

MANGALORE UNIVERSITY

Mangalagangothri, Mangalore Karnataka- 574199



An Internship Report on
“Vulnerability Assessment and Penetration Testing”

Submitted in Partial Fulfilment of the Requirement for the
Master Of Science in Cyber Security

Submitted By

Ananya N

Reg No. P05AZ23S031013

M.Sc. Cyber Security, Mangalore University

Under the Valuable Guidance Of

Internal Guide

Mr. Naveen Chandra kumar

Assistant Professor and Chairman of the Department of Electronics

Mangalore University

External Guide

Mr. Melvin John Lourdusamy

Global Practice Head and Director at Happiest Minds Technologies

Mangalore University

Mangalagangothri, Mangalore, Karnataka- 574199



CERTIFICATE

This is to certify that the internship report titled “**Vulnerability Assessment and Penetration Testing**” has been completed by **Ananya N** bearing register number **P05AZ23S031013**, in partial fulfilment for the award of degree in “**Master of Science in Cyber Security**” prescribed by Mangalore University, Mangalagangothri during the year 2023-24.

.....

Guide and Head of the Department

Mr. Naveen Chandra kumar

Assistant Professor and Chairman of the Department of Electronics

Mangalore University

Examiners

1.

2.

18th August 2025

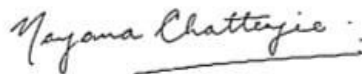
TO WHOM SO EVER IT MAY CONCERN

This is to certify that **Ananya N** from **Mangalore University, Konaje** has successfully carried out his/her project internship work in MSc in Cybersecurity as a part of his/her academic project in **Web Application Penetration Testing (VAPT, Web Application Security)** under the guidance of **Melvin John Lourdasamy, Director- IMSS**

The project is for a duration starting from **07-04-2025 to 07-08-2025**.

Her/His conduct during the period of training was found to be good. We wish him/her all success in his/her future endeavours.

for Happiest Minds Technologies Private Limited.,



Nayana Chatterjee –
Manager - PEOPLE PRACTICE CORP


Ananya N

ACKNOWLEDGEMENT

I would like to take this opportunity to sincerely thank everyone who played a role in the successful completion of my internship and this report. This experience has been an extremely valuable phase in my academic and professional journey.

First and foremost, I am deeply thankful to **Mr. Melvin John Lord Swamy**, Director and Manager at **Happiest Minds Technologies Limited**, for giving me the opportunity to be a part of the organization and for allowing me to work on real-time projects under his leadership. His expert guidance, constant encouragement, and trust gave me the confidence to explore and learn about the field of cybersecurity and Vulnerability Assessment and Penetration Testing (VAPT).

I am also truly grateful to the entire team at **Happiest Minds Technologies Limited** for their cooperation, support, and for creating a professional and collaborative work environment. The exposure I received while working alongside skilled professionals helped me gain a better understanding of how security testing is approached in the industry. Their willingness to share knowledge and involve interns in meaningful work contributed greatly to my learning.

I would like to express my heartfelt appreciation and respect to **Mr. Naveen Chandra Kumar**, Assistant Professor and Chairman of the Department of Electronics, Mangalore University, for granting me permission to carry out this internship and for his continuous support during this phase. As my internal academic guide, his encouragement and feedback helped me stay focused and committed to completing this internship with clarity and professionalism. Additionally, I am thankful to the faculty members of the **Department of Electronics, Mangalore University**, for their academic support, resources, and mentorship throughout the duration of my study, which laid the foundation for successfully applying my skills in a professional environment.

Finally, I would like to thank my family and friends for their unwavering moral support, patience, and motivation. Their encouragement kept me going during the challenging times and ensured I could complete my internship and report with dedication. I am sincerely thankful to everyone who contributed to this experience.

Sincerely,
Ananya N.

Company Introduction

Happiest Minds Technologies Limited

Happiest Minds Technologies Limited (NSE: HAPPSTMNDS) is a Mindful IT company that empowers enterprises and technology providers to embrace digital transformation. With a foundational belief in being “Born Digital. Born Agile,” the company focuses on delivering seamless customer experiences, enhanced business efficiency, and actionable insights by leveraging a wide range of advanced technologies.

Happiest Minds operates through three key business units: Product & Digital Engineering Services (PDES), Generative AI Business Services (GBS), and Infrastructure Management & Security Services (IMSS). These units enable the company to serve clients across various industries including Banking, Financial Services and Insurance (BFSI), EdTech, Healthcare & Life Sciences, Hi-Tech, Media & Entertainment, Manufacturing, Industrial, Energy & Utilities, and Retail, CPG & Logistics.

The company utilizes modern and disruptive technologies such as artificial intelligence, blockchain, cloud computing, digital process automation, internet of things (IoT), cybersecurity, robotics, drones, and virtual/augmented reality to deliver scalable digital solutions. Happiest Minds is known for its mindful work culture, strong leadership, and dedication to business ethics and governance. The company has received several prestigious recognitions, including awards from the Golden Peacock and ICSI, and is officially certified as a Great Place to Work™.

Happiest Minds has a global footprint, with operations in the United States, United Kingdom, Canada, Australia, and the Middle East. One of its major development centers is located at:

Happiest Minds Technologies Limited
Electronics City Phase 1, Adjacent to Infosys Campus,
No. 9/36, Doddathoguru Village, Begur Hobli,
Bengaluru – 560100, Karnataka, India.

With its global presence, technology expertise, and deep industry knowledge, Happiest Minds continues to be a trusted partner for organizations seeking digital innovation and excellence.

Abstract

In today's digital-first world, organizations are increasingly dependent on technology and online platforms for their operations, communication, and data management. This reliance has brought with it a growing concern about cybersecurity threats, which continue to rise in frequency and sophistication. Internet-facing systems, web applications, networks, and APIs are often prime targets for malicious attacks. In this context, **Vulnerability Assessment and Penetration Testing (VAPT)** has emerged as a vital process in securing digital assets and evaluating the effectiveness of an organization's security posture.

VAPT is a comprehensive security testing approach that includes two critical components. **Vulnerability Assessment (VA)** focuses on identifying, analyzing, and categorizing security flaws, misconfigurations, and potential loopholes in the system using automated scanners and manual inspection methods. It helps in creating an inventory of vulnerabilities that, if exploited, could lead to data breaches, system compromise, financial loss, or reputational damage. On the other hand, **Penetration Testing (PT)** simulates real-world cyberattacks to exploit the identified vulnerabilities. It aims to demonstrate how a threat actor could gain unauthorized access, escalate privileges, or exfiltrate sensitive information. This phase validates the actual risk and business impact of the vulnerabilities.

The VAPT process follows a structured lifecycle that includes information gathering, scanning, enumeration, vulnerability detection, exploitation, and detailed reporting. It helps organizations proactively detect security weaknesses before adversaries do and allows them to prioritize risks based on severity using models such as the Common Vulnerability Scoring System (CVSS). By performing VAPT, businesses can ensure compliance with security standards (e.g., OWASP, ISO 27001), minimize attack surfaces, and implement robust security controls.

This report documents the complete VAPT process undertaken during the internship, the tools and techniques used, observations made, and the real-time experience gained. The primary objective of this assessment was to identify critical flaws in web applications and infrastructure, simulate potential exploits, and recommend security enhancements. Ultimately, the internship provided deep insights into how security testing is conducted in real-world environments and reinforced the importance of a proactive approach to cybersecurity.

