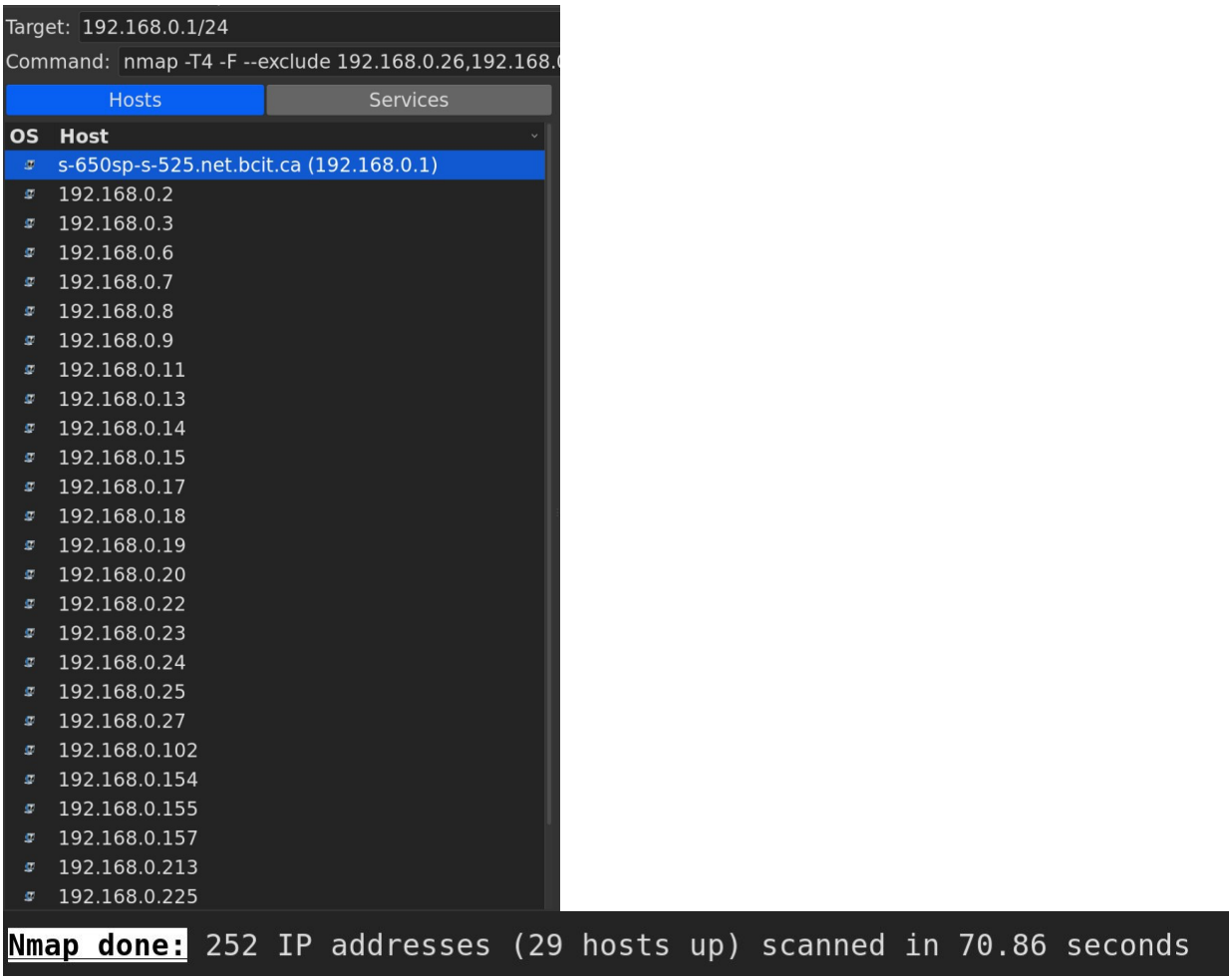
Zenmap‘s interface allows you to input the same commands that you'd use in Nmap. It comes with built-in profiles to adjust the scanning intensity and depth. The first one we tried was the quick scan.

Quick Scan

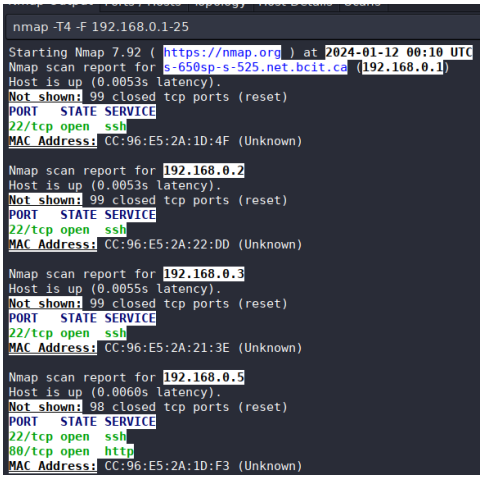
Purpose	Command
Quick Scan	nmap -T4 -F –exclude 192.168.0.26,192.168.0.100,192.168.0.244,192.168.0.238,142.232.122.3 192.168.0.1/24

This command performs a quick scan of the devices on the 192.168.0.1/24 subnet while excluding the restricted IP addresses from the scan.

- **-T4**: This option sets the timing to the 4th option which is the aggressive to accelerate the scan. This is one of the fastest timing options and is used for robust networks that can handle many packets without being overwhelmed.
- **-F**: This option is the fast scan mode, which reduces the scan to the 100 most common ports to quickly scan the open ports without performing a scan on 1000 ports(default) or a full scan of all 65535 ports. it is efficient for scans where time matters.
- **--exclude**: This parameter allows us to specify one or more IP addresses that we want to omit from the scan. In our case, **192.168.0.26,192.168.0.100,192.168.0.244,192.168.0.238,142.232.122.3** are the restricted devices in the lab and we excluded them to not getting scanned.
- **192.168.0.1/24**: This is the network range for the scan. The **/24** indicates a subnet mask of 255.255.255.0 to cover all IP addresses from **192.168.0.1** to **192.168.0.255**.



Scan of all devices in the network excluding the 3 restricted devices. 252 requests have been sent and 29 of the devices were up at that time.



Quick scan for IP addresses **192.168.0.1-5.**

```
Nmap scan report for 192.168.0.6
Host is up (0.0058s latency).
Not shown: 98 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: CC:96:E5:27:81:88 (Unknown)

Nmap scan report for 192.168.0.8
Host is up (0.0049s latency).
Not shown: 99 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: CC:96:E5:2A:20:EF (Unknown)

Nmap scan report for 192.168.0.9
Host is up (0.0055s latency).
Not shown: 98 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: CC:96:E5:2A:22:87 (Unknown)

Nmap scan report for 192.168.0.11
Host is up (0.0052s latency).
Not shown: 98 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
```

Quick scan for IP addresses **192.168.0.6-11.**

```
80/tcp open  http
MAC Address: CC:96:E5:2A:22:A6 (Unknown)

Nmap scan report for 192.168.0.13
Host is up (0.0050s latency).
Not shown: 98 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: CC:96:E5:2A:1E:39 (Unknown)

Nmap scan report for 192.168.0.14
Host is up (0.0046s latency).
Not shown: 99 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: CC:96:E5:29:CB:C2 (Unknown)

Nmap scan report for 192.168.0.15
Host is up (0.0018s latency).
Not shown: 98 filtered tcp ports (no-response)
PORT    STATE SERVICE
22/tcp  open  ssh
135/tcp open  msrpc
MAC Address: CC:96:E5:2A:23:39 (Unknown)

Nmap scan report for 192.168.0.17
Host is up (0.0048s latency).
Not shown: 98 closed tcp ports (reset)
PORT   STATE SERVICE
```

Quick scan for IP addresses **192.168.0.13-17.**

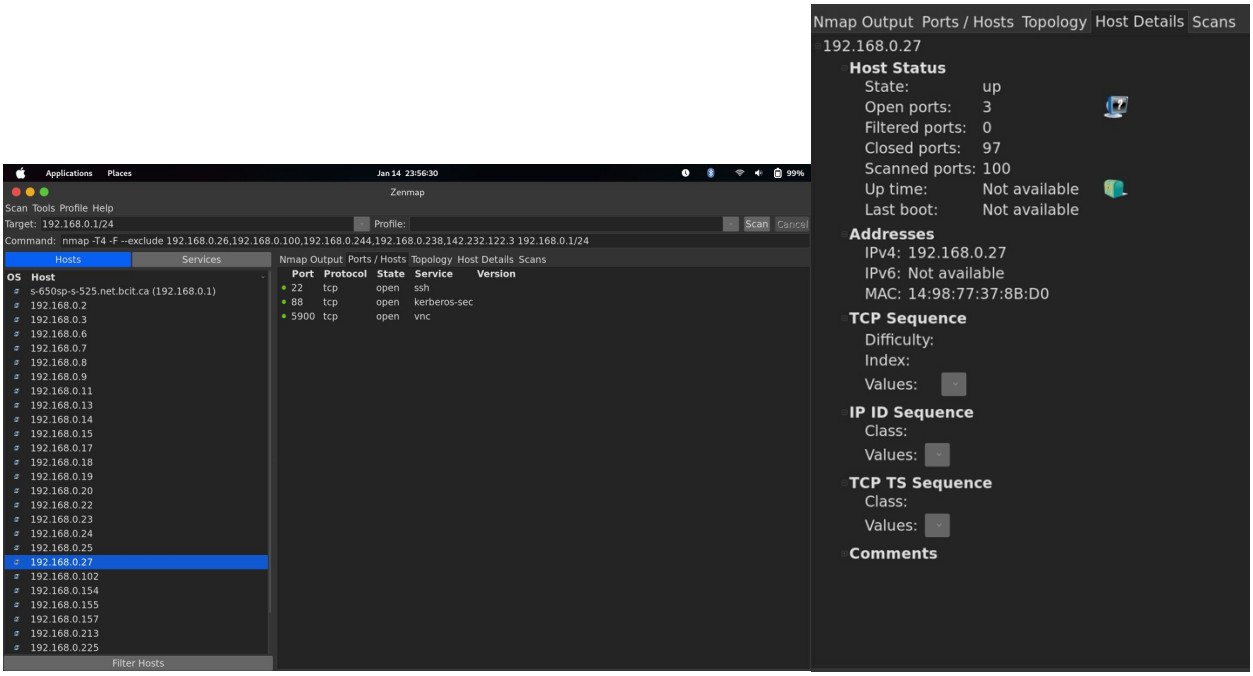
```
Nmap scan report for 192.168.0.18
Host is up (0.0041s latency).
Not shown: 98 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: CC:96:E5:21:D2:94 (Unknown)

Nmap scan report for 192.168.0.19
Host is up (0.0052s latency).
Not shown: 98 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: CC:96:E5:27:83:02 (Unknown)

Nmap scan report for 192.168.0.20
Host is up (0.0041s latency).
Not shown: 98 filtered tcp ports (no-response)
PORT    STATE SERVICE
22/tcp  open  ssh
135/tcp open  msrpc
MAC Address: CC:96:E5:2A:1E:A5 (Unknown)

Nmap scan report for 192.168.0.22
Host is up (0.0054s latency).
Not shown: 98 closed tcp ports (reset)
PORT   STATE SERVICE
```

Quick scan for IP addresses 18-22

```
Nmap scan report for 192.168.0.23
Host is up (0.013s latency).
Not shown: 98 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: CC:96:E5:2A:21:64 (Unknown)

Nmap scan report for 192.168.0.24
Host is up (0.011s latency).
Not shown: 99 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: CC:96:E5:2A:20:84 (Unknown)

Nmap scan report for 192.168.0.25
Host is up (0.012s latency).
Not shown: 98 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: CC:96:E5:2A:1D:60 (Unknown)

Nmap scan report for 192.168.0.27
Host is up (0.0042s latency).
Not shown: 97 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
88/tcp   open  kerberos-sec
5900/tcp open  vnc
MAC Address: 14:98:77:37:8B:D0 (Apple)
```

Quick scan for IP addresses 19-27

Nmap outputs give us a brief explanation of each of the IP addresses in the network such as if the device is up, their open ports and the mac address related to each IP address. we can get the Mac address from the Host details as well.



