NETWORK VULNERABILITY ASSESSMENT REPORT

01/10/2024

COMMUNICATIONS & NETWORK SECURITY

Eleftheria Goula

Contents

Executive Summary 5
Introduction 6
Details of the network 6
Scope of the test 7
Purpose of the test 8
Technical Summary 11
Vulnerability Definitions 11
Tools used for the test 12
Findings 14
Discovering the home network 14
Internal Vulnerability Scanning 19
Port Scanning 19
External Vulnerability Scanning 26
Conclusion 27
References 28
Table of Figures 30

Executive Summary

This vulnerability assessment was conducted in a controlled home lab environment designed to simulate real-world network security threats and defenses. The lab environment consisted of a Kali Linux virtual machine (VM) on a host PC and an Ubuntu laptop, with the goal to explore vulnerability scanning tools and commands and identify potential security weaknesses within the local network. Using tools like nmap and Shodan, both internal and external vulnerability scans were conducted to detect open ports, exposed services, and potential vulnerabilities. This report outlines the findings along with the network discovery and vulnerability scanning techniques and tools that were used, the expected outcomes and the importance of the key objectives of this test.

Introduction

Details of the network

The setup was composed of two main devices: a Virtual Machine (VM) hosted on a desktop computer and a separate laptop. The VM was configured with bridged networking, a method that allows the VM to function as a full-fledged machine on the local network. By using bridged networking, the VM gains a unique IP address on the network, making it indistinguishable from other physical devices. This configuration was crucial for realistic network simulations, as the VM could interact with the network in the same way as a physical machine.

Host PC and Virtual Machine:

- Host PC: The physical machine running the VM, which provided the necessary resources (CPU, memory, and network interface) for the virtual environment.

- Virtual Machine: On this VM, Kali Linux 2024.2 was installed. Kali Linux is a powerful penetration testing and ethical hacking distribution, featuring a wide array of tools for cybersecurity testing, making it ideal for this experiment.

The decision to use bridged networking ensured that the VM appeared to other devices on the network as a distinct machine with its own IP address and MAC address, as opposed to operating through the host PC's IP and MAC addresses. This allowed more realistic networking scenarios, particularly in terms of network discovery, interaction, and attacks.

Laptop:

On the laptop, Ubuntu 22.04.4 LTS was installed. Ubuntu is a widely-used Linux distribution, chosen for its stability and compatibility with various network services. In this experiment, the laptop functioned as part of the simulated network infrastructure. Ubuntu's inherent security features provided a stable test environment, allowing the Kali VM to attempt certain penetration tests or vulnerability assessments against it.

Security Considerations:

For the purposes of this experiment, the Internet Security software on the host PC was temporarily disabled. This action was necessary to prevent interference with the network traffic and security tests conducted from the Kali Linux VM. Some antivirus and firewall programs may block certain network activities, such as port scanning, vulnerability scanning, or even intercept network packets, which could have skewed the results of the experiment. Disabling these protections ensured that the Kali Linux VM could operate freely and interact with the network without restrictions.

Scope of the test

The primary objective of this lab is to explore and become proficient with various network discovery and vulnerability scanning tools and commands. The focus is on understanding the structure and weaknesses of the local network by gathering critical information, such as host addresses, open and closed ports, and identifying potential vulnerabilities. By performing both internal and external vulnerability scans, the aim is to simulate common cybersecurity practices used in real-world penetration testing and network defense.

Key Objectives:

1. **Finding Hosts in the Internal Network**: The first step of the experiment involves identifying all active devices or hosts within the internal network. This is done using network discovery tools or commands that can scan for IP addresses and map the network topology.
   - **Objective**: Identify the IP addresses, MAC addresses, and device types of all machines connected to the local network.
   - **Techniques**: Commands like nmap or ping sweep can be used for host discovery.
   - **Expected Outcome**: A list of reachable devices, including their IP addresses and possibly the services they are running.

2. **Internal Vulnerability Scanning**: Once hosts are identified, the next task is to perform an internal vulnerability scan on the network. This involves probing the discovered devices to check for open or closed ports, services running on those ports, and identifying any known vulnerabilities or misconfigurations.

    - **Objective**: Uncover potential weaknesses in devices that could be exploited by an attacker within the network, such as open ports or insecure configurations.
    - **Techniques**: Using tools like nmap with specific scanning options (e.g., port scans, service detection, OS fingerprinting).
    - **Expected Outcome**: A report detailing which devices have open ports, what services they are running, and any potential vulnerabilities or misconfigurations that could be exploited.

3. **External Vulnerability Scanning**: After scanning the internal network, the experiment extends to external vulnerability scanning. This involves checking the network's exposure to external threats by simulating attacks that could originate from outside the network (e.g., the internet). The goal is to see how well the internal network is protected against external access and to identify any open ports, services, or vulnerabilities accessible from outside the network perimeter.
    - **Objective**: Assess the external attack surface of the network by identifying externally accessible services, misconfigurations, or vulnerabilities that an attacker might exploit.
    - **Techniques**: Tools like Shodan can be used to simulate external vulnerability assessments.
    - **Expected Outcome**: A detailed report showing which ports and services are visible to the outside world, along with any potential security weaknesses that could be targeted by attackers.

Purpose of the test

The primary purpose of this lab is to provide hands-on experience with network discovery and vulnerability scanning techniques. The test is designed to introduce participants to essential cybersecurity skills, helping them understand how to identify potential threats and weaknesses within a network. By using widely-adopted tools and commands, participants gain practical exposure to methods commonly used by both security professionals and malicious actors to evaluate and target network vulnerabilities.

Key Objectives:

1. **Develop Network Security Awareness**: The test aims to increase awareness of how networks function and how they can be vulnerable to internal and external threats. Understanding network topologies, discovering hosts, and identifying exposed services are critical first steps in securing a network.
   - **Why this is important**: Many network attacks occur because administrators are unaware of all the devices or services running on their network. This test helps participants realize the importance of continuous network monitoring and discovery.

2. **Practice Using Essential Tools**: A secondary purpose is to help participants become familiar with common security tools such as nmap and other vulnerability scanners. These tools are standard in the field of cybersecurity and are used by professionals to conduct network assessments.
   - **Why this is important**: Mastery of these tools is essential for anyone entering the field of cybersecurity, as they are foundational for vulnerability assessments, penetration testing, and network defense.

3. **Understand Vulnerability Scanning**: The test also emphasizes the need to understand vulnerability scanning, which is crucial for identifying weaknesses within the network. By conducting both internal and external scans, participants will learn how to identify potential vulnerabilities that could be exploited by attackers.
   - **Why this is important**: Understanding how vulnerabilities are discovered provides insight into how attackers operate and what steps can be taken to prevent such attacks. Participants will also gain a clearer understanding of how misconfigurations, outdated software, and exposed services contribute to network vulnerabilities.

4. **Simulate Real-World Attack Scenarios**: By combining network discovery with internal and external vulnerability scans, participants simulate real-world attack scenarios. These simulations demonstrate how an attacker might gain a foothold within a network and exploit weaknesses to escalate privileges or exfiltrate sensitive information.
   - **Why this is important**: Understanding how attackers approach a network enables defenders to anticipate and mitigate threats more effectively. This simulation prepares participants to think critically about security measures and the importance of proactive defense.

5. **Prepare for Defensive Action**: The test is designed to prepare participants for future defensive actions by teaching them to think from both an offensive (attacker) and defensive (security professional) perspective. By identifying potential vulnerabilities, participants can make recommendations for improving security, including patching vulnerabilities, closing unnecessary ports, or improving network segmentation.
    - **Why this is important**: Effective cybersecurity professionals must be able to both detect threats and recommend practical steps for mitigation. The test helps develop the analytical mindset needed to identify risks and take appropriate action to secure networks.

Technical Summary

Vulnerability Definitions

- **Open Ports:** Unnecessary or unsecured open ports can expose network services to external threats. Attackers can use these ports to exploit weaknesses in the services running on them. Commonly targeted ports include those for web services (e.g., port 80/443) and remote access protocols (e.g., port 22 for SSH) [1].

- **Misconfigured Services:** Incorrectly configured services, such as default credentials, insecure settings, or improperly set access controls, can provide attackers with easy entry points. This could include database services with poor authentication or network services left open to external access [2].

- **Unpatched Software:** Software that has not been updated to address known security vulnerabilities is at significant risk. Attackers often exploit outdated software with public vulnerabilities that have documented exploits [3].

- **Weak Encryption:** Outdated or inadequate encryption protocols, such as the use of deprecated SSL/TLS versions, make it easier for attackers to intercept or manipulate sensitive data during transmission [4].

- **Exposed Services to the Internet:** Public-facing services, such as web servers, FTP servers, or email servers, are more vulnerable to external attacks. If not properly secured, attackers can exploit these to gain access to the internal network [5].

- **Weak Authentication Mechanisms:** Systems or services using weak passwords or insecure login methods can be susceptible to brute-force or dictionary attacks [6].

- **Unmonitored Network Traffic:** Networks without proper monitoring or logging may fail to detect suspicious activities, such as reconnaissance scans, port sweeps, or unauthorized access attempts [7].

- **Insufficient Firewall Rules:** Weak or overly permissive firewall configurations can allow unwanted or malicious traffic into the network, increasing the risk of exploitation [8].