TABLE OF CONTENTS

Chapter 1. Introduction.....	1
1.1 Current Cybersecurity Challenges.....	1
1.2 Importance of Web Application Security.....	1
1.3 Role of VAPT in Organization.....	2
1.4 Internship Objectives and Scope.....	3
1.5 Ethical Guidelines and Confidentiality.....	4
Chapter 2. Literature Review	5
Chapter 3. Methodology.....	7
3.1 VAPT Testing Process.....	7
3.2 VAPT Scope and Approaches.....	8
3.3 Vulnerability Management (VM)	9
3.4 Infrastructure VAPT.....	9
3.5 Basics of API Security.....	10
3.6 Risk Scoring (CVSS)	10
3.7 Report Writing Standards.....	11
Chapter 4. Complete Internship Experience.....	12
4.1 My Role and Daily Work.....	12
4.2 Web Application VAPT.....	13
4.2.1 Information Gathering	
4.2.2 Automated and Manual Testing	
4.2.3 Findings and Exploiting Vulnerabilities	
4.2.4 Testing Real Company Applications	
4.3 Basics and Concepts Learned Beyond Web Application Security	
4.3.1 Infrastructure VAPT	
4.3.2 Vulnerability Management	
4.3.3 Basics of API Security	
4.4 Security Testing Tools and Extensions.....	16

4.4.1 Web and Security Testing Tools	
4.4.2 Command Line Tools	
4.4.3 Browser Extensions	
4.5 Security Testing Skills.....	24
Chapter 5. Professional Reporting and Detailed Findings	26
5.1 Executive Summary.....	26
5.2 Vulnerability Distribution.....	27
5.3 Assessment Scope.....	28
5.4 Methodology.....	29
5.5 Detailed Findings.....	30
5.5.1 Deprecated Protocols	
5.5.2 Information Disclosure through Error Messages	
5.5.3 Account Takeover by Brute Force	
5.6 Conclusion.....	33
Chapter 6. Future Scope and Conclusion	34
References	35

LIST Of FIGURES

Fig. 1 VAPT Testing Process	17
Fig. 2 Burpsuite Scanning	18
Fig. 3 OWASP ZAP Scan	18
Fig. 4 Nessus Vulnerability Scan	19
Fig. 5 Dirbuster Directory Enumeration	19
Fig. 6 Rapid7 InsightVM Dashboard	20
Fig. 7 Postman API Testing	20
Fig. 8 Nmap Port Scanning	21
Fig. 9 Nikto Web Server Scanning	21
Fig. 10 SQLMap SQL Injection Testing	22
Fig. 11 Wappalyzer Technology Detection	22
Fig. 12 Cookie Editor View/Edit Cookies	23
Fig. 13 CSP Evaluator Security Analysis	23
Fig. 14 Security Headers Tester Results	24
Fig. 15 Distribution of Vulnerabilities	27
Fig. 17 Deprecated Protocols	28
Fig. 18 Information Disclosure	29
Fig. 19 Brute Force Vulnerability	30

Chapter 1: Introduction

1.1 Current Cybersecurity Challenges

In today's digital era, organizations face increasing threats such as data breaches, ransomware, phishing, and web application exploits. As most business processes have moved online, web applications have become primary targets for attackers. These challenges make it essential for organizations to perform continuous Vulnerability Assessment and Penetration Testing (VAPT) to proactively identify and remediate weaknesses before they can be exploited.

1.2 Importance of Web Application Security

Web applications often handle sensitive data, including user credentials, financial records, and personal information. Even a single vulnerability can result in significant financial losses and damage an organization's reputation.

During my internship, I primarily focused on Web Application Penetration Testing (WAPT), testing multiple applications, including both demo platforms and real company applications. I identified security gaps, assessed potential risks, and documented vulnerabilities in accordance with OWASP Top 10 standards. This hands-on experience enhanced my understanding of why securing web applications is one of the most critical areas in cybersecurity.

1.3 Role of VAPT in Organizations

VAPT plays a key role in strengthening the overall security posture of an organization.

- **Vulnerability Assessment (VA):** Systematic scanning of systems and applications using tools such as Nessus, Nmap, Nikto, and testssl.sh to identify known weaknesses.
- **Penetration Testing (PT):** Simulating real-world attacks to validate how an attacker could exploit those vulnerabilities. I practiced manual exploitation using tools like Burp Suite, OWASP ZAP, SQLMap, DirBuster, and browser extensions such as Wappalyzer, Cookie Editor, CSP Evaluator, Security Headers Tester, and JWT Editor.

- **Risk Rating & Reporting:** I learned how to categorize vulnerabilities based on CVSS scores and business impact. I also developed skills in professional report writing, including Executive Summary, Technical Findings, Proof-of-Concept (PoC), Impact, and Recommendations.

This combination of assessment, exploitation, and reporting ensures organizations not only detect vulnerabilities but also understand their real risk and mitigation steps.

1.4 Internship Objectives and Scope

The main objectives of my internship were:

- To gain practical experience in VAPT with a focus on Web Application Security Testing.
- To perform hands-on testing on applications and identify vulnerabilities based on OWASP Top 10.
- To get exposure to Infrastructure VAPT basics, mainly using tools such as Nmap and Nessus.
- To learn about Vulnerability Management (VM) process and get an introduction to Rapid7 tool (only at a basic level).

To enhance my skills in tools and techniques used by security professionals:

- **Scanning Tools:** Nessus, Nmap, Nikto, testssl.sh
- **Web Testing Tools:** Burp Suite, OWASP ZAP, SQLMap, DirBuster
- **Browser Extensions:** Wappalyzer, Cookie Editor, CSP Evaluator, Security Headers Tester, JWT Editor

To understand the end-to-end process of report writing, including categorization of findings, assigning severity using CVSS, and providing clear recommendations. The scope of my internship covered both web application testing (main focus) and infrastructure basics, while maintaining ethical guidelines and confidentiality of company data.

1.5 Ethical Guidelines and Confidentiality

As part of my internship, I followed strict ethical hacking guidelines:

- No unauthorized access or exploitation outside the approved scope.
- Confidential data from company was never disclosed in my documentation.

- Only safe examples, screenshots, and generic findings are included in this report for academic purposes.
- The main focus is on skills gained, methodologies followed, and tools mastered, without exposing any sensitive organizational information.

Chapter 2: Literature Review

For any security engineer, Vulnerability Assessment and Penetration testing is a way to find all the security issues in a particular application or system and list them according to the CVSS score for the risk analysis. In this section, I have provided some literature works from cybersecurity researchers and professionals, whom I took help with while working on the project.

1. Jai Narayan Goel, BM Mehtre (2015), “Vulnerability Assessment & Penetration Testing as a Cyber Defence Technology”

As we know the complexity of systems keeps on increasing rapidly with time, this also raises the possibility of an increase in the potential for vulnerabilities as well, which can be easily exploited by attackers to hack the system. This can be avoided by identifying and addressing these vulnerabilities before attackers can exploit them. In this case, Vulnerability Assessment and Penetration Testing (VAPT) counts as a powerful cyber defense technology that offers a proactive cyber defense. The paper also examines some prevalent VAPT techniques and gives an overview of some popular and useful opensource VAPT tools.

This paper also discusses the importance of VAPT as a Cyber Defence Technology. By properly removing system vulnerabilities, VAPT can be used as a cyber defense technology, lowering the chance of cyberattacks. The paper explains several Vulnerability & Penetration testing approaches, and provide a comprehensive VAPT life cycle for active defense. The detailed tutorial in this paper helps in reducing risks and cyber-attacks. The paper explains various methods of conducting Vulnerability Assessment and Penetration Testing and provides a comprehensive life cycle of VAPT for proactive defense.

2. By Zoran Davidovac and Vanja Korac (2010), “Vulnerability Management and Patching IT Systems”

The paper focuses on the common problems and weaknesses found in modern information systems. It explains that large organizations like banks and government agencies often lack enough resources and expertise, making them more vulnerable to attacks. The authors suggest

that vulnerability management and regular patching are key strategies to stop system abuse, lower costs, and improve overall security awareness.

The study also talks about different types of cyber threats, such as attacks on critical infrastructure, botnets made of infected computers, and electronic warfare carried out by organized or state-sponsored groups. Real-world cases from countries like the US, France, and Kyrgyzstan are used to show how cyberattacks are becoming more advanced and even tied to politics and national security. The authors share results from vulnerability scans done at the Mathematical Institute of the Serbian Academy of Sciences and Arts. The scans found weak points like open FTP servers and web servers vulnerable to denial-of-service attacks. The paper concludes that effective security requires both automated tools (like Rapid7 NeXpose) and strong security policies, along with continuous monitoring, timely patching, and proper awareness training to keep systems safe.

3. By Urshila Ravindran and Raghu Vamsi Potukuchi (2022) “A Review on Web Application Vulnerability Assessment and Penetration Testing”

This paper highlights the growing importance of securing web applications as cyber threats increase. The authors explain that vulnerabilities in applications can lead to severe financial and reputational losses, and emphasize that **Vulnerability Assessment and Penetration Testing (VAPT)** provides a proactive way to detect and fix weaknesses before attackers exploit them.

The study explains the **vulnerability management lifecycle**, the difference between vulnerability assessment and penetration testing, and the importance of the CIA triad (Confidentiality, Integrity, Availability). It also reviews commonly used open-source VAPT tools such as Burp Suite, Nmap, Nikto, OWASP ZAP, and Wireshark etc. In addition, testing strategies like black-box and grey-box approaches, along with the skill requirements for testers, are discussed to give a clear understanding of how VAPT is practically performed. And then the authors describe the VAPT process flow, which includes both passive and active testing modes.

These cover areas like information gathering, web server fingerprinting, authentication, authorization, client-side testing, and input validation. The paper concludes that performing VAPT regularly is essential for technology-driven organizations, as it ensures secure coding, timely vulnerability detection, and stronger protection against evolving cyber-attacks.

