INTERNSHIP ON CYBER SECURITY AND ETHICAL HACKING

A INTERNSHIP REPORT

Submitted by

VIGNESH M 110521104053

In partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GOJAN SCHOOL OF BUSINESS AND TECHNOLOGY

ANNAUNIVERSITY:: CHENNAI- 600 025

JULY 2024

GOJAN SCHOOL OF BUSINESS AND TECHNOLOGY

APPROVED BY AICTE, NEW DELHI & AFFILIATED TO ANNA UNIVERSITY, CHENNAI
ACCREDITED BY NAAC & RECOGNISED U/S 2(F) & 12(B) OF UGC ACT 1956
80 FEET ROAD, EDAPALAYAM, ALAMATHI VIA, REDHILLS, CHENNAI - 600052
MOB: +91 7010723984/85 | PH: +91-4426311001/1002/1045 | WWW.GSBT.EDU.IN

BONAFIDE CERTIFICATE

This is to certify that Mr VIGNESH M with registration number 110521104053 of the 2021-2025 batch, is a bona fide student of the Department of Computer Science and Engineering at Gojan School of Business and Technology, Redhills, Chennai 600052.

The student is currently in the **Final Year and Seventh Semester** of study. This certificate is issued to him/her to apply for an internship. During the internship, the student will represent our college and be permitted to engage in activities related to the branches of management studies.

During the due course of the internship, the respective student will abide by all the rules and regulations of the permitted company/industry/organization/institution.

We certify that the information provided in this certificate is true and accurate to the best of our knowledge.

Thanking You,

Principal
Gojan School of Business and Technology
Chennai 600052.

Dr. C. SELVAKUMAR
PRINCIPAL
GOJAN SCHOOL OF BUSINESS AND TECHNOLOGY
EDAPALAYAM, REDHILLS, CHENNAI-600 052

CODTECH IT SOLUTIONS PVT.LTD

IT SERVICES & IT CONSULTING

8-7-7/2, Plot NO.51,, Opp: Naveena School, Hasthinapuram Central, Hyderabad , 500 079., Telangana

Internship Offer Letter

Date : 13/07/2024
Inter ID :CT6CSEH1330

Dear VIGNESH M

Congratulations on being selected for the **"CYBER SECURITY&ETHICAL HACKING"** internship! We at **CODTECH IT SOLUTIONS PVT.LTD** are thrilled to have you join our team. **This internship will span 6 WEEKS from JULY 15th, 2024 to AUGUST 30th, 2024**.

This internship is designed as an educational experience, focusing on learning, skill development, and gaining practical knowledge. As an intern, we expect you to:

1. Complete all assignments to the best of your ability.
2. Follow any lawful and reasonable instructions provided by your supervisors.
3. Participate actively in team meetings and discussions.
4. Provide regular updates on your progress.
5. Adhere to company policies and maintain a professional demeanor.
6. Collaborate effectively with team members and contribute to group projects.
7. Seek feedback and apply it to improve your performance.

We trust that you will approach all tasks with diligence and enthusiasm. We are confident that this internship will be an enriching experience for you. We look forward to working with you and supporting you in achieving your career aspirations.

Best regards,

Neela Santhosh Kumar
Human Resources & Academic Head
CODTECH IT SOLUTIONS PRIVATE LIMITED

VERIFIED BY
CODTECH IT SOLUTIONS PVT LTD

www.codtechitsolutions.com

+91 9848925128
Hr@codtechitsolutions.com

CODTECH IT SOLUTIONS

8-7-7/2, Plot No: 51,Opp:Naveena School, Hasthinapuram Central, Hyderabad , 500 079., Telangana

+91 9848925128
hr@codtechitsolutions.com
www.codtechitsolutions.com

ISO 9001 CERTIFIED

CERTIFICATE

OF INTERNSHIP EXPERIENCE

This certificate is proudly presented to : CT6CSEH1330

VIGNESH M

successfully completed the Internship Program at **CODTECH IT SOLUTIONS** in **CYBER SECURITY&ETHICAL HACKING**

active participant from **15TH JULY 2024 TO 30TH AUGUST 2024** with unwavering dedication.