Tools used for the test

Several tools were used during this network vulnerability assessment to discover hosts, scan for vulnerabilities, and analyze the network traffic. Each tool served a specific purpose, allowing for a thorough evaluation of the internal and external network environment. Below are the key tools used in this assessment:

- **netstat:** A command-line tool used to examine network connections, routing tables, interface statistics, masquerade connections, and multicast memberships. This command is commonly used in network diagnostics to understand how packets are routed and which gateway or router is being used for external communication [9].
    - -r: This option shows the kernel's routing table. It lists the routes that data packets will take to reach different networks or hosts.
    - -n: This flag forces the command to display addresses and port numbers in numerical form, rather than resolving and showing them as hostnames or service names. This speeds up the display and is useful when resolving addresses is unnecessary or when DNS resolution is slow.

- **netdiscover:** A passive network scanning tool designed for discovering live hosts on a local network. It works by sending ARP requests across the specified network range and listening for responses. When devices reply, netdiscover maps their IP addresses and MAC addresses, effectively building a list of active devices. In this assessment, netdiscover was used to scan the entire local network range (192.168.1.0/16), revealing all connected devices without actively sending traffic that could alert security systems. This makes it a stealthy option for local network reconnaissance [10].

- **nmap:** Nmap (Network Mapper) is one of the most widely used network scanning tools in cybersecurity. It allows users to perform port scanning, service detection, and vulnerability discovery on local and remote hosts. Nmap supports various scan types, such as [11] [12]:
    - TCP Connect Scan (-sT): Completes the full TCP handshake, revealing open, closed, and filtered ports.
    - SYN Scan (-sS): A stealthy scan that sends SYN packets without completing the TCP handshake, identifying open ports while remaining less detectable.
    - UDP Scan (-sU): Probes UDP services, which are often more difficult to scan but necessary to identify services like DNS and SNMP.

  - Xmas Scan (-sX): A specialized stealth scan setting multiple TCP flags to determine port states (open or filtered). Specifically, the FIN, PSH, and URG flags are set in the TCP packet header to manipulate how the target system responds. The FIN (Finish) flag signals that the sender has finished sending data, tricking the target that the connection is being closed, the PSH (Push) flag forces the transmission of data immediately, and the URG (Urgent) flag indicates that the data being sent is urgent and should be processed immediately, bypassing normal processing queues.
  - Null Scan (-sN): A scan that sends packets with no flags set, confusing some firewalls and intrusion detection systems, useful for identifying how targets handle unexpected traffic.

- **Shodan:** Shodan is a powerful search engine for identifying internet-connected devices such as servers, routers, cameras, and industrial control systems. Unlike traditional search engines, which index websites, Shodan indexes exposed services and devices by scanning the internet for open ports and fingerprinting their associated software or firmware. Shodan also highlights security weaknesses by providing details about vulnerabilities related to each device. In this assessment, Shodan was used to search for any exposed services or devices corresponding to the public IP address of the network [13].

- **nmap.online:** nmap.online is a web-based interface that enables users to run Nmap scans from external networks without requiring Nmap installation on a local machine. This can be particularly useful for assessing the security of a network from an external perspective, replicating how an attacker might scan for vulnerabilities from outside the target environment. In this assessment, nmap.online was used to scan the public IP address of the network, revealing open and filtered ports that could potentially be exploited [14].

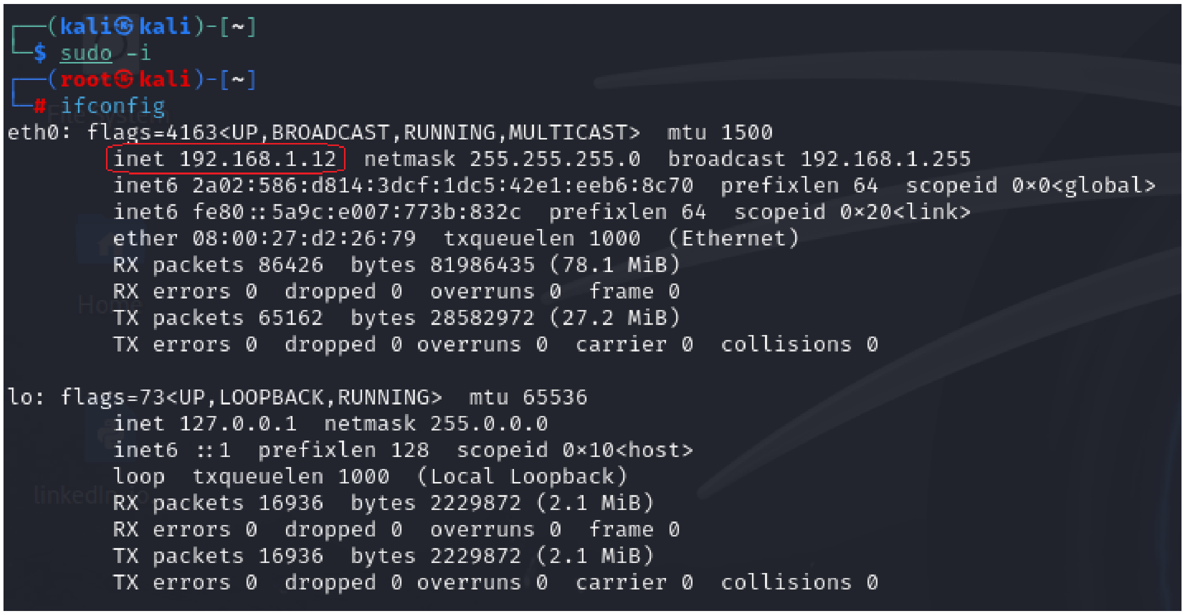
- **Wireshark:** Wireshark is a network protocol analyzer that captures and displays data traveling over a network in real-time. It allows users to dissect packets and analyze communication between devices, protocols used, and traffic patterns [15].

Findings

Discovering the home network

The process of discovering the home network began by identifying the internal and external IP addresses assigned to the devices within the network.

The first command executed was **ifconfig** on a Kali Linux system, which provided the internal IP address of the device. In this case, the IP address assigned was 192.168.1.12. This address, assigned by the router or set as a static IP and remains hidden from external networks being used only within the private local network.