4. By Omotosho et al. (2025), “Enhancing Network Security Using Vulnerability Assessment and Penetration Testing”

The paper focuses on the rising challenges of protecting modern networks. It explains that traditional firewalls and perimeter security are no longer sufficient, as networks now combine different systems like Cisco, Linux, and Windows. To deal with these risks, the authors propose using **Vulnerability Assessment and Penetration Testing (VAPT)** as a proactive method to find and fix weaknesses. They present a framework that uses tools such as Nessus, Nmap, and to scan for common issues like open ports, weak configurations, and unprotected communication channels. The framework allows organizations to identify threats before attackers exploit them, making security

Their experiments showed that attackers could exploit flaws such as ARP spoofing and Man-in-the-Middle (MITM) attacks when networks were not properly secured. However, after applying measures like encryption, secure configurations, and intrusion detection, the number of vulnerabilities dropped significantly. A systematic VAPT is essential for any organization, as it helps in quickly detecting problems, applying fixes, and improving resilience against modern cyber threats in complex network environments.

5. By Bennouk et al. (2024), “A Comprehensive Review and Assessment of Cybersecurity Vulnerability Detection Methodologies”

The paper discusses how VAPT plays a vital role in securing modern systems, with a focus on **Web Application** and **API security testing**. The authors highlight common issues such as SQLi, XSS, insecure authentication, and API misuse, stressing that these must be identified through a mix of automated scans and manual penetration testing. The study also explains the vulnerability management lifecycle, covering identification, prioritization, and remediation

It emphasizes the role of CVE, CWE, NVD databases for cataloging vulnerabilities and the use of **CVSS scores** for measuring severity. The paper notes that effective reporting should include CVE references, CVSS metrics, impact, and recommendations, feeding into Vulnerability Management Systems (VMS) for continuous monitoring. The authors conclude that integrating VAPT, web and API testing, and structured vulnerability management significantly improves an organization’s security posture, reduces exposure to cyber threats, and ensures proactive defense against future attacks.

Chapter 3: Methodology

3.1 VAPT Testing Process

The Vulnerability Assessment and Penetration Testing (VAPT) process is a structured methodology used to identify, assess, and report security weaknesses in IT systems, networks, and applications. This methodology ensures that vulnerabilities are detected in a controlled and ethical manner. The process includes the following detailed steps:



Fig 1: VAPT Testing Process

- **Planning:** Defining the objectives, scope, rules of engagement, and type of testing (Black Box, Grey Box, or White Box). Legal approvals and permissions are obtained to ensure responsible testing.
- **Information Gathering:** Collecting intelligence about the target using passive techniques (like OSINT, WHOIS, Google Dorking) and active techniques (like port scanning, banner grabbing, directory brute-forcing). This provides insights into system architecture and potential entry points.
- **Vulnerability Scanning:** Using automated tools to identify known vulnerabilities, misconfigurations, outdated software, and weak protocols. Scanners produce a list of potential issues that must later be verified for accuracy.

- **Penetration Testing:** Controlled exploitation of vulnerabilities to evaluate real-world impact. This includes attempting to bypass authentication, inject malicious inputs, or chain multiple vulnerabilities to simulate an actual attack scenario.
- **Reporting:** Preparing a structured report with vulnerability descriptions, CVSS-based severity ratings, proof-of-concept evidence, and recommendations for remediation. Reports are designed for both technical teams and management.

3.2 VAPT Scope and Approaches

Defining the scope and selecting an approach are critical steps in the VAPT methodology. The scope determines the boundaries of the engagement, including which applications, systems, networks, or IP ranges will be tested, ensuring that testing is legally compliant and aligned with organizational objectives. The approach defines how the testing will be conducted and the level of knowledge the tester has about the system.

Black Box Testing simulates an external attacker with no prior knowledge of the target environment. The tester starts from scratch, using reconnaissance techniques to discover entry points and vulnerabilities, which helps organizations understand their exposure to unknown threats.

Grey Box Testing represents a scenario where the tester has partial knowledge, such as user credentials or limited system information. This approach focuses on assessing security from the perspective of an insider or semi-privileged user and helps identify vulnerabilities that could be exploited once basic access is obtained.

White Box Testing involves comprehensive knowledge of the system, including architecture, source code, and credentials. This approach allows for a thorough internal assessment, helping detect hidden security flaws, insecure coding practices, and configuration weaknesses that may not be visible externally.

A clear definition of scope and approach ensures that testing is systematic, efficient, and aligned with business objectives while reducing the risk of legal or operational issues. It also helps in prioritizing testing efforts based on critical assets, potential threats, and business impact.

3.3 Vulnerability Management (VM)

Vulnerability Management is a structured methodology for continuously maintaining the security of IT assets. It encompasses a series of systematic steps designed to identify, assess, and prioritize vulnerabilities across systems, applications, and networks. Initially, automated scans are conducted using specialized tools such as Rapid7 InsightVM, Nessus, or Qualys to detect known vulnerabilities, misconfigurations, outdated software, and weak security controls.

Once the vulnerabilities are detected, they are categorized and prioritized based on factors such as severity, exploitability, and business impact. The methodology involves creating clear action plans for remediation, including patching, configuration changes, or mitigation strategies, and ensuring that the remediation steps are effectively applied. Verification scans are then performed to confirm that the vulnerabilities have been addressed successfully. Vulnerability Management also integrates risk assessment into the methodology. By understanding which systems are most critical and which vulnerabilities are most exploitable, organizations can allocate resources efficiently and make informed security decisions.

3.4 Infrastructure VAPT

Infrastructure VAPT is the methodology used to assess the security of an organization's IT infrastructure, including networks, servers, routers, firewalls, and databases. The process typically starts with scanning and discovery, where tools like Nmap, Nessus, or OpenVAS are used to identify live hosts, open ports, running services, and known vulnerabilities.

Following scanning, the methodology emphasizes the evaluation of network and system configurations to identify weaknesses such as default credentials, weak protocols, or exposed services. Penetration testing is performed in a controlled manner to simulate potential real-world attacks, assessing the impact of exploitable vulnerabilities. A critical part of the methodology is assessing privilege escalation paths, determining how an attacker could gain higher-level access within the infrastructure. By systematically combining scanning, assessment, exploitation, and risk analysis, this methodology ensures a comprehensive understanding of the security posture of the infrastructure and provides actionable insights for strengthening defenses.

3.5 Basics of API Security

API security methodology focuses on assessing and protecting APIs used in web, mobile, and cloud-based applications. It begins with authentication and authorization checks, ensuring that only valid and appropriately privileged users can access specific API endpoints. Input validation and sanitization are integral steps in the methodology to prevent injection attacks such as SQLi, XSS, or command injections. The methodology also involves evaluating rate limiting and throttling mechanisms, which protect APIs against brute-force or denial-of-service attacks.

Transport security is assessed to ensure that all communication is encrypted using HTTPS/TLS, preventing data leakage during transmission. Additionally, API security testing includes reviewing error handling and information disclosure, ensuring that sensitive data is not exposed in error messages or responses. Testing tools such as Postman, Burp Suite, and OWASP ZAP are integrated into the methodology to perform structured API testing. The methodology encompasses both functional testing of endpoints and security-focused testing to identify potential flaws, misconfigurations, and weaknesses in the API design and implementation.

3.6 Risk Scoring (CVSS)

The Common Vulnerability Scoring System (CVSS) provides a standardized methodology for assessing and quantifying the severity of security vulnerabilities. It combines base metrics, temporal metrics, and environmental metrics to calculate a score ranging from 0.0 (no risk) to 10.0 (critical risk). Base metrics evaluate the intrinsic characteristics of a vulnerability, such as the attack vector, complexity, required privileges, and potential impact on confidentiality, integrity, and availability. Temporal metrics take into account factors that change over time, like the availability of exploits or patches.

Environmental metrics adjust the score based on the organization's context, such as the criticality of affected assets or compensating controls. Using CVSS methodology allows security teams to prioritize vulnerabilities effectively, allocate resources strategically, and implement mitigation plans based on a quantified risk perspective.

3.7 Report Writing Standards

The methodology for report writing ensures that findings are communicated clearly and effectively to both technical teams and management. A professional VAPT report includes a structured executive summary providing a high-level overview, followed by detailed descriptions of each finding, including affected assets, severity ratings (CVSS), proof-of-concept evidence, and suggested remediation steps.

The report methodology emphasizes clarity, conciseness, and evidence-based documentation. Screenshots, payload examples, and logs are incorporated to support technical recommendations, while risk assessments help management understand business impact and prioritize actions. A well-structured report bridges the gap between security testing efforts and actionable outcomes, ensuring that vulnerabilities are addressed efficiently and that security posture improvements are measurable.