NEELA SANTHOSH KUMAR
HUMAN RESOURCES& ACADEMIC HEAD

VERIFIED BY
CODTECH IT SOLUTIONS PVT.LTD

ABSTRACT

This internship report outlines the practical experience gained and the skills developed during a cybersecurity internship. Cybersecurity is a critical field that focuses on protecting systems, networks, and data from malicious threats and vulnerabilities. The internship involved hands-on tasks such as vulnerability assessment, network monitoring, threat analysis, and implementation of security protocols.

The objectives of the internship were to strengthen theoretical knowledge through real-world applications, enhance problem-solving skills in identifying and mitigating risks, and gain familiarity with industry-standard tools and frameworks like SIEM (Security Information and Event Management) systems, firewalls, and encryption techniques.

Key accomplishments included identifying potential vulnerabilities in network configurations, responding to simulated cyber incidents, and drafting detailed reports on risk management strategies. Additionally, the internship emphasized compliance with cybersecurity standards, including GDPR and ISO 27001, to ensure adherence to legal and ethical guidelines.

This experience provided valuable insights into the dynamic challenges of cybersecurity, emphasizing the importance of continuous learning and proactive defense mechanisms in safeguarding digital assets. The knowledge and skills acquired during this period will be instrumental in pursuing a professional career in cyber security.

This internship underscored the necessity of continuous innovation and vigilance in the cybersecurity domain. The experience not only reinforced the importance of a strong technical foundation but also cultivated soft skills such as teamwork, communication, and critical thinking, all of which are essential for success in the dynamic field of cybersecurity.

Cybersecurity and Ethical Hacking Internship

Table of Contents

Internship Overview:

This internship focuses on providing practical exposure to cyber security concepts through tasks such as penetration testing, vulnerability analysis, and malware analysis. It aims to enhance technical skills and provide a solid foundation for a career in cybersecurity.

What is cyber security?

"Cyber security is primarily about people, processes, and technologies working together to encompass the full range of threat reduction, vulnerability reduction, deterrence, international engagement, incident response, resiliency, and recovery policies and activities, including computer network operations, information assurance, law enforcement, etc."

OR

Cyber security is the body of technologies, processes, and practices designed to protect networks, computers, programs and data from attack, damage or unauthorized access.

It is made up of two words one is cyber and other is security.

- Cyber is related to the technology which contains systems, network and programs or data.
- Whereas security related to the protection which includes systems security, network security and application and information security.

Why is cyber security important?

Listed below are the reasons why cyber security is so important in what's become a predominant digital world:

- Cyber attacks can be extremely expensive for businesses to endure.
- Cyber-attacks these days are becoming progressively destructive. Cybercriminals are using more sophisticated ways to initiate cyber attacks.

- In addition to financial damage suffered by the business, a data breach can also inflict untold reputational damage.

- Regulations such as GDPR are forcing organizations into taking better care of the personal data they hold.

Cyber security Fundamentals:

*Confidentiality:

*Confidentiality is about preventing the disclosure of data to unauthorized parties.

*It also means trying to keep the identity of authorized parties involved in sharing and holding data private and

*Often confidentiality is compromised by cracking poorly encrypted data, Man-in-the-middle (MITM) attacks, disclosing sensitive data.

*Standard measures to establish confidentiality include:

- Data encryption
- Two-factor authentication
- Biometric verification
- Security tokens

*Integrity:

*Integrity refers to protecting information from being modified by unauthorized parties.

*Standard measures to guarantee integrity include:

- Cryptographic checksums.
- Using file permissions.
- Uninterrupted power supplies.
- Data backups.

***Availability:**

*Availability is making sure that authorized parties are able to access the information when needed

*Standard measures to guarantee availability include:

- Backing up data to external drives.
- Implementing firewalls.
- Having backup power supplies.
- Data redundancy.

