LINUX VULNERABILITY ASSESSMENT LAB

Foundational Security Assessment of an Isolated Ubuntu Environment

Prepared by:

Charlotte Galloway

Assessment Date:

August 23, 2026

Environment:

Isolated Virtual Lab

Tools:

Linux, Nmap, Virtual Box, Python

Table of Contents

Figure Table

1.0 EXECUTIVE SUMMARY

This assessment evaluated the configuration and network of Virtual Machines using Virtual Box in an Ubuntu Linux Environment with a designated Attacker and Target. I used several techniques including network reconnaissance, service enumeration, and other security assessment tools to examine services, ports, controls, and settings. Initially there was very little obviously exposed service surface in the target. I then intentionally curated a controlled data disclosure vulnerability and validated it before remediating and re-testing to follow up and verify that it was functional. I documented the process throughout and maintained an organized system so that I could have a reproduceable procedure.

2.0 SCOPE & OBBJECTIVES

2.1 Scope

The assessment was conducted and restricted to a Virtual Home Laboratory using Ubuntu target and attacker machines on an isolated 10.10.10.0/24 network. There were several specific security areas tested in the assessment including ports, services, network connections, host configurations, and a controlled vulnerability.

2.2 Objective

1. Establish and verify an isolated network for the Attacker and Target.
2. Use reconnaissance and enumeration techniques to establish any exposure.
3. Examine the security posture of the Target VM.
4. Intentionally create a vulnerable service and safely validate the controlled data disclosure vulnerability.
5. Remediate the exposure and conduct follow up testing to verify the fix

3.0 LABRATORY ENVIORNMENT

3.1 Virtual Machines

This laboratory setting was created using Oracle Virtual Box. The two Machines had designated roles in this process. The Target was running Ubuntu 26.04 LTS with kernel 7.0.0-29-generic and was being examined by the Attacker. While the Attacker served as the assessment workstation.

3.2 Network Architecture

The laboratory used the 10.10.10.0/24 network for communication between the two machines. Separate IP addresses were configured; the target was assigned 10.10.10.20/24 and Attacker assigned 10.10.10.10/24. The two machines ran on the isolated Virtual Box network 'cyberlab' which was configured for this lab in order to keep testing contained. Connection was verified successfully using ICMP ping requests.

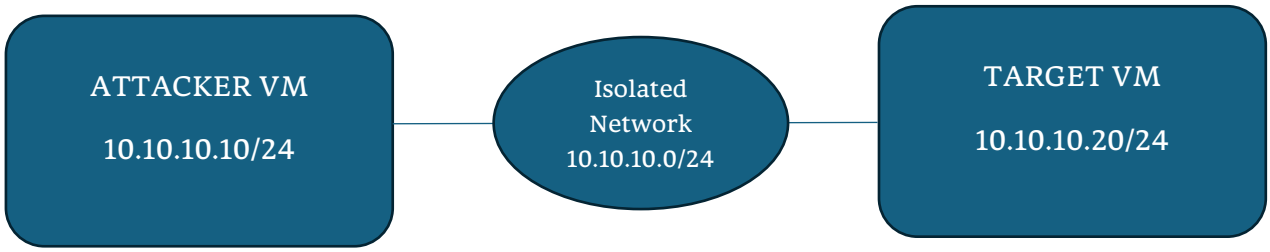

Figure 1. Isolated Lab Network Architecture Diagram

The Diagram illustrates the isolated network configuration used throughout the duration of the lab to contain communication and security testing between the Attacker and Target.

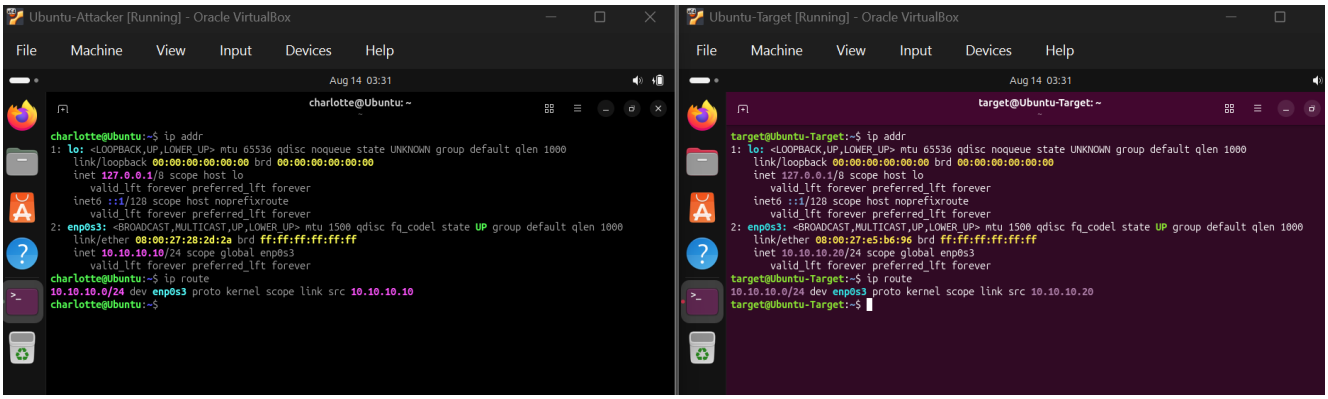

Figure 2. Network Isolation Configuration

The terminal output demonstrated the assigned IP addresses and network interfaces configured for the isolated laboratory environment.

