IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Practical Vulnerability Detection And Penetration Testing Using Wireshark, Nmap, And Metasploit

Deven Sonawane[1], Ajaz Ali[1], Swami Rajput[1], Santosh Kushwaha[1], Archana Rane[2]

[1,2]Department of MCA, K. K. Wagh Institute of Engineering Education and Research, Nashik, India

Abstract

This study examines the use of three key cybersecurity tools Wireshark, Nmap, and Metasploit for effective vulnerability analysis in IT environments across public and private organizations. Wireshark helps capture and analyse real-time network traffic, Nmap is used to discover open ports and potential weaknesses, and Metasploit supports ethical penetration testing by simulating real attack scenarios. When combined, these tools enable organizations to identify security issues early, strengthen defences, and reduce the risk of cyber threats. The paper emphasizes the importance of regularly using these tools within a structured vulnerability management framework to improve overall cybersecurity.

Keywords—Cybersecurity, Cyber Threats, Vulnerability Assessment and management, OpenVAS, Wireshark, Nmap, Metasploit.

I. INTRODUCTION

Vulnerability analysis and assessment are important processes that help organizations protect their systems from cyber threats. By identifying weaknesses in networks, applications, and infrastructure, organizations can take preventive actions before attackers exploit them. This also supports better planning of security resources, helps meet compliance standards, and reduces financial and reputational damage caused by security breaches.

Regular vulnerability assessments are necessary because cyber threats continue to evolve. They help maintain strong security measures, improve incident response, and protect the confidentiality, integrity, and availability of important data. These practices also build trust among users and stakeholders by showing a commitment to protecting digital assets.

Overall, vulnerability analysis is a key part of a strong cybersecurity strategy, enabling organizations to stay prepared and secure. To improve protection, tools such as firewalls, intrusion detection systems, encryption, and routine security audits should be used together for effective defence.

II. LITERATURE REVIEW

The rapidly evolving landscape of cybersecurity has prompted extensive research on the identification and mitigation of system vulnerabilities. Many studies focus on the various methods attackers employ to exploit weaknesses in both hardware and software, highlighting the need for advanced security protocols to protect sensitive information. As cyber threats continue to grow in complexity, vulnerability scanning, network assessments, and automated security measures have become essential tools in detecting and preventing attacks. The following are detailed studies of previously conducted research.

Irfan Yaqoob et al. [1] highlight increasing cyber threats and explain how attackers exploit system weaknesses to access sensitive information. The study stresses the need for strong security measures to protect data.

Nikita Jhala et al. [2] focus on vulnerability scanning and network assessment using tools like Nmap and OpenVAS. Their research introduces an improved scanning tool that maps networks, detects open ports, identifies vulnerabilities, and supports automated scanning for compromised devices.

Sudhanshu Raj et al. [3] discuss the role of the Metasploit Framework in penetration testing. Their work explains how Metasploit helps security experts test system weaknesses and develop exploit modules.

Wei Chen et al. [4] explore enhancements in Metasploit for bypassing antivirus detection. The study highlights advanced evasion techniques that assist penetration testers and researchers in evaluating defensive tools.

Mandeep Singh et al. [5] use Metasploitable 2 as a vulnerable testing environment to examine security tools and identify system vulnerabilities without legal risks. Their research contributes to improving system security through controlled testing.

Prashant S. Shinde et al. [6] provide an overview of Vulnerability Assessment and Penetration Testing (VAPT). The paper emphasizes the importance of identifying loopholes in IT infrastructure to protect organizations from cyber-attacks.

Alayna Kennedy et al. [7] discuss the need for ethical auditing tools in machine learning projects to avoid biased outcomes and ensure safe decision-making in government applications.

S. Khan et al. [8] present different vulnerability assessment approaches, including automated and manual methods. Their work states that automated tools are cost-effective and reduce analysis time.

S. Raj et al. [9] explain penetration testing processes using Metasploit to detect security weaknesses and document vulnerabilities through simulated attacks.

M. Muharrom et al. [10] describe OpenVAS as a flexible and powerful open-source vulnerability scanner that provides comprehensive scanning capabilities through a user-friendly interface.

III. VULNERABILITY ANALYSIS

Vulnerability analysis is the process of finding and evaluating security weaknesses in systems, networks, or applications. It involves both manual and automated testing methods to identify issues based on their severity and potential impact. This helps organizations detect weaknesses before attackers can exploit them.