2. Internship Tasks::

To Participants are required to complete at least two out of six tasks, including developing tools, conducting security audits, and performing penetration testing. Detailed tasks are outlined below:

*Task 1: Password Strength Checker.
*Task 2: Vulnerability Scanning Tool.
*Task 3: Web Application Penetration Testing.
*Task 4: Malware Analysis Sandbox.
*Task 5: Red Team vs. Blue Team Simulation.
*Task 6: Blockchain Security Audit.

EXPLANATION OF TASK:

*TASK ONE: PASSWORD STRENGTH CHECKER:

Develop a tool to assess the strength of passwords entered by users. Implement algorithms to analyze factors such as length, complexity, and uniqueness to provide feedback on password strength.

***TASK TWO: VULNERABILITY SCANNING TOOL:**

Create a simple vulnerability scanning tool that scans a network or website for common security vulnerabilities such as open ports, outdated software versions, and misconfigurations.

***TASK THREE:WEB APPLICATION PENETRATION TESTING:**

Perform penetration testing on a web application to identify and exploit security vulnerabilities such as SQL injection, cross-site scripting (XSS), and insecure authentication mechanisms.

***TASK FOUR: MALWARE ANALYSIS SANDBOX:**

Build a sandbox environment for analyzing malware samples in a controlled environment. Implement features for executing malware samples, monitoring behavior, and analyzing malicious activities.

***TASK FIVE: RED TEAM VS. BLUE TEAM SIMULATION:**

Set up a simulated cybersecurity exercise where a red team (attackers) attempts to breach the security of a network defended by a blue team (defenders). Design and execute attack scenarios while monitoring and responding to security incidents.

***TASK SIX: BLOCKCHAIN SECURITY AUDIT:**

Conduct a security audit of a blockchain network to identify vulnerabilities and potential attack vectors. Analyze smart contracts, consensus mechanisms, and network infrastructure for security weaknesses.

2.1. Task 1: Password Strength Checker:

*Objective

Develop a tool to evaluate the strength of passwords based on critical factors such as **length**, **complexity**, and **uniqueness**. The tool should provide actionable feedback to users for creating stronger passwords.

Key Concepts:

1. **Password Strength Metrics**
   - **Length**: A strong password is typically 12+ characters long.
   - **Complexity**: Incorporate a mix of:
     - Uppercase letters (A-Z)
     - Lowercase letters (a-z)
     - Numbers (0-9)
     - Special characters (!, @, #, etc.)
   - **Uniqueness**: Avoid common passwords, sequences (e.g., `1234`, `abcd`), or predictable patterns.
   -
2. **Algorithm Components**
   - Evaluate password properties:
     - **Entropy Calculation**: Measures unpredictability.
     - Check against a **common password list** (e.g., `password`, `123456`).
   - Assign a **strength score** (e.g., Weak, Medium, Strong) based on predefined criteria.
   - Provide user feedback:
     - Suggestions for improvement (e.g., "Add special characters").
     - Warnings about risky patterns (e.g., "Avoid dictionary words").

Step 1: Design:

- **Input**: User-entered password.
- **Output**: A score or rating (e.g., Weak/Strong) and feedback.

Step 2: Algorithm

1. **Length Check**:
    - Set a minimum acceptable length (e.g., 8 characters).
    - Assign points for exceeding the minimum.

2. **Character Variety:**
    - Check for uppercase, lowercase, numbers, and special characters.
    - Reward passwords containing all types.

3. **Dictionary Lookup**:
    - Cross-check password against a list of commonly used passwords.
    - Reduce score if it matches any entry.

4. **Pattern Detection**:
    - Identify patterns like repeated characters (e.g., `aaaa`) or sequences (e.g., `abcd`).

5. **Entropy Calculation**:
    - Use Shannon Entropy or similar measures to assess randomness.

Step 3: Feedback Mechanism

- Provide tips for weak passwords:
    - "Increase password length."
    - "Add numbers or symbols."
    - "Avoid common patterns."

2.2. Task 2: Vulnerability Scanning Tool:

Objective:

Develop a simple vulnerability scanning tool that evaluates networks or websites to identify common security issues, such as open ports, outdated software versions, and misconfigurations.