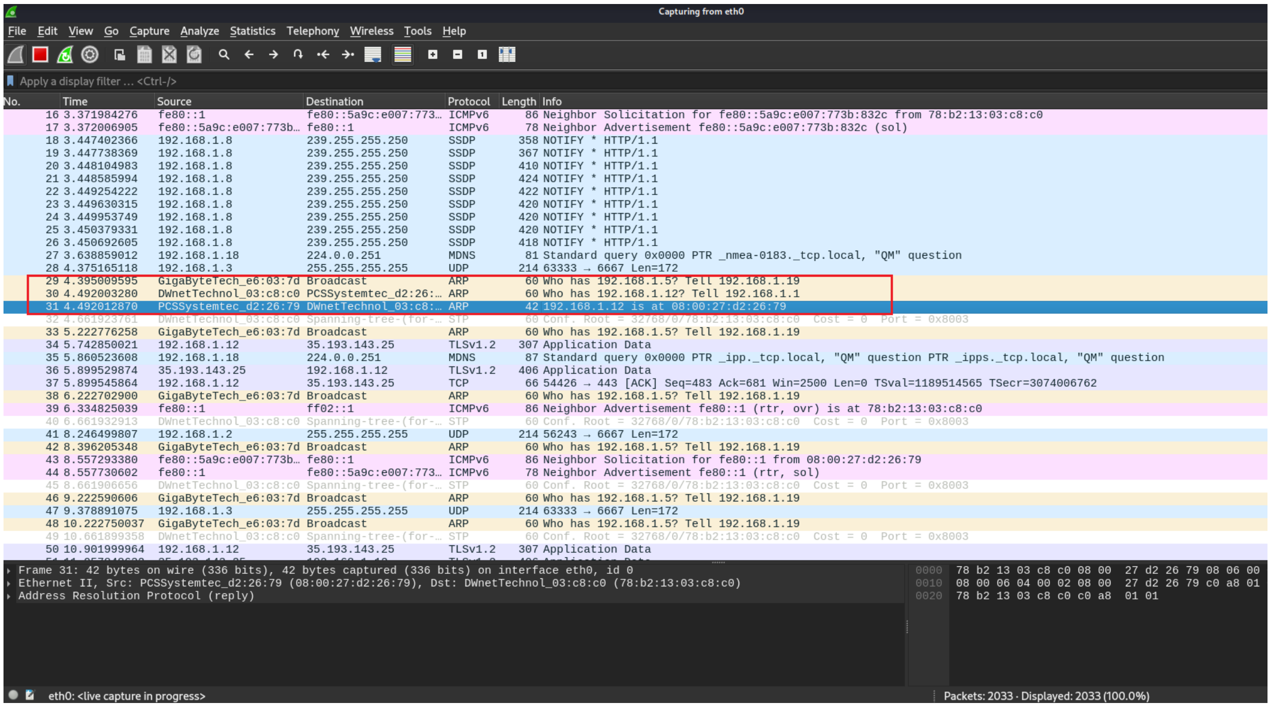


```
┌──(kali㉿kali)-[~]
└─$ sudo -i
┌──(root㉿kali)-[~]
└─# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.12  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 2a02:586:d814:3dcf:1dc5:42e1:eeb6:8c70  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::5a9c:e007:773b:832c  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:d2:26:79  txqueuelen 1000  (Ethernet)
        RX packets 86426  bytes 81986435 (78.1 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 65162  bytes 28582972 (27.2 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 16936  bytes 2229872 (2.1 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 16936  bytes 2229872 (2.1 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 1 –Command ifconfig in VM Kali Linux

By observing Wireshark in the VM it can be seen that the VM is sending broadcast traffic, ARP broadcasts, to resolve MAC addresses to IP addresses and detect devices on the local network.


Capturing from eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.  Time          Source                 Destination             Protocol Length Info
  16 3.371984276   fe80::1                fe80::5a9c:e007:773…    ICMPv6     86 Neighbor Solicitation for fe80::5a9c:e007:773b:832c from 78:b2:13:03:c8:c0
  17 3.372006905   fe80::5a9c:e007:773b…  fe80::1                 ICMPv6     78 Neighbor Advertisement fe80::5a9c:e007:773b:832c (sol)
  18 3.447402366   192.168.1.8            239.255.255.250         SSDP      358 NOTIFY * HTTP/1.1
  19 3.447738369   192.168.1.8            239.255.255.250         SSDP      367 NOTIFY * HTTP/1.1
  20 3.448104983   192.168.1.8            239.255.255.250         SSDP      410 NOTIFY * HTTP/1.1
  21 3.448585994   192.168.1.8            239.255.255.250         SSDP      424 NOTIFY * HTTP/1.1
  22 3.449254222   192.168.1.8            239.255.255.250         SSDP      422 NOTIFY * HTTP/1.1
  23 3.449630315   192.168.1.8            239.255.255.250         SSDP      420 NOTIFY * HTTP/1.1
  24 3.449953749   192.168.1.8            239.255.255.250         SSDP      420 NOTIFY * HTTP/1.1
  25 3.450379331   192.168.1.8            239.255.255.250         SSDP      420 NOTIFY * HTTP/1.1
  26 3.450692605   192.168.1.8            239.255.255.250         SSDP      418 NOTIFY * HTTP/1.1
  27 3.638859012   192.168.1.18           224.0.0.251             MDNS       81 Standard query 0x0000 PTR _nmea-0183._tcp.local, "QM" question
  28 4.375165118   192.168.1.3            255.255.255.255         UDP       214 63333 → 6667 Len=172
  29 4.395009595   GigaByteTech_e6:03:7d  Broadcast               ARP        60 Who has 192.168.1.5? Tell 192.168.1.19
  30 4.492003280   DWnetTechnol_03:c8:c0  PCSSystemtec_d2:26:…    ARP        60 Who has 192.168.1.12? Tell 192.168.1.1
  31 4.492012870   PCSSystemtec_d2:26:79  DWnetTechnol_03:c8:…    ARP        42 192.168.1.12 is at 08:00:27:d2:26:79
  32 4.661923761   DWnetTechnol_03:c8:c0  Spanning-tree-(for-…    STP        60 Conf. Root = 32768/0/78:b2:13:03:c8:c0  Cost = 0  Port = 0x8003
  33 5.222776258   GigaByteTech_e6:03:7d  Broadcast               ARP        60 Who has 192.168.1.5? Tell 192.168.1.19
  34 5.742850021   192.168.1.12           35.193.143.25           TLSv1.2   307 Application Data
  35 5.860523608   192.168.1.18           224.0.0.251             MDNS       87 Standard query 0x0000 PTR _ipp._tcp.local, "QM" question PTR _ipps._tcp.local, "QM" question
  36 5.899529874   35.193.143.25          192.168.1.12            TLSv1.2   406 Application Data
  37 5.899545864   192.168.1.12           35.193.143.25           TCP        66 54426 → 443 [ACK] Seq=483 Ack=681 Win=2500 Len=0 TSval=1189514565 TSecr=3074006762
  38 6.222702900   GigaByteTech_e6:03:7d  Broadcast               ARP        60 Who has 192.168.1.5? Tell 192.168.1.19
  39 6.334825039   fe80::1                ff02::1                 ICMPv6     86 Neighbor Advertisement fe80::1 (rtr, ovr) is at 78:b2:13:03:c8:c0
  40 6.661932913   DWnetTechnol_03:c8:c0  Spanning-tree-(for-…    STP        60 Conf. Root = 32768/0/78:b2:13:03:c8:c0  Cost = 0  Port = 0x8003
  41 8.246499807   192.168.1.2            255.255.255.255         UDP       214 56243 → 6667 Len=172
  42 8.396205348   GigaByteTech_e6:03:7d  Broadcast               ARP        60 Who has 192.168.1.5? Tell 192.168.1.19
  43 8.557293380   fe80::5a9c:e007:773b…  fe80::1                 ICMPv6     86 Neighbor Solicitation for fe80::1 from 08:00:27:d2:26:79
  44 8.557730602   fe80::1                fe80::5a9c:e007:773…    ICMPv6     78 Neighbor Advertisement fe80::1 (rtr, sol)
  45 8.661906656   DWnetTechnol_03:c8:c0  Spanning-tree-(for-…    STP        60 Conf. Root = 32768/0/78:b2:13:03:c8:c0  Cost = 0  Port = 0x8003
  46 9.222590606   GigaByteTech_e6:03:7d  Broadcast               ARP        60 Who has 192.168.1.5? Tell 192.168.1.19
  47 9.378891075   192.168.1.3            255.255.255.255         UDP       214 63333 → 6667 Len=172
  48 10.222750037  GigaByteTech_e6:03:7d  Broadcast               ARP        60 Who has 192.168.1.5? Tell 192.168.1.19
  49 10.661899358  DWnetTechnol_03:c8:c0  Spanning-tree-(for-…    STP        60 Conf. Root = 32768/0/78:b2:13:03:c8:c0  Cost = 0  Port = 0x8003
  50 10.901999964  192.168.1.12           35.193.143.25           TLSv1.2   307 Application Data

› Frame 31: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface eth0, id 0
› Ethernet II, Src: PCSSystemtec_d2:26:79 (08:00:27:d2:26:79), Dst: DWnetTechnol_03:c8:c0 (78:b2:13:03:c8:c0)
› Address Resolution Protocol (reply)

0000  78 b2 13 03 c8 c0 08 00  27 d2 26 79 08 06 00
0010  08 00 06 04 00 02 08 00  27 d2 26 79 c0 a8 01
0020  78 b2 13 03 c8 c0 c0 a8  01 01

eth0: <live capture in progress>                    Packets: 2033 · Displayed: 2033 (100.0%)
```

Figure 2 – Wireshark network traffic

Next, the external IP address (public IP) was discovered using the website https://whatismyipaddress.com. This address is assigned by the Internet Service Provider (ISP) and is visible to websites, online services, and other devices outside the local network. Unlike internal IP addresses, which are used for communication within a private network, external IPs are globally unique and allow devices to access and be accessed by internet resources.

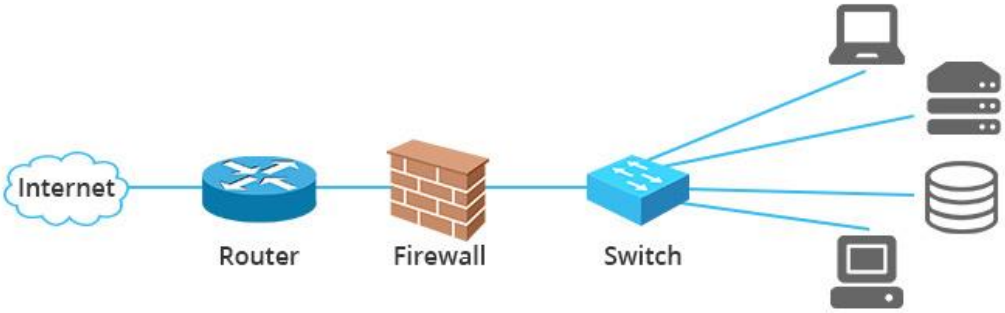
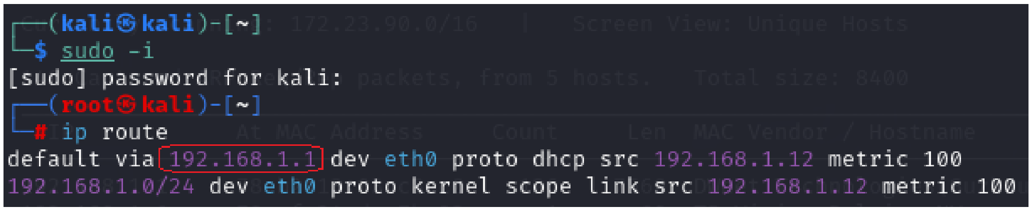


Figure 3 - How switch, router and firewall are connected in a network [16]

To identify the router's IP address, the command **ip route** was used. A router IP address (also known as the default gateway) is the local address assigned to the router within the network. It is the address that devices on the home network use in order to communicate with the router and, through it, connect to the wider internet. Common router IP addresses include:

192.168.0.1
192.168.1.1
10.0.0.1

These are private IP addresses as defined by the IPv4 address space and are used to manage the router's settings.



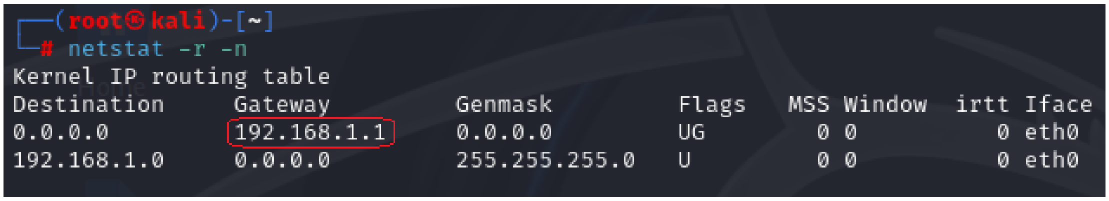
```
┌──(kali㉿kali)-[~]
└─$ sudo -i
[sudo] password for kali:
┌──(root㉿kali)-[~]
└─# ip route
default via 192.168.1.1 dev eth0 proto dhcp src 192.168.1.12 metric 100
192.168.1.0/24 dev eth0 proto kernel scope link src 192.168.1.12 metric 100
```

Figure 4 – Command ip route in Kali VM

Alternatively, **netstat -r -n** command can be used. This command is used to display the system's routing table in a numerical format.