The quick scan conducted an informative snapshot of the network's active devices such as their IP address and Mac addresses. For instance, focusing on the device at IP address 192.168.0.27, which

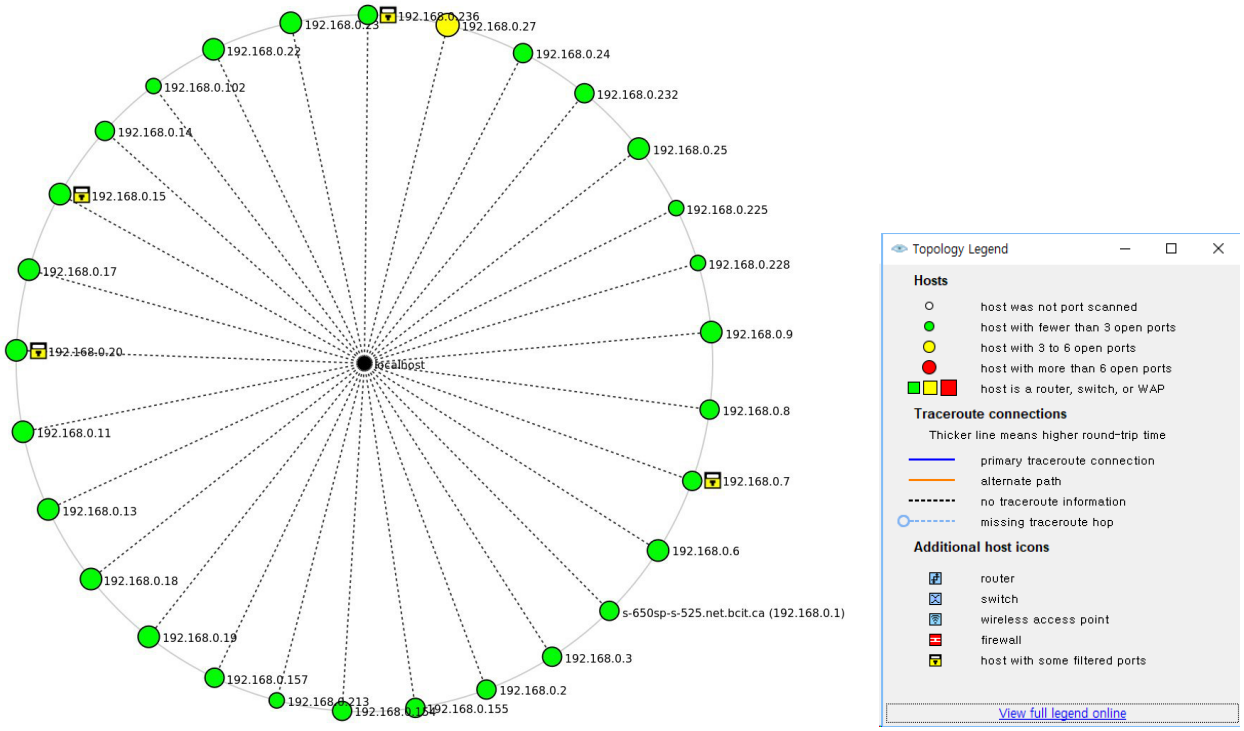
is a Mac mini, Zenmap shows that three out of the 100 most common ports scanned are open as we were doing a quick scan with -F flag set.

Port 22 is used for SSH (Secure Shell), a protocol for secure remote login.

Port 88 is associated with Kerberos-sec. It is a network authentication protocol that provides strong authentication for client/server applications.

Port 5900 is used by VNC (Virtual Network Computing). It uses the Remote Frame Buffer protocol to control another computer remotely.

In this example, open ports pose potential security risks. For example, SSH is vulnerable to brute-force attacks without strong passwords or key-based authentication. Kerberos can be susceptible to attacks if it is not correctly implemented or weaknesses in the network's configuration. VNC also should only be accessible through a secure network or VPN to prevent unauthorized remote control. Attackers can use this information in the reconnaissance phase to find the best attack.



This topology map shows all the devices that our machine (local host) scanned on the network. it shows that all of the devices on the network have less than 3 ports open and the machine with 192.168.0.27 (Mac Mini) has 3-6 ports open. Based on the locks on the topology there are 4 devices with filtered ports.

Comprehensive slow Scan

Purpose	Command
Quick Scan	nmap -sS -sU -T4 -A -v -PE -PP -PS80,443 -PA3389 -PU40125 -PY -g 53 --script "default or (discovery and safe)" 192.168.0.3 192.168.0.11 192.168.0.15 192.168.0.24 192.168.0.27

This command is for a comprehensive slow scan, targeting 5 specific IP addresses within the network. This scan is more detailed compared to the quick scan:

- **-sS**: This option is called half-open scan. It's a stealthy scan because it doesn't fully establish a TCP connection. It sends a SYN packet and waits for a response, identifying open ports based on whether a SYN-ACK is received. 3 Options:
  First, if the ACK is received the port is open.
  Second, if the RST is received the port is closed.
  Third, if there is no response the port is filtered.
- **-sU**: This option enables UDP scanning, since UDP is connectionless, this scan sends UDP packets to different ports and listens for replies. It's useful for discovering open UDP ports and the services running on them it is more challenging to scan because of the connectionless nature and can't determine an open port based on the response only.
  The port is closed if an ICMP port unreachable error (type 3, code 3) is received.
  The port is open and is being used by a service if a service responds to the probe.
  The port is filtered if there is no response.
  UDP scans are slower than TCP scans due to the uncertainty of responses and the retransmissions. A comprehensive -sU scan is essential for a full network audit as important services may run over UDP, and unsecured ports could be vectors for exploitation.
- **-T4**: This option sets the timing to the 4th option which is the aggressive to accelerate the scan. This is one of the fastest timing options and is used for robust networks that can handle many packets without being overwhelmed.
- **-A**: This option is for OS detection, version detection, script scanning, and traceroute. It tries to identify the operating system of the target, and versions of the running services, runs a set of scripts, and traces the path packets take.

- **-v**: Increases verbosity to more detailed output about the scan.
- **-PE:** This flag performs an ICMP echo request, and if an echo reply is received, it indicates the host is up.
- **-PP:** This flag sends an ICMP timestamp request, which is another type of ICMP message, and if a timestamp reply is received, the host is up. This method is used when echo requests are blocked.
- **-PS80,443:** The -PS flag is for TCP SYN discovery. It sends a TCP SYN packet to open a connection to the specified ports (80 and 443, are for web traffic). If a SYN-ACK is received, the host is up. This method is stealthier than an ICMP ping because some systems may not log it as a full connection.
- **-PA3389:** This option uses a TCP ACK packet. It's useful for identifying hosts that are not responding to SYN requests. The ACK packet is sent to port 3389 by default, which is commonly used for Microsoft's Remote Desktop Protocol (RDP). A response indicates the host is up, but no response could mean the packet was filtered or the host is down.
- **-PU40125**: This option is a UDP discovery on port 40125. It sends a UDP packet to the port, and it looks for an ICMP port unreachable error to determine if the host is down. If there is no response, or if there's a specific response from a UDP service, the host is considered up.
- **-PY:** This option is used for SCTP (Stream Control Transmission Protocol) discovery. SCTP is primarily used in telephony networks. If an INIT-ACK chunk is received in response, the host is marked as up.
- **-g 53**: Uses port 53 for the scan, which is used for DNS and can sometimes bypass firewalls that expect DNS traffic on this port.
- **--script "default or (discovery and safe)"**: Tells Nmap to run predefined scripts categorized as "default" or those in the "discovery" and "safe" categories. These scripts detect additional information about the targets.
- **192.168.0.3**, **192.168.0.11**, **192.168.0.15**, **192.168.0.24**, **192.168.0.27**: These are the specific IP addresses targeted by the scan. this scan focuses on these 5 hosts for an in-depth analysis.

Kali Linux Machine

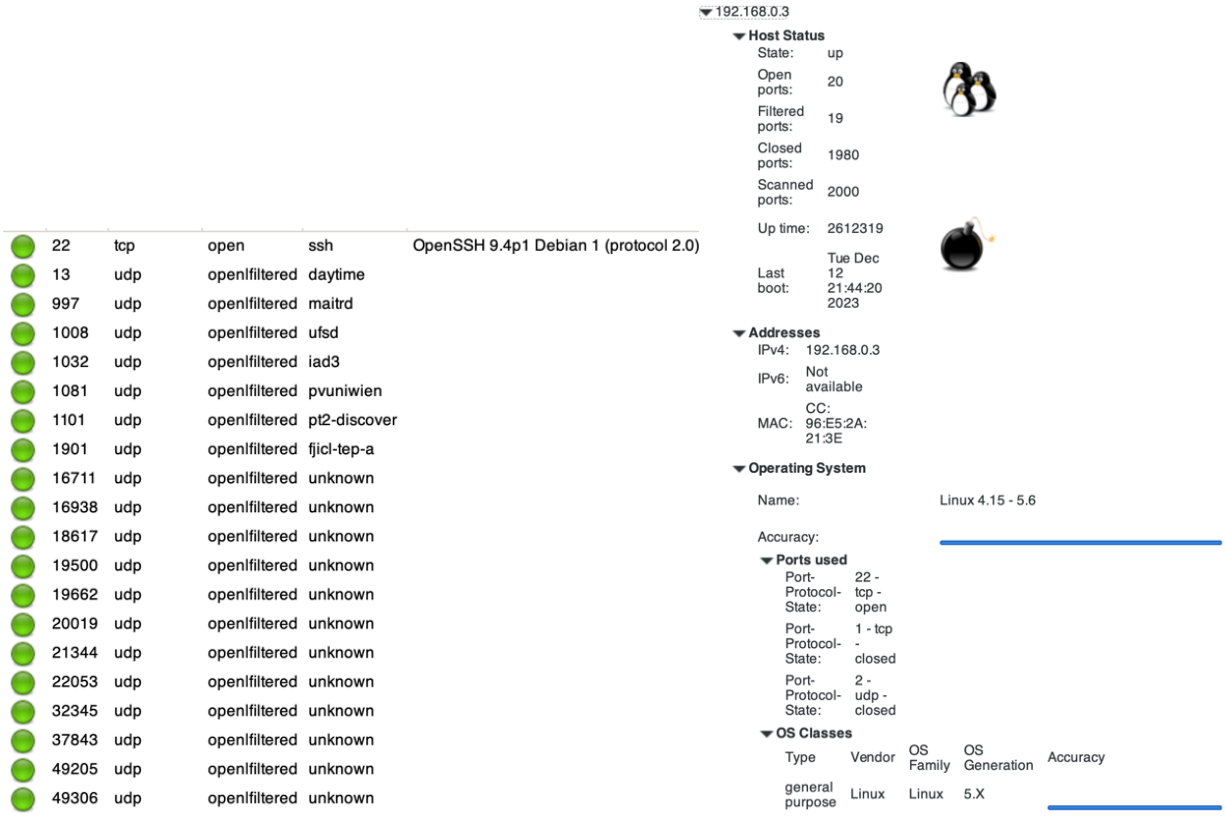
```
Nmap scan report for 192.168.0.3
Host is up (0.0026s latency).
Not shown: 999 closed tcp ports (reset), 981 closed udp ports (port-unreach)
PORT      STATE         SERVICE      VERSION
22/tcp    open          ssh          OpenSSH 9.4p1 Debian 1 (protocol 2.0)
|_banner: SSH-2.0-OpenSSH_9.4p1 Debian-1
| ssh2-enum-algos:
|   kex_algorithms: (10)
|       sntrup761x25519-sha512@openssh.com
|       curve25519-sha256
|       curve25519-sha256@libssh.org
|       ecdh-sha2-nistp256
|       ecdh-sha2-nistp384
|       ecdh-sha2-nistp521
|       diffie-hellman-group-exchange-sha256
|       diffie-hellman-group16-sha512
|       diffie-hellman-group18-sha512
|       diffie-hellman-group14-sha256
|   server_host_key_algorithms: (4)
|       rsa-sha2-512
|       rsa-sha2-256
|       ecdsa-sha2-nistp256
|       ssh-ed25519
|   encryption_algorithms: (6)
|       chacha20-poly1305@openssh.com
|       aes128-ctr
|       aes192-ctr
|       aes256-ctr
|       aes128-gcm@openssh.com
|       aes256-gcm@openssh.com
|   mac_algorithms: (10)
|       umac-64-etm@openssh.com
|       umac-128-etm@openssh.com
|       hmac-sha2-256-etm@openssh.com
|       hmac-sha2-512-etm@openssh.com
|       hmac-sha1-etm@openssh.com
|       umac-64@openssh.com
|       umac-128@openssh.com
|       hmac-sha2-256
|       hmac-sha2-512
|       hmac-sha1
|   compression_algorithms: (2)
|       none
|_      zlib@openssh.com
| ssh-hostkey:
|   256 e5:4c:fe:4e:b5:40:7c:8b:ea:59:dc:95:3d:c6:2c:eb (ECDSA)
|_  256 f9:67:15:7d:cb:b4:ab:60:bf:d9:f9:8a:a7:ae:ef:b4 (ED25519)
13/udp    open|filtered daytime
|_daytime: The script encountered an error: Failed to read
997/udp   open|filtered maitrd
1008/udp  open|filtered ufsd
1032/udp  open|filtered iad3
1081/udp  open|filtered pvuniwien
1101/udp  open|filtered pt2-discover
1901/udp  open|filtered fjicl-tep-a
16711/udp open|filtered unknown
16938/udp open|filtered unknown
```

