

# Sai Venkat Muthyala

## idp-final-10dec-13-19pm.pdf

 SRM University AP Amravati

### Document Details

**Submission ID****trn:oid:::8044:73630853****Submission Date****Dec 10, 2024, 3:36 PM GMT+5:30****Download Date****Dec 10, 2024, 3:37 PM GMT+5:30****File Name****idp-final-10dec-13-19pm.pdf****File Size****1.3 MB****33 Pages****5,810 Words****34,876 Characters**





# 7% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.




## Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 10 words)

## Match Groups

-  **23 Not Cited or Quoted 7%**  
Matches with neither in-text citation nor quotation marks
-  **0 Missing Quotations 0%**  
Matches that are still very similar to source material
-  **0 Missing Citation 0%**  
Matches that have quotation marks, but no in-text citation
-  **0 Cited and Quoted 0%**  
Matches with in-text citation present, but no quotation marks

## Top Sources

- 5%  Internet sources
- 2%  Publications
- 4%  Submitted works (Student Papers)

## Integrity Flags

### 0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

## Match Groups

- 23 Not Cited or Quoted 7%**  
Matches with neither in-text citation nor quotation marks
- 0 Missing Quotations 0%**  
Matches that are still very similar to source material
- 0 Missing Citation 0%**  
Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%**  
Matches with in-text citation present, but no quotation marks

## Top Sources

- 5% Internet sources
- 2% Publications
- 4% Submitted works (Student Papers)

## Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	
	www.ncbi.nlm.nih.gov	2%
2	Submitted works	
	srmap on 2024-05-10	2%
3	Submitted works	
	srmap on 2024-05-12	1%
4	Submitted works	
	srmap on 2024-05-20	1%
5	Submitted works	
	srmap on 2024-05-12	0%
6	Submitted works	
	National Institute of Technology, Agartala on 2024-05-13	0%
7	Submitted works	
	srmap on 2024-05-14	0%
8	Publication	
	Andrei-Daniel Tudosi, Adrian Gaur, Doru Gabriel Balan, Alin Dan Potorac. "Resea...	0%
9	Internet	
	nsgl.gso.uri.edu	0%
10	Internet	
	www.mdpi.com	0%

11

Internet

sdiwc.net

0%

# ENHANCING SECURITY IN DISTRIBUTED FIREWALLS: A PENETRATION TESTING APPROACH

3  
Project Submitted to the  
SRM University AP, Andhra Pradesh  
for the partial fulfillment of the requirements to award the degree of  
Bachelor of Technology  
in  
Computer Science & Engineering  
School of Engineering & Sciences

Submitted by  
Muthyala Sai Venkat (AP21110011582)  
Ravuri Aravind Kumar (AP21110011605)  
Killi Madhuri (AP21110011536)  
Shaik Nayeem (AP21110011604)

Under the Guidance of  
(Dr. Kakumani K C Deepthi)



2  
Department of Computer Science & Engineering  
SRM University-AP  
Neerukonda, Mangalgiri, Guntur  
Andhra Pradesh - 522 240  
May 2024

## DECLARATION

I undersigned hereby declare that the project report titled **Enhancing Security in Distributed Firewalls: A Penetration Testing Approach** submitted for the partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering at SRM University- AP, is a bonafide work done by me under the supervision of Dr. Kakumani K C Deepthi. This submission reflects my original ideas and, where the ideas or words of others have been used, they have been properly cited and referenced. I affirm that I have adhered to academic honesty and integrity, ensuring no misrepresentation or fabrication of data, ideas, facts, or sources. I acknowledge that any breach of these principles may result in disciplinary action by the institute or university and could lead to legal consequences if proper citations or permissions have not been secured. Furthermore, I confirm that this report has not been submitted previously for any degree or diploma at any institution or university.

Place : Neerukonda

Date : Nov 14, 2024

Name of the Student: Muthyala Sai Venkat

Signature:

Name of the Student: Ravuri Aravind Kumar

Signature:

Name of the Student: Killi Madhuri

Signature:

Name of the Student: Shaik Nayeem

Signature:

**DEPARTMENT OF COMPUTER SCIENCE &  
ENGINEERING  
SRM University-AP  
Neerukonda, Mangalgiri, Guntur  
Andhra Pradesh - 522 240**



**CERTIFICATE**

This is to certify that the report entitled **ENHANCING SECURITY IN DISTRIBUTED FIREWALLS: A PENETRATION TESTING APPROACH** submitted by Muthyala Sai Venkat (AP21110011582), Ravuri Ravuri Aravind Kumar (AP21110011605), Killi Madhuri (AP21110011536), Shaik Nayeem (AP21110011604) to SRM University-AP as part of the requirements for obtaining the Degree of Master of Technology. It represents an authentic account of the project work conducted under my/our guidance and supervision. This report has not been submitted to any other University or Institute for any purpose in any form.

Project Guide

Head of Department

Name: Dr.Kakumani K C Deepthi

Name: Dr. Murali Krishna Enduri

Signature: .....

Signature: .....

## ACKNOWLEDGMENT

I wish to record my indebtedness and thankfulness to all who helped me prepare this Project Report **ENHANCING SECURITY IN DISTRIBUTED FIREWALLS: A PENETRATION TESTING APPROACH** and present it satisfactorily.

I extend my heartfelt gratitude to my guide and supervisor, Dr. Kakumani K. C. Deepthi from the Department of Computer Science and Engineering, for providing invaluable suggestions and constructive feedback throughout the preparation of this report.

I am also deeply thankful to my classmates, whose constant support and willingness to listen to my project presentations have been a great source of encouragement.

M Sai Venkat, R Aravind Kumar, Madhuri Killi, Shaik Nayeem  
(Reg.No. AP21110011582, AP21110011605, AP21110011536, AP21110011604)  
B.Tech., Department of Computer Science & Engineering  
SRM University-AP



## ABSTRACT

In the contemporary digital landscape, cybersecurity is a pivotal concern, as organizations across industries grapple with escalating threats to sensitive information and critical infrastructure. This paper delves into the security vulnerabilities inherent in distributed firewall systems and presents a comprehensive methodology for identifying and mitigating these weaknesses through penetration testing. Distributed firewalls, widely adopted for their ability to enhance network security in complex architectures, are not immune to sophisticated attacks that exploit their configuration and implementation gaps.

The research employs a systematic approach to conduct security audits on distributed firewalls, integrating manual and automated penetration testing techniques. Tools such as Nmap, Nessus, and Firewalk are leveraged to uncover vulnerabilities, while remediation strategies are devised to enhance the resilience of the firewall systems. The analysis also highlights the pivotal role of regular audits in maintaining robust network defenses, given the evolving nature of cyber threats.

Through extensive testing, this study identifies common security lapses in distributed firewalls and proposes practical solutions to address them. The findings underscore the efficacy of combining distributed firewalls with periodic security audits and advanced penetration testing techniques. Furthermore, the research provides insights into configuring firewalls to withstand reconnaissance, unauthorized access, and data breaches, ensuring compliance with modern cybersecurity standards.

By bridging the gap between theoretical security models and practical applications, this work establishes a robust framework for safeguarding distributed network infrastructures. The proposed methodology not only fortifies firewall defenses but also serves as a benchmark for future research and implementation in the field of network security. This study thereby contributes to advancing cybersecurity resilience in an era marked by increasingly complex and pervasive cyber threats.

# CONTENTS

<b>ACKNOWLEDGMENT</b>	<b>i</b>
<b>ABSTRACT</b>	<b>ii</b>
<b>LIST OF FIGURES</b>	<b>iv</b>
<b>Chapter 1. INTRODUCTION</b>	<b>1</b>
1.1 Penetration Testing. . . . .	3
1.2 Intrusion Detection System .. . . .	3
<b>Chapter 2. MOTIVATION</b>	<b>5</b>
<b>Chapter 3. LITERATURE SURVEY</b>	<b>6</b>
<b>Chapter 4. METHODOLOGY</b>	<b>15</b>
4.1 Automated pentesting using Nessus. . . . .	15
4.2 Third Party Software Vulnerabilities .. . . .	23
<b>Chapter 5. IMPLEMENTATION</b>	<b>27</b>
<b>Chapter 6. HARDWARE/SOFTWARE TOOLS USED</b>	<b>30</b>
6.1 Hardware. . . . .	30
<b>Chapter 7. RESULTS &amp; DISCUSSION</b>	<b>31</b>
<b>Chapter 8. CONCLUSION</b>	<b>33</b>
<b>REFERENCES</b>	<b>34</b>

# Chapter 1

## INTRODUCTION

Our study focuses on addressing issues raised in publicly-available firewall, which have deployed at points in our network, each with unique settings. While the firewall offers strong network security by default, there are several vulnerabilities that could be exploited. We aim to identify and resolve these weaknesses using various methods, making unauthorized access to the network more difficult. Since new security threats emerge regularly, it is crucial for us to perform continuous security audits to stay ahead. When a system's vulnerabilities are known, it becomes easier for attackers to exploit them, but by eliminating these weaknesses, the potential for cyber-attacks is significantly reduced.

