Network Environment Analysis Report

Table of Contents

1. Introduction

This project aimed to perform a comprehensive network analysis within a controlled lab environment, focusing on identifying active devices, open ports, and running services. The process involved utilizing Nmap for detailed network scanning and Wireshark for traffic analysis to validate findings and gain deeper insights into the network's behavior.

An initial network-wide scan was conducted to map out all active devices within the network, followed by more intense, targeted scans on each device. This included full TCP port scans, as well as combined TCP and UDP scans to ensure thorough detection of all services. The scans provided key information such as operating system identification, open ports, and service versions, helping to assess potential vulnerabilities.

Wireshark was employed to capture network traffic during these scans, allowing for real-time analysis of device responses and communication patterns. The findings were meticulously documented, cross-referenced with known configurations, and compiled into a final report. The overall effort provided a complete picture of the lab network's structure and highlighted areas for potential security improvements.

2. Network Devices Information

2.1 Device Overview

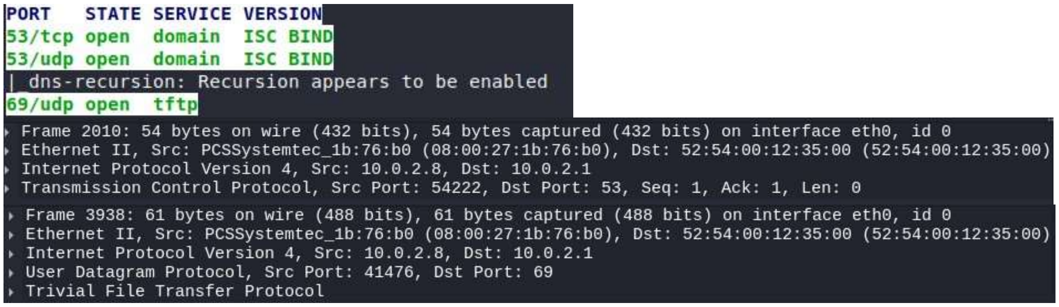
The following table summarizes the key attributes of the devices discovered within the lab network:

Machine Designation	Host Name	IP Address	MAC Address	Operating System	Open Ports
Network Gateway	Network Gateway	10.0.2.1	52:54:00:12:35:00	GrandStream GXP1105 VoIP phone	53/tcp, 69/udp
Windows11	WINDOWS11-DESKT	10.0.2.4	08:00:27:CB:20:4A	Windows 11	80/tcp, 7680/tcp
LinuxServer1	LinuxServer	10.0.2.15	08:00:27:DD:D8:F8	Linux 4.15 - 5.6	80/tcp, 3306/tcp
KaliOpenVas	Kali	10.0.2.15	08:00:27:1B:76:B0	Linux	None

2.2 Detailed Device Information

2.2.1 Network Gateway

- **Host Name:** QEMU virtual NIC (indicating a virtualized environment, likely running on a virtual machine).
- **IP Address:** 10.0.2.1
- **MAC Address:** 52:54:00:12:35:00
- **Operating System:** No exact OS matches found
- **Open Ports and Services**
  - **53 (TCP and UDP):** DNS with service ISC BIND
  - **69 (UDP):** TFTP (Trivial File Transfer Protocol)
- **ARP Ping Scan Elapsed Time:** 0.05s
- **OSI Layer Headers:**
  - **IP Address (10.0.2.1):** Layer 3**(**Internet Protocol Version 4**)**
  - **MAC Address (52:54:00:12:35:00):** Layer 2(Ethernet II)
  - **Port Numbers (53,69):** Layer 4 (TCP, UDP)



```
PORT    STATE SERVICE VERSION
53/tcp open   domain  ISC BIND
53/udp open   domain  ISC BIND
| dns-recursion: Recursion appears to be enabled
69/udp open   tftp
```