Summary of Output:

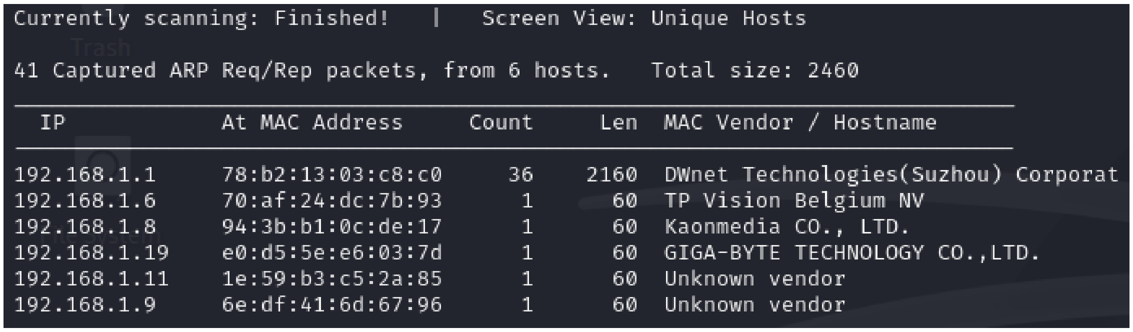
- **Destination:** The destination network or host.
- **Gateway:** The gateway through which the destination is reached (often the router).
- **Genmask:** The network mask for the destination.
- **Flags:** Indicates the route's status (e.g., U for up, G for gateway).
- **Iface:** The network interface associated with the route.



```
┌──(root㉿kali)-[~]
└─# netstat -r -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         192.168.1.1     0.0.0.0         UG        0 0          0 eth0
192.168.1.0     0.0.0.0         255.255.255.0   U         0 0          0 eth0
```

Figure 5 – Command netstat –r –n in Kali VM

Finally, to discover other live hosts on the internal network, the **netdiscover -r 192.168.1.0/24** command was executed. This command scanned the local network range, identifying all devices within the specified IP address range. Specifically, 192.168.1.0/24 was the subnet range to be scanned. The /24 part indicates a subnet mask of 255.255.255.0, meaning it will scan IP addresses from 192.168.1.1 to 192.168.1.254 within the local network. This is typically a default range for home or small office networks.

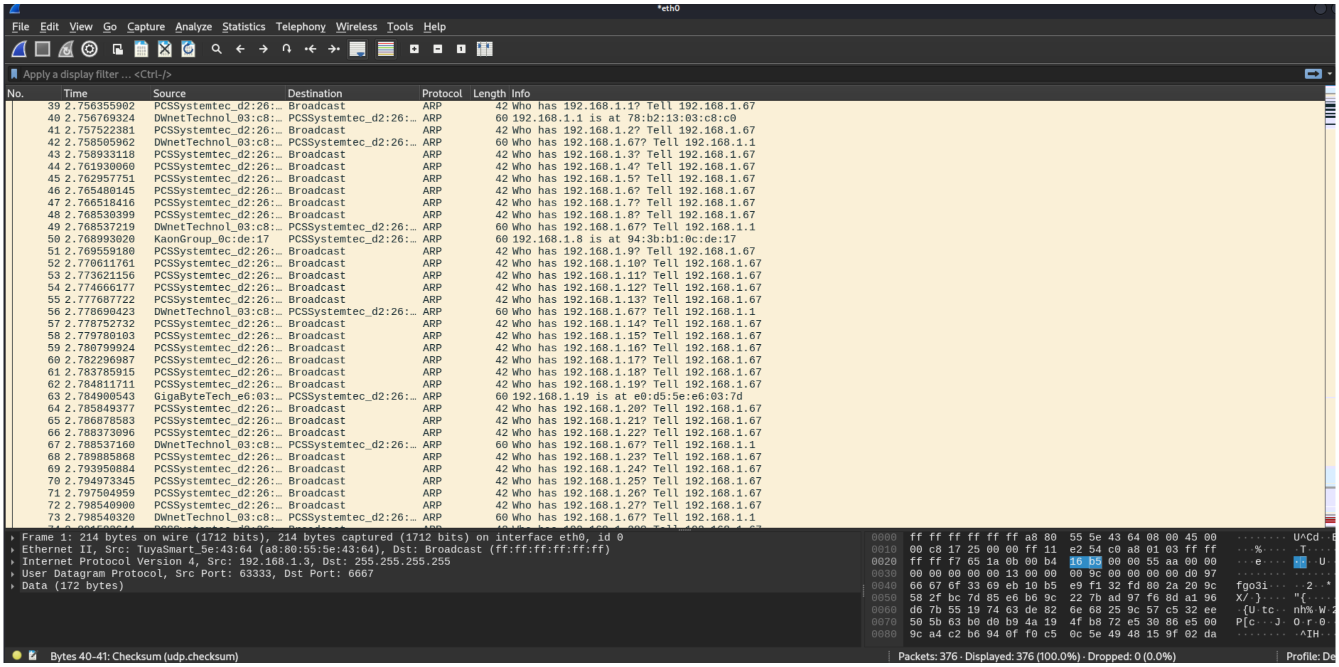


```
Currently scanning: Finished!   |   Screen View: Unique Hosts

41 Captured ARP Req/Rep packets, from 6 hosts.   Total size: 2460
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.1.1     78:b2:13:03:c8:c0     36    2160  DWnet Technologies(Suzhou) Corporat
192.168.1.6     70:af:24:dc:7b:93      1      60  TP Vision Belgium NV
192.168.1.8     94:3b:b1:0c:de:17      1      60  Kaonmedia CO., LTD.
192.168.1.19    e0:d5:5e:e6:03:7d      1      60  GIGA-BYTE TECHNOLOGY CO.,LTD.
192.168.1.11    1e:59:b3:c5:2a:85      1      60  Unknown vendor
192.168.1.9     6e:df:41:6d:67:96      1      60  Unknown vendor
```

Figure 6 – Command netdiscover -r 192.168.1.0/24

Again, the VM is sending broadcast traffic, ARP broadcasts, to resolve MAC addresses to IP addresses and detect devices on the local network.


*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.   Time         Source               Destination          Protocol Length Info
   39 2.756355902  PCSSystemtec_d2:26:… Broadcast            ARP        42 Who has 192.168.1.1? Tell 192.168.1.67
   40 2.756769324  DWnetTechnol_03:c8:… PCSSystemtec_d2:26:… ARP        60 192.168.1.1 is at 78:b2:13:03:c8:c0
   41 2.757522381  PCSSystemtec_d2:26:… Broadcast            ARP        42 Who has 192.168.1.2? Tell 192.168.1.67
   42 2.758505962  DWnetTechnol_03:c8:… PCSSystemtec_d2:26:… ARP        60 Who has 192.168.1.67? Tell 192.168.1.1
   43 2.758933118  PCSSystemtec_d2:26:… Broadcast            ARP        42 Who has 192.168.1.3? Tell 192.168.1.67
   44 2.761930060  PCSSystemtec_d2:26:… Broadcast            ARP        42 Who has 192.168.1.4? Tell 192.168.1.67
   45 2.762957751  PCSSystemtec_d2:26:… Broadcast            ARP        42 Who has 192.168.1.5? Tell 192.168.1.67
   46 2.765480145  PCSSystemtec_d2:26:… Broadcast            ARP        42 Who has 192.168.1.6? Tell 192.168.1.67
   47 2.766518416  PCSSystemtec_d2:26:… Broadcast            ARP        42 Who has 192.168.1.7? Tell 192.168.1.67
   48 2.768530399  PCSSystemtec_d2:26:… Broadcast            ARP        42 Who has 192.168.1.8? Tell 192.168.1.67
   49 2.768537219  DWnetTechnol_03:c8:… PCSSystemtec_d2:26:… ARP        60 Who has 192.168.1.67? Tell 192.168.1.1
   50 2.768993020  KaonGroup_0c:de:17   PCSSystemtec_d2:26:… ARP        60 192.168.1.8 is at 94:3b:b1:0c:de:17
   51 2.769559180  PCSSystemtec_d2:26:… Broadcast            ARP        42 Who has 192.168.1.9? Tell 192.168.1.67
   52 2.770611761  PCSSystemtec_d2:26:… Broadcast            ARP        42 Who has 192.168.1.10? Tell 192.168.1.67
   53 2.773621156  PCSSystemtec_d2:26:… Broadcast            ARP        42 Who has 192.168.1.11? Tell 192.168.1.67
   54 2.774666177  PCSSystemtec_d2:26:… Broadcast            ARP        42 Who has 192.168.1.12? Tell 192.168.1.67
   55 2.777687722  PCSSystemtec_d2:26:… Broadcast            ARP        42 Who has 192.168.1.13? Tell 192.168.1.67
   56 2.778690423  DWnetTechnol_03:c8:… PCSSystemtec_d2:26:… ARP        60 Who has 192.168.1.67? Tell 192.168.1.1
   57 2.778752732  PCSSystemtec_d2:26:… Broadcast            ARP        42 Who has 192.168.1.14? Tell 192.168.1.67
   58 2.779780103  PCSSystemtec_d2:26:… Broadcast            ARP        42 Who has 192.168.1.15? Tell 192.168.1.67
   59 2.780799924  PCSSystemtec_d2:26:… Broadcast            ARP        42 Who has 192.168.1.16? Tell 192.168.1.67
   60 2.782296987  PCSSystemtec_d2:26:… Broadcast            ARP        42 Who has 192.168.1.17? Tell 192.168.1.67
   61 2.783785915  PCSSystemtec_d2:26:… Broadcast            ARP        42 Who has 192.168.1.18? Tell 192.168.1.67
   62 2.784811711  PCSSystemtec_d2:26:… Broadcast            ARP        42 Who has 192.168.1.19? Tell 192.168.1.67
   63 2.784900543  GigaByteTech_e6:03:… PCSSystemtec_d2:26:… ARP        60 192.168.1.19 is at e0:d5:5e:e6:03:7d
   64 2.785849377  PCSSystemtec_d2:26:… Broadcast            ARP        42 Who has 192.168.1.20? Tell 192.168.1.67
   65 2.786878583  PCSSystemtec_d2:26:… Broadcast            ARP        42 Who has 192.168.1.21? Tell 192.168.1.67
   66 2.788373096  PCSSystemtec_d2:26:… Broadcast            ARP        42 Who has 192.168.1.22? Tell 192.168.1.67
   67 2.788537160  DWnetTechnol_03:c8:… PCSSystemtec_d2:26:… ARP        60 Who has 192.168.1.67? Tell 192.168.1.1
   68 2.789885868  PCSSystemtec_d2:26:… Broadcast            ARP        42 Who has 192.168.1.23? Tell 192.168.1.67
   69 2.793950884  PCSSystemtec_d2:26:… Broadcast            ARP        42 Who has 192.168.1.24? Tell 192.168.1.67
   70 2.794973345  PCSSystemtec_d2:26:… Broadcast            ARP        42 Who has 192.168.1.25? Tell 192.168.1.67
   71 2.797504959  PCSSystemtec_d2:26:… Broadcast            ARP        42 Who has 192.168.1.26? Tell 192.168.1.67
   72 2.798540900  PCSSystemtec_d2:26:… Broadcast            ARP        42 Who has 192.168.1.27? Tell 192.168.1.67
   73 2.798540320  DWnetTechnol_03:c8:… PCSSystemtec_d2:26:… ARP        60 Who has 192.168.1.67? Tell 192.168.1.1
```