The primary goal of security audits is to assess and review computing systems, whether in full or in part, and evaluate an organization's procedures. Audits involve identifying vulnerabilities and offering solutions to address them. During the detection phase, the audit team identifies gaps in existing defenses, assesses opportunities to update policies, and reviews employee training. The audit process includes several steps: first, defining the main objectives, which helps to set clear expectations for the audit results. Next, the organization defines the scope of the audit, including the systems to be tested and the audit's timeline. Often, a task book is created by both parties to ensure clarity. Once the audit begins, the team gains access to the organization's information and starts identifying vulnerabilities. Following this, a report is generated, outlining the identified flaws and proposed solutions. However, the audit is not complete after the report; updates and improvements are often necessary.

Our proposed solution involves a distributed firewall, which we believe offers enhanced protection compared to traditional firewalls. This approach provides several advantages. Vulnerabilities in the supply chain can lead to widespread user impacts, making it essential for organizations to deepen their understanding of the technologies within their networks. Beyond preventive measures, organizations must also develop and rehearse response scenarios in case of an attack. Cyber-attacks typically involve infiltrating systems to encrypt data until a ransom is paid. This often results in significant downtime, with affected companies experiencing an average of 22 days of disruption. The high cost of paying ransoms, the extended shut time, and the missed data can cause businesses to go under after their first ransomware attack.

A cybersecurity audit offers several key benefits, including giving the organization a comprehensive look of strong threats and uncovering Instabilities. Audits help identify security gaps and areas of non-compliance, leading to improvements in security, efficiency, and consistency. They also contribute to the development of better documentation practices and the establishment of more robust policies and procedures. Additionally, audits provide assurance regarding the integrity of sensitive data, helping to safeguard it against cyber-attacks. As remote work becomes more common, improving network security protocols is crucial since unsecured networks present opportunities for cyber criminals to exploit vulnerabilities.

The following chapters outline our study: Chapter 2 provides a literature review, presenting related publications to give context to our research. Chapter 3 focuses on our proposed method, detailing the tools and techniques used to identify security flaws and highlighting the public vulnerabilities discovered and resolved to date. In Chapter four, we present our findings and suggest solutions for addressing the identified issues. Finally, last chapter offers a conclusion, discussing the outcomes and the relevance of our work.

## 1.1 PENETRATION TESTING

Penetration Testing, is a proactive approach to cybersecurity where experts mimic cyber-attacks on systems, networks to uncover potential vulnerabilities that will be exploited by attackers. Its main goal is to assess the strength of existing security measures and detect weaknesses before they can be targeted by malicious entities. The process generally includes stages such as reconnaissance, scanning, exploiting discovered vulnerabilities, and documenting the findings. Penetration testing can be performed manually or with automated tools such as Nmap, Nessus, and Metasploit. This practice aids organizations in evaluating their security framework, addressing identified risks, and fortifying their defenses against potential threats. This method is essential in identifying issues such as misconfigurations, unpatched systems, and potential points of unauthorized access, ensuring that systems are secure against evolving threats.

## 1.2 INTRUSION DETECTION SYSTEM (IDS)

Intrusion Detection System is an essential cyber system used to monitor network traffic and activities to detect malicious activity, policy breaches, unauthorized access attempts. It works by analyzing data in real-time, comparing it with a database of known attack signs, or employing anomaly detection algorithms to spot irregular patterns indicative of a potential security compromise.

IDS is divided into two types: Network-based IDS, which observes and also analyzes network traffic for malicious activity, and Host-based IDS, which checks the activities and behavior of specific hosts or system devices within a network. By detecting and alerting administrators about potential threats, IDS plays a vital role in preventing data breaches, mitigating potential attacks, and ensuring the integrity and security of networked systems.

## Chapter 2

### MOTIVATION

Penetration testing for distributed firewalls addresses the pressing need for enhanced security and resilience in modern network infrastructures. Traditional security measures often fall short in managing the complexities of decentralized systems, leaving them vulnerable to sophisticated attacks. By applying penetration testing, this project seeks to identify vulnerabilities within distributed firewalls, enabling organizations to proactively address security gaps and improve network defenses. This approach not only ensures the integrity of data and systems but also optimizes firewall configurations to withstand emerging threats. Through this study, the project contributes to the evolving field of cybersecurity by offering practical solutions to enhance the performance, reliability, and protection of distributed networks. By integrating advanced testing techniques, this work aims to foster a more secure digital environment, meeting the growing demand for effective cybersecurity solutions in a rapidly changing technological landscape.

## Chapter 3

### LITERATURE SURVEY

In this research report, we continuously examined previous penetration testing (pentesting) methods to offer a more comprehensive and varied investigation. A crucial component of network is the ability to categorize between suspicious things and normal network traffic. When evaluating network security solutions like protocol types, files, or URLs, identifying entities that are explicitly harmful or unquestionably safe is relatively straightforward. However, a significant challenge arises in the gray area between these extremes, where differentiating between potentially malicious and legitimate content becomes more complex and requires deeper analysis. In this ambiguous zone, content might appear suspicious, even if it is not malicious in nature. Nonetheless, even benign content may cause issues if it triggers a false alarm or leads to an endpoint application failure.

False positives happen when a network security tool erroneously flags a benign entity as harmful. These misclassifications can happen across various types of network traffic, including files, URLs, and network protocol patterns. The vulnerability scanning process is a key approach for addressing these challenges, consisting of three main stages. The initial phase, known as reconnaissance, involves scanning targets to detect active services by identifying open and accessible ports. This step often includes sending ICMP requests to addresses within a defined range. The second phase zeroes in on locating open ports on the active hosts identified during reconnaissance. In the final phase, carefully designed data packets are transmitted to these open ports to extract detailed information about the services running on them, including their specific versions. This information is then matched against established vulnerability databases to pinpoint any potential security flaws. Subsequently, in the vulnerability analysis stage, the gathered data is meticulously reviewed, compiled into a comprehensive report, and provided to cybersecurity specialists for further assessment.

The study in [9] discusses the use of port scanning as a technique in network reconnaissance. In a test involving pfSense, a widely used open-source router, pfSense successfully detected the attack in every case but was unable to block the originating source. To improve its ability to block attacks, pfSense would benefit from being integrated with an IDS or IPS. In [10], the importance of risk assessment is discussed, which involves the identification, evaluation, and management of risks. This process is integral to risk analysis, which is an ongoing effort to detect and analyze potential risks that could harm a business. Effective risk management often requires information that vulnerability scanners cannot automatically retrieve, as it is often organization-specific.

Additionally, In [11] This framework leverages graphs to visualize potential attack paths and security. It incorporates a state-of-the-art reinforcement learning algorithm to identify the most effective strategies for performing penetration testing. ASAP autonomously generates attack tactics and conducts security analyses on real networks, uncovering attack line in manual penetration testing. By automating, ASAP reduces the need for labor while providing effective penetration testing. Penetration testing is essential for evaluating the health of a network from cyber-attacks and helps prioritize security measures and ensure compliance with industry standards.

A recurring issue in network-based threat detection is the lack of context in many solutions. Various methods, such as active scanners, have been developed to address this, though each has its limitations. Active scanners, such as Nmap and Nessus, are widely used to gain endpoint visibility. Nmap is used to check if systems are online, identify network issues, and detect network stacks, OS current version, and favours. Nessus, another popular tool, is used for vulnerability detection, asset discovery, and high-speed profiling, and it quickly updates to address newly disclosed vulnerabilities.

The Nmap tool, integral to the security and IT industries, is essential for assessing system security by identifying open network ports. Without having knowledge of which disclosed ports, it is difficult to measure a system's integrity. Professional penetration testers and hackers also commonly use tools like Nessus



for target profiling and finding sensitive data. Nessus provides updated plugins to customers within 24 hours of the disclosure of new vulnerabilities[8].

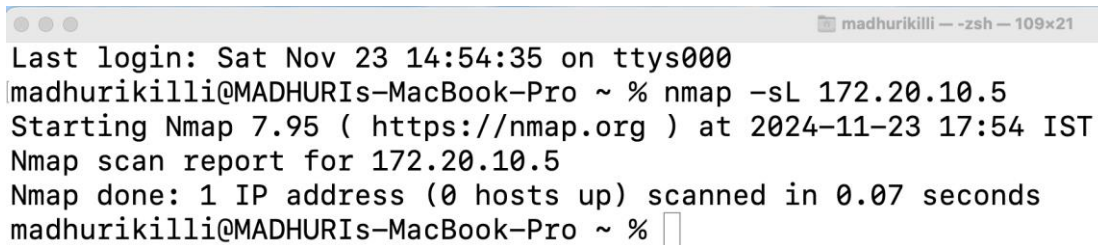
Penetration testing can be conducted manually or automatically. Manual penetration testing requires specialized experts and takes longer, whereas automated testing can be performed with minimal expertise and delivers consistent results. Both methods have their advantages, and their use depends on the specific circumstances. Manual testing, though time-consuming, provides more in-depth analysis, while automated testing offers quicker and repeatable results, making it an effective tool for routine assessments.