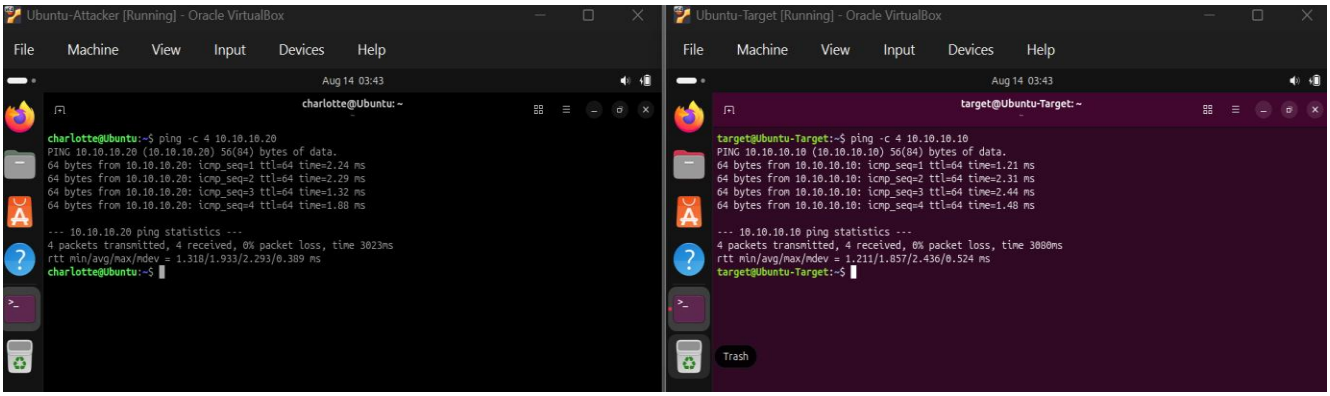

Figure 3. ICMP Connectivity Verification

Successful ICMP echo responses confirming communication between the Attacker and Target within the isolated network.

3.3 Tools

Virtualization: Oracle VirtualBox

Operating System: Ubuntu 26.04 LTS

Reconnaissance: Nmap

Network Analysis: ip, ss, ping

Host Assessment: Linux CLI, chage, UFW, package update utilities

Vulnerability Validation: Python HTTP server

4.0 METHODOLOGY

4.1 Assessment Approach

For the Assessment I determined a structured well documented approach was necessary to begin. First, I would observe the network and perform reconnaissance. Then analyze and Assess the Target's security configurations. After the initial assessment revealed no obvious exploitable vulnerabilities, I introduced a controlled vulnerability. I then Validated the vulnerability, remediated the information disclosure, and finally verified the remediation with follow up testing.

4.2 Network Reconnaissance and Enumeration

I performed the reconnaissance to determine if there were any obvious vulnerabilities or network accessible exposures. I observed listening services and the network sockets. Nmap allowed me to observe the ports and their current states and ss provided me with the local socket information. I then enumerated the observed services to analyze further. This gave me a good understanding of the target's attack surface.

4.3 Host & Service Assessment

After the initial enumeration I identified and investigated DNS and CUPS services for any potential security implications. After finding no remotely exploitable vulnerabilities I began the host assessment. The local security assessment contained the OS baseline, patch Status, host firewall configuration, and the password-aging configuration there were some hardening findings but no directly exploitable vulnerabilities from this assessment.

4.4 Vulnerability Validation & Remediation

I then introduced a controlled vulnerability in order to demonstrate a full vulnerability-management life cycle in the virtual environment. I started a Python HTTP server and exposed a controlled sensitive file to demonstrate an information-disclosure vulnerability on the Target which was then discovered using Nmap on the Attacker to validate the accessible port. A browser was then opened, and the information disclosure vulnerability was validated. I then began remediation by stopping the accessible HTTP service then performing follow up testing through the browser and the Attacker's direct service access to determine whether the remediation was effective.

5.0 ASSESSMENT FINDINGS

5.1 Initial Assessment Findings

The Target's remotely accessible attack surface was limited in my initial assessment. I investigated further for any potential exposure and found no directly exploitable vulnerabilities through these services. In the Host Security assessment, I found the Target's Ubuntu Version and kernel, 7 upgradable packages, the password-aging configuration, and the inactive host firewall. These were hardening findings in the Target's security posture.

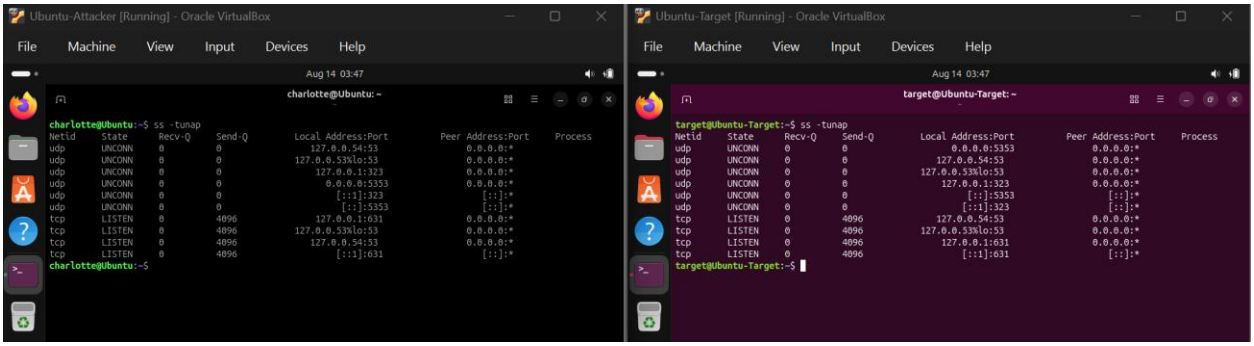

Figure 4. Network Socket Observation

The socket output provided a local view of the Target's active network connections and listening services.