```
› Frame 1: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits) on interface eth0, id 0
› Ethernet II, Src: TuyaSmart_5e:43:64 (a8:80:55:5e:43:64), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
› Internet Protocol Version 4, Src: 192.168.1.3, Dst: 255.255.255.255
› User Datagram Protocol, Src Port: 63333, Dst Port: 6667
› Data (172 bytes)
```

```
0000  ff ff ff ff ff ff a8 80  55 5e 43 64 08 00 45 00   ········ U^Cd··E
0010  00 c8 17 25 00 00 ff 11  e2 54 c0 a8 01 03 ff ff   ···%···· ·T·····
0020  ff ff f7 65 1a 0b 00 b4  16 b5 00 00 55 aa 00 00   ···e···· ····U···
0030  00 00 00 00 00 13 00 00  00 9c 00 00 00 00 d0 97   ········ ········
0040  66 67 6f 33 69 eb 10 b5  e9 f1 32 fd 80 2a 20 9c   fgo3i··· ··2··* ·
0050  58 2f bc 7d 85 e6 b6 9c  22 7b ad 97 f6 8d a1 96   X/·}···· "{······
0060  d6 7b 55 19 74 63 de 82  6e 68 25 9c 57 c5 32 ee   ·{U·tc·· nh%·W·2·
0070  50 5b 63 b0 d0 b9 4a 19  4f b8 72 e5 30 86 e5 00   P[c···J· O·r·0···
0080  9c a4 c2 b6 94 0f f0 c5  0c 5e 49 48 15 9f 02 da   ········ ·^IH····
```

```
Bytes 40-41: Checksum (udp.checksum)    Packets: 376 · Displayed: 376 (100.0%) · Dropped: 0 (0.0%)    Profile: De
```

Figure 7 – Wireshark network traffic for netdiscover

Use Cases:

- Identifying Unauthorized Devices: Helps network administrators spot devices that shouldn't be connected to the network.
- Preparing for Security Testing: Before launching more in-depth scans (like using nmap), netdiscover can provide an overview of the network.
- Low-Noise Scanning: Unlike some other network scanners, netdiscover is relatively low-noise, meaning it's less likely to trigger alarms on intrusion detection systems.

Internal Vulnerability Scanning

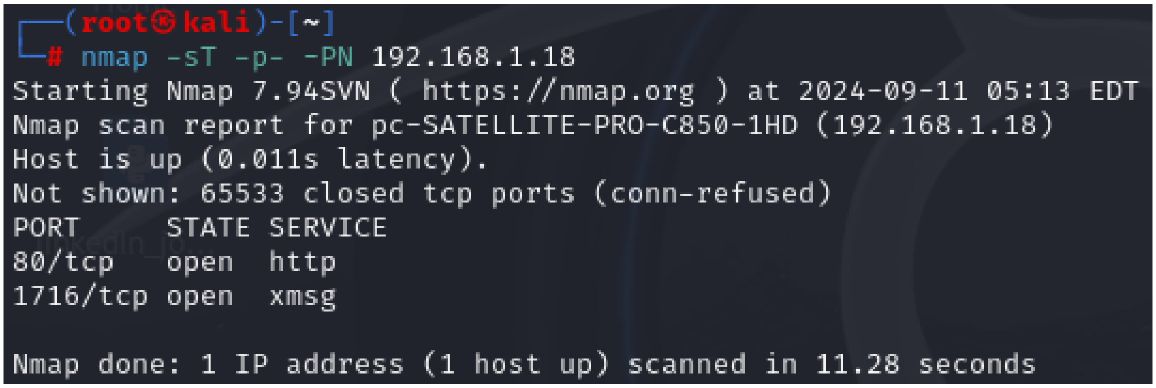
Port Scanning

TCP Connect Scan

During the internal vulnerability assessment, a TCP connect scan was executed from the VM against the host 192.168.1.18 (Ubuntu laptop) using the command **nmap -sT -p- -PN 192.168.1.18**.

The -sT flag initiates a full TCP connect scan, where a complete three-way handshake (SYN, SYN-ACK, ACK) is established with the target system for each open port. This method provides a reliable indication of open ports, but generates a larger number of network packets compared to stealthier scan types.

The -p- flag was utilized to scan all available TCP ports, while the -PN was argument was used to bypass the host discovery phase. As this scan involves full connection establishment, it tends to take longer to complete due to the higher number of interactions between the scanning and target systems.

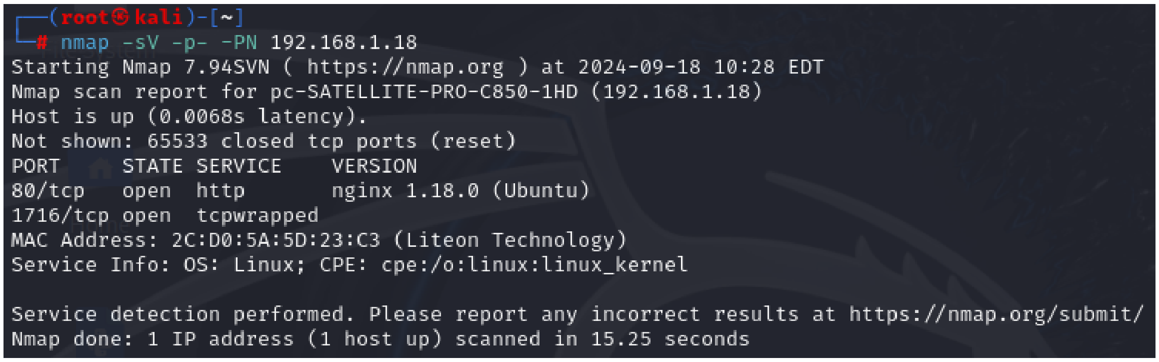


```
┌──(root㉿kali)-[~]
└─# nmap -sT -p- -PN 192.168.1.18
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-11 05:13 EDT
Nmap scan report for pc-SATELLITE-PRO-C850-1HD (192.168.1.18)
Host is up (0.011s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT     STATE SERVICE
80/tcp   open  http
1716/tcp open  xmsg

Nmap done: 1 IP address (1 host up) scanned in 11.28 seconds
```

Figure 8 – TCP Connect scan using nmap

Since more info is needed about the services that use that ports, another command can also be used **nmap -sV -p- -PN 192.168.1.18.** The -sV argument enables the service version detection.

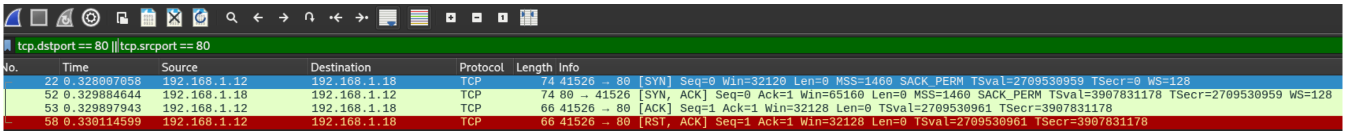