### *Manual Pentesting in Kali Linux*

Penetration testing should start with commonly available and straightforward approaches. In this section, considering that these tools are widely used and not expected to cause significant disruption. We begin the security audit by performing a manual test using Nmap.

Nmap comes with a tool for identifying firewall organizing, using ACK probe responses to detect port filtering. This feature allows us to check the filtering status on individual ports or across a range of ports. For this to work, Nmap requires a automated system that is scanning the network so that the audits can be used for firewall identification. Nmap's versatility makes it a popular choice for security testing, offering users the ability to fully customize their scans. This scanner can do numerous tasks, including host finding, port probing, and OS detection. Additionally, Nmap is highly extensible, enabling users to create features through the Nmap Scripting System. Since most scan operations which typically require root access on Unix systems, these operations are limited to some users.

1 The starting stage in our testing is to understand the firewall. In this view, we take that IP data is left undetected. To begin, a straightforward Nmap-scan helps to detect what is active. By using -sL option with Nmap, which displays the results as depicted in Figure 1.



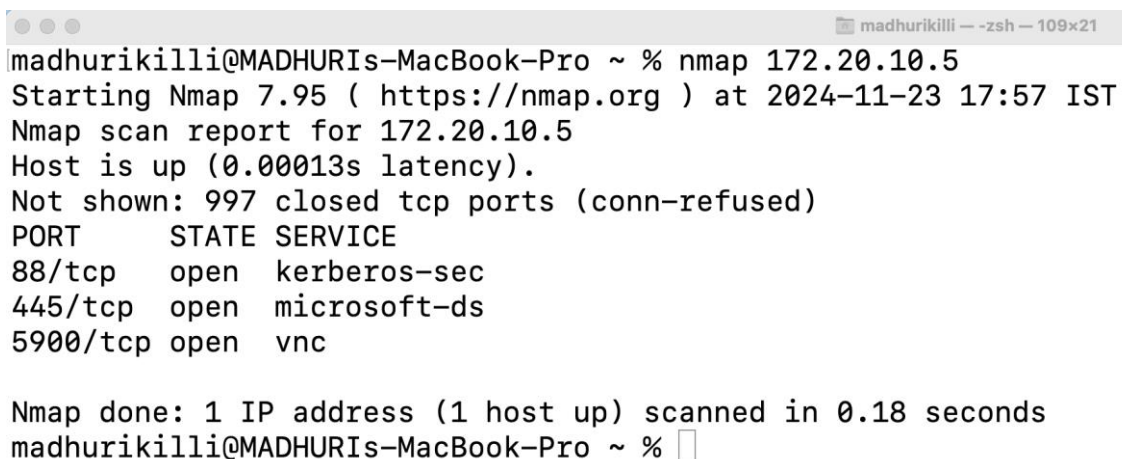
```

Last login: Sat Nov 23 14:54:35 on ttys000
madhurikilli@MADHURIs-MacBook-Pro ~ % nmap -sL 172.20.10.5
Starting Nmap 7.95 ( https://nmap.org ) at 2024-11-23 17:54 IST
Nmap scan report for 172.20.10.5
Nmap done: 1 IP address (0 hosts up) scanned in 0.07 seconds
madhurikilli@MADHURIs-MacBook-Pro ~ %

```

**Figure 1.** Nmap scan of -sL command

Zero hosts were found in first stage. This can possible due to the way certain OS manage network traffic during this type of scans. However, It provides multiple techniques to locate these machines. In this scenario, we instruct this to detect every address in the 172.20.10.5/24 network. By detecting through -sn option, Nmap pings the host without performing a port scan, which is its default behavior. The scan returned a few potential hosts, as shown below.



```

madhurikilli@MADHURIs-MacBook-Pro ~ % nmap 172.20.10.5
Starting Nmap 7.95 ( https://nmap.org ) at 2024-11-23 17:57 IST
Nmap scan report for 172.20.10.5
Host is up (0.00013s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE
88/tcp    open  kerberos-sec
445/tcp   open  microsoft-ds
5900/tcp  open  vnc

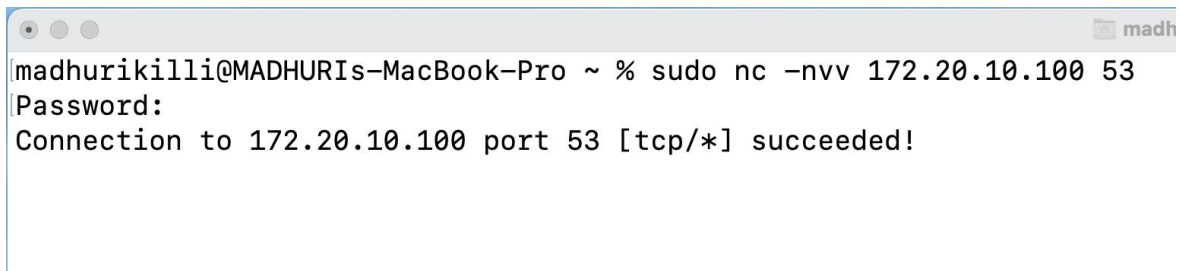
Nmap done: 1 IP address (1 host up) scanned in 0.18 seconds
madhurikilli@MADHURIs-MacBook-Pro ~ %

```

**Figure 2.** Nmap scan for opened and closed ports

In succeeding step, we will use Nmap scans capabilities to identify connected systems on particular hosts. This total indicates the availability of a service on this specific system. As noted before, the cause for having number of unclosed ports on this IP address 172.10.5.100 corresponds to a metasploitable machine. On most systems, having such an large list of unfolded ports, so it's worth taking deeper look at this device.

Getting knowledge on which network or systems are positioned between your system and a goal is essential, as it helps in mapping out the network's touch. For instance, Security Auditors can't audit a server with undetermining whether firewall is placed in starting of that. Below figure illustrates this information.

A screenshot of a macOS terminal window. The title bar shows three window control buttons and a tab labeled 'madh'. The terminal text shows a user 'madhurikilli' at a 'MADHURIs-MacBook-Pro' prompt. The command 'sudo nc -nv 172.20.10.100 53' is entered. The prompt 'Password:' is shown, followed by the output 'Connection to 172.20.10.100 port 53 [tcp/\*] succeeded!'.

**Figure 4.** Scan Report for 53.

The traceroute kit is valuable here, as it sends packets to the target machine, tracking each action along the way. This helps identify entangled machines in transmitting network data and their IP addresses. If a device responds with stars in the traceroute output, as seen in below, it typically indicates the machine is not configured. However, it doesn't necessarily mean there's no traffic; it could also result from network issues that cause packets to be dropped, such as packet timeouts or traffic being blocked by a firewall, as observed in our setup.

```

madhurikilli@MADHURIs-MacBook-Pro ~ % traceroute -F 172.20.10.100
traceroute to 172.20.10.100 (172.20.10.100), 64 hops max, 52 byte packets
 1 172.20.10.1 (172.20.10.1)  4.962 ms  2.391 ms  2.274 ms
 2 192.168.29.10 (192.168.29.10)  271.413 ms  86.302 ms  288.456 ms
 3 * * *
 4 192.168.31.20 (192.168.31.20)  68.476 ms  35.678 ms  38.265 ms
 5 * * *
 6 * * 125.21.125.154 (125.21.125.154)  63.767 ms
 7 125.21.125.153 (125.21.125.153)  21.581 ms
 125.21.125.149 (125.21.125.149)  50.217 ms
 125.21.125.153 (125.21.125.153)  23.808 ms
 8 182.79.137.2 (182.79.137.2)  85.517 ms
 116.119.57.140 (116.119.57.140)  155.480 ms
 116.119.57.160 (116.119.57.160)  63.864 ms
 9 * * *
10 * * *
11 * * *
12 * * *
13 * * *
14 * * *
15 * * *
16 * * *
17 * * *
18 * * *
19 * * *
20 * * *
21 * * *
22 * * *
23 * * *
24 * * *
25 * * *
26 * * *
27 * * *
28 * * *
29 * * *
30 * * *
31 * * *
32 * * *
33 * * *
34 * * *
35 * * *
36 * * *
37 * * *

```

**Figure 5.** Traceout result

Succeeding phase will use Netcat for banner grabbing. Banner grabbing is a approach used to identify unclosed ports and features running on a machine. While admins often employ this method to inventory the machines and features on their networks, it can also be exploited by attackers to determine the OS that may contain ideal vulnerabilities. Need to check with the results from above with those obtained while utilizing Nmap, as plotted in below figure.

```

madhurikilli@MADHURIs-MacBook-Pro ~ % sudo nc -nvv 172.20.10.100 53
Password:
Connection to 172.20.10.100 port 53 [tcp/*] succeeded!