Chapter 4: Complete Internship Experience

4.1 My Role and Daily Work

During my internship, I worked as a **Threat and Vulnerability Management (TVM) Intern**, with my primary focus on **Web Application Security Testing (VAPT)**. My main objective was to understand the complete workflow of identifying and analyzing vulnerabilities in web applications. Although I did not work on live projects, I engaged in structured learning and hands-on practice using lab-based vulnerable applications to simulate real-world scenarios.

I spent considerable time exploring and practicing **automated and manual testing techniques**. This included using tools such as Burp Suite, OWASP ZAP, Nessus, Nmap, and browser extensions to intercept, analyze, and test web application traffic. I learned how to perform tasks such as scanning applications, analyzing responses, testing input fields, and identifying common vulnerabilities, including those listed in the **OWASP Top 10**. These exercises gave me a practical understanding of how vulnerabilities are discovered and the steps required to validate them.

To reinforce my learning, I practiced on vulnerable applications like demo.testfire.net, DVWA, and similar test environments. Through these exercises, I attempted to exploit vulnerabilities in a safe, controlled manner, helping me understand how attacks could be carried out and how mitigation measures could be applied. I also experimented with **browser extensions and manual testing techniques** to explore edge cases and deepen my understanding of web application security.

In addition to practical testing, I focused on **analyzing findings and documenting them professionally**. I learned to record evidence of each vulnerability, assess its potential impact, and relate it to risk metrics such as CVSS scores.

Alongside web application testing, I gained exposure to API testing, Infrastructure VAPT, and Vulnerability Management (VM). I learned the theoretical concepts of tools like Rapid7 InsightVM and Postman, and understood how vulnerabilities in applications and networks are monitored, tracked, and reported. Although I did not perform practical exercises with these tools, this knowledge allowed me to grasp the broader VAPT and VM workflow, preparing me to discuss these topics confidently in professional or academic contexts.

4.2 Web Application VAPT

During my internship, my main focus was on **Web Application Penetration Testing (VAPT)**, where I learned to systematically identify, analyze, and understand security weaknesses in web applications. This involved following a structured workflow from information gathering to vulnerability analysis and reporting, providing me with hands-on exposure to real-world web security concepts. Although I primarily practiced in lab environments, these exercises gave me practical insight into how vulnerabilities can be identified, tested, and mitigated in professional scenarios.

4.2.1 Information Gathering

The first step in the VAPT process was information gathering, which involved mapping the application's functionality, identifying input points, and analyzing user interactions. I explored application URLs, headers, and parameters to find potential attack vectors, and I practiced capturing and inspecting HTTP requests using tools like Burp Suite and OWASP ZAP, along with various browser extensions for deeper insights.

I also learned to identify hidden or less obvious endpoints, parameters, and form fields that could be potential security risks. This stage required careful observation, research, and note-taking, as it formed the foundation for effective testing. Understanding the **application's workflow and attack surface** was essential before performing automated scans or manual exploitation, ensuring that all critical areas were assessed comprehensively.

4.2.2 Automated and Manual Testing

Once sufficient information was gathered, I performed automated testing using Nessus and Nmap to quickly detect common security issues such as misconfigurations, open ports, and outdated components. Automated scanning provided an overview of potential vulnerabilities, helping prioritize areas for deeper investigation.

In parallel, I carried out manual testing using Burp Suite, OWASP ZAP, and browser extensions, which allowed me to verify vulnerabilities that scanners might miss. I practiced testing for a variety of issues, including input validation errors, session management flaws, cross-site scripting (XSS),

SQL injection, HTML injection, and IDOR vulnerabilities. Manual testing helped me understand the nuances of security flaws, how they can be exploited, and why automated tools alone are not sufficient for thorough assessment.

4.2.3 Finding and Exploiting Vulnerabilities

After testing, I focused on analyzing and exploiting identified vulnerabilities in controlled lab environments. Using `demo.testfire.net` and DVWA, I practiced exploiting vulnerabilities safely, understanding their potential impact on application confidentiality, integrity, and availability. This helped me learn how attackers could exploit these weaknesses and the steps required to mitigate them effectively.

During this stage, I also documented every finding in detail, including screenshots, request/response evidence, CVSS severity scores, and references to CVEs. I learned how to interpret each vulnerability's impact and how to present it clearly in a report. This process emphasized the importance of accurate documentation and clear communication of technical findings, which is critical in professional security assessments.

4.2.4 Testing Real Company Applications

Although my practical testing was mainly in lab environments, I was exposed to the methodology applied in real company applications under senior analyst supervision. I observed how web applications are scoped, scanned, and validated for vulnerabilities before reporting, and I learned about the professional standards and protocols followed in client-facing assessments.

This exposure helped me understand the workflow of real-world web application security assessments, including prioritizing vulnerabilities, evaluating risk using CVSS scores, referencing CVEs, and following proper reporting procedures. It also gave me insight into how security findings are communicated to development teams and stakeholders, providing a holistic view of professional VAPT processes.

4.3 Basics and Concepts Learned Beyond Web Application Security

4.3.1 Infrastructure VAPT Basics

During my internship, I learned the basics of **Infrastructure Vulnerability Assessment and Penetration Testing (Infra VAPT)**, which focuses on evaluating the security of an organization's IT infrastructure including servers, routers, firewalls, databases, and network components. Infra VAPT simulates potential cyberattacks to identify weaknesses, understand their impact, and demonstrate how an attacker might exploit them. Although I did not perform practical testing, I gained theoretical knowledge of the typical workflow used by security teams.

In infrastructure VAPT, the testing process generally starts with scanning the defined scope, followed by analysis, validation, and reporting. Tools like Nessus are used for automated vulnerability scanning, while Nmap is often used for network discovery and port scanning. Security teams also conduct manual verification to remove false positives and validate findings before reporting to the client.

Key Steps and Processes in Infrastructure VAPT

- **Scanning:** Tools like Nessus scan servers, endpoints, and network devices to identify known vulnerabilities and misconfigurations.
- **Manual Verification:** Security analysts review automated scan results to remove false positives and confirm actual vulnerabilities.
- **Network Enumeration:** Nmap is used to discover open ports, running services, and potential entry points.
- **Reporting:** Verified vulnerabilities are documented and communicated to clients in a professional report, including proof of concept and suggested remediation.
- **Outcome:** The process helps organizations identify weaknesses, understand real-world risks, and prioritize fixes to improve network security.

4.3.2 Vulnerability Management Learning (Rapid7)

I also learned the basics of **Vulnerability Management (VM)**, which ensures continuous monitoring and remediation of vulnerabilities in IT systems and applications. Unlike one-time penetration tests, VM focuses on **maintaining security over time** by tracking vulnerabilities and ensuring they are resolved before they can be exploited. During the learning process, I understood how **Rapid7 InsightVM** works in real-world scenarios:

Rapid7 VM Process (Conceptual Understanding):

- **Scanning:** Rapid7 performs automated scans on client infrastructure and applications to detect vulnerabilities.
- **Initial Ticketing:** Instead of sending a full report with CVSS scores and remediation steps, Rapid7 initially raises a single ticket for each identified vulnerability, alerting the client that the issue exists and requires attention.
- **Verification & Tracking:** Once the client applies remediation steps, Rapid7 verifies whether the vulnerability has been fixed and updates the ticket accordingly.
- **Outcome:** This workflow allows organizations to quickly respond to critical vulnerabilities without being overwhelmed by detailed reports for every issue. It ensures that remediation is tracked and nothing is missed.

Through this process, I learned that VM tools like Rapid7 are essential for continuous security monitoring, prioritizing vulnerabilities, and ensuring accountability in patch management. Even though I did not operate Rapid7 directly, I understood the tool's purpose, workflow, and the way tickets are used for efficient client communication.

4.3.3 Basics of API Security

In addition to web applications, I learned the basics of API security (VAPI). APIs (Application Programming Interfaces) allow communication between different software systems and are commonly used in web, mobile, and cloud applications. API security differs from web application security because it focuses on endpoints, data exchange, and authorization mechanisms, rather than forms or user interfaces.

Key Concepts and Learning Points:

- **Testing Focus:** API testing involves checking endpoints for vulnerabilities, rather than pages and input fields as in web applications.
- **Common Tools:** Postman is the primary tool used to send requests, test inputs, and observe responses. Burp Suite can also be integrated to intercept API traffic if needed.
- **Vulnerabilities Covered:** Authentication and authorization issues, input validation flaws, rate limiting problems, error handling weaknesses, and insecure transport (HTTPS/TLS).
- **Process Overview:** Identify endpoints and required request parameters. Send test requests using Postman to check for improper access, injection attacks, or sensitive data exposure. Document any vulnerabilities and understand the potential impact on confidentiality, integrity, and availability.
- **Outcome:** Learning API security helps understand how attackers could exploit API endpoints differently from traditional web applications, and emphasizes the importance of securing data flows and access control.