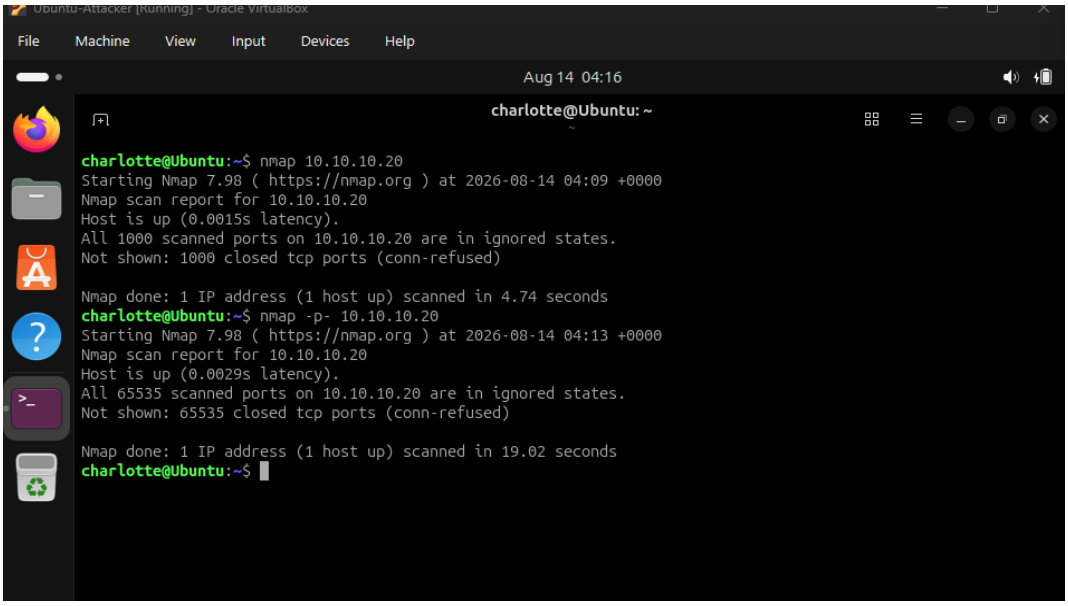

Figure 5. Network Reconnaissance

The Nmap scan revealed the Target's remotely accessible network surface for further investigation.

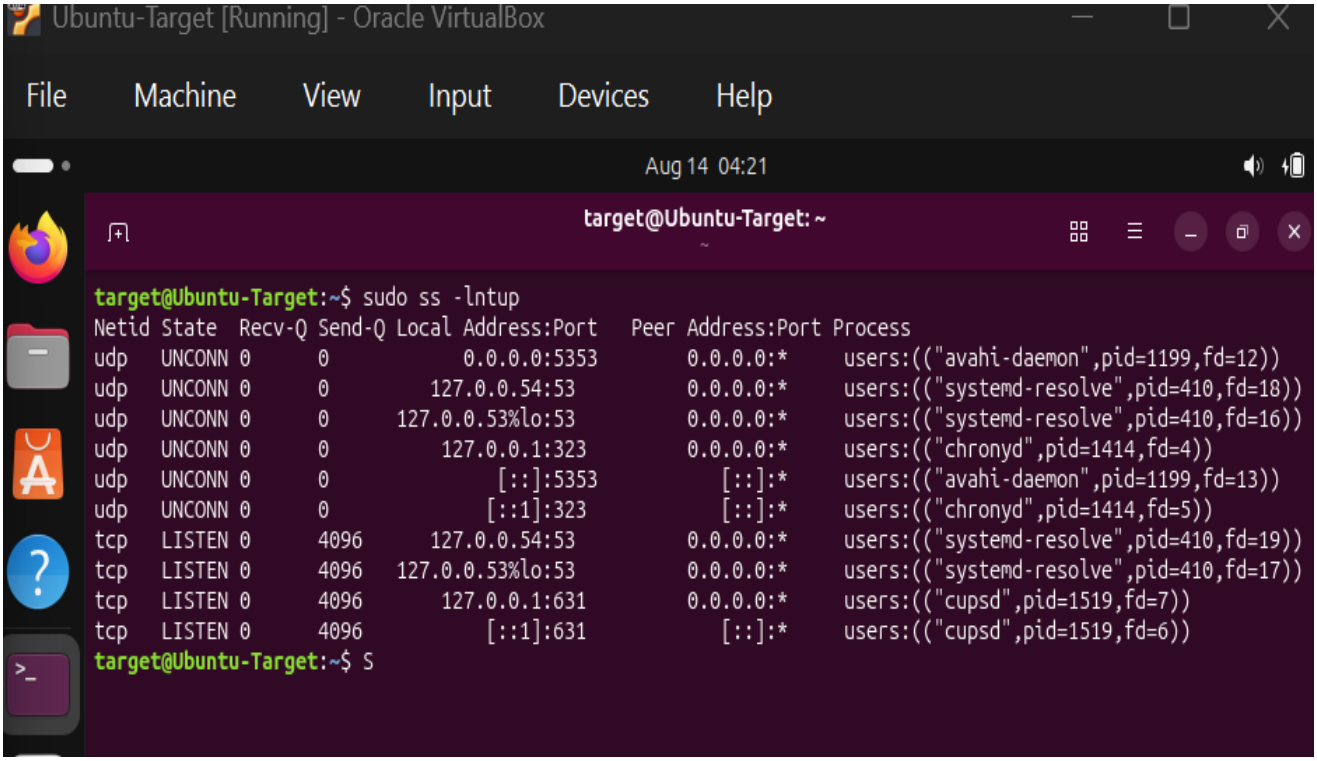

Figure 6. Service Enumeration – Target

Service Enumeration associated the identified ports with specific services for further security analysis.