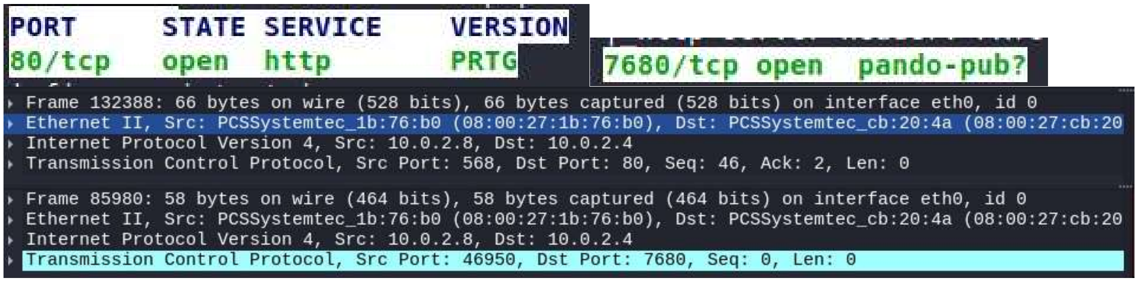
```
▸ Frame 2010: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface eth0, id 0
▸ Ethernet II, Src: PCSSystemtec_1b:76:b0 (08:00:27:1b:76:b0), Dst: 52:54:00:12:35:00 (52:54:00:12:35:00)
▸ Internet Protocol Version 4, Src: 10.0.2.8, Dst: 10.0.2.1
▸ Transmission Control Protocol, Src Port: 54222, Dst Port: 53, Seq: 1, Ack: 1, Len: 0
```

```
▸ Frame 3938: 61 bytes on wire (488 bits), 61 bytes captured (488 bits) on interface eth0, id 0
▸ Ethernet II, Src: PCSSystemtec_1b:76:b0 (08:00:27:1b:76:b0), Dst: 52:54:00:12:35:00 (52:54:00:12:35:00)
▸ Internet Protocol Version 4, Src: 10.0.2.8, Dst: 10.0.2.1
▸ User Datagram Protocol, Src Port: 41476, Dst Port: 69
▸ Trivial File Transfer Protocol
```

2.2.2 Windows11

- **Host Name:** WINDOWS11-DESKT
- **IP Address:** 10.0.2.4
- **MAC Address:** 08:00:27:CB:20:4A (Oracle VirtualBox virtual NIC)

- **Operating System:** Microsoft Windows 11
- **Open Ports and Services:**
  - **80 (TCP):** Microsoft HTTP PRTG
  - **7680(TCP):** Pando Media Booster(pando-pub)?
- **ARP Ping Scan Elapsed Time:** 0.09s
- **OSI Layer Headers:**
  - **IP Address (10.0.2.4):** Layer 3(Internet Protocol Version 4)
  - **MAC Address (08:00:27:CB:20:4A):** Layer 2**(**Ethernet II**)**
  - **Port Numbers (80,7680):** Layer 4 (TCP)