```

**Figure 6.** Netcat Mapping

```

Last login: Sat Nov 23 21:08:17 on ttys001
madhurikilli@MADHURIs-MacBook-Pro ~ % nmap -sV -script=banner 172.20.10.100
Starting Nmap 7.95 ( https://nmap.org ) at 2024-11-23 21:11 IST
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.73 seconds
madhurikilli@MADHURIs-MacBook-Pro ~ % nmap -Pn -script=banner 172.20.10.100
Starting Nmap 7.95 ( https://nmap.org ) at 2024-11-23 21:13 IST
Nmap scan report for 172.20.10.100
Host is up (0.016s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT      STATE SERVICE
53/tcp    open  domain

Nmap done: 1 IP address (1 host up) scanned in 57.41 seconds
madhurikilli@MADHURIs-MacBook-Pro ~ %

```

**Figure 7.** Banner Grabbing.

The outcome procedure shown in above figures didn't provide any additional insights beyond what was discovered in before. N map confirmed that host, specifically the first firewall used in our analysis, remains active.

**Firewalking [12]** is another tool we can utilize in this setup. It operates by sending TCP/UDP packets with a Time to Live (TTL) value incremented by one beyond the target gateway. Depending on whether the gateway accepts or rejects the traffic revealing information about the routing path. Conversely, if the gateway blocks the traffic, no response is received. Firewalk is an open-source project available under the BSD license, providing a cost-effective and versatile solution for network security reconnaissance by routers or firewalls. During penetration testing, this tool is especially valuable for examining firewall configurations.

While there are also scripts available, most of them did not yield useful information in our experiment. We selected this script due to its open-source nature, cost-effectiveness, and the supportive community backing it. This section underscores that manual penetesting is money-time. Manuals often needs preffered time to conduct thorough investigations, particularly in big enterprises and software industries. For this analyzation, had to delve for these tools, analyze the situations that would yield useful results, and fine-tune the tests accordingly

to produce meaningful findings.

In the following section, we will conduct an automated penetration test to collect information that will help us gain a better understanding of the security vulnerabilities present in our setup.

## Chapter 4

# METHODOLOGY

### 4.1 Automatic Penetration Testing with Nessus

The effectiveness of manual pentesting can't match the fastness and levels of automatic pentests, primarily due to their automated nature. Given this, we conducted an automated penetration test to gain a clearer understanding of the security challenges that might arise in our proposed distributed firewall setup. To ensure consistency and reliability in our results, we repeated the penetration test multiple times and monitored any changes. Each scan took approximately fifteen minutes, and outcomes provided below.

The insecurities are provided non-static Vulnerability Priority Rating, which is updated regularly by Tenable to reflect the latest threat environment. The VPR ranges from 0.1 to 10.0, with higher values indicating greater exploitation risk.

In our case, a medium VPR, which corresponds to a range of 4.0 to 6.9, was observed for many vulnerabilities, while vulnerabilities lacking CVEs[7] (such as many of the Info severity types) do not receive a VPR. Based on Tenable's recommendations, vulnerabilities should be prioritized according to their severity as indicated by the CVSS scores.

In below, the highest threats identified in analysis are presented. For every vulnerability founded, Nessus provides a note and suggested solutions. In this scenario, the issues are manageable and could be addressed through a straightforward procedure, thanks to the high performance and robust security features provided by pfSense. This allowed us to efficiently resolve the vulnerabilities identified in our distributed firewall setup.