4.4 Security Testing Tools and Extensions Overview

During my internship, I gained practical exposure to several security tools and browser extensions, which are commonly used in web application testing, infrastructure VAPT, and vulnerability assessment. These tools helped me understand how vulnerabilities are identified, tested, and analyzed in real-world scenarios.

4.4.1 Web & Security Testing Tools

1. Burp Suite

It is a comprehensive web application security testing platform used to identify vulnerabilities such as XSS, SQL injection, and session management flaws. It includes several modules like Proxy, Repeater, Intruder, Decoder, and Scanner, which support both automated scanning and manual testing. Burp Suite helps security testers intercept, analyze, and manipulate web traffic to validate vulnerabilities effectively.

3.Nessus

Nessus is a network and system vulnerability scanner that identifies known vulnerabilities, outdated software, and misconfigurations. It performs automated scans across hosts and services, generating detailed reports that highlight critical issues. Nessus helps security teams prioritize fixes based on risk and severity.

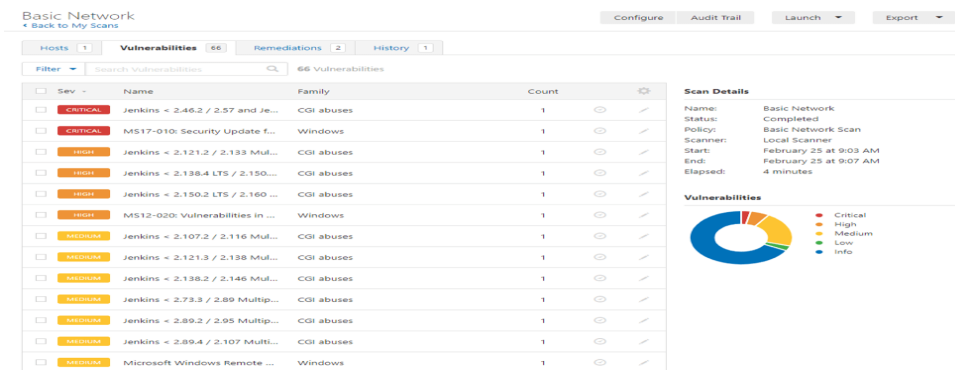


Fig 4: Nessus Vulnerability Scan

4.DirBuster

It is a web application directory and file brute-forcing tool used to discover hidden files or directories that may expose sensitive information. It sends multiple requests using wordlists and reports existing paths. DirBuster is valuable for uncovering areas of a web application that are not publicly visible.

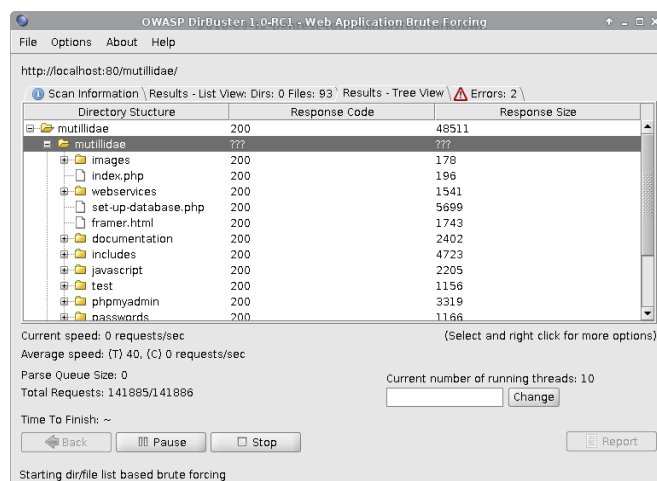


Fig 5: Dirbuster Directory Enumeration

5. Rapid7 InsightVM

Rapid7 is a vulnerability management platform that performs automated scans across infrastructure and applications. It detects vulnerabilities and generates tickets for clients, ensuring that security issues are tracked and remediated. Rapid7 also provides dashboards for risk prioritization and monitoring remediation progress over time.

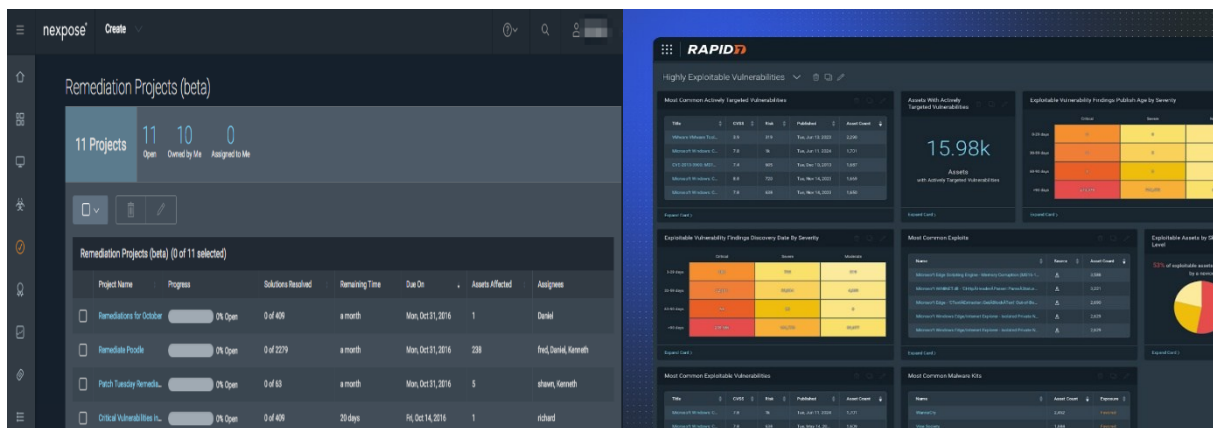


Fig 6: Rapid7 InsightVM Dashboard

6. Postman

It is a tool used for security testing, particularly for web applications and APIs. It allows testers to perform request manipulation, analyze responses, and identify potential vulnerabilities in input fields and endpoints. Postman simplifies testing by providing a user-friendly interface for both beginners and experienced testers.

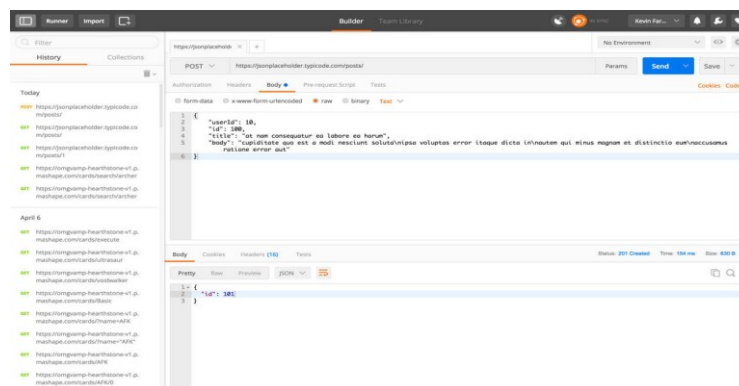


Fig 7: Postman API Testing

4.4.2 Command-line Tools

1.Nmap

Network scanning and discovery tool; helps identify open ports, running services, and potential network entry points; works by sending network probes and mapping hosts.

```
ravi@tecmint:~$ nmap 192.168.0.*
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-14 12:59 IST
Nmap scan report for _gateway (192.168.0.1)
Host is up (0.00053s latency).
Not shown: 995 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
53/tcp    open  domain
80/tcp    open  http
1900/tcp   open  upnp
20005/tcp  open  btx

Nmap scan report for tecmint (192.168.0.162)
Host is up (0.00052s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE
```

Fig 8: Nmap Port Scanning

2.Nikto

Web server scanner; detects misconfigurations, outdated software, and known vulnerabilities; performs automated server checks and reports security risks.