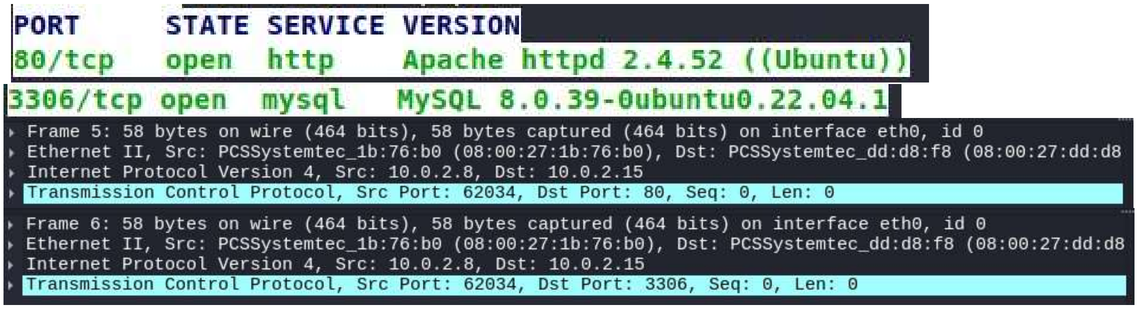
```
PORT     STATE SERVICE     VERSION
80/tcp   open  http        PRTG
7680/tcp open  pando-pub?
```

```
Frame 132388: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface eth0, id 0
Ethernet II, Src: PCSSystemtec_1b:76:b0 (08:00:27:1b:76:b0), Dst: PCSSystemtec_cb:20:4a (08:00:27:cb:20
Internet Protocol Version 4, Src: 10.0.2.8, Dst: 10.0.2.4
Transmission Control Protocol, Src Port: 568, Dst Port: 80, Seq: 46, Ack: 2, Len: 0

Frame 85980: 58 bytes on wire (464 bits), 58 bytes captured (464 bits) on interface eth0, id 0
Ethernet II, Src: PCSSystemtec_1b:76:b0 (08:00:27:1b:76:b0), Dst: PCSSystemtec_cb:20:4a (08:00:27:cb:20
Internet Protocol Version 4, Src: 10.0.2.8, Dst: 10.0.2.4
Transmission Control Protocol, Src Port: 46950, Dst Port: 7680, Seq: 0, Len: 0
```

2.2.2 Linux Server

- **Host Name:** LINUX-SERVER01
- **IP Address:** 10.0.2.15
- **MAC Address:** 08:00:27:DD:D8:F8 (Oracle VirtualBox virtual NIC)
- **Operating System:** Linux 4.15 - 5.6
- **Open Ports and Services:**
  - **80 (HTTP):** Apache HTTP Server 2.4.52 (Ubuntu)
  - **3306 (MySQL):** MySQL 8.0.39-0ubuntu0.22.04.1
- **ARP Ping Scan Elapsed Time:** 0.04s
- **OSI Layer Headers:**
  - **IP Address (10.0.2.15):** Layer 3(Internet Protocol Version 4)
  - **MAC Address (08:00:27:DD:D8:F8):** Layer 2 **(**Ethernet II**)**
  - **Port Numbers (80, 3306):** Layer 4(TCP)



```
PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.4.52 ((Ubuntu))
3306/tcp open  mysql   MySQL 8.0.39-0ubuntu0.22.04.1
```

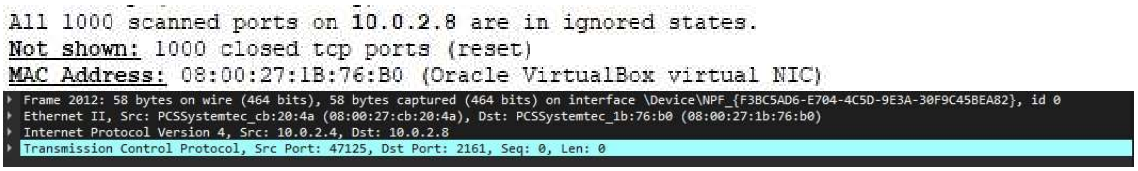
```
Frame 5: 58 bytes on wire (464 bits), 58 bytes captured (464 bits) on interface eth0, id 0
Ethernet II, Src: PCSSystemtec_1b:76:b0 (08:00:27:1b:76:b0), Dst: PCSSystemtec_dd:d8:f8 (08:00:27:dd:d8
Internet Protocol Version 4, Src: 10.0.2.8, Dst: 10.0.2.15
Transmission Control Protocol, Src Port: 62034, Dst Port: 80, Seq: 0, Len: 0

Frame 6: 58 bytes on wire (464 bits), 58 bytes captured (464 bits) on interface eth0, id 0
Ethernet II, Src: PCSSystemtec_1b:76:b0 (08:00:27:1b:76:b0), Dst: PCSSystemtec_dd:d8:f8 (08:00:27:dd:d8
Internet Protocol Version 4, Src: 10.0.2.8, Dst: 10.0.2.15
Transmission Control Protocol, Src Port: 62034, Dst Port: 3306, Seq: 0, Len: 0
```