The scan output gives a view of the network services and configuration for the host with the IP address 192.168.0.3. The host's latency is 0.0026 seconds. Most ports are closed, indicating a secured configuration.

SSH service is running with OpenSSH version 9.4p1 Debian 1. The details of SSH algorithms and cryptographic methods are enumerated. The ECDSA and ED25519 SSH keys indicate that key-based authentication is enabled.

Port 13 of the UDP protocol indicates that the daytime service is detected but not identified due to an error. It's marked as open/filtered, which means the scanner couldn't find out if the port is open or filtered.

Several UDP ports are listed as open/filtered. These could be custom applications or services that didn't respond to the scan, or they could be filtered by a firewall.



22	tcp	open	ssh	OpenSSH 9.4p1 Debian 1 (protocol 2.0)
13	udp	open|filtered	daytime	
997	udp	open|filtered	maitrd	
1008	udp	open|filtered	ufsd	
1032	udp	open|filtered	iad3	
1081	udp	open|filtered	pvuniwien	
1101	udp	open|filtered	pt2-discover	
1901	udp	open|filtered	fjicl-tep-a	
16711	udp	open|filtered	unknown	
16938	udp	open|filtered	unknown	
18617	udp	open|filtered	unknown	
19500	udp	open|filtered	unknown	
19662	udp	open|filtered	unknown	
20019	udp	open|filtered	unknown	
21344	udp	open|filtered	unknown	
22053	udp	open|filtered	unknown	
32345	udp	open|filtered	unknown	
37843	udp	open|filtered	unknown	
49205	udp	open|filtered	unknown	
49306	udp	open|filtered	unknown	

192.168.0.3

Host Status

State:	up
Open ports:	20
Filtered ports:	19
Closed ports:	1980
Scanned ports:	2000
Up time:	2612319
Last boot:	Tue Dec 12 21:44:20 2023

Addresses

IPv4:	192.168.0.3
IPv6:	Not available
MAC:	CC: 96:E5:2A: 21:3E

Operating System

Name: Linux 4.15 - 5.6

Accuracy:

Ports used

Port-Protocol-State:	22 - tcp - open
Port-Protocol-State:	1 - tcp - closed
Port-Protocol-State:	2 - udp - closed

OS Classes

Type	Vendor	OS Family	OS Generation	Accuracy
general purpose	Linux	Linux	5.X	

The MAC address of the host is provided but the vendor is not identified, which is common for local network devices. we can find the vendor with the Mac address with the Mac address vendor lookup websites.

The device is for the general-purpose device. It can be a computer or server.

It's running a Linux operating system with a kernel version between 4.15 and 5.6.

```
OS details: Linux 4.15 - 5.6
Uptime guess: 30.235 days (since Tue Dec 12 21:44:20 2023)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=261 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

The uptime guess indicates the device has been running without a reboot for 30 days since Dec 12 at 9:44 pm.

The network distance is one hop.

TCP sequence prediction difficulty indicates a high difficulty in predicting TCP sequence numbers, which is a good security measure.

```
Host script results:
|_fcrdns: FAIL (No PTR record)
| traceroute-geolocation:
|   HOP  RTT   ADDRESS      GEOLOCATION
|_  1    2.55  192.168.0.3  - ,-
|_ipidseq: ERROR: Script execution failed (use -d to debug)
|_path-mtu: PMTU == 1500
| firewalk:
| HOP  HOST           PROTOCOL  BLOCKED PORTS
|_0    192.168.0.228  udp       13,997,1008,1032,1081,1101,1901,16711,16938,18617
| qscan:
| PORT  FAMILY  MEAN (us)  STDDEV  LOSS (%)
| 1     0       1839.50    162.47  0.0%
|_22    1       2073.40    357.97  0.0%

TRACEROUTE
HOP RTT     ADDRESS
1   2.55 ms 192.168.0.3
```

The **fcrdns** script failed to find a PTR record. It means reverse DNS is not configured.

The **traceroute-geolocation** attempts to provide geolocation information for each hop in the traceroute, but here only the local hop is present with the block ports.

The **ipidseq** reports an error, it can illustrate a secure configuration that prevents the script from predicting the IP ID sequence.

The **path-mtu** confirms the Path Maximum Transmission Unit is 1500 bytes (standard for Ethernet).

The **firewalk** script suggests that certain UDP ports may be blocked by a device with the IP address 192.168.0.228, possibly a firewall or another security appliance on the network.

The **qscan** provides meantime and packet loss information for scanned ports to use to understand the network's performance and reliability.

The **Traceroute** indicates a single hop with a round-trip time (RTT) of 2.55 ms. It is the host's closeness to the scanning machine.

Manjaro Linux Machine

```
Nmap scan report for 192.168.0.11
Host is up (0.0023s latency).
Not shown: 998 closed tcp ports (reset), 996 closed udp ports (port-unreach)
Bug in http-security-headers: no string output.
PORT      STATE         SERVICE   VERSION
22/tcp    open          ssh       OpenSSH 9.6 (protocol 2.0)
| ssh2-enum-algos:
|   kex_algorithms: (12)
|       sntrup761x25519-sha512@openssh.com
|       curve25519-sha256
|       curve25519-sha256@libssh.org
|       ecdh-sha2-nistp256
|       ecdh-sha2-nistp384
|       ecdh-sha2-nistp521
|       diffie-hellman-group-exchange-sha256
|       diffie-hellman-group16-sha512
|       diffie-hellman-group18-sha512
|       diffie-hellman-group14-sha256
|       ext-info-s
|       kex-strict-s-v00@openssh.com
|   server_host_key_algorithms: (4)
|       rsa-sha2-512
|       rsa-sha2-256
|       ecdsa-sha2-nistp256
|       ssh-ed25519
|   encryption_algorithms: (6)
|       chacha20-poly1305@openssh.com
|       aes128-ctr
|       aes192-ctr
|       aes256-ctr
|       aes128-gcm@openssh.com
|       aes256-gcm@openssh.com
|   mac_algorithms: (10)
|       umac-64-etm@openssh.com
|       umac-128-etm@openssh.com
|       hmac-sha2-256-etm@openssh.com
|       hmac-sha2-512-etm@openssh.com
|       hmac-sha1-etm@openssh.com
|       umac-64@openssh.com
|       umac-128@openssh.com
|       hmac-sha2-256
|       hmac-sha2-512
|       hmac-sha1
|   compression_algorithms: (2)
|       none
|_      zlib@openssh.com
| ssh-hostkey:
|   256 b1:17:2a:04:b7:8c:ad:8d:72:50:5a:37:df:fb:c1:22 (ECDSA)
|_  256 a5:3b:16:63:68:95:c1:f4:de:4b:86:e1:0c:79:78:60 (ED25519)
|_banner: SSH-2.0-OpenSSH_9.6
80/tcp    open          http      Apache httpd 2.4.58 ((Unix))
|_http-date: Fri, 12 Jan 2024 02:19:19 GMT; -1s from local time.
|_http-comments-displayer: Couldn't find any comments.
|_http-mobileversion-checker: No mobile version detected.
| http-headers:
|   Date: Fri, 12 Jan 2024 02:19:19 GMT
|   Server: Apache/2.4.58 (Unix)
|   Last-Modified: Tue, 09 Jan 2024 23:49:54 GMT
|   ETag: "ea-60e8bf89cd213"
|   Accept-Ranges: bytes
|   Content-Length: 234
|   Connection: close
```

The scan output gives a view of the network services and configuration for the host with the IP address 192.168.0.11. The host's latency is 0.0026 seconds. Most ports are closed, indicating a secured configuration.

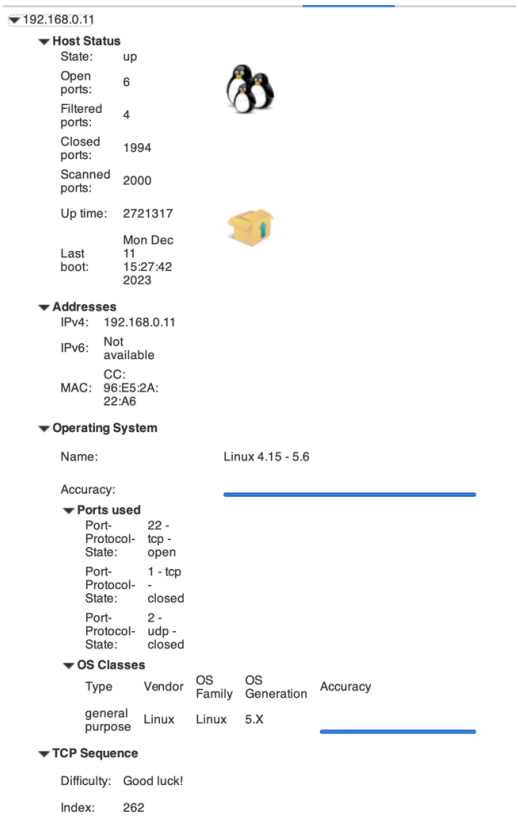
The SSH service is running on OpenSSH version 9.6. The service supports a robust selection of key exchange algorithms. The details of SSH algorithms and cryptographic methods are enumerated. The web server headers reveal that it's an Apache server.

There is an indication of a possible reverse proxy detected by the HTTP traceroute script.

An Apache HTTP server version 2.4.58 is running, serving web content. The HTTP headers and the server provide information about the server's software and configuration.

22	tcp	open	ssh	OpenSSH 9.6 (protocol 2.0)
80	tcp	open	http	Apache httpd 2.4.58 ((Unix))
13	udp	openlfiltered	daytime	
5003	udp	openlfiltered	filemaker	
16711	udp	openlfiltered	unknown	
21344	udp	openlfiltered	unknown	

Some of the UDP ports are marked as open|filtered for the daytime and FileMaker services. The scan could not determine if these UDP ports are open due to UDP being connectionless or firewall filtration.



The MAC address of the host is provided but the vendor is not identified, which is common for local network devices. we can find the vendor with the Mac address with the Mac address vendor lookup websites.

The device is for the general-purpose device. It can be a computer or server.

It's running a Linux operating system with a kernel version between 4.15 and 5.6.