Key Concepts:

1. **Vulnerability Scanning:**
    - A method to identify weaknesses in systems, applications, or configurations that could be exploited by attackers.
    - Focuses on finding potential entry points without actively exploiting them.
2. **Common Vulnerabilities to Scan:**
    - **Open Ports**: Ports that should be closed or restricted but are accessible.
    - **Outdated Software**: Applications or services running on unpatched versions.
    - **Misconfigurations**: Weak permissions, default credentials, or unsecured services.
    - **SSL/TLS Issues**: Weak encryption protocols or expired certificates.
3. **Scanning Techniques:**
    - **Port Scanning**: Use tools like `nmap` or `socket` libraries to find open ports.
    - **Banner Grabbing**: Extract service and version information from open ports.
    - **Configuration Checks**: Inspect server headers, default settings, or security flags (e.g., HTTP headers like `X-Frame-Options`).

Implementation Steps:

Step 1: Design:

- **Input**:
  - IP address or domain name.
- **Output**:
  - List of identified vulnerabilities with severity levels (e.g., Low, Medium, High).

Step 2: Tool Features:

1. **Port Scanner**:
   - Identify open ports and associated services.
   - Highlight unnecessary open ports that could be closed.
2. **Software Version Check**:
   - Compare version information against known vulnerabilities (CVE database or manual checks).
3. **Basic Configuration Checks**:
   - Check for default credentials or missing security headers.
4. **SSL/TLS Testing**:
   - Verify certificate validity and strength of encryption protocols.

Step 3: Algorithms and Tools:

Use existing Python libraries for scanning tasks:

- `socket` and `nmap` for port scanning.
- `requests` for HTTP headers and SSL checks.
- `ssl` for encryption tests.

Advanced Features:

To enhance the functionality and usability of the vulnerability scanning tool.

2.3. Task 3: Web Application Penetration Testing:

Objective

Perform penetration testing on a web application to identify and exploit vulnerabilities such as **SQL injection**, **cross-site scripting (XSS)**, and **insecure authentication mechanisms**. The goal is to evaluate the application's security posture and recommend measures to mitigate risks.

Key Concepts

1. **Penetration Testing:**
   - A systematic approach to simulate real-world attacks on a web application.
   - Aims to identify vulnerabilities and assess their exploitability without causing harm to the system.
2. **Common Vulnerabilities:**
   - **SQL Injection**: Exploits flaws in query handling to manipulate databases (e.g., bypassing authentication or extracting data).
   - **Cross-Site Scripting (XSS)**: Injects malicious scripts into web pages viewed by other users.
   - **Insecure Authentication**: Weak or flawed mechanisms that allow unauthorized access (e.g., password reuse, lack of multi-factor authentication).
   - **CSRF (Cross-Site Request Forgery)**: Forces authenticated users to perform actions without their consent.
3. **Testing Methodologies:**

OWASP Top 10: Focus on common and critical vulnerabilities.

Black-box Testing: No prior knowledge of the application.

Implementation Steps

Step 1: Reconnaissance:

- Gather information about the web application, including its:
  - Framework and technologies (e.g., PHP, JavaScript).
  - Publicly accessible endpoints.
  - Subdomains and server details.

Step 2: Vulnerability Scanning:

- Use automated tools like:
  - **Burp Suite**: For intercepting requests and testing vulnerabilities.
  - **OWASP ZAP**: For automated security scanning.
  - **Nikto**: To identify server-side vulnerabilities.

Step 3: Manual Testing:

1. **SQL Injection**:
   - Test form fields, URLs, and cookies for SQL payloads (e.g., `' OR 1=1;--`).
   - Verify if the database is leaking data or being manipulated.
2. **XSS**:
   - Inject scripts into input fields or URLs (e.g., `<script>alert('XSS')</script>`).
   - Check if the script executes in other users' browsers.
3. **Authentication Testing**:
   - Attempt to bypass authentication (e.g., weak password brute-forcing, session hijacking).
   - Check for secure storage and handling of credentials.
4. **CSRF Testing**:
   - Analyze if the application validates requests for authenticated actions.