2.2.3 KaliOpenVAS

- **Host Name:** KALI-OPENVAS
- **IP Address:** 10.0.2.8
- **MAC Address:** 08:00:27:1B:76:B0
- **Operating System:** Kali Linux

- **Open Ports and Services:** None
- **ARP Ping Scan Elapsed Time:** 0.15s
- **OSI Layer Headers:**
  - **IP Address (10.0.2.8):** Layer 3 (Internet Protocol Version 4)
  - **MAC Address (08:00:27:1B:76:B0):** Layer 2 (Ethernet II)
  - **Port Numbers (None):** Layer 4



```
All 1000 scanned ports on 10.0.2.8 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: 08:00:27:1B:76:B0 (Oracle VirtualBox virtual NIC)
```

```
Frame 2012: 58 bytes on wire (464 bits), 58 bytes captured (464 bits) on interface \Device\NPF_{F3BC5AD6-E704-4C5D-9E3A-30F9C45BEA82}, id 0
Ethernet II, Src: PCSSystemtec_cb:20:4a (08:00:27:cb:20:4a), Dst: PCSSystemtec_1b:76:b0 (08:00:27:1b:76:b0)
Internet Protocol Version 4, Src: 10.0.2.4, Dst: 10.0.2.8
Transmission Control Protocol, Src Port: 47125, Dst Port: 2161, Seq: 0, Len: 0
```

3. Information Collection Methodology

3.1 Network Scanning with Nmap

To systematically identify active devices and services within the lab network, I conducted an extensive scan using **Nmap** with multiple scan types and methodologies. The initial network-wide scan was followed by targeted scans on individual devices for a more thorough examination.

1. **Initial Network Scan**: The first step was to perform an intense network scan to discover all devices on the network and gather general information about their open ports and services. The following command was used: `nmap -T4 -A -v 10.0.2.0/24`
   - **-T4**: Adjusts the timing template to speed up the scan.
   - **-A**: Enables OS detection, version detection, script scanning, and traceroute.
   - **-v**: Increases verbosity to provide detailed output.

   This scan allowed me to identify the IP addresses of all active devices on the network and provided basic information about the operating systems, open ports, and services on each device.

2. **Targeted Device Scans**: After identifying the IP addresses of each device from the initial scan, I conducted a series of more detailed scans on each device:
   - **Intense Scan on Each Device**: `nmap -T4 -A -v <IP address>`

     This scan provided more detailed information about the specific device, including OS detection, service versions, and running services.

   - **Full TCP Port Scan**: `nmap -p 1-65535 -T4 -A -v <IP address>`

     I performed a complete TCP port scan to ensure that no open ports were overlooked by the default scan (which only scans the top 1000 common ports).

   - **Intense + UDP Scan**: `nmap -sS -sU -T4 -A -v <IP address>`

This scan combined both TCP and UDP scans to detect services running on both protocols.

3. Each scan provided a comprehensive look at the services and potential vulnerabilities on each device.
4. **Documentation**: All Nmap output was recorded in my personal knowledge management (PKM) system, and I cross-referenced the results with known configurations for each device. The findings were then compiled into a final report, which included details such as:
   - Device IP addresses
   - Open ports (TCP/UDP)
   - Detected services and their versions
   - Operating systems and related information