```
┌──(root㉿kali)-[~]
└─# nmap -sV -p- -PN 192.168.1.18
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-18 10:28 EDT
Nmap scan report for pc-SATELLITE-PRO-C850-1HD (192.168.1.18)
Host is up (0.0068s latency).
Not shown: 65533 closed tcp ports (reset)
PORT     STATE SERVICE    VERSION
80/tcp   open  http       nginx 1.18.0 (Ubuntu)
1716/tcp open  tcpwrapped
MAC Address: 2C:D0:5A:5D:23:C3 (Liteon Technology)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 15.25 seconds
```

Figure 9 – TCP Connect scan including service version detection

Wireshark observation example (port 80 TCP connection):



tcp.dstport == 80 || tcp.srcport == 80

No.	Time	Source	Destination	Protocol	Length	Info
22	0.328007058	192.168.1.12	192.168.1.18	TCP	74	41526 → 80 [SYN] Seq=0 Win=32120 Len=0 MSS=1460 SACK_PERM TSval=2709530959 TSecr=0 WS=128
52	0.329884644	192.168.1.18	192.168.1.12	TCP	74	80 → 41526 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=3907831178 TSecr=2709530959 WS=128
53	0.329897943	192.168.1.12	192.168.1.18	TCP	66	41526 → 80 [ACK] Seq=1 Ack=1 Win=32128 Len=0 TSval=2709530961 TSecr=3907831178
58	0.330114599	192.168.1.12	192.168.1.18	TCP	66	41526 → 80 [RST, ACK] Seq=1 Ack=1 Win=32128 Len=0 TSval=2709530961 TSecr=3907831178

Figure 10 – Wireshark traffic during TCP scan

The Nmap scan results for the host 192.168.1.18 reveal the following findings:

Host Status: The host is active, with a latency of 0.068 seconds, indicating it is responding to network requests.

Open Ports:

Port 80/tcp (HTTP): The HTTP service is running, typically indicating a web server is accessible via this port. It may host a website or some other web-based service on this device. In this particular case it is indicated that the device is running an Nginx web server version 1.18.0 on Ubuntu.

Port 1716/tcp (Xmsg): This port is open and commonly associated with the Xmsg service. It may relate to messaging or a custom application. Upon further investigation, it appears that the service running on port 1716 is likely kdeconnectd, a daemon associated with KDE Connect. This application facilitates communication between devices, including PCs and Android phones, by enabling functionalities such as file sharing, remote control, and notifications across the connected devices.

Security Implications**:**

There are known vulnerabilities associated with nginx 1.18.0 on Ubuntu:

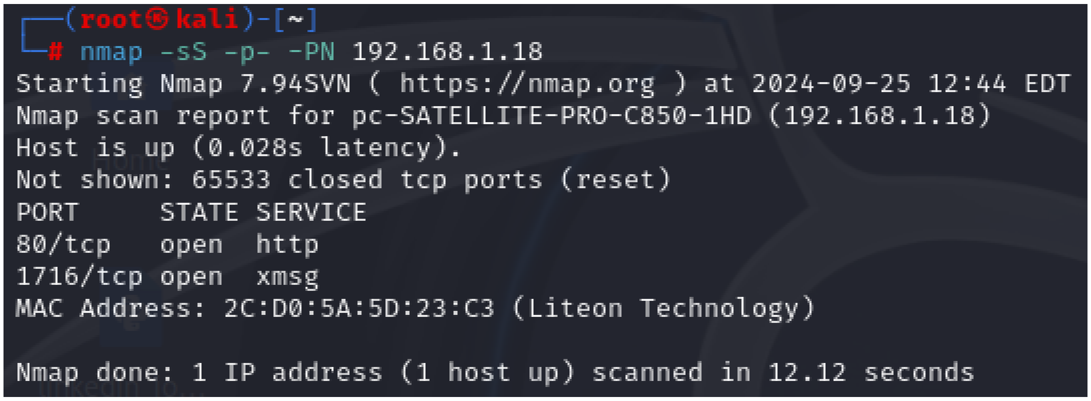
For example, CVE-2021-23017 highlights a critical vulnerability related to DNS resolver handling, which could be exploited remotely by attackers to execute arbitrary code. This specific vulnerability affects all Nginx versions up to 1.18.0, and a patch is available to address the issue [17].

To mitigate this kind of vulnerabilities, it is imperative that the nginx installation is regularly updated, or upgraded to a more secure version.

Port 1716/tcp: KDE Connect's background service, kdeconnectd, has had known vulnerabilities, such as CVE-2020-26164, which allows an attacker on the same local network to send crafted packets to overwhelm the system, causing excessive CPU, memory, or network slot usage—essentially a Denial of Service (DoS) attack. This vulnerability has been patched in later versions, but it's important to ensure that the system is updated to avoid such issues [18].

SYN Scan

A SYN scan was performed on the same host using the command **nmap -sS -p- -PN 192.168.1.18**. This scan is considered stealthier than the TCP connect scan, only completing half the TCP connection (a reset packet is being sent instead of an ACK packet). It also allows clear, reliable differentiation between the open, closed, and filtered state of the ports.



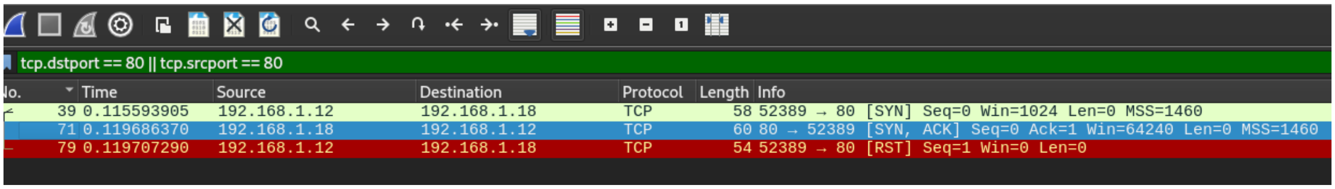
```
┌──(root㉿kali)-[~]
└─# nmap -sS -p- -PN 192.168.1.18
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-25 12:44 EDT
Nmap scan report for pc-SATELLITE-PRO-C850-1HD (192.168.1.18)
Host is up (0.028s latency).
Not shown: 65533 closed tcp ports (reset)
PORT     STATE SERVICE
80/tcp   open  http
1716/tcp open  xmsg
MAC Address: 2C:D0:5A:5D:23:C3 (Liteon Technology)

Nmap done: 1 IP address (1 host up) scanned in 12.12 seconds
```

Figure 11 – SYN Scan using nmap

In this particular case, the findings were exactly the same as the previous scanning.

Wireshark observation example (port 80 TCP connection):

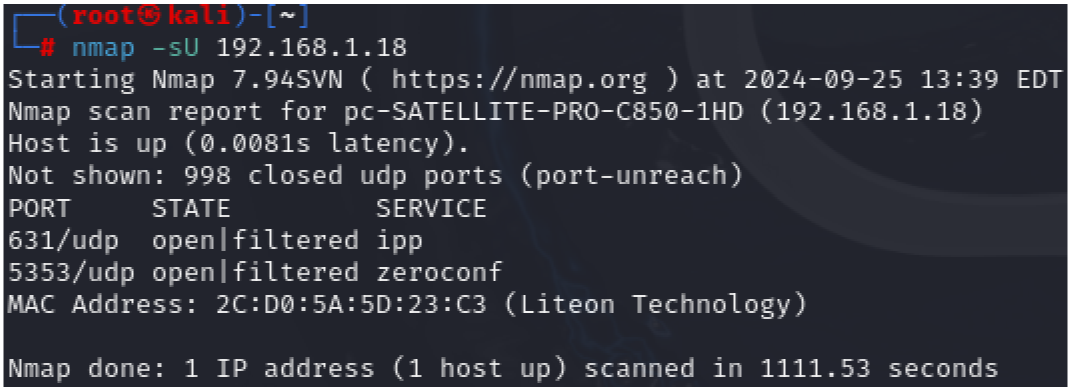


```
tcp.dstport == 80 || tcp.srcport == 80
No.      Time         Source        Destination   Protocol Length Info
      39 0.115593905  192.168.1.12  192.168.1.18  TCP        58 52389 → 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
      71 0.119686370  192.168.1.18  192.168.1.12  TCP        60 80 → 52389 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
      79 0.119707290  192.168.1.12  192.168.1.18  TCP        54 52389 → 80 [RST] Seq=1 Win=0 Len=0
```

Figure 12 – Wireshark traffic during SYN scan

UDP Scan

Following the TCP Connect and SYN scans, a UDP scan was performed using the command **nmap -sU 192.168.1.18** (-sU for UDP scan). This type of scan generally takes more time compared to the previous scans, as open and filtered UDP ports rarely send responses. Consequently, nmap may either time out or receive numerous ICMP error packets from closed ports, causing it to slow down the scanning process. This is done to avoid overwhelming the network with excessive packets that do not yield useful information.

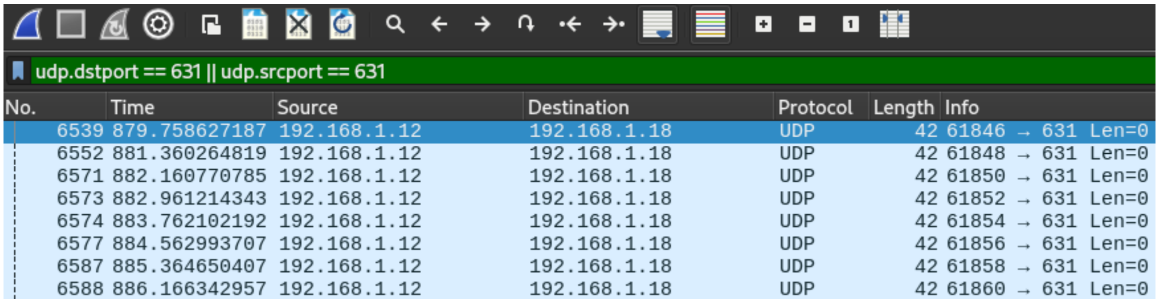