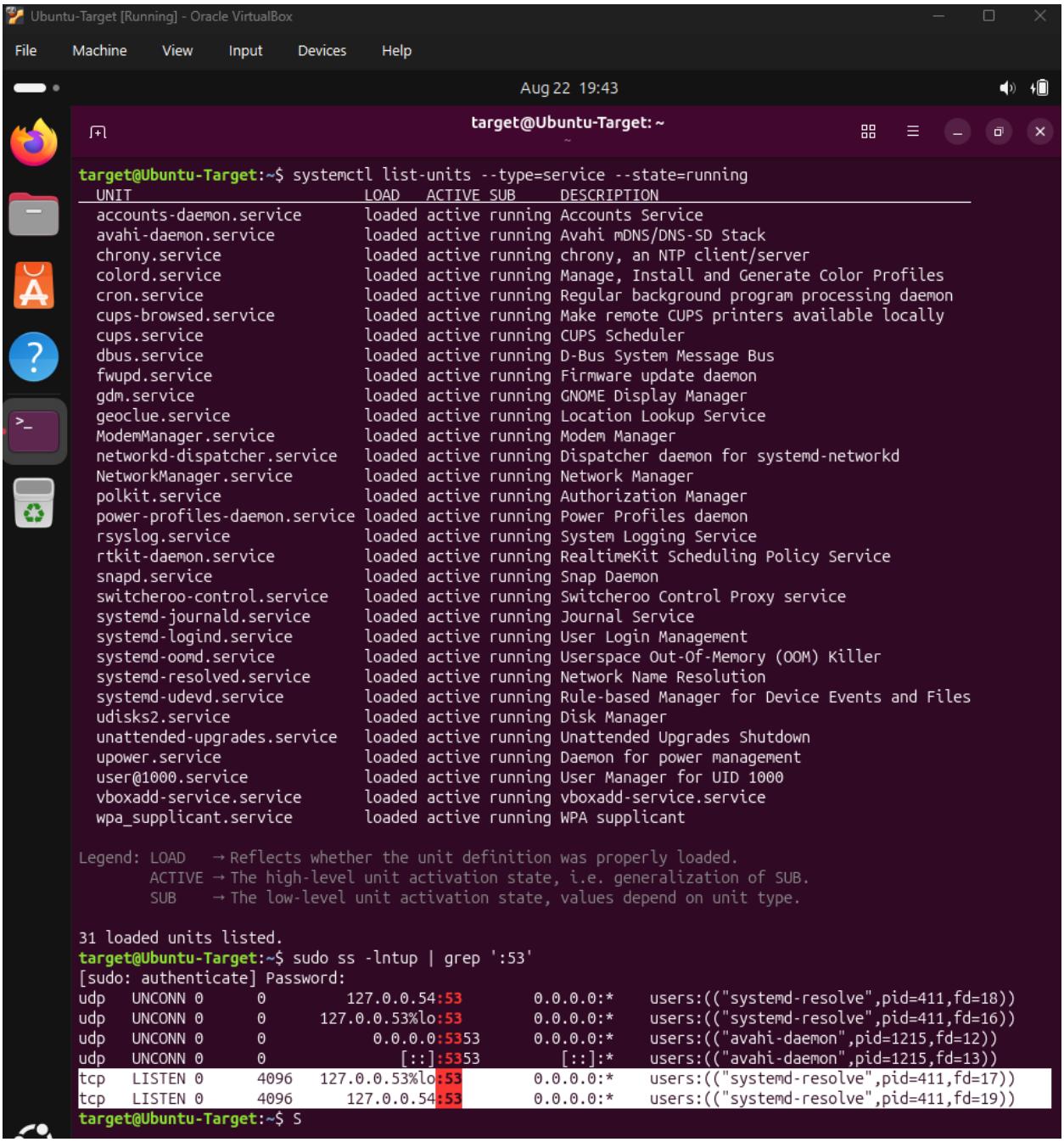

Figure 7. DNS Service Analysis

The DNS service was investigated to determine whether its configuration or accessibility presented a security exposure.