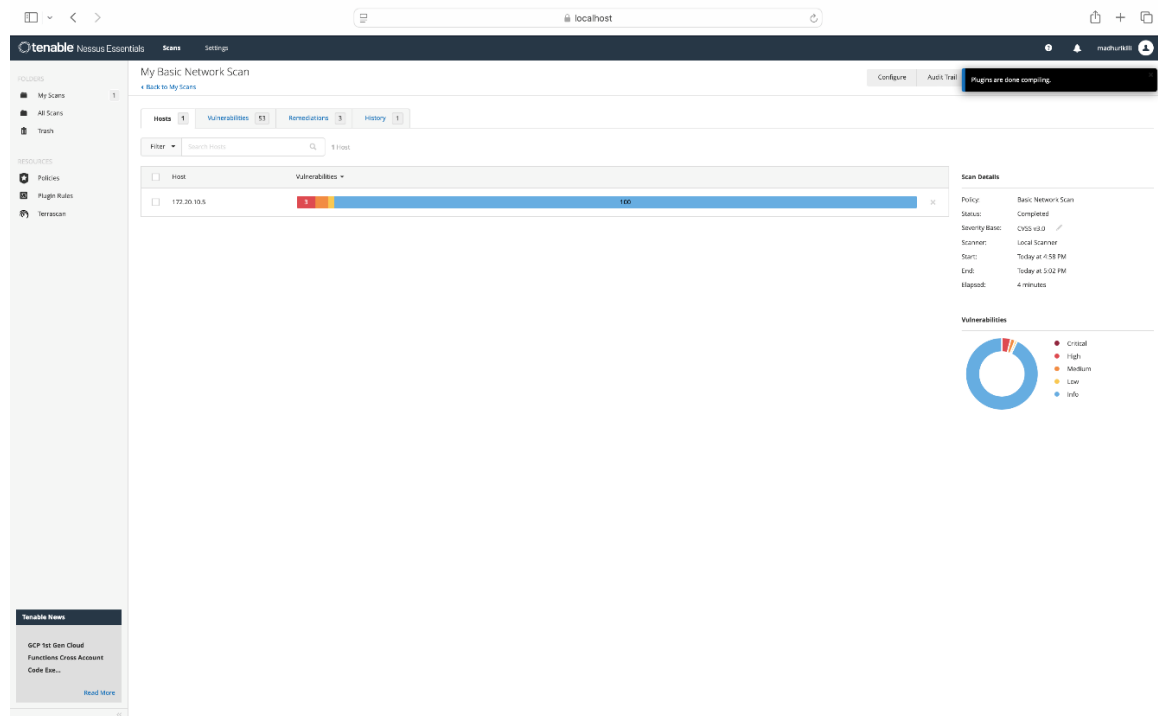


Figure 9. Basic Network Scan

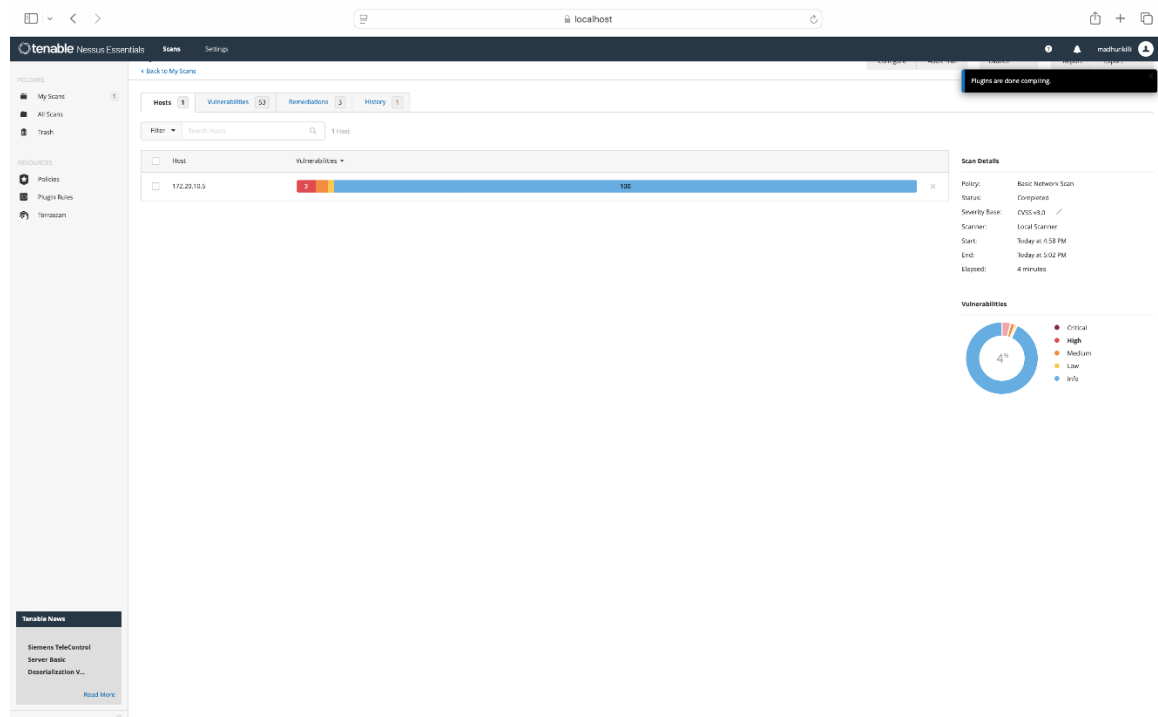


Figure 10. Results – High vulnerabilities



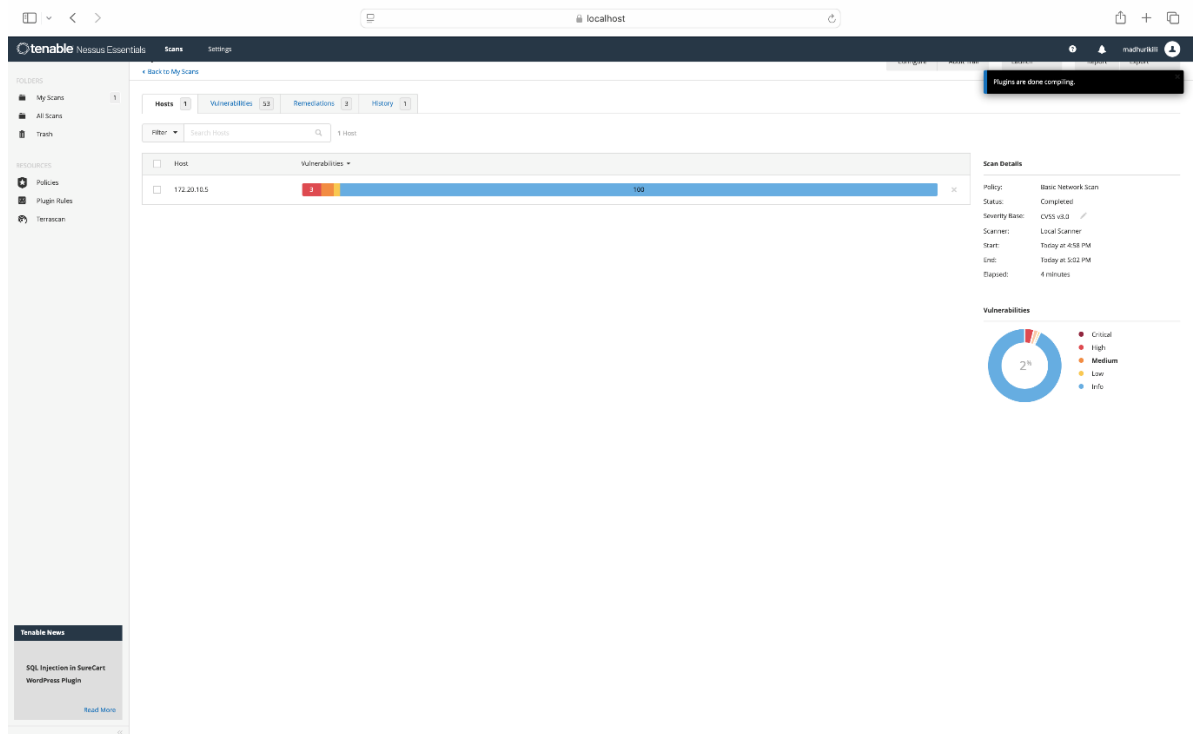


Figure 11. Results – Medium vulnerabilities

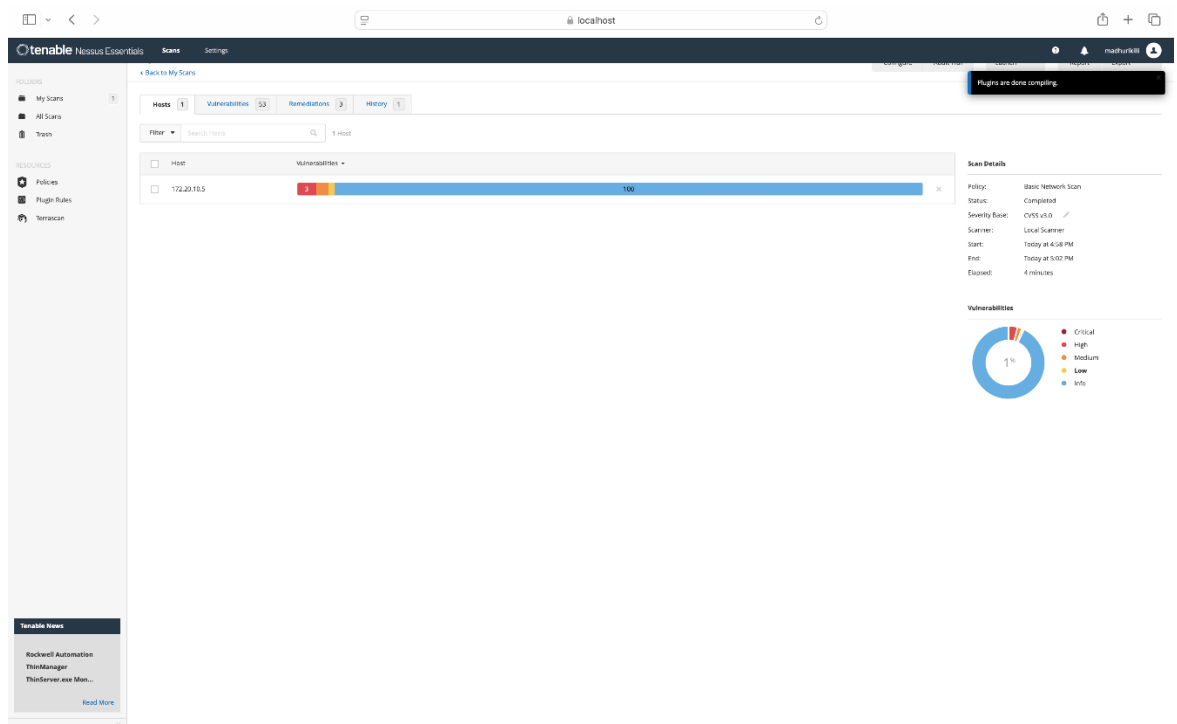


Figure 12. Scanning Result – Low vulnerabilities

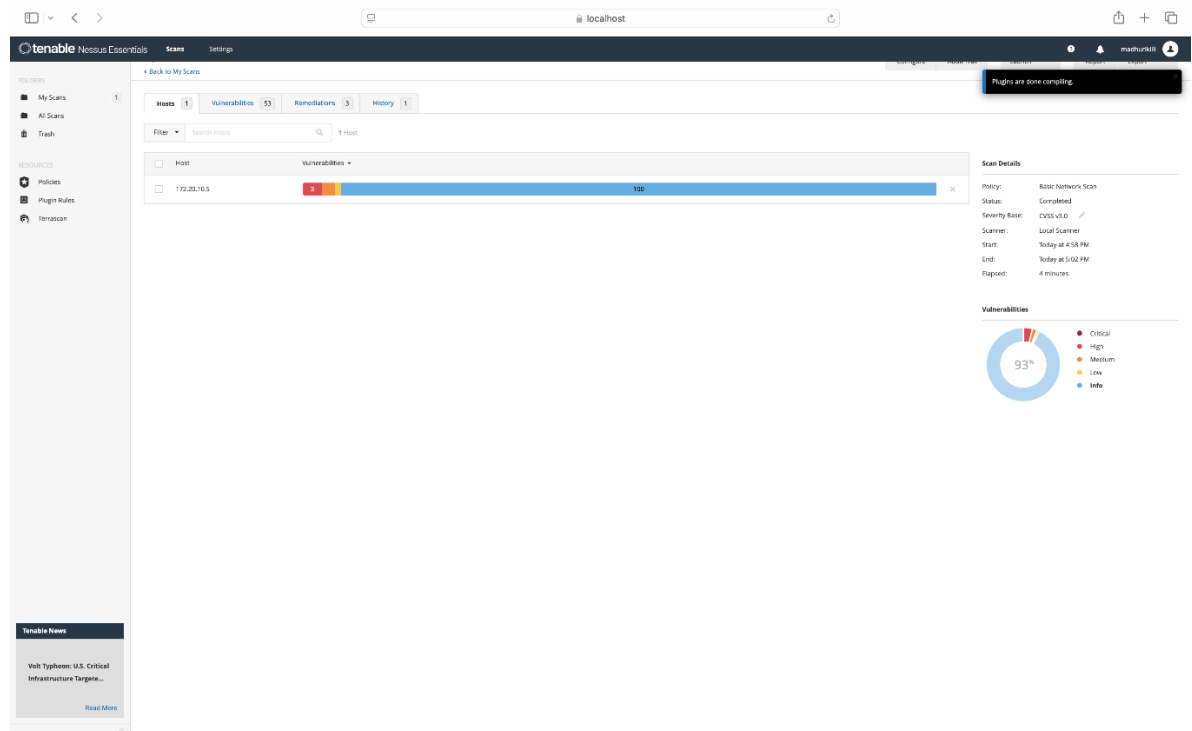


Figure 13. Scanning Result – Info vulnerabilities

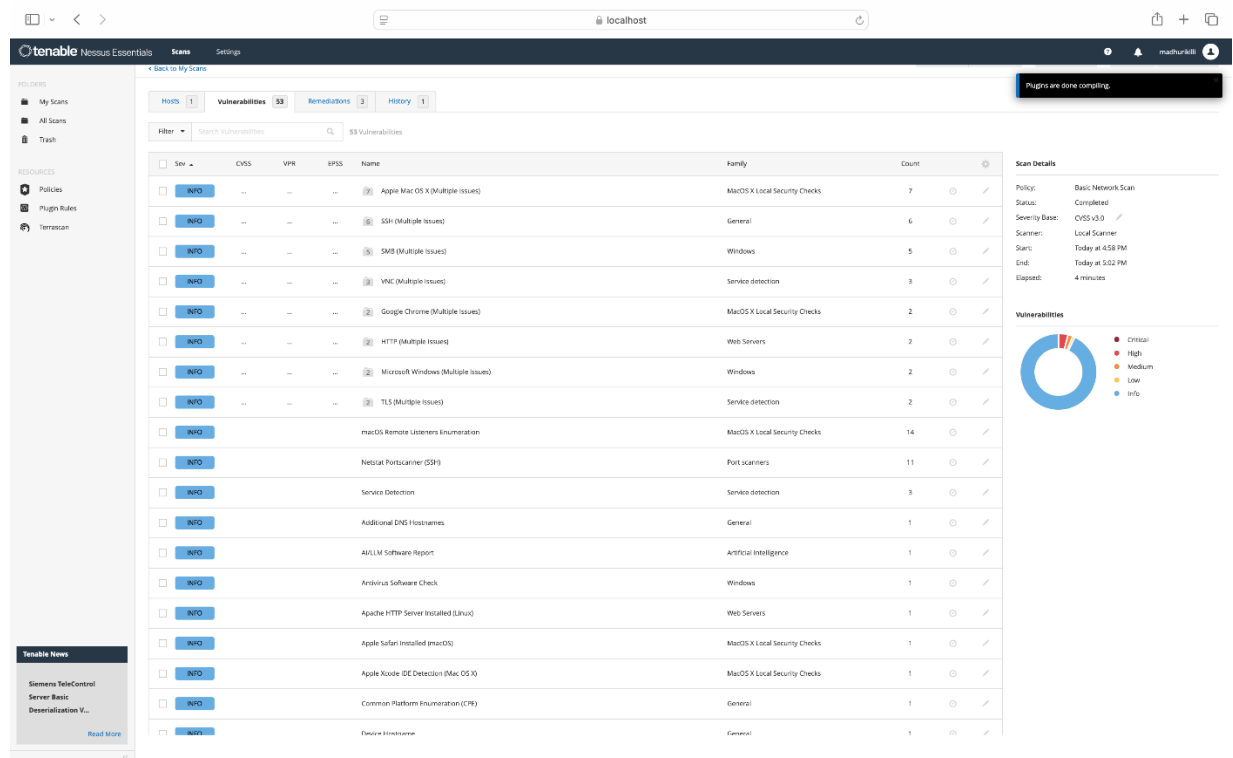


Figure 14. Detected Vulnerabilities – Info

The first medium-severity vulnerability discovered affects the Network Time Protocol (NTP). Mode 6 queries are handled by the NTP server, and devices set up to process these requests may be susceptible to NTP amplification attacks. By submitting a specifically constructed mode 6 query, a remote attacker might take advantage of this vulnerability and perhaps start a reflected Distributed Denial of Service (DDoS) attack. In order to reduce this risk, Nessus suggests limiting how NTP mode 6 requests are handled.

The remaining three medium vulnerabilities are connected to SSL certificates. SSL certificates are critical for protecting website communication since they enable the transition from HTTP to HTTPS. These certificates contain critical information that devices may use to validate the server's validity. However, the X. 509 outcomes is untrustworthy due to flaws in the certificate chain. Possible reasons for this include:

- When a self-signed certificate is the top certificate or when there are no intermediate certificates linking the root certificate to a trusted authority, server's root couldn't have been issued by a acknowledged or allowed authority.
- A mismatch or invalid signature in the certificate network, possibly due to the use of an unsupported signature algorithm, which renders the certificate invalid.

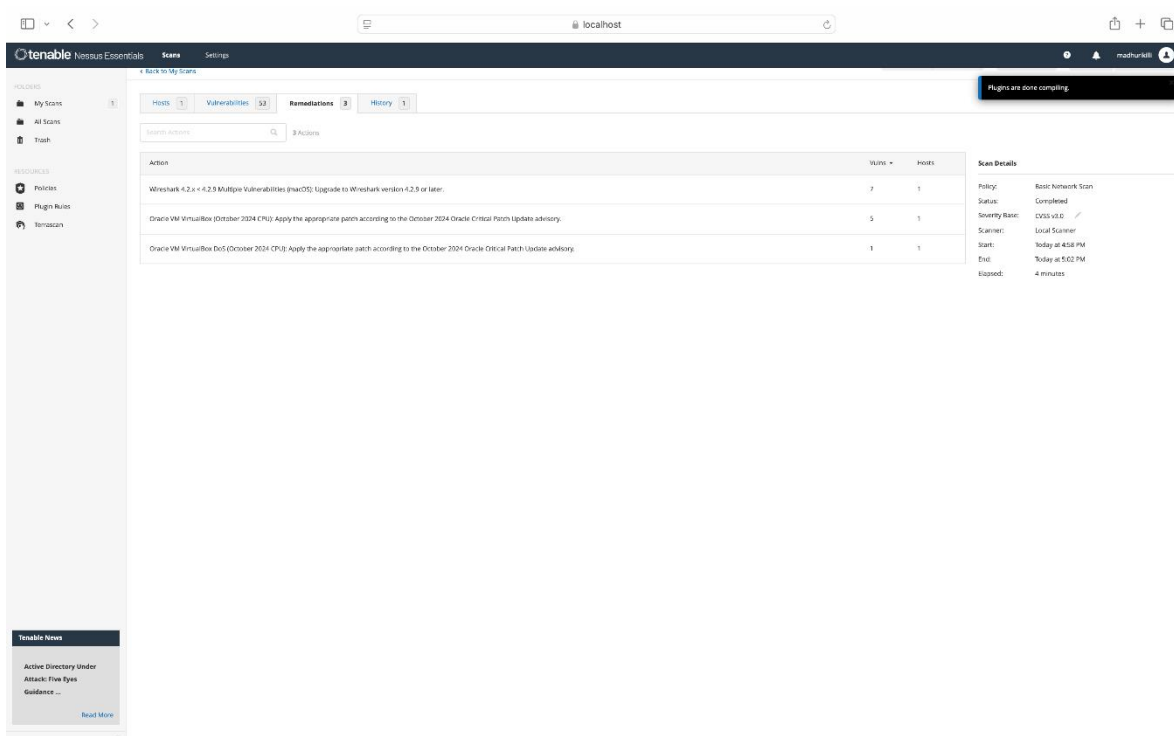