```
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Uptime guess: 31.497 days (since Mon Dec 11 15:27:42 2023)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=262 (Good luck!)
IP ID Sequence Generation: All zeros
```

The uptime guess indicates the device has been running without a reboot for 31 days since Dec 11 at 3:27 pm.
The network distance is one hop.

TCP sequence prediction difficulty indicates a high difficulty in predicting TCP sequence numbers, which is a good security measure.

```
Host script results:
|_ipidseq: ERROR: Script execution failed (use -d to debug)
|_clock-skew: -1s
|_fcrdns: FAIL (No PTR record)
| traceroute-geolocation:
|   HOP  RTT   ADDRESS       GEOLOCATION
|_  1    2.32  192.168.0.11  - ,-
| qscan:
| PORT  FAMILY  MEAN (us)  STDDEV    LOSS (%)
| 1     0       3084.50    2335.46   0.0%
| 22    0       15823.90   42789.38  0.0%
|_80    0       2119.80    269.74    0.0%
|_path-mtu: PMTU == 1500
| firewalk:
| HOP  HOST           PROTOCOL  BLOCKED PORTS
|_0    192.168.0.228  udp       13,5003,16711,21344

TRACEROUTE
HOP RTT     ADDRESS
1   2.32 ms 192.168.0.11
```

The **ipidseq** reports an error, it can illustrate a secure configuration that prevents the script from predicting the IP ID sequence.

The **clock-skew** script notes a minor clock skew of -1s, which can be used for fingerprinting or coordinating attacks.

The **fcrdns s**cript failed to find a PTR record. It means reverse DNS is not configured.

The **qscan** script provides meantime and packet loss information for scanned ports to use to understand the network's performance and reliability.

The **path-mtu** confirms the Path Maximum Transmission Unit is 1500 bytes (standard for Ethernet).

The **firewalk** script suggests that certain UDP ports may be blocked by a device with the IP address 192.168.0.228, possibly a firewall or another security appliance on the network.

the **Traceroute** indicates a single hop with a round-trip time (RTT) of 2.32 ms. It is the host's closeness to the scanning machine.

Windows Machine

```
Nmap scan report for 192.168.0.15
Host is up (0.0040s latency).
Not shown: 999 open|filtered udp ports (no-response), 997 filtered tcp ports (no-response), 1 filtered udp ports (port-unreach)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH for_Windows_8.6 (protocol 2.0)
| ssh-hostkey:
|   3072 36:18:45:c2:4d:48:7a:bc:28:19:dd:b3:0c:a5:37:01 (RSA)
|   256 d2:00:66:b7:31:3c:45:79:67:fe:cc:82:a8:5a:49:35 (ECDSA)
|_  256 0f:18:82:c6:53:bd:29:3c:04:85:86:7e:53:32:c0:30 (ED25519)
| ssh2-enum-algos:
|   kex_algorithms: (9)
|       curve25519-sha256
|       curve25519-sha256@libssh.org
|       ecdh-sha2-nistp256
|       ecdh-sha2-nistp384
|       ecdh-sha2-nistp521
|       diffie-hellman-group-exchange-sha256
|       diffie-hellman-group16-sha512
|       diffie-hellman-group18-sha512
|       diffie-hellman-group14-sha256
|   server_host_key_algorithms: (5)
|       rsa-sha2-512
|       rsa-sha2-256
|       ssh-rsa
|       ecdsa-sha2-nistp256
|       ssh-ed25519
|   encryption_algorithms: (6)
|       chacha20-poly1305@openssh.com
|       aes128-ctr
|       aes192-ctr
|       aes256-ctr
|       aes128-gcm@openssh.com
|       aes256-gcm@openssh.com
|   mac_algorithms: (10)
|       umac-64-etm@openssh.com
|       umac-128-etm@openssh.com
|       hmac-sha2-256-etm@openssh.com
|       hmac-sha2-512-etm@openssh.com
|       hmac-sha1-etm@openssh.com
|       umac-64@openssh.com
|       umac-128@openssh.com
|       hmac-sha2-256
|       hmac-sha2-512
|       hmac-sha1
|   compression_algorithms: (2)
|       none
|_      zlib@openssh.com
|_banner: SSH-2.0-OpenSSH_for_Windows_8.6
135/tcp  open  msrpc   Microsoft Windows RPC
2179/tcp open  vmrdp?
MAC Address: CC:96:E5:2A:23:39 (Unknown)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
```

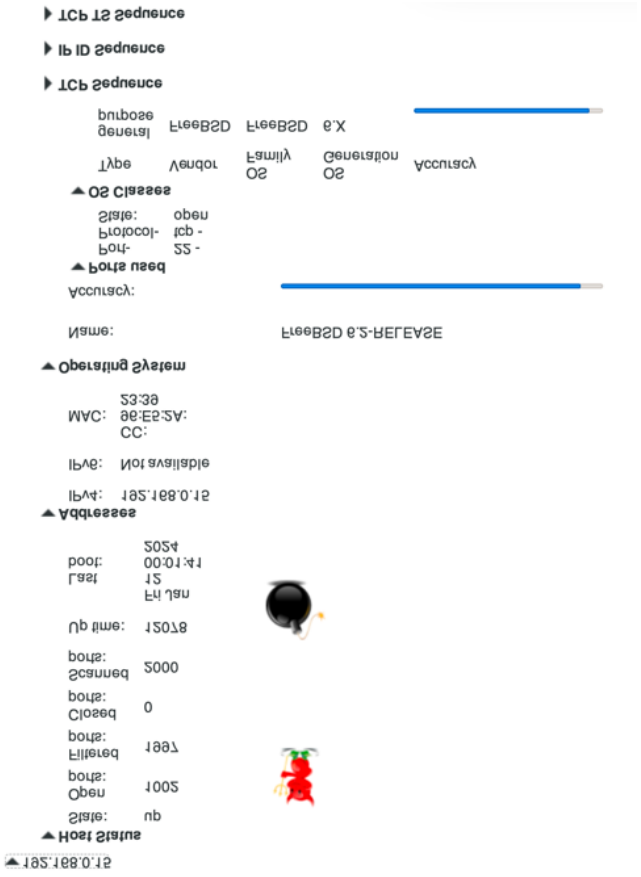
The scan output gives a view of the network services and configuration for the host with the IP address 192.168.0.15. The host is active, with a response latency of 0.0040 seconds. The scan shows a large number of TCP and UDP ports as either open|filtered or filtered, indicating a secured configuration.

SSH service is running with OpenSSH for Windows 8.6. The details of SSH algorithms and cryptographic methods are enumerated.

The Microsoft Windows RPC service is open on port 135, which is used for remote procedure calls and is a component of the Windows operating system. It is a potential vector for attacks as it has historically been vulnerable.

22	tcp	open	ssh	OpenSSH for_Windows_8.6 (protocol 2.0)
135	tcp	open	msrpc	Microsoft Windows RPC
2179	tcp	open	vmrdp	

This port 2179 is open and is related to vmrdp (Virtual Machine Remote Desktop Protocol), the version isn't specified. It suggests the host may be used for virtualization management.



The MAC address of the host is provided but the vendor is not identified, which is common for local network devices. we can find the vendor with the Mac address with the Mac address vendor lookup websites.

```
Aggressive OS guesses: FreeBSD 6.2-RELEASE (93%), Microsoft Windows 10 (92%), Microsoft Windows Server 2008 or 2008
Beta 3 (89%), Microsoft Windows Server 2008 SP1 (87%), m0n0wall 1.3b11 - 1.3b15 (FreeBSD 6.3) (86%), Juniper Networks
JUNOS 12 (86%), Juniper Networks JUNOS 9.0R2.10 (86%), Microsoft Windows 10 1703 (86%), Microsoft Windows 10 1511 -
1607 (86%), Juniper SRX100-series or SRX200-series firewall (JUNOS 10.4 - 12.1) (85%)
No exact OS matches for host (test conditions non-ideal).
```

Aggressive OS guesses range from FreeBSD to various versions of Microsoft Windows, indicating uncertainty in pinpointing the exact operating system. It could be due to the host filtering as this is the Windows device in the lab and we had to turn off the firewall to be able to finish the scan.

```
Uptime guess: 0.140 days (since Fri Jan 12 00:01:41 2024)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=255 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

The uptime guess is very short, indicating the host was booted recently.

The network distance is one hop, which is typical for devices on a local network.

TCP sequence prediction difficulty indicates a high difficulty in predicting TCP sequence numbers, which is a good security measure.

```
Host script results:
|_ipidseq: ERROR: Script execution failed (use -d to debug)
| qscan:
| PORT  FAMILY  MEAN (us)  STDDEV   LOSS (%)
| 22    0       2313.10    330.87   0.0%
| 135   0       2716.10    832.41   0.0%
|_2179  0       3672.90    3583.52  0.0%
|_path-mtu: PMTU == 1500
|_fcrdns: FAIL (No PTR record)
| traceroute-geolocation:
|   HOP  RTT   ADDRESS       GEOLOCATION
|_  1    3.98  192.168.0.15  - ,-
| firewalk:
| HOP  HOST           PROTOCOL  BLOCKED PORTS
| 0    192.168.0.228  tcp       1,3-4,6-7,9,13,17,19-20
|_                    udp       2-3,7,9,13,17,19-22

TRACEROUTE
HOP RTT     ADDRESS
1   3.98 ms 192.168.0.15
```

The **ipidseq** reports an error, it can illustrate a secure configuration that prevents the script from predicting the IP ID sequence.

The **qscan** provides meantime and packet loss information for scanned ports to use to understand the network's performance and reliability.

The **path-mtu** confirms the Path Maximum Transmission Unit is 1500 bytes (standard for Ethernet).

The **fcrdns** script failed to find a PTR record. It means reverse DNS is not configured.

The **traceroute-geolocation** attempts to provide geolocation information for each hop in the traceroute, but here only the local hop is present with the block ports.

The **firewalk** script suggests that certain UDP ports may be blocked by a device with the IP address 192.168.0.228, possibly a firewall or another security appliance on the network.

The **Traceroute** indicates a single hop with a round-trip time (RTT) of 3.98 ms. It is the host's closeness to the scanning machine.