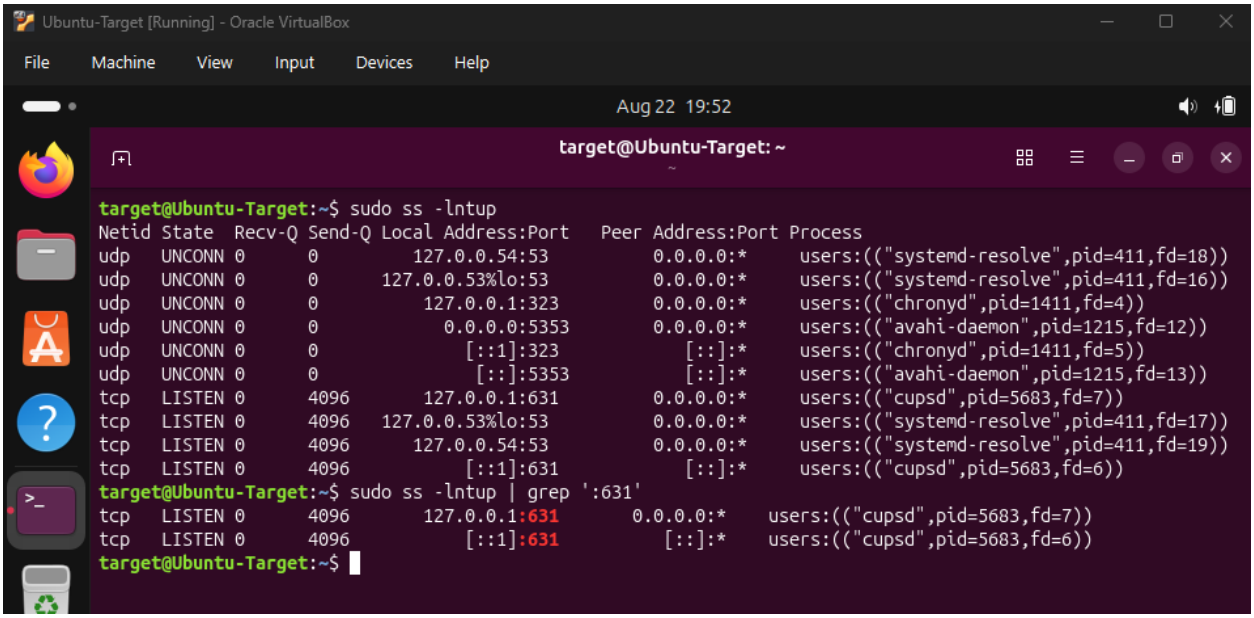

Figure 8. CUPS Service Analysis

The CUPS service analysis was investigated to determine whether its network exposure presented an exploitable condition

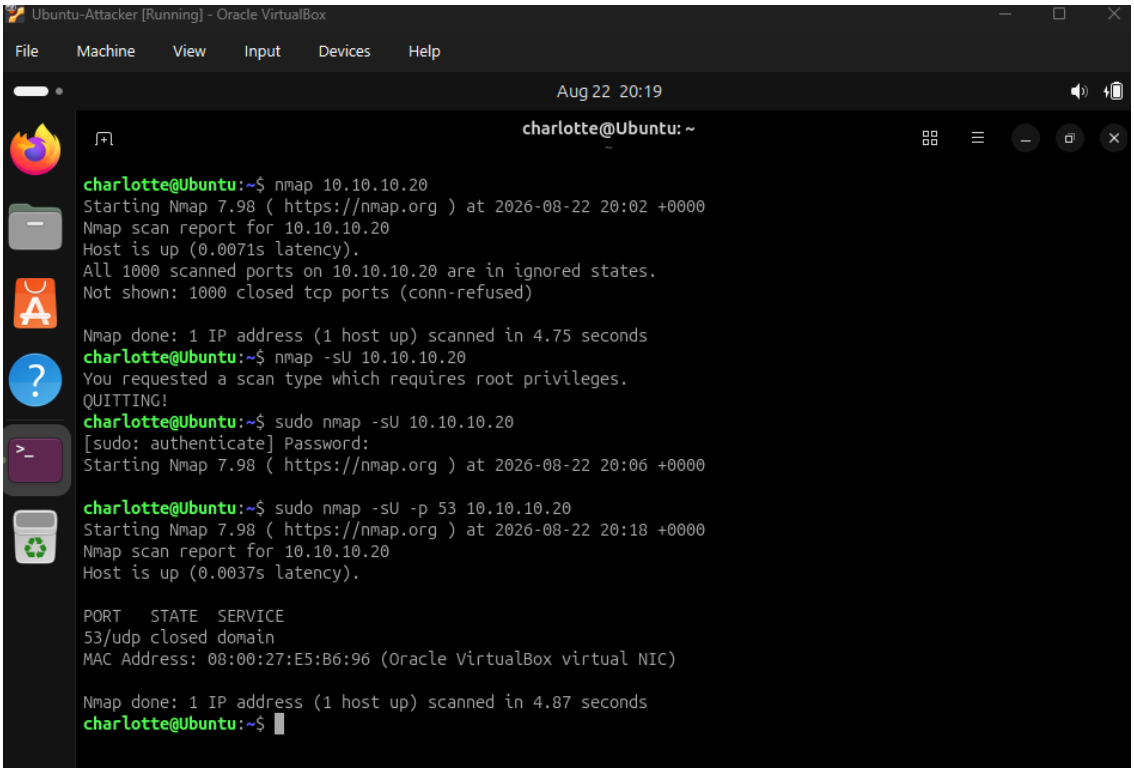

Figure 9. DNS UDP Service Analysis

The UDP scan was used to investigate DNS accessibility and determine whether the service presented an additional network exposure.