NIKTO - WEB SCANNER

```
Note: This is the short help output. Use -H for full help text.

root@kali:~# nikto -h 192.168.1.104
- Nikto v2.1.5
-----
- Target IP:      192.168.1.104
- Target Hostname: 192.168.1.104
- Target Port:    80
- Start Time:     2014-03-16 13:12:38 (GMT0)
-----
- Server: Apache/2.2.14 (Ubuntu)
- Server leaks inodes via ETags, header found with file /, inode: 294235, size: 177, etime: 0x4a4e4a1000a00
- The anti-clickjacking X-Frame-Options header is not present.
- Apache/2.2.14 appears to be outdated (current is at least Apache/2.2.22). Apache 1.3.42 (final release) and 2.0.64 are also current.
- Allowed HTTP Methods: GET, HEAD, POST, OPTIONS
- OSVDB-3268: /icons/: Directory indexing found.
- OSVDB-3223: /icons/README: Apache default file found.
```

Fig 9: Nikto Web Server Scanning

3.SQLMap

Automated SQL injection testing tool; identifies **SQLi vulnerabilities** in web applications and APIs; works by sending crafted queries and analyzing server responses.

```
$ python sqlmap.py -u "http://debiandev/sqlmap/mysql/get_int.php?id=1" --batch

[+] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 10:44:53 /2019-04-30/

[10:44:54] [INFO] testing connection to the target URL
[10:44:54] [INFO] heuristics detected web page charset 'ascii'
[10:44:54] [INFO] checking if the target is protected by some kind of WAF/IPS
[10:44:54] [INFO] testing if the target URL content is stable
[10:44:55] [INFO] target URL content is stable
[10:44:55] [INFO] testing if GET parameter 'id' is dynamic
[10:44:55] [INFO] GET parameter 'id' appears to be dynamic
[10:44:55] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable (possible DBMS: 'MySQL')
```

Fig 10: SQLMap SQL Injection Testing

4.4.3 Browser Extensions (for analyzing headers, cookies, and technologies)

1.Wappalyzer

Detects technologies used on websites, such as frameworks, servers, and CMS. It helps understand the tech stack and potential attack surfaces.

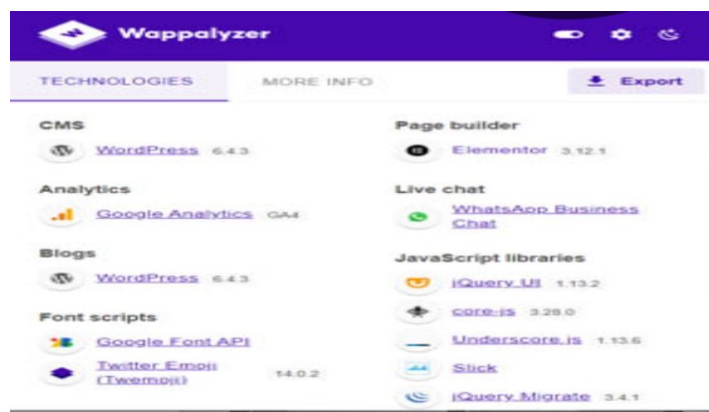


Fig 11: Wappalyzer Technology Detection

2.Cookie Editor

Allows inspecting and modifying browser cookies to test session management and security. It helps identify insecure cookie settings.

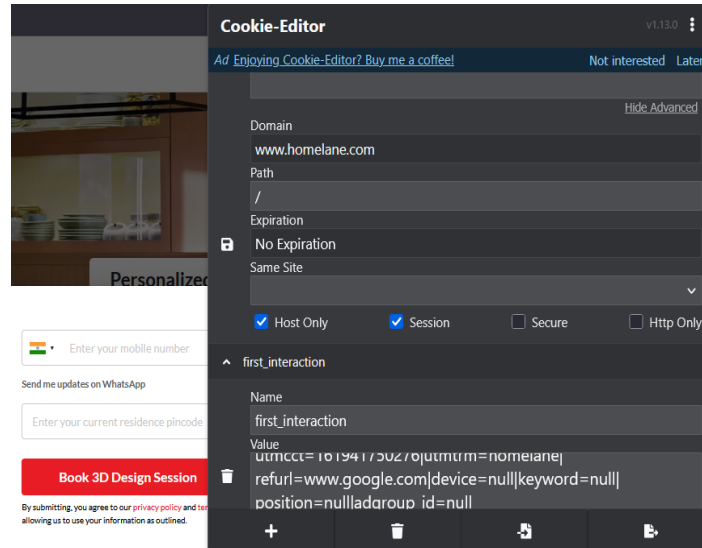


Fig 12: Cookie Editor View/Edit Cookies

3.CSP Evaluator

Checks Content Security Policy headers for misconfigurations or missing directives. It ensures proper protection against XSS attacks.



Fig 13: CSP Evaluator Security Analysis

4.Security Headers Tester

Verifies security-related HTTP headers like HSTS, X-Frame-Options, and CSP. It helps ensure that web applications follow best security practices.

Security Headers by snyk

Scan your site now

https://testfire.net/ Scan

Hide results Follow redirects

Security Report Summary

F

Site: <https://testfire.net/>
IP Address: 65.61.137.117
Report Time: 18 Aug 2025 17:09:18 UTC

Headers: **Strict-Transport-Security** **Content-Security-Policy** **X-Frame-Options** **X-Content-Type-Options** **Referrer-Policy** **Permissions-Policy**

Advanced: Ouch, you should work on your security posture immediately. [Start Now](#)

Missing Headers

Strict-Transport-Security	HTTP Strict Transport Security is an excellent feature to support on your site and strengthens your implementation of TLS by getting the User Agent to enforce the use of HTTPS. Recommended value "Strict-Transport-Security: max-age=31536000; includeSubDomains".
Content-Security-Policy	Content Security Policy is an effective measure to protect your site from XSS attacks. By whitelisting sources of approved content, you can prevent the browser from loading malicious assets.
X-Frame-Options	X-Frame-Options tells the browser whether you want to allow your site to be framed or not. By preventing a browser from framing your site you can defend against attacks like clickjacking. Recommended value "X-Frame-Options: SAMEORIGIN".
X-Content-Type-Options	X-Content-Type-Options stops a browser from trying to MIME-sniff the content type and forces it to stick with the declared content-type. The only valid value for this header is "X-Content-Type-Options: nosniff".
Referrer-Policy	Referrer Policy is a new header that allows a site to control how much information the browser includes with navigations away from a document and should be set by all sites.
Permissions-Policy	Permissions Policy is a new header that allows a site to control which features and APIs can be used in the browser.

Raw Headers

HTTP/1.1	200 OK
Server	Apache/2.4.18
Set-Cookie	SESSIONID=ECDF8A5496C19FC887CD3A58FE886CDD; Path=/; Secure; HttpOnly
Content-Type	text/html; charset=ISO-8859-1
Transfer-Encoding	chunked
Date	Mon, 18 Aug 2025 17:09:17 GMT

Fig 14: Security Headers Tester Results

4.5 Security Testing Skills

During my internship, I gained hands-on experience in web application security testing, performing both manual and automated assessments to identify and analyze vulnerabilities. This allowed me to understand how security weaknesses occur, how they can be exploited in a controlled environment, and how to verify them effectively. Working directly with applications helped me develop analytical skills and a practical approach to testing, ensuring that findings were accurate and reproducible.

I worked on several **OWASP Top 10 vulnerabilities**, including SQL Injection (SQLi), Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), Insecure Direct Object References (IDOR), Privilege Escalation, Session Fixation, and Session Hijacking etc. For each vulnerability, I learned not only to identify and exploit the issue but also to analyze its potential impact in a real-world

context, including the risk to sensitive data, user sessions, and administrative privileges. This provided a realistic understanding of how attackers could compromise an application.

A significant part of my learning involved “creating detailed Proof-of-Concepts (PoCs)” for vulnerabilities. For each finding, I documented step-by-step reproduction instructions, captured screenshots, and explained the potential impact on the system and users. This process reinforced the importance of clear, structured documentation and taught me how to demonstrate exploitability in a professional and controlled manner. I also performed SSL/TLS security testing and security headers analysis, using tools like `testssl.sh` to check TLS versions, cipher strengths, certificate validity, and algorithms. I verified the proper configuration of security headers such as Content Security Policy (CSP), HSTS, X-Frame-Options, and X-XSS-Protection, which together improve the application’s defense against attacks and strengthen overall web security.

During the internship, “I learned the basics of professional security reporting”, including how to categorize vulnerabilities, assign CVSS scores, and assess risk levels. I practiced organizing findings into structured reports that clearly highlight each issue’s severity, potential business impact, and recommended remediation steps. By linking PoCs and evidence to each vulnerability, I learned to produce reports that are informative, clear, and actionable for both technical teams and management. Overall, this experience allowed me to combine technical testing skills with the foundational principles of professional reporting. I developed the ability to analyze vulnerabilities, document them thoroughly, and communicate security risks effectively, ensuring that my work provides practical value to an organization’s security posture.

Chapter 5: Professional Reporting and Detailed Findings

In this chapter, I present the detailed findings from my security assessments along with the professional reporting practices I followed. It highlights how vulnerabilities were documented, categorized, and analyzed using CVSS scoring and risk ratings, and how evidence such as screenshots and Proof-of-Concepts (PoCs) was systematically recorded. The chapter also outlines the recommendations provided for remediation, demonstrating a structured approach to communicating security risks effectively to both technical teams and management.