2.4. Task 4: Malware Analysis Sandbox :

Objective:

Build a **sandbox environment** to safely analyze malware samples by executing them in a controlled and isolated setup. This tool helps identify the malware's behavior, attack vectors, and potential impact without affecting real-world systems.

Key Concepts:

1. **Malware Analysis:**
    - The process of dissecting malicious software to understand its purpose, functionality, and origin.
    - Types:
        - **Static Analysis**: Examining malware without execution (e.g., inspecting code or binary).
        - **Dynamic Analysis**: Observing malware behavior during execution in a controlled environment.
2. **Sandbox Environment:**
    - An isolated system or virtual machine designed for safely running and analyzing untrusted code.
    - Prevents the analyzed malware from interacting with external systems or causing damage.
3. **Use Cases:**
    - Identifying malicious payloads or behaviors.
    - Detecting indicators of compromise (IOCs), such as files, registry changes, or network activity.
    - Reverse engineering malware to develop mitigation strategies.

Automated Execution

- Automate malware deployment, monitoring, and data collection.

Implementation Steps

Step 1: Environment Setup

1. **Virtualization Tools**:
   - Use tools like **VirtualBox**, **VMware**, or **Hyper-V** to create virtual machines.
   - Set up multiple VMs with different operating systems (e.g., Windows, Linux).
2. **Isolation**:
   - Disable internet access or route traffic through a proxy for monitoring.
   - Use snapshots for reverting to a clean state after analysis.
3. **Instrumentation**:
   - Install tools for monitoring system activity:
     - File activity: Process Monitor (ProcMon).
     - Network activity: Wireshark, Fiddler.
     - Memory analysis: Volatility.

Step 2: Malware Execution and Monitoring

1. **Execution Tools**:
   - Deploy malware samples in the VM using controlled scripts.
   - Use time delays or triggers to analyze dormant or time-based malware.
2. **Monitoring**:
   - Capture key metrics during execution:
     - **Process Creation**: Identify new or unusual processes.
     - **File Changes**: Check for dropped files or modifications.
     - **Registry Changes** (on Windows): Observe added or modified registry keys.

2.5. Task 5: Red Team vs. Blue Team Simulation:

Objective:

Create a simulated cybersecurity exercise where a **red team** (attackers) attempts to compromise a network while the **blue team** (defenders) protects and responds. This simulation enhances the ability to detect, mitigate, and respond to real-world cyber threats.

Key Concepts:

1. **Red Team**:
    - Simulates attackers aiming to exploit vulnerabilities and breach the network.
    - Tests the effectiveness of the organization's defense mechanisms.
    - Employs techniques like phishing, malware deployment, and lateral movement.
2. **Blue Team**:
    - Represents defenders responsible for detecting, mitigating, and responding to attacks.
    - Utilizes tools like SIEMs (Security Information and Event Management) and IDS/IPS (Intrusion Detection/Prevention Systems).
    - Focuses on incident response and system recovery.
3. **Purple Team** (Optional):
    - Combines the efforts of red and blue teams to improve collaboration and overall security.

Network Simulation:

- Use virtualized environments (e.g., VirtualBox, VMware, or cloud platforms like AWS/Azure) to create the simulated network.

Implementation Steps

Step 1: Planning the Exercise

1. **Define Objectives**:
    - Test network defenses and detection mechanisms.
    - Evaluate incident response plans.
    - Identify weaknesses in both offensive and defensive strategies.
2. **Establish Scope**:
    - Specify the systems, applications, and data to be included in the simulation.
    - Define attack scenarios and rules of engagement (e.g., no permanent damage or data exfiltration).
3. **Roles and Teams**:
    - Assign participants to the red team, blue team, and observers.
    - Clearly outline responsibilities and communication channels.