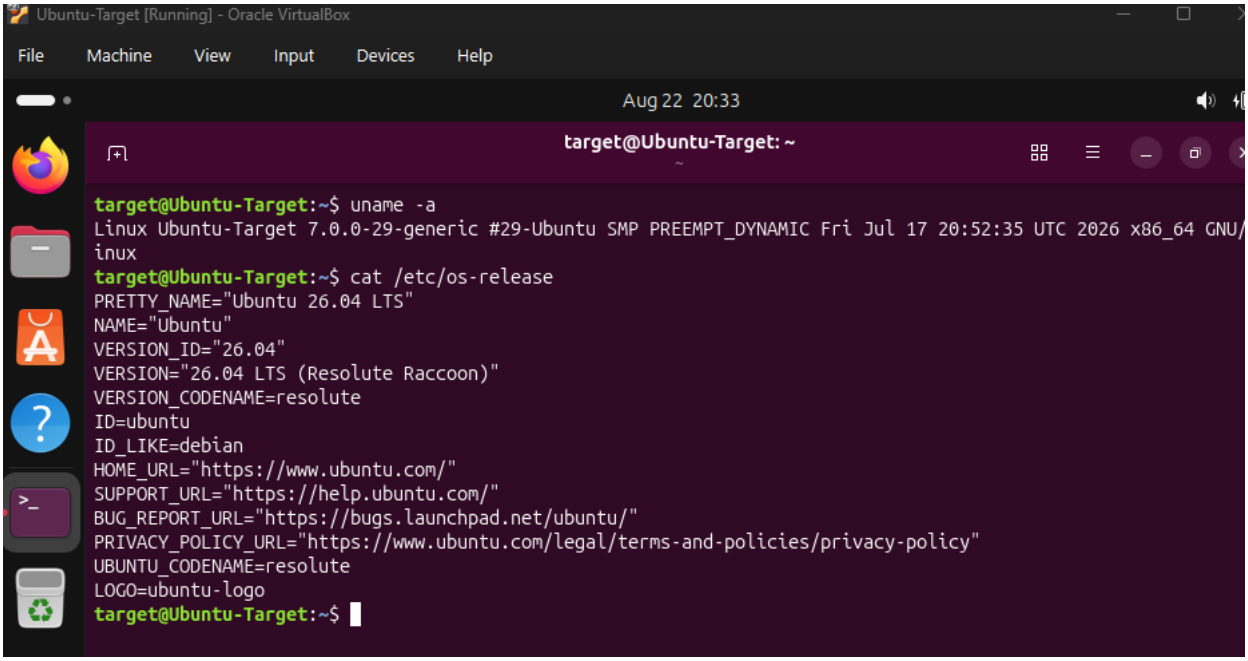

Figure 10. Target Operating System and Kernel Baseline

This output gave baseline information to establish the Target's Ubuntu version and kernel configuration.

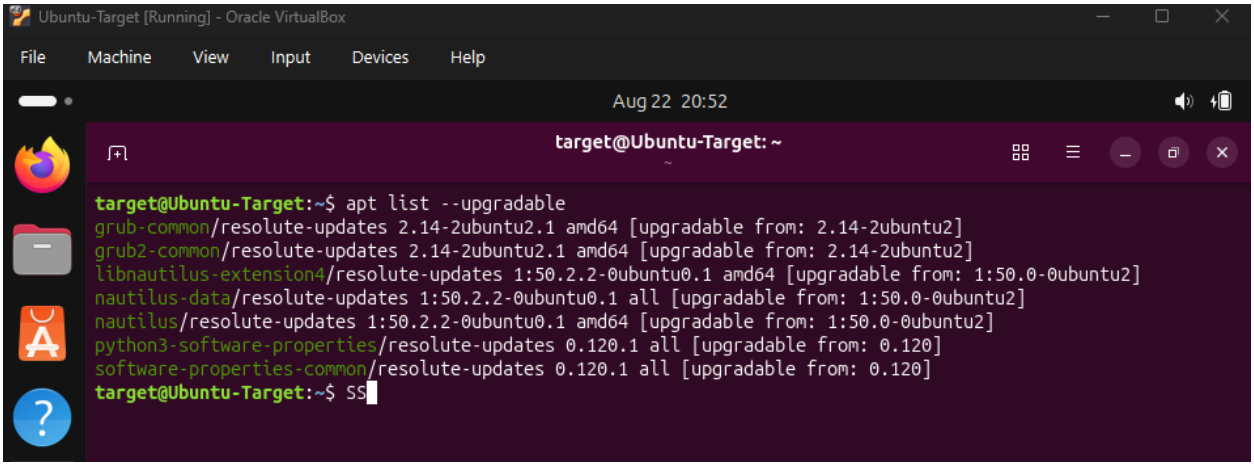

Figure 11. Target Patch Status

The patch status output identified seven available package updates as a system-hardening consideration.

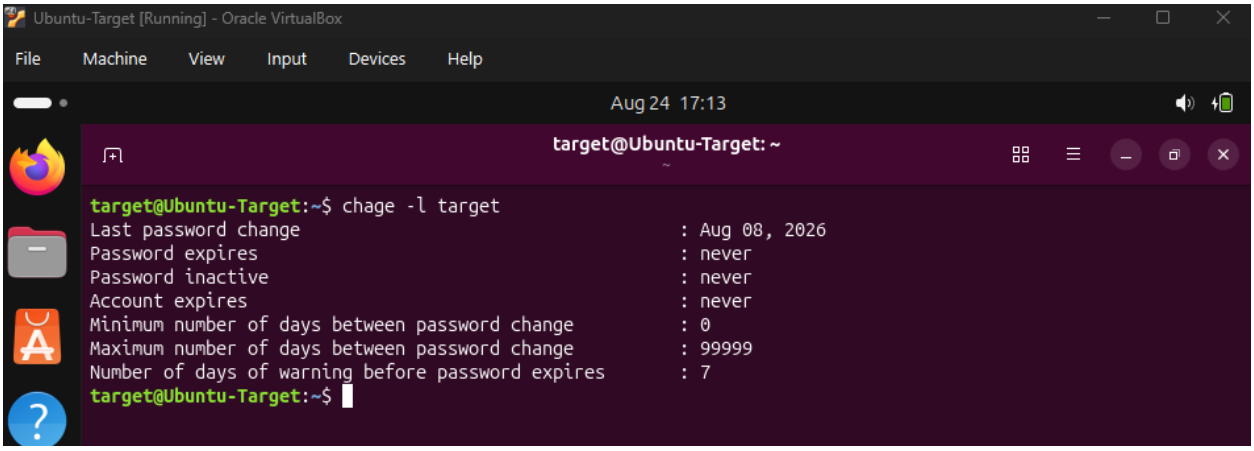

Figure 12. Password Aging Configuration

The password-aging configuration was reviewed to evaluate the Target's account password expiration policy.