```
┌──(root㉿kali)-[~]
└─# nmap -sU 192.168.1.18
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-25 13:39 EDT
Nmap scan report for pc-SATELLITE-PRO-C850-1HD (192.168.1.18)
Host is up (0.0081s latency).
Not shown: 998 closed udp ports (port-unreach)
PORT     STATE         SERVICE
631/udp  open|filtered ipp
5353/udp open|filtered zeroconf
MAC Address: 2C:D0:5A:5D:23:C3 (Liteon Technology)

Nmap done: 1 IP address (1 host up) scanned in 1111.53 seconds
```

Figure 13 – UDP scan using nmap

Wireshark observation example (port 631 UDP connection):



```
udp.dstport == 631 || udp.srcport == 631
No.   Time          Source        Destination   Protocol Length Info
 6539 879.758627187 192.168.1.12  192.168.1.18  UDP          42 61846 → 631 Len=0
 6552 881.360264819 192.168.1.12  192.168.1.18  UDP          42 61848 → 631 Len=0
 6571 882.160770785 192.168.1.12  192.168.1.18  UDP          42 61850 → 631 Len=0
 6573 882.961214343 192.168.1.12  192.168.1.18  UDP          42 61852 → 631 Len=0
 6574 883.762102192 192.168.1.12  192.168.1.18  UDP          42 61854 → 631 Len=0
 6577 884.562993707 192.168.1.12  192.168.1.18  UDP          42 61856 → 631 Len=0
 6587 885.364650407 192.168.1.12  192.168.1.18  UDP          42 61858 → 631 Len=0
 6588 886.166342957 192.168.1.12  192.168.1.18  UDP          42 61860 → 631 Len=0
```

Figure 14 – Wireshark traffic during UPD scan

Open Ports:

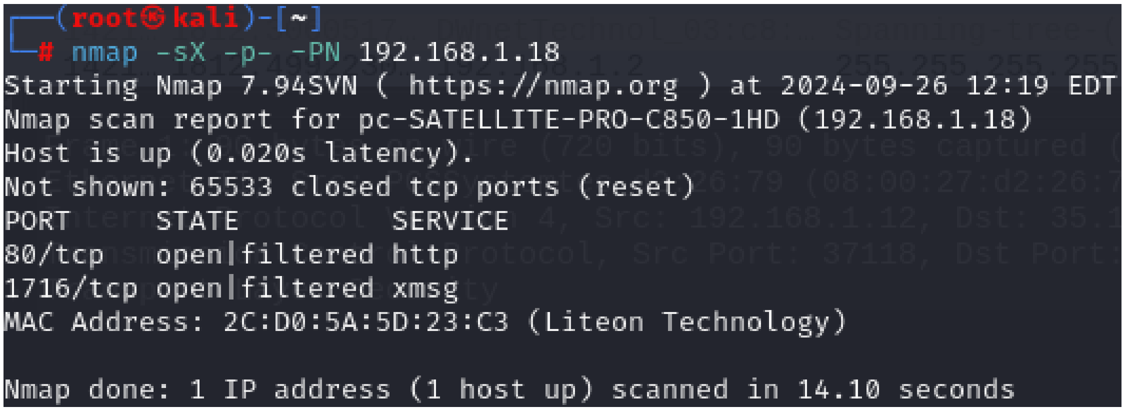
Port 631/udp: This port is associated with the Internet Printing Protocol (IPP), which is commonly used for networked printers. The status "open|filtered" suggests that nmap was unable to definitively determine whether the port is open or filtered by a firewall. This ambiguity occurs because UDP does not require acknowledgment packets, making it difficult for nmap to receive a definitive response [19].

Port 5353/udp: This port is typically used by the mDNS (Multicast DNS) service, also known as Zeroconf, which enables device discovery on local networks without the need for configuration. Like port 631, the "open|filtered" state indicates that nmap could not fully confirm whether the port is open or being filtered by a security mechanism [20].

Both open|filtered ports may be subject to DoS (Denial of Service) or information leakage attacks, especially if not properly secured. It is essential to review the configurations of these services, apply the latest patches, and restrict external access to reduce potential exposure to threats.

Xmas Scan

After conducting the TCP Connect, SYN, and UDP scans, an Xmas scan was executed to further assess the target's network state using the command **nmap -sX -p- -PN 192.168.1.18.** This type of scan, known for its stealth characteristics, sets the FIN, PSH, and URG flags, which typically elicit more detailed information about the target system. The Xmas scan is considered even more discrete than the SYN scan, as it avoids establishing a full TCP connection.



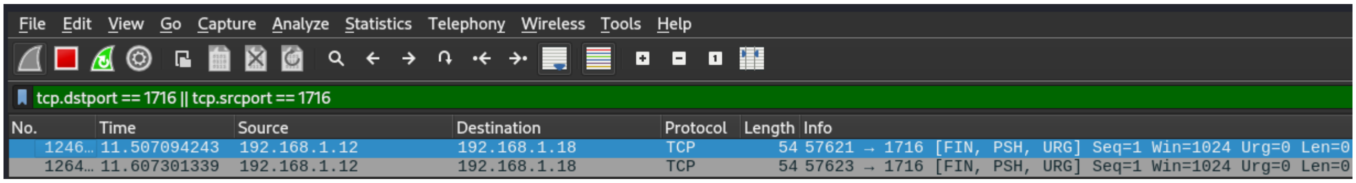
```
┌──(root㉿kali)-[~]
└─# nmap -sX -p- -PN 192.168.1.18
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-26 12:19 EDT
Nmap scan report for pc-SATELLITE-PRO-C850-1HD (192.168.1.18)
Host is up (0.020s latency).
Not shown: 65533 closed tcp ports (reset)
PORT       STATE          SERVICE
80/tcp     open|filtered  http
1716/tcp   open|filtered  xmsg
MAC Address: 2C:D0:5A:5D:23:C3 (Liteon Technology)

Nmap done: 1 IP address (1 host up) scanned in 14.10 seconds
```

Figure 15 – Xmas scan using nmap

65533 TCP ports are closed: These ports sent a TCP reset packet (RST), indicating that they are not open and actively rejecting connections.

Wireshark observation example (port 1716 TCP connection):



```
tcp.dstport == 1716 || tcp.srcport == 1716

No.     Time          Source         Destination    Protocol Length Info
1246…   11.507094243  192.168.1.12   192.168.1.18   TCP      54 57621 → 1716 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
1264…   11.607301339  192.168.1.12   192.168.1.18   TCP      54 57623 → 1716 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
```

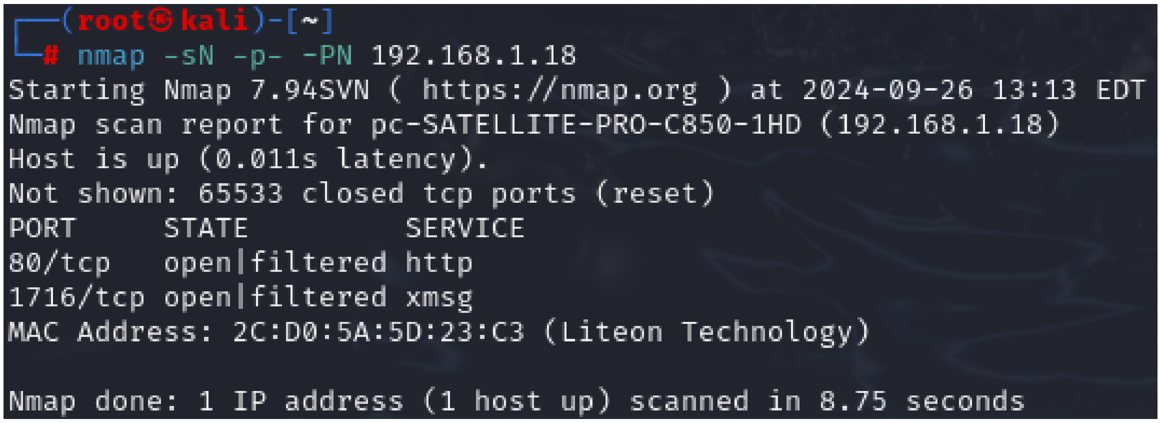
Figure 16 – Wireshark traffic during Xmas scan

80/tcp and 1716/tcp are marked as open|filtered. The scan did not retrieve detailed version information for the services running on these ports, likely due to the open|filtered state. This could be a result of firewalls or filtering mechanisms that block deeper probes or responses. More information was retrieved in the previous scans.

Null Scan

To conclude the internal vulnerability assessment, a Null scan was conducted using the command **nmap -sN -p- -PN 192.168.1.18**. The Null scan does not set any flags in the TCP header (i.e., the TCP flag header is 0). This lack of flags creates a situation where the target system is uncertain how to process the incoming packet. If the scanned port is open, no response is returned, as the system does not acknowledge the packet. However, if the port is closed, the system sends a TCP reset (RST) packet as a response, indicating the port is not accepting connections.

This technique provides a stealthier approach compared to traditional scans, as the absence of flags reduces the likelihood of detection by some firewalls or intrusion detection systems. However, it is generally more effective on Unix-based systems, as Windows systems tend to respond differently, making it less useful in those environments.

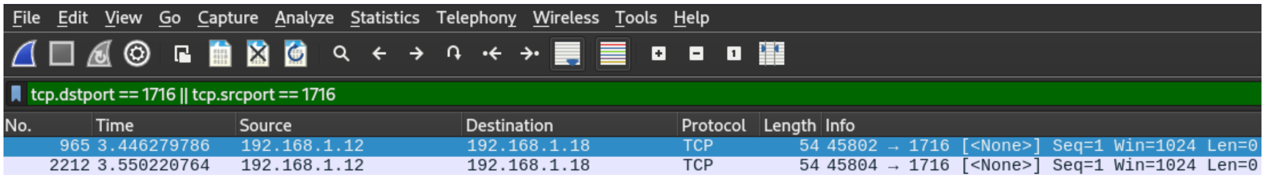