Step 2: Setting Up the Environment:

1. **Deploy Tools**:
    - **Red Team**:
        - Exploitation tools: Metasploit, Cobalt Strike.
        - Reconnaissance tools: Nmap, Shodan.
        - Phishing simulation: GoPhish.
    - **Blue Team**:
        - Monitoring tools: Wireshark, Splunk, or ELK stack.
        - Endpoint protection: Antivirus, EDR solutions.
        - Response tools: Playbooks, forensic kits.

2.6. Task 6: Blockchain Security Audit :

Objective

Conduct a security audit of a blockchain network to identify vulnerabilities and potential attack vectors in its **smart contracts**, **consensus mechanisms**, and **network infrastructure**. The aim is to ensure the security, reliability, and integrity of the blockchain system.

Key Concepts

1. **Blockchain Components**:
    - **Smart Contracts**: Self-executing contracts with terms directly written in code.
    - **Consensus Mechanisms**: Methods to validate and agree on transactions (e.g., Proof of Work, Proof of Stake).
    - **Network Infrastructure**: Nodes, peer-to-peer communication, and underlying protocols.
2. **Types of vulnerabilities:**
    - **Smart Contracts**:
        - Logical bugs (e.g., reentrancy, integer overflow/underflow).
        - Unauthorized access (poor role management).
        - Gas inefficiencies leading to excessive costs.
    - **Consensus Mechanisms**:
        - Double spending.
        - 51% attacks or Sybil attacks.
        - Fork manipulation.
    - **Network Infrastructure**:
        - Denial of Service (DoS) attacks.
        - Man-in-the-Middle (MitM) attacks on node communications.
        - Outdated or misconfigured nodes.

Implementation Steps

Step 1: Smart Contract Analysis

1. **Automated Tools**:
   - Use tools like **Mythril**, **Slither**, or **Oyente** to scan for common vulnerabilities (e.g., reentrancy or overflows).
2. **Testnet Deployment**:
   - Deploy the smart contracts in a controlled testnet environment.
   - Simulate potential attack scenarios to test behavior under different conditions.
3. **Auditing Standards**:
   - Follow frameworks like the **SWC Registry (Smart Contract Weakness Classification)** for comprehensive coverage.

Step 2: Consensus Mechanism Review

1. **Fault Tolerance Testing**:
   - Simulate scenarios of node failure or malicious behavior.
   - Assess the network's ability to achieve consensus under adverse conditions.

Step 3: Network Infrastructure Audit

1. **Node Security**:
   - Check node configurations for default credentials or open ports.
   - Ensure nodes are running the latest, secure versions of the blockchain software.

Step 4: Reporting

1. **Risk Assessment**:
   - Categorize vulnerabilities by severity (e.g., Critical, High, Medium, Low).

***Advanced Features**

1. **Decentralized Security Testing**:
   - Test interactions between multiple smart contracts and external services (e.g., oracles).
2. **Penetration Testing**:
   - Simulate attacks such as **front-running** or **oracle manipulation**.
3. **Monitoring and Alerting**:
   - Implement tools to monitor blockchain activity for abnormal patterns (e.g., Etherscan for Ethereum).

3. Benefits of the Internship :

_1).PLACEMENT SUPPORT.
• INTERVIEW TRAINING.
• RESUME & LINKEDIN BUILDING TOOLS.
• SELF INTRODUCTION EXAMPLES.

2) **EXPERIENCE CERTIFICATE.**
3) **MENTOR SUPPORT TO DO GIVEN PROJECTS.**
4) **FLEXIBLE WORKING HOURS{WORK WHEN YOU ARE FREE AND SUBMIT THE TASKS WITHIN THE DEADLINE .**

***Submission Guidelines:**

Participants are required to submit their completed tasks via GitHub and share the repository links as per instructions.

***Blockchain Security Audit project:**

***GitHub LINK: https://github.com/VIGNESH-M-2003/CODTECH_INTERNSHIP-TASK1.git**

***Vulnerability Scanning Tool project:**

***GitHub LINK: https://github.com/VIGNESH-M-2003/CODTECH_INTERNSHIP-TASK1.git**

*Conclusion:

This internship offers a comprehensive platform to apply cybersecurity knowledge in practical scenarios, paving the way for a successful career in the field.