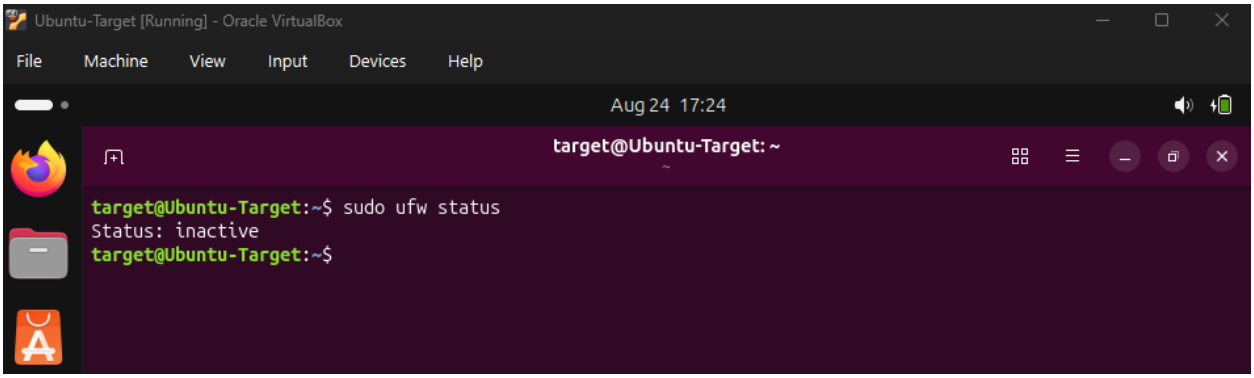

Figure 13. Host Firewall Status

The firewall status showed that the host firewall was inactive, representing a hardening concern.

5.2 Controlled Vulnerability Findings

A HTTP service exposed a sensitive file containing information that should not have been openly accessible through the service. I introduced this controlled vulnerability to perform and practice a proper remediation technique. The browser had full remote access to the sensitive data, an information-disclosure vulnerability. I found the open port responsible for the accessible vulnerability using Nmap.

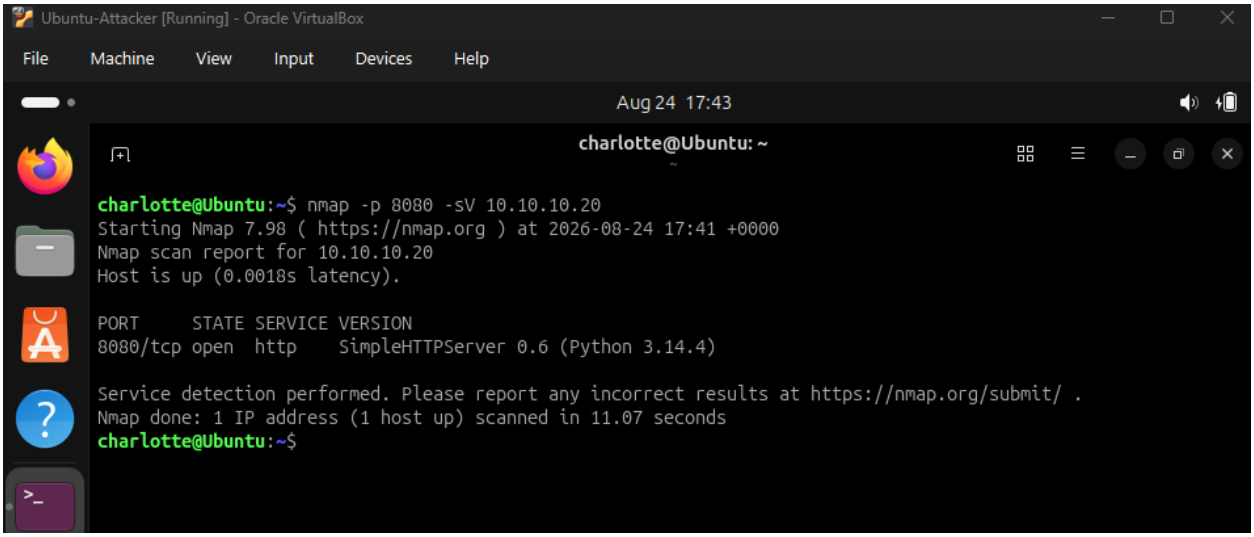

Figure 14. HTTP Service Discovery

The Nmap results confirmed that the intentionally introduced Python HTTP service was accessible through port 8080.