5.1 Executive Summary

This report outlines the results of a penetration test conducted on two web applications: <https://demo.testfire.net> and <http://dvwa.local>. The goal was to identify any security flaws that could be exploited by attackers. The goal was to identify any security flaws that could be exploited by attackers. The test was performed using a black-box approach, simulating how an external attacker would interact with the application without any credentials or backend access.

5.2 Vulnerability Distribution

This section provides detailed information about each identified vulnerability, including its type, location, and how it can be exploited by an attacker. It explains the root cause of the issue, the conditions under which it occurs, and its potential impact on the application or system. Each description includes technical insights to help understand the nature of the vulnerability. The goal is to provide clear context for both technical and non-technical stakeholders. Supporting evidence such as request/response samples or screenshots is included wherever applicable.

Severity	CVSS v3 Range	Description
Critical	9.0– 10.0	Complete system compromise, remote code execution, full admin access, data destruction, and total loss of confidentiality or availability. Attackers may gain persistent access or take control of critical infrastructure.
High	7.0– 8.9	High business impact, sensitive data leakage, authentication bypass, privilege escalation, or lateral movement. Attackers could access user accounts, manipulate critical data, or disrupt core functionalities.
Medium	4.0– 6.9	Moderate issues such as IDOR, weak access control, insecure configurations, or partial data disclosure. Could assist attackers in preparing more impactful attacks.
Low	0.1– 3.9	Minor flaws such as verbose error messages, fingerprintable components, missing security headers, or outdated server banners. Useful for reconnaissance or chaining with other issues.
Informational	N/A	No direct risk; observations such as software versions, exposed technologies, or best practice suggestions. Helpful in improving overall security hygiene.

Fig 15: Distribution of Vulnerability

5.3 Assessment Scope

The assessment was restricted to the following:

- **Target URL:** <https://demo.testfire.net> and <http://dvwa.local>.
- **Test Type:** Grey Box
- **Environment:** Production
- **Application Type:** Online Banking Demo Application
- **Technologies Used:** ASP.NET, JavaScript
- **Tools Used:** Burp Suite, manual verification, browser extensions, Nmap, SQLMap.

5.4 Methodology

The assessment followed a grey-box penetration testing approach based on the OWASP Testing Guide. This means the tester had limited knowledge about the system, such as access to the application interface but not the source code. The methodology was designed to simulate how an attacker with partial internal knowledge could identify and exploit vulnerabilities. The process was broken down into the following stages:



Fig 16: Vulnerability Testing Process

- 1 **Reconnaissance** – Passive and active information gathering techniques were used to understand the target application's structure, pages, technologies, and exposed components.
- 2 **Enumeration** – Identified entry points, user input fields, hidden parameters, and available APIs to map out attack surfaces. Directory brute-forcing can be performed.
- 3 **Vulnerability Scanning** – Used automated tools and browser extensions to scan for known vulnerabilities like outdated components, security misconfigurations, and input flaws.
- 4 **Manual Verification** – Verified and validated scanner results manually. Conducted deeper inspection for complex issues like broken access controls and session mismanagement.
- 5 **Exploitation** – Actively attempted to exploit identified vulnerabilities such as authentication bypass, IDOR, and HTML injection to determine real-world impact.
- 6 **Reporting** – Documented all findings, proof of concept, affected components, business impact, and recommendations in this structured report.

5.5 Detailed finding:

5.5.1 Deprecated Protocols Enabled

Severity: Medium	CVSS v3 Score:5.9	OWASP Category: A5 Security Misconfiguration
------------------	-------------------	--

CVSS v3.1 Base Score Vector String

Affected Url: <https://demo.testfire.net>

Description

Deprecated protocols are old versions of secure communication protocols that are no longer considered safe. They have known vulnerabilities, weak encryption, and are not recommended for use.

Proof of Concept (PoC)

```

kali@kali:~$ nmap -sS -sV -p- -oN nmap.txt 65.61.137.117
Start 2025-07-01 13:35:35 --> 65.61.137.117:443 (demo.testfire.net) <--
rDNS (65.61.137.117): --
Service detected: HTTP

Testing protocols via sockets except NPN+ALPN

SSLv2      not offered (OK)
SSLv3      not offered (OK)
TLS 1      offered (deprecated)
TLS 1.1    offered (deprecated)
TLS 1.2    offered (OK)
TLS 1.3    not offered and downgraded to a weaker protocol
NPN/SPDY   not offered
ALPN/HTTP2 not offered

Testing cipher categories

NULL ciphers (no encryption)          not offered (OK)
Anonymous NULL Ciphers (no authentication) not offered (OK)
Export ciphers (w/o ADH+NULL)         not offered (OK)
LOW: 64 Bit + DES, RC2, 41, MD5 (w/o export) not offered (OK)
Triple DES Ciphers / IDEA             not offered
Obsoleted CBC ciphers (AES, ARIA etc.) offered
Strong encryption (AEAD ciphers) with no FS not offered
Forward Secrecy strong encryption (AEAD ciphers) offered (OK)

```

Fig 17: The above screenshot shows that the server supports deprecated protocols, which are considered insecure.

Impact

Use of deprecated protocols makes the site vulnerable to attacks like POODLE, downgrade attacks, or MITM. It also affects performance and compliance with modern security standards.

Recommendation

The website should disable support for SSL 2.0, SSL 3.0, and TLS 1.0/1.1, and only allow TLS 1.2 or TLS 1.3. HTTP/2 or HTTP/3 should be enabled instead of HTTP 1.0/1.1 for better speed and security. The server should enforce HTTPS and enable HTTP Strict Transport Security (HSTS) headers. Regular scanning should be done to ensure deprecated protocols remain disabled.

5.5.2 Information Disclosure through Error Messages

Severity:Medium	CVSS:5.3	OWASP Category: A7 Identification and authentication failures
-----------------	----------	---

CVSS v3.1 Base Score Vector String

Affected Url: <https://demo.testfire.net/search.jsp>

Description

Detailed error messages can help attackers find security flaws, guess file paths, identify database technologies, or prepare for further attacks like SQL injection or path traversal.

Proof of Concept

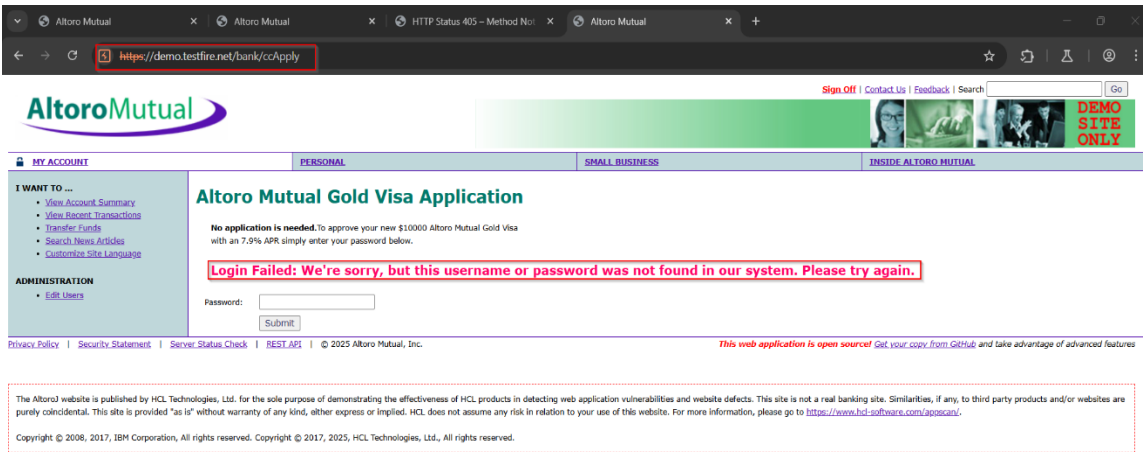


Fig 18: The above screenshot shows an error message revealing backend details, indicating information disclosure.

Impact

Detailed error messages can help attackers find security flaws, guess file paths, identify database technologies, or prepare for further attacks like SQL injection or path traversal.

Recommendation

The application should not display detailed error messages to users. Instead, it should show generic error responses and log the technical details on the server side only. Proper input validation and exception handling should be implemented to avoid leaking backend information.

5.5.3 Account Takeover by Brute Force

Severity: Critical	CVSS:9.1	OWASP Category: A7 Identification and authentication failures
--------------------	----------	---

CVSS v3.1 Base Score Vector String

Affected Url: <https://demo.testfire.net>

Description

Brute force attack is when an attacker tries to guess a valid username and password by submitting many login attempts using different combinations. If the application does not block or limit repeated login attempts, it allows attackers to guess weak passwords and take over user accounts.