Any disruption in the certificate chain can hinder users from verifying the web server's identity, especially when the server is a public production host. This vulnerability could facilitate man-in-the-middle attacks [17], where attackers intercept or impersonate a legitimate participant in a communication. By inserting themselves in the middle, attackers can secretly capture the exchanged information, compromising the integrity and confidentiality of the data.

The third detected problem, SSL Certificate Expiry, is related to the plugin's ability to verify the expiry of SSL certificates connected. If any of the certificates have expired, the plugin will identify this as a vulnerability.

Nessus provided simple solutions to address these issues, such as acquiring or obtaining a valid SSL certificate for the service, ensuring the certificates are

signed by a recognized certificate authority and are kept up to date.

During the automation penetration test, we analyzed the distributed firewall and collected logs from pfSense. **Figure-12** plots the used-ports by the scanner during analysis, highlighting the difference between TCP and UDP protocols on port 53684. Transmission Control Protocol ensures the reliable packets delivery in priority, while UDP on the same port is less reliable, with the possibility of datagrams arriving out of sequence, duplicated, or missing entirely without notification. This discrepancy arises because UDP does not guarantee error checking or correction at the network interface level, unlike TCP.



**Figure 15.** Scan details

Firewall-logging is an integral feature of a comprehensive audit strategy. Serving as a protective shield, a firewall prevents unauthorized access to your computer or network. A critical feature of this system is its capability to log details of each connection attempt, including the identity of the individual making the attempt and the timestamp. This data is vital for solving the issues, audit assessments, and various apps. In our testing, we observed no collisions or errors concerning traffic on the Interface-Statistics page of pfSense device.