A vulnerability can be:

- A software bug or design flaw that can be misused to harm a system.
- A weakness in internal infrastructure that may lead to a security breach.

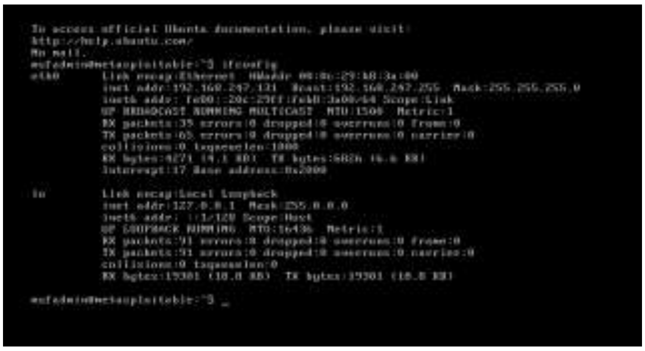
Vulnerability analysis is important for several reasons:

- **Risk Reduction:** It helps organizations discover security flaws early and take action to prevent attacks.
- **Strengthening Security:** Regular assessments improve the overall security framework by fixing weak points in systems and processes.
- **Regulatory Compliance:** Many industries require regular vulnerability assessments to meet security standards and legal policies.
- **Incident Prevention:** Finding vulnerabilities early helps avoid data leaks, system failures, and cyber-attacks.
- **Better Resource Planning:** By ranking vulnerabilities according to risk, organizations can focus on the most critical threats first.
- **Protecting Sensitive Data:** Reducing weaknesses helps prevent unauthorized access and protects confidential information.
- **Continuous Improvement:** Since cyber threats constantly change, ongoing assessments help organizations stay updated and improve defences over time.

IV. METHODOLOGY

The methodology adopted for vulnerability assessment follows a structured, multi-step approach to identify, evaluate, and mitigate potential security risks within the system. The process ensures a thorough evaluation of network and system vulnerabilities to help strengthen overall security posture. includes the following steps:

1. **Planning and Preparation:** Define the scope, objectives, and requirements of the assessment. Arrange necessary tools, team members, and access permissions.
2. **Data Collection:** Gather information about network assets, system components, and data flow. Identify possible internal and external threats.
3. **Vulnerability Identification:** Use automated scanners and manual testing to detect weaknesses. Review system settings and configurations to find security gaps.
4. **Vulnerability Analysis:** Assess the severity and potential impact of each vulnerability. Prioritize them based on the level of risk.
5. **Reporting:** Document all identified vulnerabilities along with their risk levels and effects. Provide clear recommendations for fixing issues.
6. **Remediation:** Apply patches, configuration changes, or updates to fix vulnerabilities. Verify that the corrective actions reduce the associated risks.
7. **Continuous Monitoring:** Perform routine scans to detect new vulnerabilities and keep systems updated. Link assessment results with incident response processes.
8. **Review and Improvement:** Evaluate the effectiveness of the assessment, update security policies, and train staff to improve overall security awareness.



```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:b8:3a:00
          inet addr:192.168.247.131  Bcast:192.168.247.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:feb8:3a00/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:35 errors:0 dropped:0 overruns:0 frame:0
          TX packets:65 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4271 (4.1 KB)  TX bytes:6826 (6.6 KB)
          Interrupt:17 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:91 errors:0 dropped:0 overruns:0 frame:0
          TX packets:91 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19301 (18.8 KB)  TX bytes:19301 (18.8 KB)

msfadmin@metasploitable:~$ _
```

Figure 1 shows the Metasploitable2 interface, which provides a vulnerable environment used for testing and evaluating security tools.

V. SIMULATION AND ANALYSIS

This section presents the simulated security environment and analyzes the results obtained from vulnerability scanning and penetration testing activities. Metasploitable2 is a deliberately vulnerable Linux-based virtual machine used to simulate real-world attack scenarios and analyze system weaknesses during security testing. In this research, all security testing, vulnerability scanning, and penetration testing activities were carried out on a Metasploitable2 virtual machine. Metasploitable2 is a purposely vulnerable Linux-based environment designed for learning and evaluating cybersecurity tools and techniques in a controlled and legal setup. It provides multiple built-in vulnerabilities that allow researchers to perform real-world attack simulations without risking production systems.