Proof of Concept

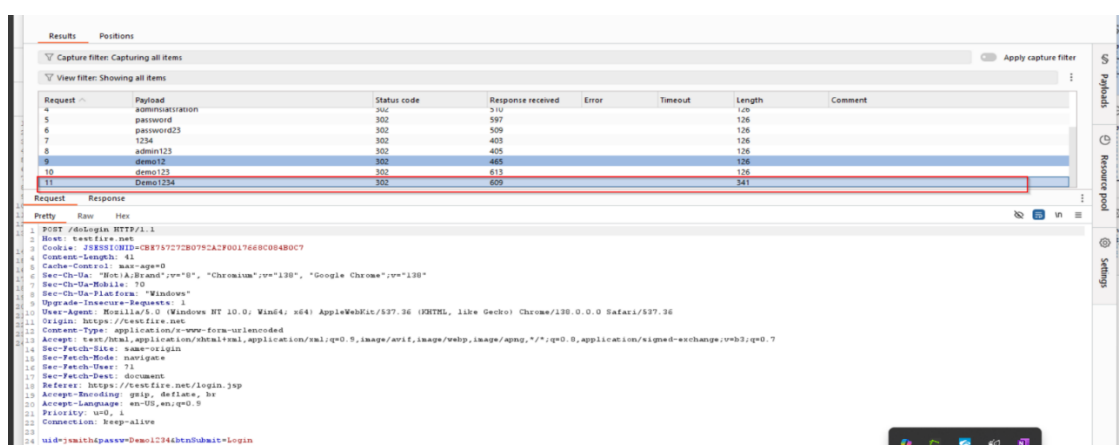


Fig 19: The screenshot shows multiple rapid login attempts made without any lockout or CAPTCHA, indicating that brute force protection is missing

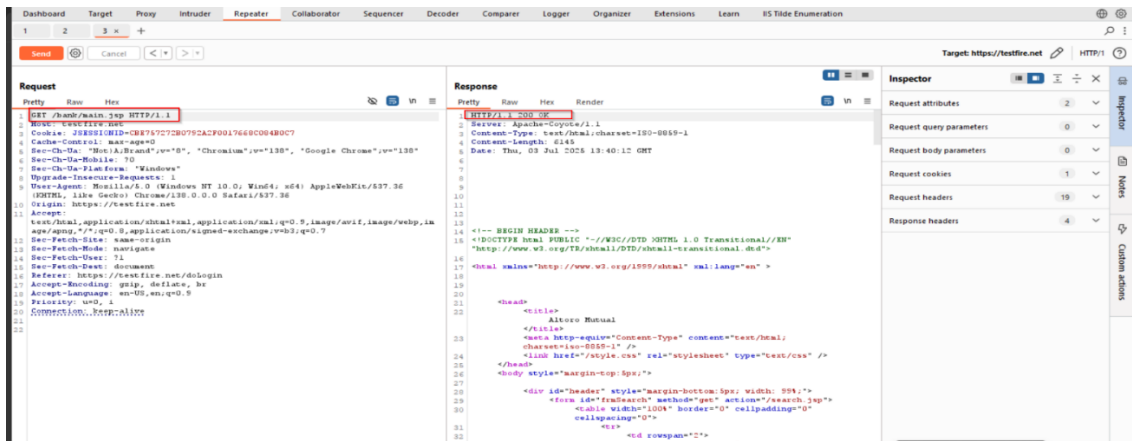


Fig 20: Shows a valid password being guessed and accepted after several attempts.

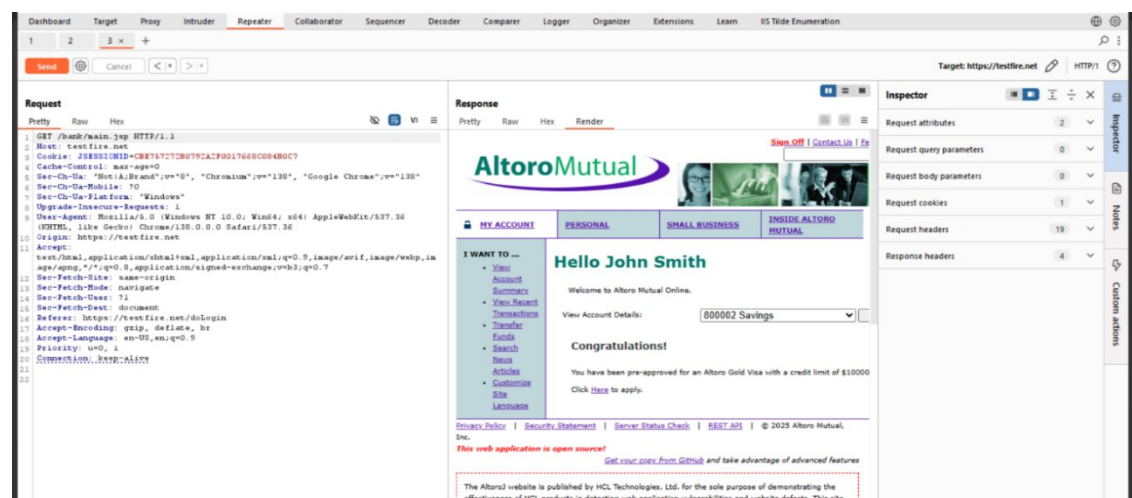


Fig 2: Confirms successful login into the user's account, proving account takeover through brute force.

Impact

If the application allows unlimited login attempts, attackers can exploit this to guess credentials and gain unauthorized access to user accounts. This leads to full account takeover, sensitive data exposure, identity misuse, or privilege escalation. If admin accounts are affected, complete system compromise may occur.

Recommendation

Implement strong account lockout mechanisms after a certain number of failed login attempts. Use CAPTCHA to deter automated tools. Enforce strong password policies and implement multi-factor authentication (MFA) to reduce the impact of credential guessing. Monitor login patterns for suspicious behavior and alert administrators accordingly.

5.6 Conclusion

In conclusion, this report presents the comprehensive findings from the web application penetration testing conducted during the internship/project. The testing process involved a combination of automated scanning and manual techniques to identify various vulnerabilities, assess their severity, and understand their potential impact. By documenting each finding and providing recommended remediation, the report serves as a practical guide for improving the security of the application. Overall, this exercise reinforced the importance of systematic testing, thorough analysis, and detailed reporting in ensuring robust web application security.

Chapter 6 Future scope and Conclusion

The internship provided a comprehensive and practical learning experience in web application and infrastructure security. I gained hands-on exposure to both manual and automated testing, learning to identify, analyze, and verify vulnerabilities effectively. Working directly with applications and performing controlled security assessments strengthened my technical understanding, analytical skills, and problem-solving abilities, providing a solid foundation for real-world cybersecurity challenges.

Throughout the internship, I developed professional documentation and reporting skills. I learned the basics of categorizing vulnerabilities, assigning CVSS scores, assessing risk levels, and recording evidence such as screenshots and Proof-of-Concepts (PoCs). This experience taught me how to organize findings into structured reports that clearly highlight severity, business impact, and recommended remediation steps, ensuring that the documentation is both informative and actionable for technical teams and management.

I also gained significant technical and professional growth, including deeper understanding of VAPT methodology, web application security principles, SSL/TLS configurations, and security headers analysis. Creating PoCs and documenting evidence systematically improved my ability to communicate technical findings effectively, while the exposure to practical testing scenarios enhanced my confidence in handling real-world security assessments.

The internship has clarified my career aspirations in cybersecurity, particularly in the areas of VAPT and web application security. Moving forward, I aim to deepen my expertise in advanced vulnerability assessment, API security testing, cloud security, and infrastructure hardening. I plan to pursue relevant certifications, stay updated with emerging attack vectors and defensive strategies, and continue building my ability to strengthen organizational security posture, ensuring that I can contribute effectively to proactive cybersecurity measures.

References

1. Goel, J. N., & Mehtre, B. M. (2015). Vulnerability Assessment & Penetration Testing as a Cyber Defence Technology.
2. Davidovac, Z., & Korac, V. (2010). Vulnerability Management and Patching IT Systems.
3. Ravindran, U., & Potukuchi, R. V. (2022). A Review on Web Application Vulnerability Assessment and Penetration Testing.
4. Omotosho, T., et al. (2025). Enhancing Network Security Using Vulnerability Assessment and Penetration Testing.
5. Bennouk, S., et al. (2024). A Comprehensive Review and Assessment of Cybersecurity Vulnerability Detection Methodologies.
6. OpenAI. 2025. ChatGPT GPT-5 Mini version. Available at <https://chat.openai.com/chat>
7. Microsoft. 2025. Microsoft Copilot. Available at <https://copilot.microsoft.com/>