Furthermore, pfSense facilitated the vulnerability scanning process without any disruptions. In proposed setting, the firewall gets utilized to guarantee effective performance.

Nessus is a highly regarded penetration-testing tool utilized by numerous organizations globally. It is specifically designed to scan IP addresses, websites, and sensitive information to detect issues such as outdated software, malware, and mobile device vulnerabilities. Nessus provides a comprehensive dashboard, robust scanning functionalities, and a variety of reporting formats to assist users in identifying and addressing vulnerabilities within their systems.

While there are numerous penetration testing and vulnerability scanning tools available, the overarching goal remains the same: to protect a company's assets from unauthorized access. Experienced penetration testers can uncover vulnerabilities, allowing for improvements in system security. Nessus, for example, is regularly updated with over 70,000 plugins, providing thorough coverage for remote and local security checks. The system is structured around a client-server architecture and features an integrated scripting language for the analysis of plugins.

Nessus can produce reports in different file types. It accommodates two main categories of scans:

1. **Traditional Active Network-based Scans**
2. **Agent-based scans**

Both scan types have their advantages and disadvantages. While agent scans are useful for environments with intermittent network connectivity, traditional scans are often sufficient if network-wide visibility is not a concern. To attain thorough visibility across networks, Tenable suggests that most organizations employ a hybrid approach that integrates both agents and conventional scanning methods.

## 4.2 Third-Party Software Vulnerabilities

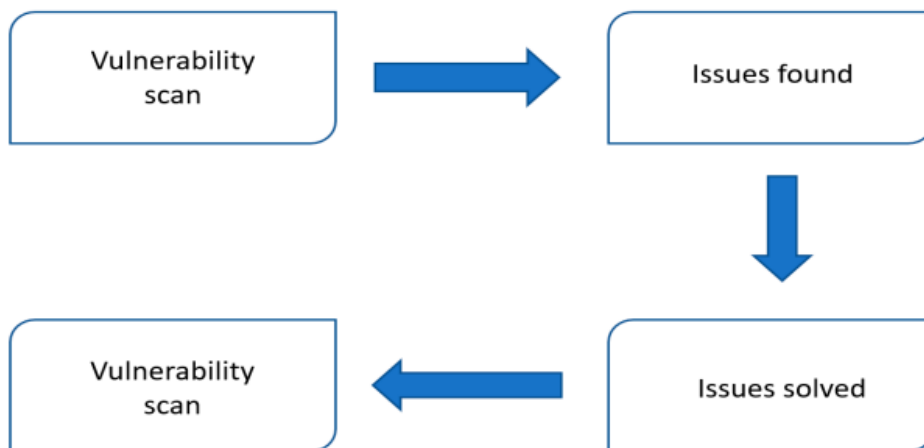
A fundamental framework often employed in vulnerabilities initiatives is the CVSS. Numerous vulnerabilities scanning tools leverage CVSS to evaluate severity of vulnerabilities. The CVE list, catalogs known vulnerabilities with exposures that are publicly disclosed. Additionally, the NVD, aligns its records with the M-CVE list.

A notable public concern is CVE-2022-42247, which relates to vulnerabilities associated with Cross-site scripting. These vulnerabilities frequently emerge when erroneous data is introduced creates web pages using unreliable or incorrect data. This problem generally occurs during the page generation process, where a program does not adequately filter data that may contain executable content for browsers, such as JScript, HTML elements and more. Consequently, the web page may harbor harmful scripts that execute within the victim's browser, posing significant security risks. To detect and address XSS vulnerabilities, several methods can be employed. Automated static analysis tools designed to target this vulnerability can help, but they are not always fully accurate, especially when multiple components are involved. Modern solutions often use data flow analysis to reduce false positives, but achieving 100% accuracy remains challenging. To further mitigate XSS risks, resources and automated tools can be used. The Cheat Sheet provides a range of strategies. The **Nessus scanner**, combining both static analysis and test-generation techniques, helped identify this issue during the security audit conducted for our experiment.

## Chapter 5

### IMPLEMENTATION

1 In the preceding section, we examined several techniques for pinpointing security vulnerabilities in our infrastructure. Figure 13 depicts the methodology for identifying these weaknesses within our proposed architecture. In the prior chapter, we performed security scans utilizing various tools to reveal possible security concerns.



**Fig.** Evaluation flow chart

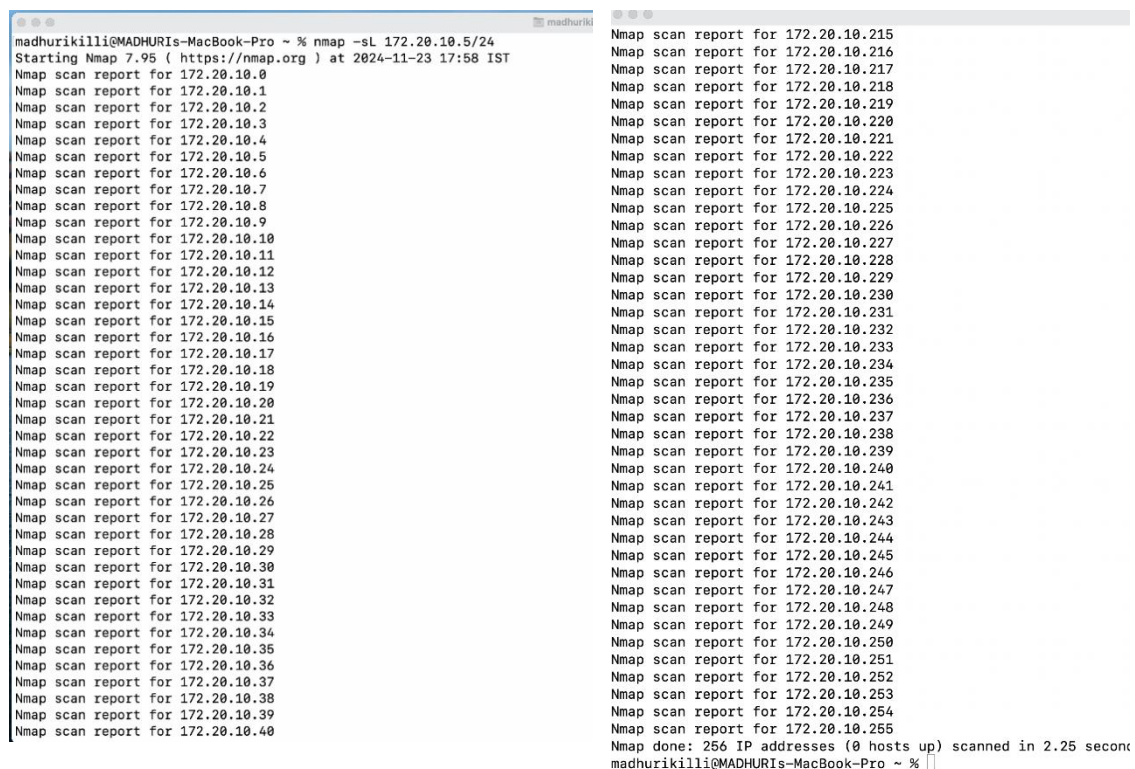
Here, We address solutions to identified challenges and emphasize advantages of firewall.

To address the discovered vulnerabilities, multiple approaches can be employed:

- **Packet Analysis and Rule Creation:** Tools like Wireshark [4], can capture incoming packets for analysis. By inspecting the traffic during scanning, we can create custom Suricata[5] or Snort rules[6] to block certain scanning methods. While this approach is effective, other scanning tools employing different techniques may bypass these rules.
- **Custom Firewall Configurations:** Another solution involves applying custom

configurations within the main firewall to resolve the identified issues.