The machine was configured with a network interface (eth0) operating on a private subnet. As shown in the system information, the IP address assigned to the machine is 192.168.247.131, enabling communication within the local virtual network environment. The ifconfig output also confirms that the network interface is active, transmitting and receiving packets successfully. A loopback interface (lo) is also present for internal system communication.

This controlled setup ensures a safe environment to execute scanning tools like Nmap, OpenVAS, Wireshark, and Metasploit, allowing the detection and exploitation of vulnerabilities for research and educational purposes. Using Metasploitable2 prevents any legal or ethical concerns because all testing is performed on a deliberately insecure system intended for cybersecurity practice.

A. Nmap

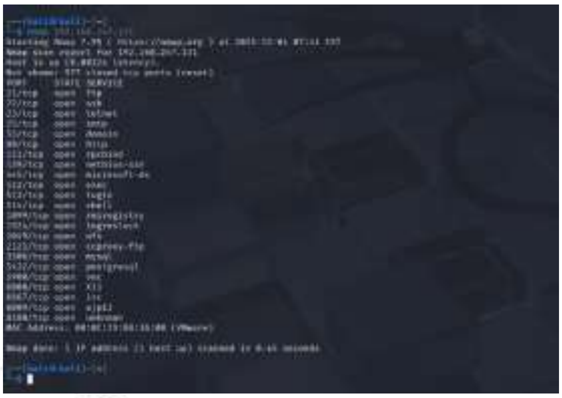
Nmap (Network Mapper) is a widely used open-source tool for network scanning and security auditing. It helps identify active devices on a network, the services they are running, operating systems, open ports, and firewall configurations. Nmap works using different protocols such as TCP, UDP, IP, and ICMP to gather information about the target system.

Using the **"-sV"** option, Nmap can detect the version of running services, which helps determine vulnerabilities by comparing results with online security databases.

Below are some common Nmap commands used during security assessment:

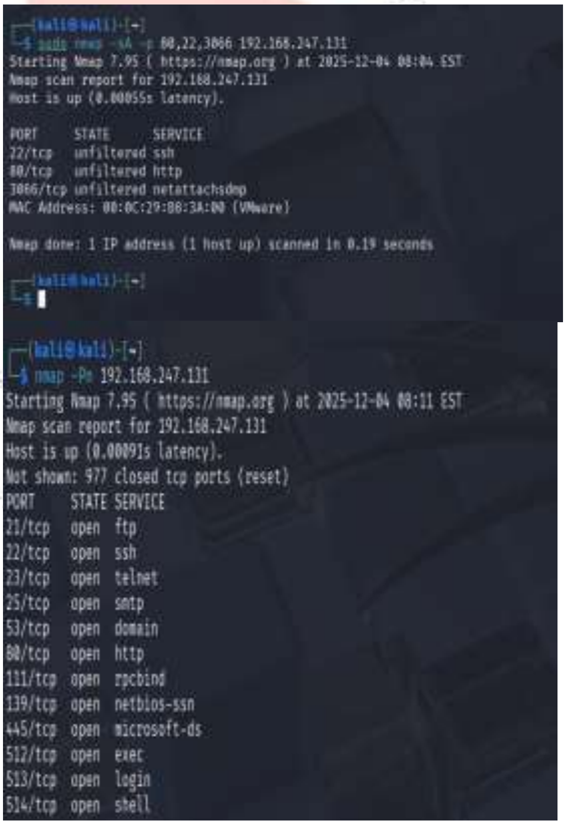
Command	**Purpose**
nmap -sn <target>	Discovers active hosts within a network (ping scan).
nmap -p- <target>	Scans all ports to identify which ones are open.
nmap -sV <target>	Detects the service and version running on each open port.
nmap -A <target>	Performs an advanced scan including OS detection, version detection, scripting, and traceroute.
nmap --script=<script-name> <target>	Runs NSE scripts to check for specific vulnerabilities or deeper system details.
nmap -oA <output-file> <target>	Saves the scan results in different formats (TXT, XML, etc.).