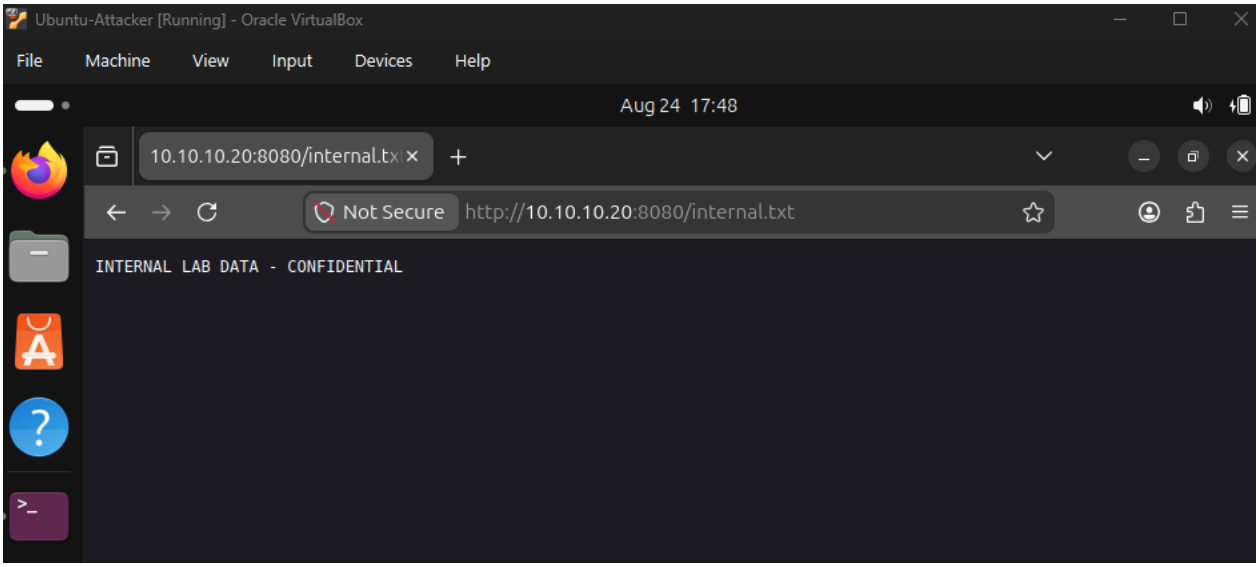

Figure 15. Sensitive File Exposure

The exposed resource was successfully retrieved through the HTTP service, confirming the controlled information-disclosure vulnerability.

5.3 Remediation

In order to remediate the HTTP server vulnerability, I stopped the server which in turn closed the port. Then followed up the remediation with Nmap testing to see if the service remained accessible, it did not. Then went on the browser to validate my Nmap findings further and was no longer able to see the sensitive file.

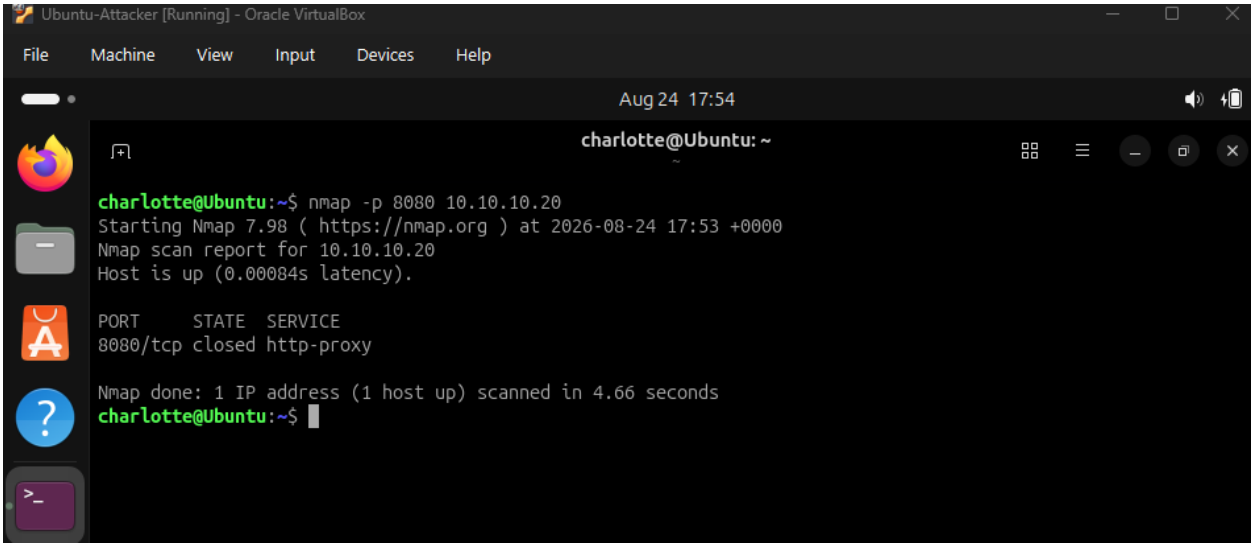

Figure 16. HTTP Service Remediation Verification

Follow-up testing confirmed that port 8080 was closed and the previously exposed resource was no longer accessible.

6.0 LESSONS LEARNED

I learned that conducting cybersecurity assessments are iterative and to actual find any security weaknesses you first must get a good understanding of the attack surface. I also learned an important new concept; presence of an open service does not directly indicate a vulnerability. The procedure is beginning to become clearer identify, validate, remediate, and reverify. I also see the value in having organized well documented work so that it can be easily reviewed or reproduced. Perhaps in my next lab I will investigate more deeply and build a different type of vulnerability so that I can try out a variety of remediation techniques. The biggest takeaway I received from this lab was strengthening my ability to think through the assessment and the right questions to ask

7.0 LIMITATIONS

This assessment was conducted in an entirely isolated virtual lab environment; this limited it to a protected network rather than a public production network. I also only used two VMs, the Attacker and Target so the surface and scale was relatively contained. The information-disclosure was a controlled vulnerability created for the purpose of learning, and not an organically discovered threat. The scope was small and in future labs I plan on using a wider range of techniques to conduct a much broader assessment.

8.0 CONCLUSION

This was an isolated Linux Ubuntu environment where I assessed the attack surface of a Target VM. I initially found limited exposure and several hardening observations but no obvious vulnerability. I introduced a controlled HTTP information-disclosure vulnerability and then performed thorough and successful remediation and verification testing sequences.