```
┌──(root㉿kali)-[~]
└─# nmap -sN -p- -PN 192.168.1.18
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-26 13:13 EDT
Nmap scan report for pc-SATELLITE-PRO-C850-1HD (192.168.1.18)
Host is up (0.011s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE         SERVICE
80/tcp    open|filtered http
1716/tcp  open|filtered xmsg
MAC Address: 2C:D0:5A:5D:23:C3 (Liteon Technology)

Nmap done: 1 IP address (1 host up) scanned in 8.75 seconds
```

Figure 17 – Null scan using nmap

Wireshark observation example (port 1716 TCP connection):


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
tcp.dstport == 1716 || tcp.srcport == 1716
```

```
No.   Time         Source        Destination   Protocol Length Info
  965 3.446279786  192.168.1.12  192.168.1.18  TCP          54 45802 → 1716 [<None>] Seq=1 Win=1024 Len=0
 2212 3.550220764  192.168.1.12  192.168.1.18  TCP          54 45804 → 1716 [<None>] Seq=1 Win=1024 Len=0
```

Figure 18 – Wireshark traffic during Null scan

Again the results were the same as the previous scan. Since there was no response, the port may probably be open.

External Vulnerability Scanning

For the external vulnerability scanning, the following process was conducted:

Shodan, a search engine specifically designed for internet-connected devices, was employed to assess any vulnerabilities corresponding to the findings from the previous scans. Various search filters were applied, such as port:1716, net:<public_IP>, and city:Volos, in an attempt to identify devices associated with the public IP address. However, no connected devices associated with the network were found on Shodan, suggesting that none of the devices were exposed to the wider internet or were not indexed by the search engine.

Additionally, external scans were conducted using nmap.online and the Nmap tool from a remote network to assess the public IP address further. During these scans, no open ports were found which suggests that all unnecessary services are either disabled or properly secured, reducing the attack surface.

Conclusion

The internal and external vulnerability assessments conducted on the local network and public-facing IP address revealed several important insights. Internally, various Nmap scans—such as TCP Connect, SYN, Xmas, Null, and UDP scans—provided valuable data about open, filtered, and closed ports. The primary host, identified as 192.168.1.18, had services running on ports such as 80/tcp (HTTP via nginx 1.18.0) and 1716/tcp (likely linked to KDE Connect), both of which were open or filtered. These services, while functional, require thorough monitoring and potential patching due to the known vulnerabilities, especially with nginx 1.18.0.

Externally, Shodan scans returned no indexed devices related to the public IP, suggesting that none of the internal devices were exposed to the internet. Moreover, further external Nmap scans conducted indicating the absence of open ports on the public, which suggests that the security posture of the network is enhanced.

The findings highlight several security concerns that that are primarily within the local network and require prompt action, such as securing open ports, verifying firewall configurations, and applying patches to potentially vulnerable services like nginx. While no significant external threats were detected, maintaining continuous monitoring and routine vulnerability assessments is essential to protect the internal network from risks posed by connected devices and unpatched services.

References

[1] J. Weiss, "Vulnerabilities by Common Ports Dashboard," Tenable, 15 September 2022. [Online]. Available: https://www.tenable.com/blog/vulnerabilities-by-common-ports-dashboard. [Accessed 1 October 2024].

[2] Cypress Data Defense, "The Impact of Security Misconfiguration and Its Mitigation," Cypress Data Defense, 29 April 2020. [Online]. Available: https://cypressdatadefense.com/blog/impact-of-security-misconfiguration/. [Accessed 1 October 2024].

[3] T. Jackins, "Risks & Vulnerabilities of Unpatched Software," splashtop, 28 August 2024. [Online]. Available: https://www.splashtop.com/blog/risks-and-vulnerabilities-of-unpatched-software. [Accessed 1 October 2024].

[4] ThreatMon Team, "What is Weak SSL Algorithms?," ThreatMon, 21 June 2024. [Online]. Available: https://threatmon.io/blog/what-is-weak-ssl-algorithms/. [Accessed 1 October 2024].

[5] Picus Labs , "What Is External Attack Surface Management (EASM) ?," Picus Security, 27 December 2023. [Online]. Available: https://www.picussecurity.com/resource/glossary/what-is-external-attack-surface-management. [Accessed 1 October 2024].

[6] J. Martinez, "11 Common Authentication Vulnerabilities You Need to Know," strongdm, 30 September 2024. [Online]. Available: https://www.strongdm.com/blog/authentication-vulnerabilities. [Accessed 1 October 2024].

[7] L. Yacono, "Monitoring for Suspicious Network Activity: Key Tips to Secure Your Network," CIMCOR, 29 December 2022. [Online]. Available: https://www.cimcor.com/blog/monitoring-for-suspicious-network-activity. [Accessed 1 October 2024].

[8] C. Klein, "7 Common Security Misconfigurations and How to Avoid Them," Jit, 17 September 2024. [Online]. Available: https://www.jit.io/resources/app-security/common-security-misconfigurations-and-how-to-avoid-them. [Accessed 1 October 2024].

[9] A. Khadka, "Guide to Linux netstat Command With Examples," Baeldung, 15 July 2024. [Online]. Available: https://www.baeldung.com/linux/netstat-command-tutorial. [Accessed 1 October 2024].

[10] Kali, "netdiscover," KALI, 23 May 2024. [Online]. Available: https://www.kali.org/tools/netdiscover/. [Accessed 1 October 2024].

[11] nmap, "Chapter 1. Getting Started with Nmap," nmap.org, [Online]. Available: https://nmap.org/book/intro.html. [Accessed 1 October 2024].

[12] nmap, "Chapter 5. Port Scanning Techniques and Algorithms," nmap.org, [Online]. Available: https://nmap.org/book/scan-methods.html. [Accessed 1 October 2024].

[13] E. Drosopoulou, "Shodan: A Peek into the Internet's Forbidden Zone," Java Code Geeks, 11 September 2024. [Online]. Available: https://www.javacodegeeks.com/2024/09/shodan-a-peek-into-the-internets-forbidden-zone.html. [Accessed 1 October 2024].

[14] Nmap Online, "Nmap Online - Web Based Nmap Scanner," Nmap Online, 2024. [Online]. Available: https://nmap.online/. [Accessed 1 October 2024].

[15] S. Gautam, "What is Wireshark? Applications, Features & How It Works," KnowledgeHut, 5 September 2023. [Online]. Available: https://www.knowledgehut.com/blog/security/what-is-wireshark. [Accessed 1 October 2024].

[16] "Network Switch vs Network Router vs Network Firewall," FS, [Online]. Available: https://community.fs.com/article/network-switch-router-firewall-why-need-all-three.html. [Accessed 1 October 2024].

[17] Ubuntu, "CVE-2021-23017," Canonical Ubuntu, 24 May 2021. [Online]. Available: https://ubuntu.com/security/CVE-2021-23017. [Accessed 1 October 2024].

[18] "CVE-2020-26164," cve.mitre, 2020. [Online]. Available: https://cve.mitre.org/cgi-bin/cvename.cgi?name=2020-26164. [Accessed 1 October 2024].

[19] E. Mualem and L. Shelly, "Analyzing the Latest CUPS RCE Vulnerability: Threats and Mitigations," UpWind, 27 September 2024. [Online]. Available: https://www.upwind.io/feed/analyzing-the-latest-cups-rce-vulnerability-threats-and-mitigations. [Accessed 1 October 2024].

[20] CQR, "Multicast Domain Name System (MDNS)," CQR, 6 April 2023. [Online]. Available: https://cqr.company/wiki/protocols/multicast-domain-name-system/. [Accessed 1 October 2024].

Table of Figures

Figure 1 – ifconfig in VM Kali Linux 14
Figure 2 – Wireshark network traffic 15
Figure 3 - How switch, router and firewall are connected in a network 15
Figure 4 – Command ip route in Kali VM 16
Figure 5 – Command netstat –r –n in Kali VM 17
Figure 6 – Command netdiscover -r 192.168.1.0/24 17
Figure 7 – Wireshark network traffic for netdiscover 18
Figure 8 – TCP Connect scan using nmap 19
Figure 9 – TCP Connect scan including service version detection 20
Figure 10 – Wireshark traffic during TCP scan 20
Figure 11 – SYN Scan using nmap 21
Figure 12 – Wireshark traffic during SYN scan 22
Figure 13 – UDP scan using nmap 22
Figure 14 – Wireshark traffic during UPD scan 23
Figure 15 – Xmas scan using nmap 24
Figure 16 – Wireshark traffic during Xmas scan 24
Figure 17 – Null scan using nmap 25
Figure 18 – Wireshark traffic during Null scan 25