Nmap is a powerful and flexible tool for ethical hackers, network administrators, and security analysts to evaluate system security and uncover potential risks.



```
┌──(kali㉿kali)-[~]
└─$ nmap 192.168.247.131
Starting Nmap 7.95 ( https://nmap.org ) at [illegible]
Nmap scan report for 192.168.247.131
Host is up ([illegible] latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: [illegible] (VMware)

Nmap done: 1 IP address (1 host up) scanned in [illegible] seconds

┌──(kali㉿kali)-[~]
└─$ 
```

Figure 1.1. Demonstration of basic nmap scan



```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sA -p 80,22,3066 192.168.247.131
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-04 08:04 EST
Nmap scan report for 192.168.247.131
Host is up (0.00055s latency).

PORT     STATE      SERVICE
22/tcp   unfiltered ssh
80/tcp   unfiltered http
3066/tcp unfiltered netattachsdmp
MAC Address: 00:0C:29:88:3A:00 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds

┌──(kali㉿kali)-[~]
└─$ 
```

```
┌──(kali㉿kali)-[~]
└─$ nmap -Pn 192.168.247.131
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-04 08:11 EST
Nmap scan report for 192.168.247.131
Host is up (0.00091s latency).
Not shown: 977 closed tcp ports (reset)
PORT    STATE SERVICE
21/tcp  open  ftp
22/tcp  open  ssh
23/tcp  open  telnet
25/tcp  open  smtp
53/tcp  open  domain
80/tcp  open  http
111/tcp open  rpcbind
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
512/tcp open  exec
513/tcp open  login
514/tcp open  shell
```

Figure 2.2. no ping scan

Figure 2.1 shows scanning of a metasploitable2 machine. The first command scans the machine's IP address "192.168.247.131", hence giving the information about port number, its state(open/closed), and the service running on that port. In figure 2.2 the second command uses "-Pn" on a subnet, hence skipping the host discovery stage and directly starts scanning.

In Figure 3.1, the first command performs an aggressive scan on an IP address, hence giving the information about the operating system, version of services running on the open ports, scans the scripts and performs a traceroute on the target host. The second command (In Figure 3.2) makes use of TCP ACK scan, on particular ports (here, 80, 22,3306) on an IP address, hence giving the state of the port (unfiltered), protocol used, and the services running on those ports.

Figure 3.1. Scanning an IP address using stealth scan

Figure 4. Scan on a IP address with aggressive scan and scan delay

Figure 3.2. TCP ACK scan

Figure 4 is a compilation of multiple options available such as, -sT, performing TCP connect scan, -T5, setting the timing of the scan to be "aggressive", hence speeding up the scanning process. The results of the scan are stored in an XML file, using -oX. The scan delay is used to control the speed of the scan, and to rescue network congestion. Figure 5 shows the port scanning options along with the description.

Command	Scan Type Discovery	Description
nmap	Default Scan	Checks if the host is up without scanning ports.
nmap -sT	TCP Connect Scan	Full TCP connect scan with service heuristics.
nmap -sS	SYN/Half-open Scan	Stealth scan, easy to detect.
nmap -sU	UDP Scan	Scans UDP ports (slow but important).
nmap -sN	Null Scan	Firewall rugs; stealth detection.
nmap -sF	FIN Scan	FIN flag stealth scan.
nmap -sX	Xmas Scan	FIN/PSH/URG: firewall evasion.

Figure 5. Port scanning options

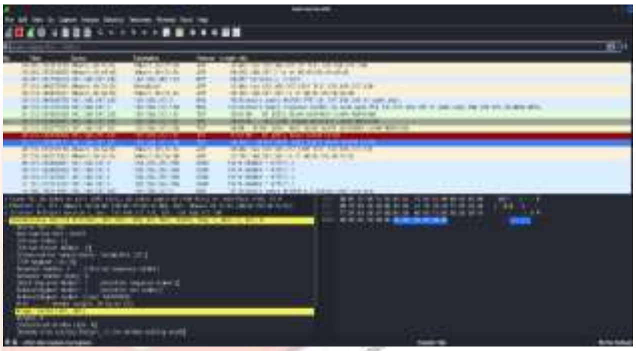


Figure 7. Wireshark demonstrates the packet captures in scanning a subnet

B. Wireshark

Wireshark is a widely used open-source tool for analysing network traffic. It allows users to capture, view, and inspect data packets traveling across a network in real time. It is commonly used for troubleshooting network issues, studying network behaviour, improving software performance, and learning about networking concepts.