Through the utilization of Nmap, we successfully detected the router and identified unclosed-ports that were susceptible to exploitation. Subsequent enhancements to pfSense led to notable improvements in security outcomes, as evidenced by the revised Nmap scans. These modifications effectively obscured our firewall in reconnaissance, as plotted below. The absence of router visibility significantly hampers the efficacy of further penetration testing, as it limits access to essential information. PfSense provides multiple layers of defense against a variety of challenges encountered in local networks, thereby affirming its effectiveness as a firewall solution. In relation to automated penetration testing conducted with Nessus, we addressed the identified vulnerabilities and achieved enhanced results, which are elaborated upon in the following sections.



```

madhurikilli@MADHURIs-MacBook-Pro ~ % nmap -sL 172.20.10.5/24
Starting Nmap 7.95 ( https://nmap.org ) at 2024-11-23 17:58 IST
Nmap scan report for 172.20.10.0
Nmap scan report for 172.20.10.1
Nmap scan report for 172.20.10.2
Nmap scan report for 172.20.10.3
Nmap scan report for 172.20.10.4
Nmap scan report for 172.20.10.5
Nmap scan report for 172.20.10.6
Nmap scan report for 172.20.10.7
Nmap scan report for 172.20.10.8
Nmap scan report for 172.20.10.9
Nmap scan report for 172.20.10.10
Nmap scan report for 172.20.10.11
Nmap scan report for 172.20.10.12
Nmap scan report for 172.20.10.13
Nmap scan report for 172.20.10.14
Nmap scan report for 172.20.10.15
Nmap scan report for 172.20.10.16
Nmap scan report for 172.20.10.17
Nmap scan report for 172.20.10.18
Nmap scan report for 172.20.10.19
Nmap scan report for 172.20.10.20
Nmap scan report for 172.20.10.21
Nmap scan report for 172.20.10.22
Nmap scan report for 172.20.10.23
Nmap scan report for 172.20.10.24
Nmap scan report for 172.20.10.25
Nmap scan report for 172.20.10.26
Nmap scan report for 172.20.10.27
Nmap scan report for 172.20.10.28
Nmap scan report for 172.20.10.29
Nmap scan report for 172.20.10.30
Nmap scan report for 172.20.10.31
Nmap scan report for 172.20.10.32
Nmap scan report for 172.20.10.33
Nmap scan report for 172.20.10.34
Nmap scan report for 172.20.10.35
Nmap scan report for 172.20.10.36
Nmap scan report for 172.20.10.37
Nmap scan report for 172.20.10.38
Nmap scan report for 172.20.10.39
Nmap scan report for 172.20.10.40
Nmap scan report for 172.20.10.215
Nmap scan report for 172.20.10.216
Nmap scan report for 172.20.10.217
Nmap scan report for 172.20.10.218
Nmap scan report for 172.20.10.219
Nmap scan report for 172.20.10.220
Nmap scan report for 172.20.10.221
Nmap scan report for 172.20.10.222
Nmap scan report for 172.20.10.223
Nmap scan report for 172.20.10.224
Nmap scan report for 172.20.10.225
Nmap scan report for 172.20.10.226
Nmap scan report for 172.20.10.227
Nmap scan report for 172.20.10.228
Nmap scan report for 172.20.10.229
Nmap scan report for 172.20.10.230
Nmap scan report for 172.20.10.231
Nmap scan report for 172.20.10.232
Nmap scan report for 172.20.10.233
Nmap scan report for 172.20.10.234
Nmap scan report for 172.20.10.235
Nmap scan report for 172.20.10.236
Nmap scan report for 172.20.10.237
Nmap scan report for 172.20.10.238
Nmap scan report for 172.20.10.239
Nmap scan report for 172.20.10.240
Nmap scan report for 172.20.10.241
Nmap scan report for 172.20.10.242
Nmap scan report for 172.20.10.243
Nmap scan report for 172.20.10.244
Nmap scan report for 172.20.10.245
Nmap scan report for 172.20.10.246
Nmap scan report for 172.20.10.247
Nmap scan report for 172.20.10.248
Nmap scan report for 172.20.10.249
Nmap scan report for 172.20.10.250
Nmap scan report for 172.20.10.251
Nmap scan report for 172.20.10.252
Nmap scan report for 172.20.10.253
Nmap scan report for 172.20.10.254
Nmap scan report for 172.20.10.255
Nmap done: 256 IP addresses (0 hosts up) scanned in 2.25 seconds
madhurikilli@MADHURIs-MacBook-Pro ~ %

```

Fig. Scan with -sL command with issues fabricated.



```
madhurikilli@MADHURIs-MacBook-Pro ~ % nmap -sn 172.20.10.5/24
Starting Nmap 7.95 ( https://nmap.org ) at 2024-11-29 11:45 IST
Nmap scan report for 172.20.10.5
Host is up (0.00071s latency).
Nmap done: 256 IP addresses (1 host up) scanned in 21.52 seconds
madhurikilli@MADHURIs-MacBook-Pro ~ %
```

**Fig.** Nmap scan with -sn

Above plot illustrates the report produced following the resolution of the previously identified issues. The majority of vulnerabilities were mitigated by enhancing firewall through range of strategies, adjusting the audit rules, deactivating unnecessary operations, and modifying particular files. Given that pfSense provides a variety of packages that can be directly implemented within firewall, there was no requirement for supplementary software solutions to address these concerns. After rectifying all medium-level vulnerabilities, only two informational issues persist :

1. **MAC Address**
2. **Scan Information:** This refers to details about the scan itself, which do not impact the functionality or security of our proposed solution.

The concluding phase of the penetration testing process involves the effective remediation of vulnerabilities identified on the Internet through the updating of pfSense to its most recent stable version. This approach is characterized by its lack of associated costs and its efficiency in terms of time.

## Chapter 6

### HARDWARE/SOFTWARE TOOLS USED

#### 6.1 HARDWARE:

##### 6.1.1 Specifications

Depending on the scale, typical specifications might include multiple CPUs, 16GB+ RAM, and SSD storage.

## Chapter 7

### RESULTS & DISCUSSION

This section presents a summary of the findings and assessments related to the proposed audit methodology, with an emphasis on the results derived from previous penetration tests. The audit process is structured into four distinct phases: performing both manual and automated vulnerability assessments, aligning the results with the vulnerability database to rectify identified concerns, and ultimately evaluating the overall risk to the system.

Prior to exploring analysis outcomes, it's crucial to create a framework for breach assessment and the evaluations utilized. Effective cybersecurity risk management encompasses the ongoing identification, evaluation, analysis, and mitigation of security threats faced by an organization. Importantly, the responsibility for managing these risks is a shared obligation that involves not only the security team but the entire organization as a whole.

Employees and business unit leaders often view risk management through the narrow scope of their specific roles, lacking the comprehensive perspective required for consistent and holistic risk handling. This challenge is compounded by the complexity of enterprise-wide risk management due to factors such as the growing reliance on third-party vendors, evolving technologies, and increasingly stringent regulatory landscapes. The COVID-19 pandemic and the accompanying economic downturn have further heightened the demands on compliance teams, expanding their things need to do while reducing resources.

Risk reporting is a crucial aspect of our study, providing scenarios that assess threats based on their likelihood and severity. Unlike financial evaluations, it relies on professional judgment, intuition, and experience to prioritize risks. The following formula outlines the risk assessment approach used in our evaluation

$$\text{Risk} = \text{Impact} \times \text{Threat} \times \text{Vulnerability}$$

1 We applied the following scale for evaluating vulnerabilities: **Low-level vulnerabilities** have an impact score between 1 and 16, **Moderate-level vulnerabilities** range from 17 to 32, **High-level vulnerabilities** from 33 to 48, and **Critical vulnerabilities** from 49 to 64.

Through our research, we identified several key insights.

In the **manual testing phase**, minor adjustments addressed the detected issues effectively. For instance, filtering open ports significantly enhanced security, allowing us to classify the associated risks within the **Low-level vulnerability range**.

In the **automated testing phase**, we uncovered informational and medium-level vulnerabilities. These were similarly resolved using straightforward methods, as demonstrated. Consequently, the risks in this phase also fall under the **Low-level vulnerability range**.

1 Lastly, during our **internet-based research** on the software supporting our proposed solution, we observed that all identified vulnerabilities could be resolved through timely updates and upgrades. Ensuring the software is up to date delivers essential patches that mitigate known issues. However, we acknowledge the potential risk of zero-day vulnerabilities, which may exploit unknown flaws not yet identified by the broader community. Addressing such vulnerabilities promptly through collaboration between the community and the provider is critical. Considering the effectiveness of updates, this aspect of our solution is also categorized in the **Low-level vulnerability range**.

## Chapter 8

### CONCLUSION

1 In our study, we conducted a security audit of an open-source distributed firewall to highlight the challenges of achieving comprehensive network protection. The firewalls in our solution utilize extensions to define custom firewall rules for IPv4 and IPv6 address spaces, enabling precise control over incoming and outgoing traffic across single or multiple interfaces. We emphasized the importance of combining manual and automated testing to achieve an efficient and thorough audit, as each method has its unique strengths and limitations.

Our findings demonstrate that the proposed solution achieves a high-security status based on the tests conducted. While we relied on open-source tools, we acknowledge that more advanced solutions for penetration testing are available in the market, though they are costly and require significant expertise. Cyber-attackers often start with commonly available open-source tools, like those we used, which highlights the importance of addressing fundamental vulnerabilities to enhance network security significantly.

1 For future work, we propose utilizing additional open-source tools to uncover further vulnerabilities and enhance the efficiency of pfSense. We also plan to test various cyber-attack scenarios on our distributed firewall to validate its robustness. While previous studies have explored different cyber-attacks and their solutions, this research aims to address emerging challenges in the field. Our solution integrates multiple layers of protection, including optimized IDS/IPS and dynamic firewall rules. These measures provide real-time data, such as logs, which can be analyzed to deepen the understanding of network traffic and potential threats.