FreeBSD Machine

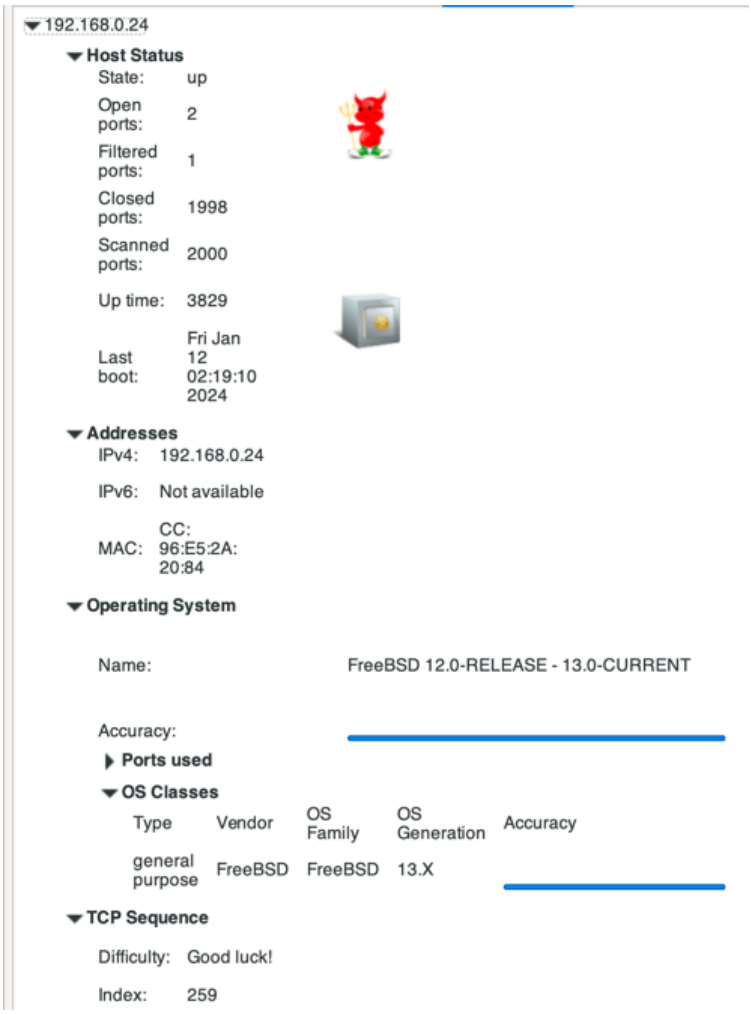
```
Nmap scan report for 192.168.0.24
Host is up (0.0027s latency).
Not shown: 999 closed udp ports (port-unreach), 999 closed tcp ports (reset)
PORT    STATE         SERVICE VERSION
22/tcp  open          ssh     OpenSSH 9.5 (FreeBSD 20231004; protocol 2.0)
| banner: SSH-2.0-OpenSSH_9.5 FreeBSD-20231004
| ssh-hostkey:
|   256 54:c3:9c:60:53:e4:a1:b1:b3:cf:5c:9b:db:81:3d:84 (ECDSA)
|_  256 c8:67:f3:c7:4f:a0:df:3e:a5:d8:a7:e4:cc:b7:cd:d5 (ED25519)
| ssh2-enum-algos:
|   kex_algorithms: (10)
|       sntrup761x25519-sha512@openssh.com
|       curve25519-sha256
|       curve25519-sha256@libssh.org
|       ecdh-sha2-nistp256
|       ecdh-sha2-nistp384
|       ecdh-sha2-nistp521
|       diffie-hellman-group-exchange-sha256
|       diffie-hellman-group16-sha512
|       diffie-hellman-group18-sha512
|       diffie-hellman-group14-sha256
|   server_host_key_algorithms: (4)
|       rsa-sha2-512
|       rsa-sha2-256
|       ecdsa-sha2-nistp256
|       ssh-ed25519
|   encryption_algorithms: (6)
|       chacha20-poly1305@openssh.com
|       aes128-ctr
|       aes192-ctr
|       aes256-ctr
|       aes128-gcm@openssh.com
|       aes256-gcm@openssh.com
|   mac_algorithms: (10)
|       umac-64-etm@openssh.com
|       umac-128-etm@openssh.com
|       hmac-sha2-256-etm@openssh.com
|       hmac-sha2-512-etm@openssh.com
|       hmac-sha1-etm@openssh.com
|       umac-64@openssh.com
|       umac-128@openssh.com
|       hmac-sha2-256
|       hmac-sha2-512
|       hmac-sha1
|   compression_algorithms: (2)
|       none
|_      zlib@openssh.com
514/udp open|filtered syslog
MAC Address: CC:96:E5:2A:20:84 (Unknown)
Device type: general purpose
Running: FreeBSD 12.X|13.X
OS CPE: cpe:/o:freebsd:freebsd:12 cpe:/o:freebsd:freebsd:13
OS details: FreeBSD 12.0-RELEASE - 13.0-CURRENT
Uptime guess: 0.044 days (since Fri Jan 12 02:19:10 2024)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=259 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: OS: FreeBSD; CPE: cpe:/o:freebsd:freebsd
```

The scan output gives a view of the network services and configuration for the host with the IP address 192.168.0.24. The host is active with a latency of 0.0027 seconds. Nearly all TCP and UDP ports are closed, indicating a secured configuration.

SSH service is running with OpenSSH version 9.5 on FreeBSD. The SSH is configured with various modern key exchange, host key, encryption, and MAC algorithms. The details of SSH algorithms and cryptographic methods are enumerated.

The port 514 which is used for syslog services, is marked as open|filtered, meaning the service didn't respond to the scan, or they could be filtered by a firewall.

	22	tcp	open	ssh	OpenSSH 9.5 (FreeBSD 20231004; protocol 2.0)
	514	udp	open|filtered	syslog	



The MAC address of the host is provided but the vendor is not identified, which is common for local network devices. we can find the vendor with the Mac address with the Mac address vendor lookup websites.

The device is for the general-purpose device. It can be a computer or server.

It is running FreeBSD version 12.x or 13.x, as indicated by the OS details.

```
OS details: FreeBSD 12.0-RELEASE - 13.0-CURRENT
Uptime guess: 0.044 days (since Fri Jan 12 02:19:10 2024)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=259 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: OS: FreeBSD; CPE: cpe:/o:freebsd:freebsd
```

The uptime guess is very short, indicating the host was booted recently.

The network distance is one hop.

TCP sequence prediction difficulty indicates a high difficulty in predicting TCP sequence numbers, which is a good security measure.

```
Host script results:
| traceroute-geolocation:
|   HOP  RTT   ADDRESS       GEOLOCATION
|_  1    2.73  192.168.0.24  - ,-
| qscan:
| PORT  FAMILY  MEAN (us)  STDDEV   LOSS (%)
| 1     0       1862.10    694.56   0.0%
|_22    0       2665.20    1913.61  0.0%
|_ipidseq: ERROR: Script execution failed (use -d to debug)
|_path-mtu: PMTU == 1500
| firewalk:
| HOP  HOST           PROTOCOL  BLOCKED PORTS
|_0    192.168.0.228  udp       514
|_fcrdns: FAIL (No PTR record)

TRACEROUTE
HOP RTT     ADDRESS
1   2.73 ms 192.168.0.24
```

The **traceroute-geolocation** attempts to provide geolocation information for each hop in the traceroute, but here only the local hop is present with the block ports.

The **qscan** provides meantime and packet loss information for scanned ports to use to understand the network's performance and reliability.

The **ipidseq** reports an error, it can illustrate a secure configuration that prevents the script from predicting the IP ID sequence.

The **path-mtu** confirms the Path Maximum Transmission Unit is 1500 bytes (standard for Ethernet).

The **firewalk** script suggests that certain UDP ports may be blocked by a device with the IP address 192.168.0.228, possibly a firewall or another security appliance on the network.

The **fcrdns** script failed to find a PTR record. It means reverse DNS is not configured.

The **Traceroute** It indicates a single hop with a round-trip time (RTT) of 2.73 ms. It is the host's closeness to the scanning machine.

MacOs Machine

```
Nmap scan report for 192.168.0.27
Host is up (0.0042s latency).
Not shown: 1000 filtered tcp ports (no-response), 704 closed udp ports (port-unreach), 293 open|filtered udp ports (no-response)
PORT     STATE SERVICE      VERSION
88/udp   open  kerberos-sec Heimdal Kerberos (server time: 2024-01-12 02:17:33Z)
137/udp  open  netbios-ns   Apple Mac OS X netbios-ns
5353/udp open  mdns         DNS-based service discovery
| dns-service-discovery:
|   22/tcp sftp-ssh
|     Address=192.168.0.27 fe80::1c47:a7e5:9ca2:319e
|   22/tcp ssh
|     Address=192.168.0.27 fe80::1c47:a7e5:9ca2:319e
|   5900/tcp rfb
|_    Address=192.168.0.27 fe80::1c47:a7e5:9ca2:319e
MAC Address: 14:98:77:37:8B:D0 (Apple)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop
Service Info: Host: DARCYS-MAC-MINI; OS: Mac OS X; CPE: cpe:/o:apple:mac_os_x

Host script results:
|_fcrdns: FAIL (No PTR record)
|_path-mtu: PMTU == 1500
| nbstat: NetBIOS name: DARCYS-MAC-MINI, NetBIOS user: <unknown>, NetBIOS MAC: 14:98:77:37:8b:d0 (Apple)
| Names:
|_  DARCYS-MAC-MINI<00>  Flags: <unique><active>
| firewalk:
| HOP  HOST           PROTOCOL  BLOCKED PORTS
| 0    192.168.0.228  tcp       1,3-4,6-7,9,13,17,19-20
|_                    udp       3,13,38,49,112,120,138,199,389,402
| traceroute-geolocation:
|   HOP  RTT   ADDRESS       GEOLOCATION
|_  1    4.20  192.168.0.27  - ,-

TRACEROUTE
HOP RTT     ADDRESS
1   4.20 ms 192.168.0.27
```

The scan output gives a view of the network services and configuration for the host with the IP address 192.168.0.27. The host is active with a latency of 0.0042 seconds. Most ports are closed, indicating a secured configuration.

	Port	Protocol	State	Service	Version
	88	udp	open	kerberos-sec	Heimdal Kerberos (server time: 2024-01-12 02:17:33Z)
	5353	udp	open	mdns	DNS-based service discovery
	137	udp	open	netbios-ns	Apple Mac OS X netbios-ns

An open UDP port 83 is running Kerberos (Heimdal Kerberos). It is a network authentication protocol that uses secret-key cryptography for secure authentication.

An open UDP port 137 is running NetBIOS. It is used on Apple Mac OS X for local network name registration.

An open UDP port 5353 is running Multicast DNS (mDNS). It is used for DNS-based service discovery in local networks.

The dns-service-discovery script has found services like SFTP over SSH, SSH, and RFB on port 22/tcp and 5900/tcp, with IPv6 addresses provided for the host.

192.168.0.27

Host Status

State:	up
Open ports:	296
Filtered ports:	1293
Closed ports:	704
Scanned ports:	2000
Up time:	Not available
Last boot:	Not available

Addresses

IPv4: 192.168.0.27

IPv6: Not available

MAC: 14:98:77:37:8B:D0

TCP Sequence

Difficulty:

Index:

Values:

IP ID Sequence

Class:

Values:

TCP TS Sequence

Class:

Values:

Comments

The MAC address is associated with Apple, Inc.

The service info identifies the host by the name "DARCYS-MAC-MINI" and the OS is Mac OS X.

```
MAC Address: 14:98:77:37:8B:D0 (Apple)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop
Service Info: Host: DARCYS-MAC-MINI; OS: Mac OS X; CPE: cpe:/o:apple:mac_os_x
```

The network distance is one hop.
The exact OS version could not be determined.

```
Host script results:
|_fcrdns: FAIL (No PTR record)
|_path-mtu: PMTU == 1500
| nbstat: NetBIOS name: DARCYS-MAC-MINI, NetBIOS user: <unknown>, NetBIOS MAC: 14:98:77:37:8b:d0 (Apple)
| Names:
|_  DARCYS-MAC-MINI<00>  Flags: <unique><active>
| firewalk:
| HOP  HOST           PROTOCOL  BLOCKED PORTS
| 0    192.168.0.228  tcp       1,3-4,6-7,9,13,17,19-20
|_                    udp       3,13,38,49,112,120,138,199,389,402
| traceroute-geolocation:
|   HOP  RTT   ADDRESS       GEOLOCATION
|_  1    4.20  192.168.0.27  - ,-

TRACEROUTE
HOP RTT     ADDRESS
1   4.20 ms 192.168.0.27
```

The **fcrdns** script failed to find a PTR record. It means reverse DNS is not configured.

The **path-mtu** confirms the Path Maximum Transmission Unit is 1500 bytes (standard for Ethernet).

The **nbstat** script output provides the NetBIOS name "DARCYS-MAC-MINI" but NetBIOS username is undetermined and the MAC address belongs to Apple.

The **firewalk** script suggests that certain UDP ports may be blocked by a device with the IP address 192.168.0.228, possibly a firewall or another security appliance on the network.

The **traceroute-geolocation** attempts to provide geolocation information for each hop in the traceroute, but here only the local hop is present with the block ports.