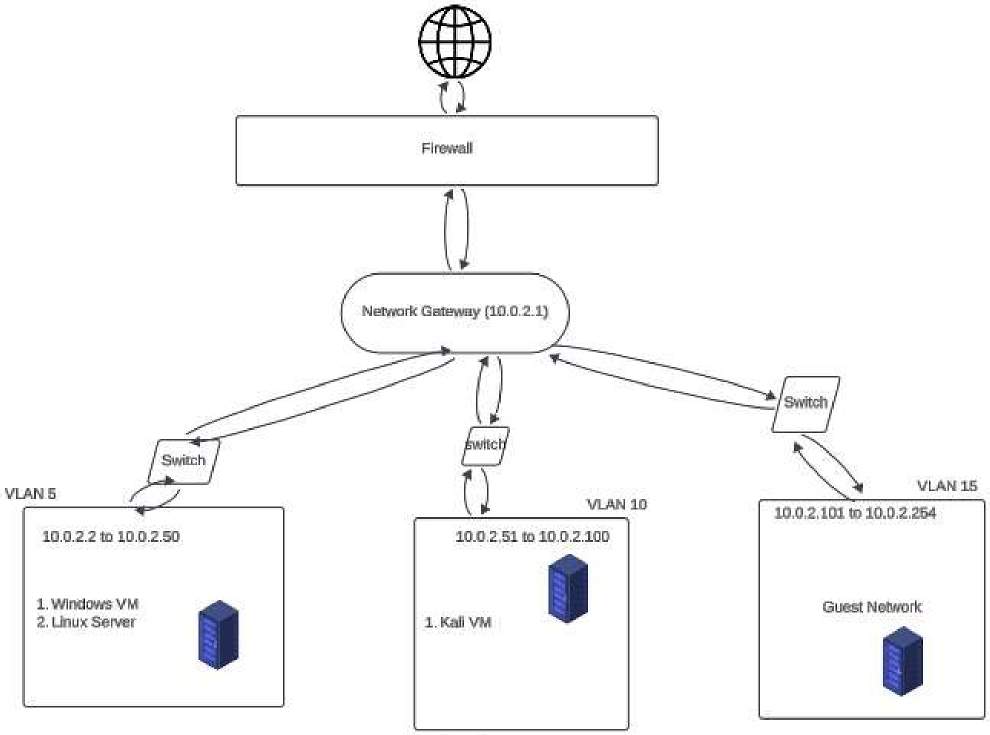
3.2 Traffic Analysis with Wireshark

Wireshark was used to monitor and capture all network traffic during the Nmap scans, enabling me to validate the findings and observe network behavior in real time.

1. **Capture Process**:
   - I started a packet capture on the **eth0** interface of the scanning machine before initiating each Nmap scan.
   - Wireshark continuously captured traffic, including SYN packets, port scan attempts, and service responses from the target devices.
2. **Traffic Analysis**:
   - **SYN Packet Filtering**: Applied filters such as `tcp.flags.syn == 1 && tcp.flags.ack == 0` to isolate SYN packets generated during the scans, indicating port scanning activity.
   - **MAC Address Collection**: Extracted MAC addresses from ARP requests to match IP addresses with physical devices.
   - **Behavioral Analysis**: Analyzed traffic patterns to understand how each device responded to scan attempts, and identified potential vulnerabilities or misconfigurations.
3. **Recording and Reporting**:
   - Screenshots and Wireshark capture data were recorded for key moments when service responses revealed detailed information about the target devices.
   - The captured traffic was used to corroborate the Nmap findings and provide additional context for the report.

---

4. Network Topology and Segmentation Recommendations Diagram



Network Topology Diagram

5. Project Learning Outcomes

Upon completing this project, the following learning outcomes were achieved:

- **Exploration of Processes and Software:** Gained insights into the various processes and software running on Windows and Linux systems by analyzing active services and open ports.
- **Virtual Machine Connectivity:** Selected appropriate virtual machine connectivity configurations to ensure seamless communication between different VMs and secure access to external networks based on the lab scenario.
- **Network Traffic Interpretation:** Developed the ability to interpret captured network traffic using packet sniffing software like Wireshark, identifying patterns and potential security issues.

6. References

1. Nmap.org. (n.d.). Chapter 15. Nmap Reference Guide. https://nmap.org/book/man-briefoptions.html
2. Wireshark.org. (n.d.). *Wireshark User's Guide*. https://www.wireshark.org/docs/wsug_html_chunked/
3. Bitsight. (n.d.). *Cybersecurity executive summary example*. https://www.bitsight.com/glossary/cybersecurity-executive-summary-example