Integrating wireshark with n-map for packet sniffing and network scanning:

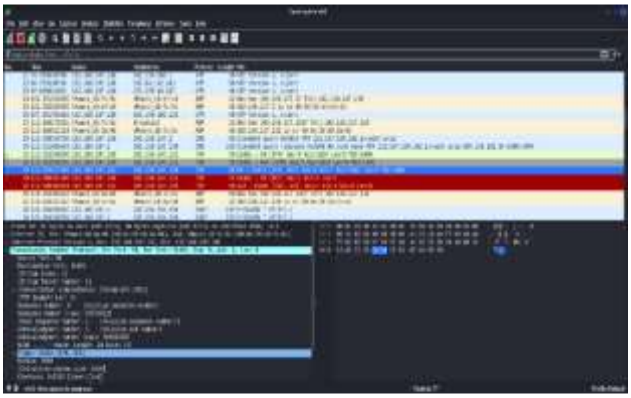


Figure 6. Wireshark demonstrating packet capture

Figure 6 shows the packet capture using wireshark along with a nmap scan performed on http (port 80), and https (port 443), on the subnet. Wireshark is used along with the scan to analyse the packets. The source IP address, destination IP address, the protocol used, and the length of the packet sent/received can be found from the capture. By capturing the packets, any anomaly can be easily detected, hence helping in assessing the nature and the origin of the vulnerabilities.
Figure 7 has the RST, reset TCP flag, indicating the abrupt termination of the connection, highlighted in red. This may happen due to connection reset, closed port, or firewall rejection.

C. Metasploit Framework

Metasploit is a widely used penetration testing framework designed to find and exploit security weaknesses in networks, systems, and applications. It was first created by **H.D. Moore in 2003** and originally developed using Perl. Later, in **2007**, the entire framework was rewritten in **Ruby**, and in **2009**, **Rapid7** acquired the Metasploit project and continues to maintain and enhance it. For this research work, **Metasploitable2** is used as the testing platform. It is a deliberately vulnerable Linux-based virtual machine designed for learning and practicing penetration testing techniques. It works seamlessly with Metasploit, making it ideal for performing real-world attack simulations and security testing.

Metasploit helps security professionals:

- Identify system vulnerabilities
- Test and validate security controls
- Develop and run exploit modules
- Practice ethical hacking techniques

By scanning the target system, Metasploit reveals open ports and services running on them, which can be analysed to determine possible attack paths and improve defensive strategies. Exploiting metasploitable2 using common vulnerability:

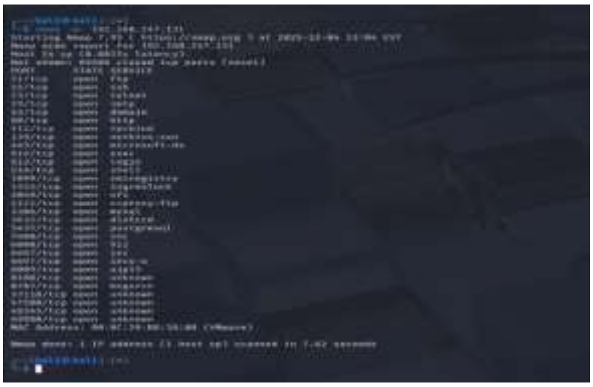
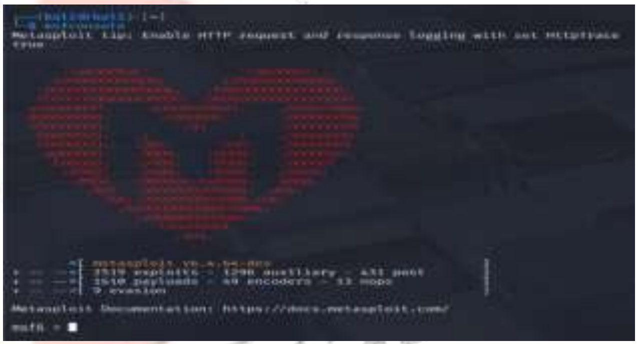


Figure 8. N-map scan for the metasploitable2 machine

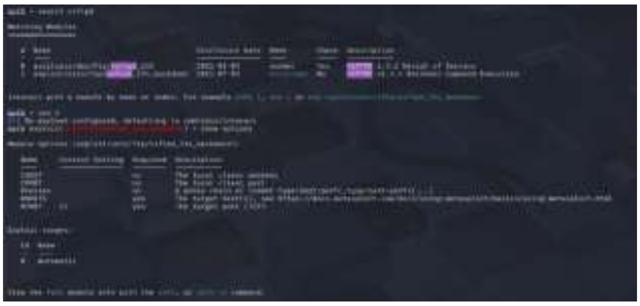


```
Metasploit tip: Enable HTTP request and response logging with set HttpTrace
true

       =[ metasploit v6.4.x-dev                          ]
+ -- --=[ 2519 exploits - 1298 auxiliary - 431 post      ]
+ -- --=[ 1510 payloads - 49 encoders - 13 nops          ]
+ -- --=[ 9 evasion                                      ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 >
```