The **Traceroute** indicates a single hop with a round-trip time (RTT) of 4.20 ms. It is the host's closeness to the scanning machine.

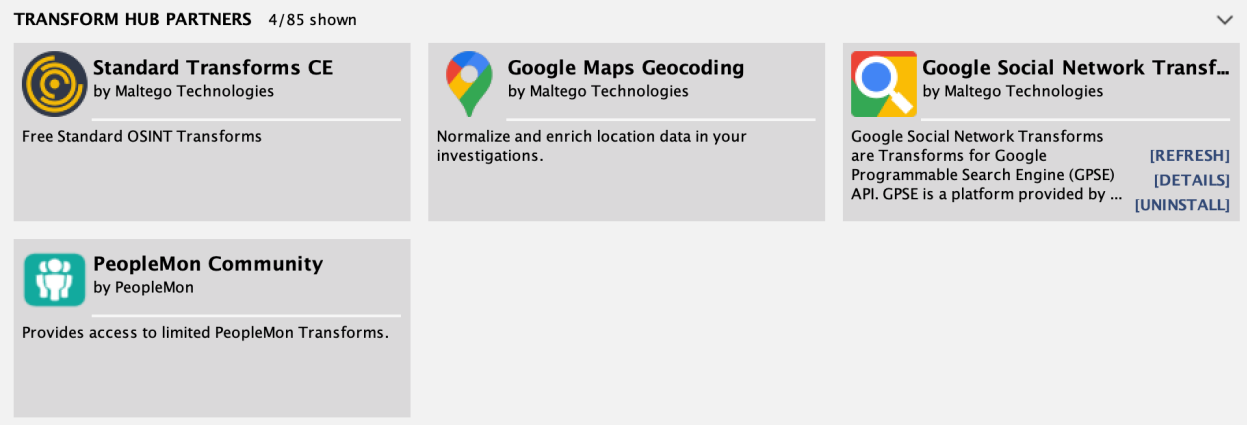
Maltego

Maltego is an open-source intelligence (OSINT) and graphical link analysis tool for gathering information and connecting the information for investigations from public sources. Public information includes the Person's name, social media usernames, email addresses, IP addresses, Personal devices, websites, etc. It is highly effective in piecing together digital footprints across the internet.

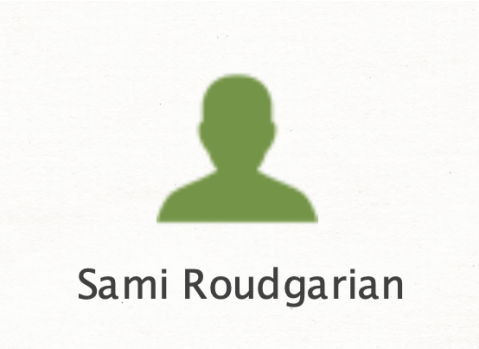
Maltego uses “transforms” – scripts that extract and visualize data -to automatically fetch and parse the data from different resources and display them on the graph.

Maltego has different subscriptions to meet different user needs. Each version offers a set of capabilities designed for the user's investigative requirements.

For the purpose of this assignment, I have to use the Maltego to look for my digital footprints using the free community edition of the Maltego and its APIs. Despite the limitations, such as restricted API calls on services like Google Social Network Transform, it still provides valuable information.



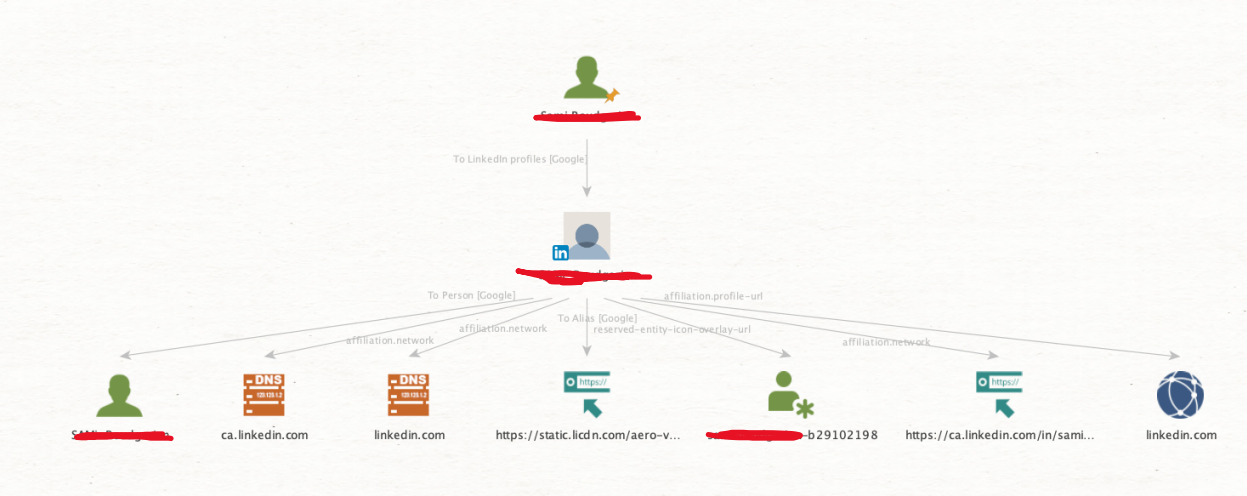
These are the APIs that I installed.



I started the investigation by selecting a 'Person' entity and inputting my name.

```
Running transform To Quora profiles [Google] on 1 entities (from entity "[redacted]")
Running transform To Facebook profiles [Google] on 1 entities (from entity "[redacted]")
Running transform To Website [Bing] on 1 entities (from entity "[redacted]")
Running transform To Snapchat profiles [Google] on 1 entities (from entity "[redacted]")
Running transform To Phone Numbers [within Properties] on 1 entities (from entity "[redacted]")
No results found (from entity "[redacted]")
ING Please Input the BING API Key (from entity "Sami Roudgarian")
Transform To Website [Bing] returned with 0 entities (from entity "[redacted]")
Transform To Phone Numbers [within Properties] returned with 0 entities (from entity "[redacted]")
Transform To Website [Bing] done (from entity "[redacted]")
Transform To Phone Numbers [within Properties] done (from entity "[redacted]")
Free Google Custom Search Engine Transform runs : 17 of 20 credits remaining. Current quota period ends at February 1,
Free Google Custom Search Engine Transform runs : 18 of 20 credits remaining. Current quota period ends at February 1,
Free Google Custom Search Engine Transform runs : 19 of 20 credits remaining. Current quota period ends at February 1,
Running transform To GitHub profiles [Google] on 1 entities (from entity "[redacted]")
Running transform To YouTube profiles [Google] on 1 entities (from entity "[redacted]")
Transform To Facebook profiles [Google] completed in 2 s 392 ms (from entity "[redacted]")
Transform To Snapchat profiles [Google] completed in 2 s 390 ms (from entity "[redacted]")
Transform To Quora profiles [Google] completed in 2 s 392 ms (from entity "[redacted]")
Running transform To Weibo profiles [Google] on 1 entities (from entity "[redacted]")[1/16/24, 6:23 PM] INFO Runn
Running transform To VK profiles [Google] on 1 entities (from entity

Free Google Custom Search Engine Transform runs : 16 of 20 credits remaining. Current quota period ends at February 1,
Free Google Custom Search Engine Transform runs : 15 of 20 credits remaining. Current quota period ends at February 1,
```

The search started primarily utilizing the Google API and was successfully able to find my LinkedIn profile on the Internet.

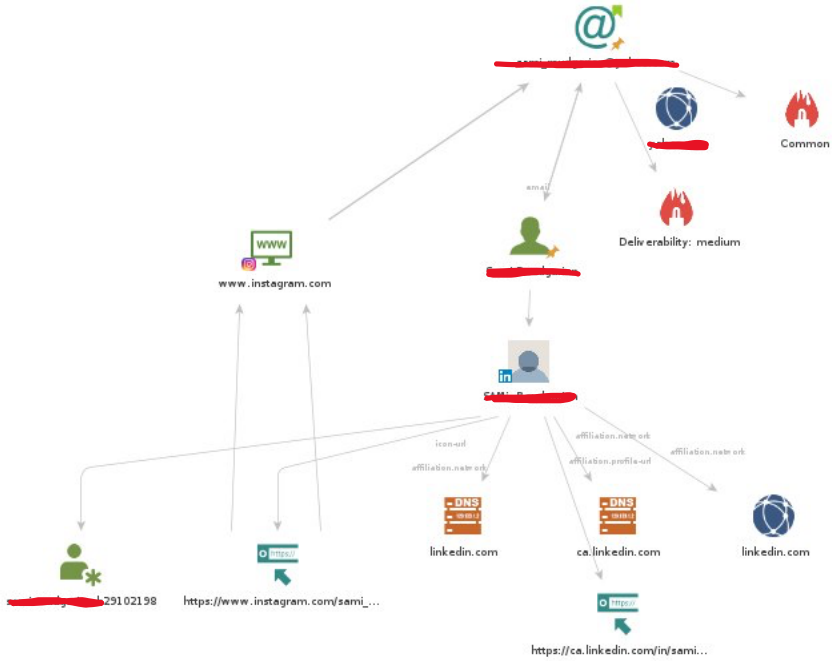


As it shows the central entity is my first and last name.

The social media account that represents my online presence is my LinkedIn profile.

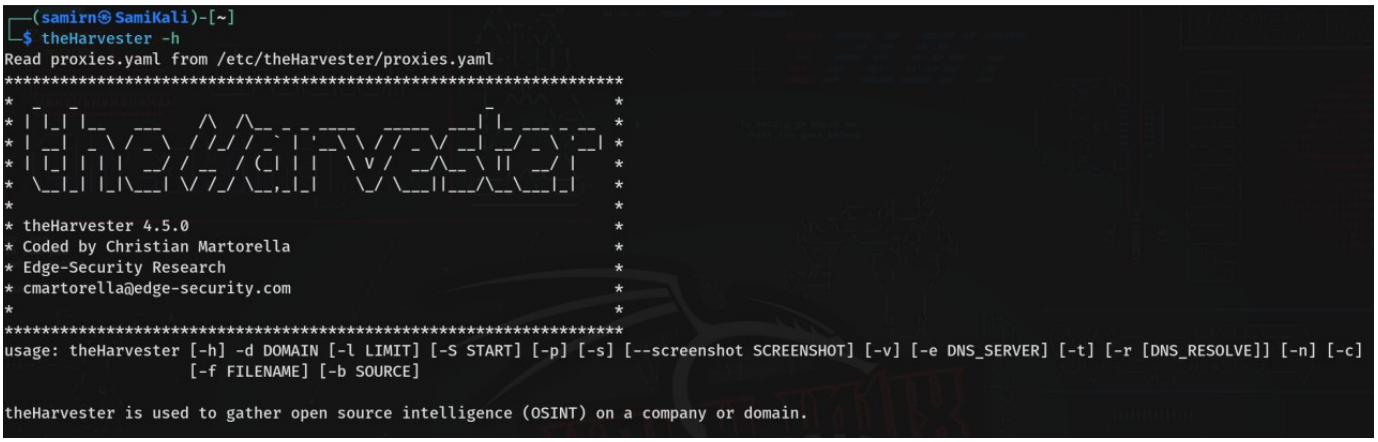
Maltego has fetched DNS information related to the LinkedIn URL of my account.

However, to gain a more comprehensive view. I added additional entities manually, such as my yahoo email address. By transforming the email entity, it was possible to map DNS information related to the associated Yahoo mail servers and link them to an active Instagram account that utilizes my email.

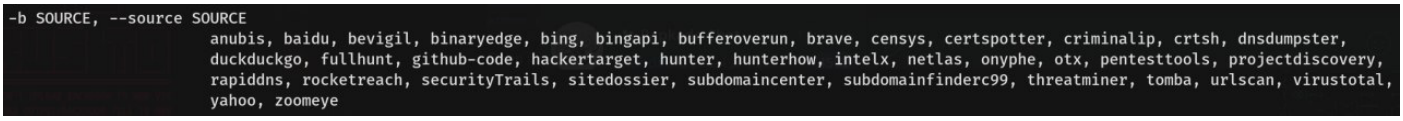


Overall, Maltego is not just a tool for investigating personal data points; it is a comprehensive platform that enables a complete view of digital footprints. Maltego provides a framework for navigating available digital information. Maltego empowers users to conduct their specific investigative needs with its analytical capabilities.

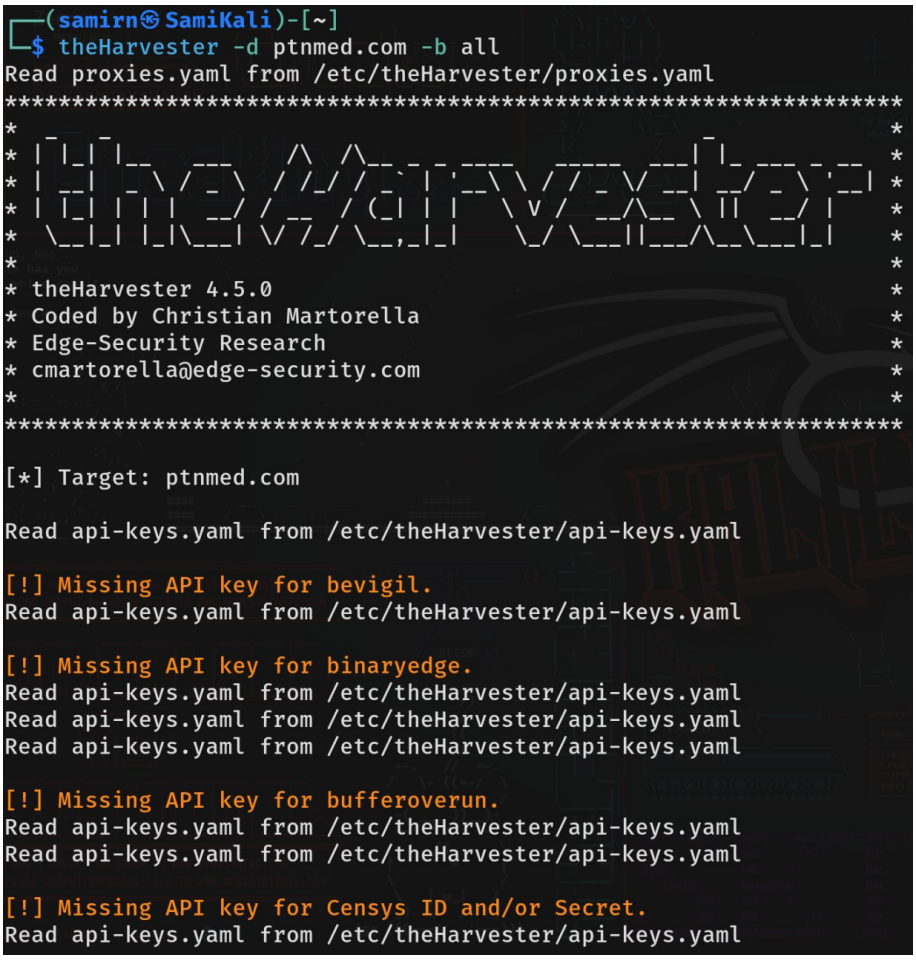
theHarvester



theHarvester is another tool designed to gather open-source intelligence (OSINT) about a given target, which could be a domain, a company, or an individual.



theHarvester includes various public sources like search engines, social media, and other internet resources to gather information.
Same as Maltego, we can try different data types such as email addresses, subdomains, company names, etc.
It also uses APIs from different data sources to enhance the data collection. Some of these APIs require API keys for functionality.



A typical command specifies the domain to investigate and the sources we want to use. For this example, I am using the website of our company which is based in Iran. **-d ptnmed.com** sets the domain and **-b all** specifies to use all available sources.

Setting the sources to all will give us the Missing API key warning for most of the sources as I do not have the API key for them.

In the previous picture, the output was not suitable as we had all the sources running with their warnings and errors for the API key.

In this output, theHarvester investigated the domain **ptnmed.com** using the Bing and Yahoo search engines.

The **-b bing, yahoo** flags did not find any IP addresses but discovered one email (info@ptnmed.com) and one host (azmoon.ptnmed.com).

Overall, theHarvester is a useful tool for quickly gathering public data, which can be used in various investigative scenarios such as cybersecurity assessments or competitive analysis. However, to maximize its effectiveness, we should obtain the necessary API keys to access a wider range of data sources. These APIs ensure that not everyone can access the data freely. It protects against misuse or collection of information for personal use.

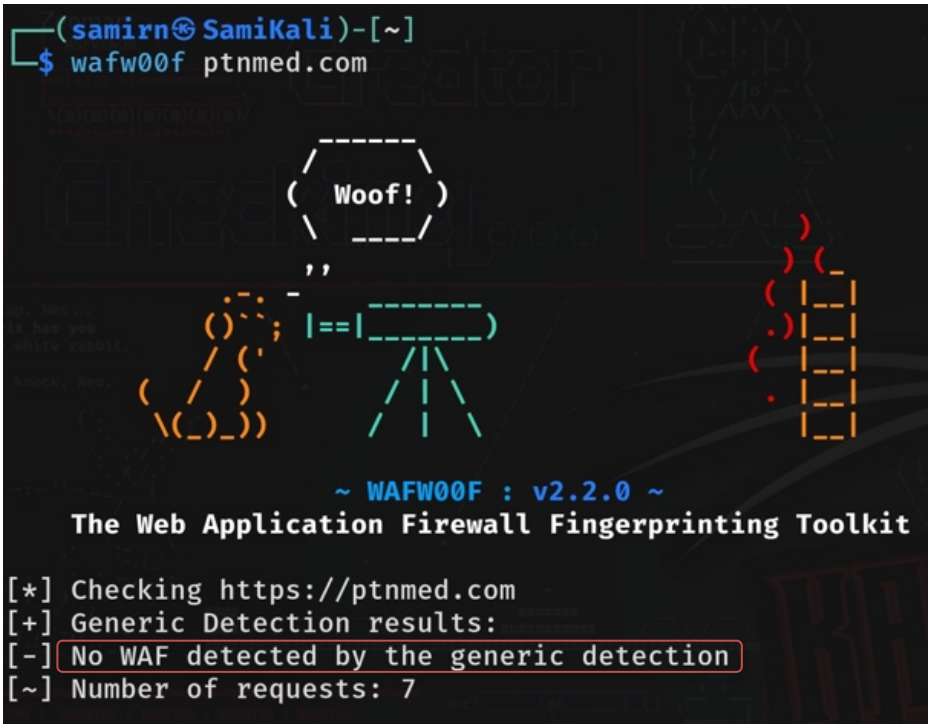
Wafw00f

```
┌──(samirn㉿SamiKali)-[~]
└─$ wafw00f -h
Usage: wafw00f url1 [url2 [url3 ... ]]
example: wafw00f http://www.victim.org/
```

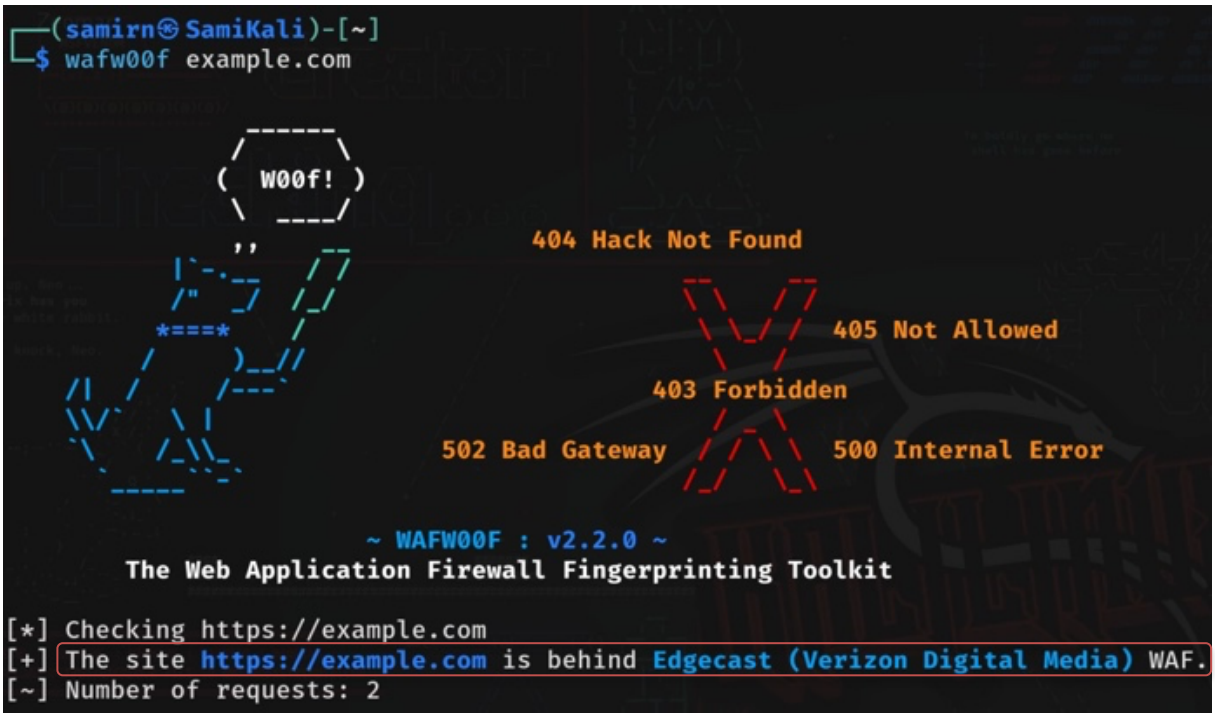
Wafw00f is designed to identify and fingerprint Web Application Firewalls (WAFs) that are deployed to protect websites. Wafw00f can determine whether a WAF is present by analyzing web servers' responses.

Run the command **wafw00f** followed by the domain name of the target website to use it. The tool will then perform its checks and report its findings. Wafw00f should be used responsibly and ethically like all the other tools. In this case, **ptnmed.com** is our own company's domain, and I have the authority to conduct such a test.

The output illustrates that made a total of 7 HTTP requests during the detection process and did not detect the presence of a WAF using its detection methods. which means our website does not have a web application firewalls setup or Wafw00f was not able to detect them.



In another example, I will run example.com as it shows their web application Firewalls. I used example.com as they had "These domains may be used as illustrative examples in documents without prior coordination with us." on their website.



The output illustrates that made a total of 2 HTTP requests during the detection process and detected Edgecast WAF on the domain. which means example.com is behind Edgecast Web Application Firewalls.

In conclusion, Wafw00f is a proactive security measure which can help to understand the defence strategy of a website.

whatweb

WhatWeb identifies web technologies used on websites, such as their content management systems (CMS), web servers, JavaScript libraries, etc. It examines a website's HTTP headers, HTML code, and often JavaScript files to determine signatures and patterns that match known technologies.

In our example, WhatWeb makes an HTTP request to ptnmed.com and analyzes the HTTP response headers, the HTML body, and maybe the CSS or JavaScript files. Then it checks the patterns that match their pattern in the database against the response content to identify software, plugins, and versions.

The 200 OK illustrates that the website is accessible.

ASP.NET: The site is using the ASP.NET framework.

MVC5.2: The Model-View-Controller framework version 5.2.

Bootstrap: The front-end framework

Country: It is hosted in Iran.

Email: info@ptnmed.com.

HTML5: The markup language version.

HTTP Server: The web server is Microsoft-IIS/10.0.