Vsftpd 2.3.4:

Figure 9 shows the basic steps used to perform an exploit in the Metasploit framework. First, the tool is launched using the **msfconsole** command. Next, the **search vsftpd** command is used to locate the exploit module for the VSFTPD backdoor vulnerability. The command **use 1** is then entered to load the selected module. After that, the target machine's IP address is configured using **set RHOSTS**. Finally, the attack is executed using the **exploit** command. If the exploit is successful, it provides remote access to the

Figure 9. steps to exploit Vsftpd 2.3.4 vulnerability



target system, allowing further actions such as interacting with the system or attempting privilege escalation.

VI. FUTURE RESEARCH DIRECTIONS

Future research can focus on improving vulnerability assessment tools like OpenVAS to make them more effective and adaptable to modern security needs. One key direction is the use of machine learning to predict and prioritize vulnerabilities by analysing past data and current threat trends.

Another direction is to increase automation, allowing OpenVAS to integrate smoothly with other security tools for faster and more efficient vulnerability management. Enhancing support for cloud environments and container platforms such as Docker and Kubernetes is also important, as organizations increasingly rely on these technologies.

Further improvements can include strengthening IoT security to identify risks in connected devices, updating the tool for real-time threat intelligence, and developing a more user-friendly interface so both technical and non-technical users can operate it easily.

With the growing role of Artificial Intelligence (AI) in cybersecurity, future work may explore AI-based systems that analyze large volumes of data, detect unusual behavior, and anticipate attacks before they occur. AI-driven solutions can greatly improve threat detection and response, supporting stronger and smarter security strategies.

Conclusion

This research has provided an in-depth analysis of vulnerability scanning, penetration testing, and security measures using various tools and techniques. The use of the Metasploitable2 virtual machine as a testing environment allowed for practical exploration of common system vulnerabilities and security exploits. Through tools such as Nmap and Metasploit, the study demonstrated how attackers exploit system weaknesses and the importance of robust security practices to protect sensitive information. The findings underscore the need for continuous security assessments, the use of automated scanning tools, and effective network monitoring to mitigate the risks posed by ever-evolving cyber threats. Ultimately, implementing comprehensive security measures and staying vigilant against vulnerabilities is crucial in safeguarding systems and networks from potential attacks.

REFERENCES

[1] M. Alhamed, “A Systematic Literature Review on Penetration Testing in Network Security,” *Applied Sciences*, vol. 13, no. 12, 2023.

[2] Z. Al-Khazaali, “Characteristics of Port Scan Traffic: A Case Study Using Nmap-Generated Scans,” *JEASD Journal*, 2025.

[3] O. I. Omotosho, Y. P. Lawal, B. I. Ogunrinola, O. L. Ajayi, and A. A. Akintunde, “Enhancing Network Security Using Vulnerability Assessment and Penetration Testing,” *FUOYE Journal of Engineering and Technology (FUOYEJET)*, vol. 10, no. 1, pp. 76–83, 2025.

[4] S. H. V. Sanne, “Investigations into Security Testing Techniques, Tools, and Methodologies for Identifying and Mitigating Security Vulnerabilities,” *Journal of Artificial Intelligence, Machine Learning and Data Science*, vol. 1, no. 1, pp. 626–631, 2024.

[5] D. Everson, “A Survey on Network Attack Surface Mapping,” *ACM Computing Surveys*, 2024.

[6] “Penetration Testing for System Security: Methods and Practical Approaches,” Wei Zhang, Ju Xing, Xiaoqi Li, *arXiv preprint*, May 2025.

[7] “A Study on Vulnerability Scanning Tools for Network Security,” D. N. Railkar and S. Joshi, *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 2022.

[8] “Vulnerability Assessment & Penetration Testing for Applications Security,” A. S. Popuri et al., *IJERT*, 2023.

[9] “An Experimental Study on Detecting and Mitigating Web Application Vulnerabilities,” *IJSSE*, 2023.

[10] “Smart Home, It’s Vulnerability Assessment Through Penetration Testing,” K. Kondru, P. Kancharla, M. Darlanka, B. C. Sathuluri, *Journal of Informatics Electrical and Electronics Engineering (JIEEE)*, vol. 6, no. 1, 2025