JavaScript Libraries: jQuery and Modernizr

Script: The presence of inline scripts or external JavaScript files.

Title: The title of the homepage in Farsi

Uncommon Headers: **X-AspNetMvc-Version** and **X-Powered-By**

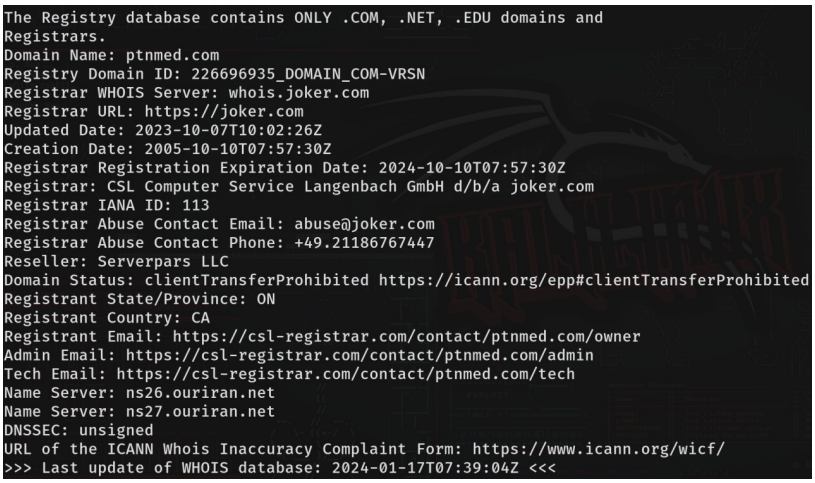
whois

```
┌──(samirn㉿SamiKali)-[~]
└─$ whois -h
whois: option requires an argument -- 'h'
Usage: whois [OPTION]... OBJECT...
```

The whois command is an Internet record listing that identifies who owns a domain and how to get in contact with it. The Internet Corporation for Assigned Names and Numbers (ICANN) regulates domain name registration and ownership. People often use it when they are deciding on buying a domain.

whois searches databases that store information about domain ownership and affiliation. It gives information about the domain's registration, the domain's creation and expiration dates, nameservers, current registrar, and administrative contacts.

```
┌──(samirn㉿SamiKali)-[~]
└─$ whois ptnmed.com
   Domain Name: PTNMED.COM
   Registry Domain ID: 226696935_DOMAIN_COM-VRSN
   Registrar WHOIS Server: whois.joker.com
   Registrar URL: http://www.joker.com
   Updated Date: 2023-10-07T10:02:22Z
   Creation Date: 2005-10-10T07:57:30Z
   Registry Expiry Date: 2024-10-10T07:57:30Z
   Registrar: CSL Computer Service Langenbach GmbH d/b/a joker.com
   Registrar IANA ID: 113
   Registrar Abuse Contact Email: abuse@joker.com
   Registrar Abuse Contact Phone: +49.21186767447
   Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
   Name Server: NS26.OURIRAN.NET
   Name Server: NS27.OURIRAN.NET
   DNSSEC: unsigned
   URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2024-01-17T07:38:45Z <<<
```



The pictures show the whois lookup results for the domain **ptnmed.com**.

It shows the domain's updated date, original creation date, and expiration date.

The registrar's name, IANA ID, contact email, and abuse contact phone number are given, which indicate the organization responsible for the registration of the domain.

The status 'clientTransferProhibited' is a security measure to prevent unauthorized domain hijacking and indicates restrictions on domain transfers.

Servers (NS26.OURIRAN.NET and NS27.OURIRAN.NET) handle the DNS operations for the domain.

Overall, whois is for domain investigation, providing information that can support activities such as cybersecurity analysis, domain purchasing decisions, and legal investigations.

Dmitry

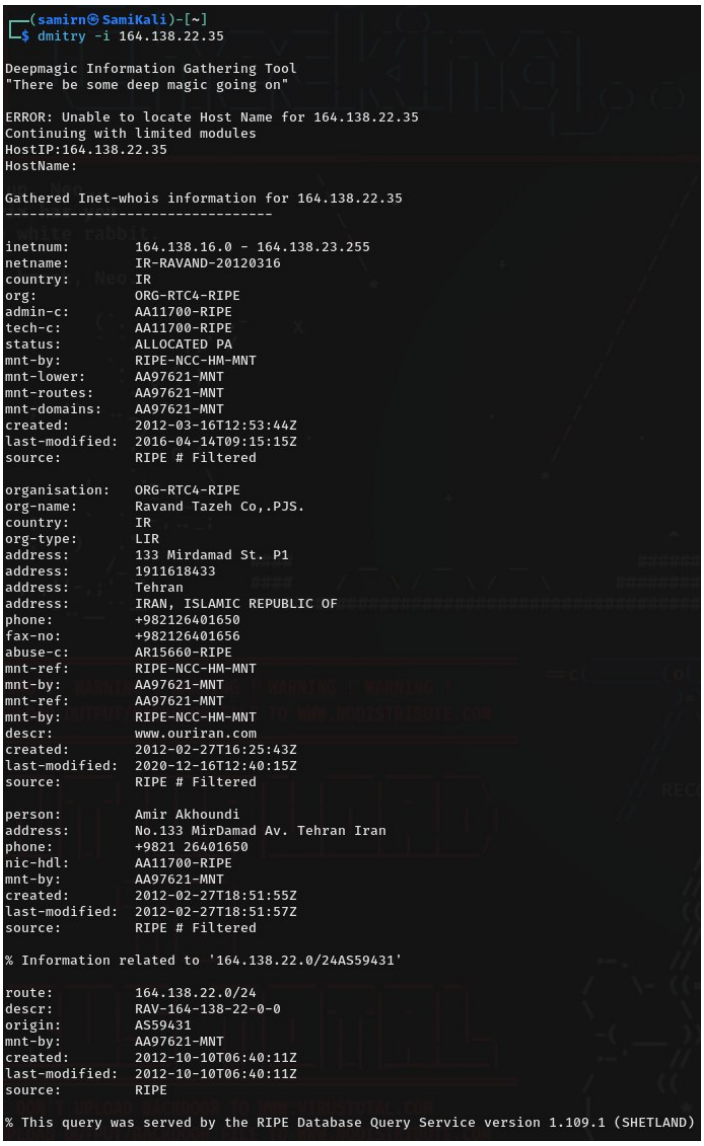
```
┌──(samirn㉿SamiKali)-[~]
└─$ dmitry -h
Deepmagic Information Gathering Tool
"There be some deep magic going on"

dmitry: invalid option -- 'h'
Usage: dmitry [-winsepfb] [-t 0-9] [-o %host.txt] host
  -o     Save output to %host.txt or to file specified by -o file
  -i     Perform a whois lookup on the IP address of a host
  -w     Perform a whois lookup on the domain name of a host
  -n     Retrieve Netcraft.com information on a host
  -s     Perform a search for possible subdomains
  -e     Perform a search for possible email addresses
  -p     Perform a TCP port scan on a host
* -f     Perform a TCP port scan on a host showing output reporting filtered ports
* -b     Read in the banner received from the scanned port
* -t 0-9 Set the TTL in seconds when scanning a TCP port ( Default 2 )
*Requires the -p flagged to be passed
```

DMitry (Deepmagic Information Gathering Tool) is capable of gathering as much information as possible about a host. It's a reconnaissance tool that penetration testers use to gather public information that can be useful for analysis.

It can perform basic whois queries that we did in previous example, port scans, and search for email addresses within a domain. it can also reveal the structure of an organization's external network.

```
┌──(samirn㉿SamiKali)-[~]
└─$ dmitry -w ptnmed.com
Deepmagic Information Gathering Tool
"There be some deep magic going on"

HostIP:164.138.22.35
HostName:ptnmed.com

Gathered Inic-whois information for ptnmed.com
---------------------------------
   Domain Name: PTNMED.COM
   Registry Domain ID: 226696935_DOMAIN_COM-VRSN
   Registrar WHOIS Server: whois.joker.com
   Registrar URL: http://www.joker.com
   Updated Date: 2023-10-07T10:02:22Z
   Creation Date: 2005-10-10T07:57:30Z
   Registry Expiry Date: 2024-10-10T07:57:30Z
   Registrar: CSL Computer Service Langenbach GmbH d/b/a joker.com
   Registrar IANA ID: 113
   Registrar Abuse Contact Email: abuse@joker.com
   Registrar Abuse Contact Phone: +49.21186767447
   Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
   Name Server: NS26.OURIRAN.NET
   Name Server: NS27.OURIRAN.NET
   DNSSEC: unsigned
   URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2024-01-17T06:57:21Z <<<
```

In this picture, I used the -w flag which performs a whois lookup on the domain name of our host to gather information about the domain ptnmed.com. This information is from the “Whois” server. The reason I used this command is to get the HostIp and use it for HostIP lookup on Dmitry to gather more insights about the domain and server.



Here we looked up the host IP associated with ptnmed.com.

- **inetnum**: This is the range of IP addresses managed by the organization.
- **Netname**: The name assigned to the network.
- **Country**: Iran (IR).
- **Organization**: Details of the organization responsible for the IP address, its name (Ravand Tazeh Co., PJS.), address, and contact information.
- **Admin**: Administrative contact information, individual who is responsible for managing the IP address or network.
- **Created**: The date when the IP range was assigned.
- **Last Modified**: The last date when the IP range information was updated.
- **Source**: RIPE is the registry or database which we got the information from.

Conclusion

In conclusion, the comprehensive network reconnaissance task is undertaken with tools to get deep insight into the security posture of the targeted network environment. The findings from Zenmap, Maltego, theHarvester, Wafw00f, WhatWeb, whois, and Dmitry have illustrated a detailed picture of the vulnerabilities, strengths, and potential areas for improvement about the devices and domain we used and their network structure.

Zenmap's quick and comprehensive scans revealed information about active devices, open ports, and potential security risks associated with common services such as SSH, Kerberos-sec, and VNC. These findings emphasize the need for severe security protocols, including robust password policies and secure network configurations to mitigate unauthorized access. From the attacker's perspective, performing a detailed scan is risky as they are actively scanning ports sequentially which some systems can detect those patterns. Attackers would use the information gathered to find the best tools and attacking strategies against the open ports and running services on the specified target.

Maltego's analysis of digital footprints emphasized the importance of monitoring and managing digital information that can be leveraged in cyber-attacks. Similarly, theHarvester extracts valuable data from public sources about the domain or a company which can be used to refine attack strategies and identify additional targets within the network.

Wafw00f's investigation into Web Application Firewalls. It indicates a potential vulnerability in web server defences which can easily arise from the lack of effective WAF solutions and can guide attackers to exploit vulnerable web servers to gain unauthorized access to sensitive data.

WhatWeb's identification of web technologies and server configurations provides a roadmap for addressing software-related security issues. It gives vital information about the version and configurations of the system that can be used in tailoring their exploits to increase the likelihood of successful breaches.

whois and Dmitry gather domain information to reveal the importance of robust domain management practices, including secure domain registration and transfer processes. Based on this information, attackers can plan domain hijacking or intercept communications.

The collective findings from these tools offer actionable recommendations for enhancing network security. The reconnaissance task has uncovered areas that could be exploited by attackers. These weaknesses can arise from network services, vulnerabilities in web applications,

exploitable software configurations, etc. The findings highlight the importance of robust security measures and constant maintenance and monitoring to protect networks from potential cyber threats. The need for network administrators is crucial to continuously assess and enhance their cybersecurity defences with the evolution of attack strategies.

Appendix A: Input Data

- These are the input data used during the different tasks and this information are related to me.

Data	Info	Value
Name	My first and last name which was used during the Maltego task.	Sami
Email	My email was used during the Maltego task.	
Website	My Company's website which is based in Iran and was used for other kali linux tools.	ptnmed.com
Website	Used during the WafW00f task to show the web application firewall that they are using.	example.com