ADDIS ABABA UNIVERSITY

ADDIS ABABA INSTITUTE OF TECHNOLOGY

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

COMPUTER NETWORK AND SECURITY PROJECT

PHASE I SUBMISSION

Topic : Security Shepherd

GROUP NAME	ID NO
1. Bernabas getachew	UGR/1850/13
2.Birhan Kabtamu	UGR/8064/13
3.Daniel Adeba	UGR/4326/13
4.Tekalign Mesfin	UGR/7226/13
5.Siyamregn yeshidagna	UGR/3779/13

SUBMISSION DATE MAY 9,2024

SUBMITTED TO Mr KINDE M.

INTRODUCTION

- ✓ During this phase, we will thoroughly examine the Security Shepherd environment to identify any potential challenges and vulnerabilities. We will carefully analyze the system to uncover any security weaknesses, misconfigurations, or potential ways for attackers to exploit it. Our goal is to assess these vulnerabilities in terms of how severe they are, how likely they are to be exploited, and what impact they could have on the system's integrity and confidentiality. This process will help us gain a comprehensive understanding of the risks present and allow us to effectively mitigate them.

OBJECTIVIES

- ✓ Our objective in this phase is focus on conducting a detailed analysis of the Security Shepherd environment to identify and address any security vulnerabilities, weaknesses, and potential attack vectors in order to ensure the system's integrity and confidentiality.

What is a *Security shepherd ?*

- *Security Shepherd* is a web application developed by OWASP (Open Web Application Security Project) designed to teach web application security concepts in a hands-on, interactive manner. It provides a safe environment for individuals to practice and improve their skills in identifying and mitigating various security vulnerabilities commonly found in web applications.

Here below is a short overviews of Security Shepherd's key features and components

Vulnerable Web Applications: Security Shepherd includes a variety of intentionally vulnerable web applications that simulate real-world scenarios. These applications are riddled with security vulnerabilities such as *cross-site scripting (XSS)*, *SQL injection, CSRF (Cross-Site Request Forgery), insecure direct object references*, and more.

Learning Paths: Users can follow structured learning paths tailored to their skill levels and interests. These paths guide participants through a series of challenges, gradually increasing in complexity, to help them understand and mitigate different types of security vulnerabilities.

Challenge-based Learning: Security Shepherd offers a hands-on learning experience through challenges and exercises. Participants are tasked with identifying and exploiting vulnerabilities within the provided web applications, gaining practical experience in securing web applications.

Scoring and Progress Tracking: Users can track their progress and skill development as they complete challenges and learning paths. Security Shepherd provides scoring and feedback mechanisms to help users understand their strengths and areas for improvement.

Community Engagement: Security Shepherd encourages collaboration and knowledge sharing among users. Participants can discuss challenges, share insights, and seek assistance from the community through forums and other communication channels.

Open Source and Customization: As an open-source project, Security Shepherd allows users to contribute to its development and customize the platform to suit their specific needs. Organizations can tailor Security Shepherd to align with their training objectives and integrate it into their existing security education programs.

- ❖ **Vulnerability and challenges**

- ✓ let's break down the **security weaknesses** ,**misconfigurations**, and **potential attack vectors** that could affect the Security Shepherd environment:

1.***Security Weaknesses*** *(Vulnerabilities*)

a. Injection Vulnerabilities:

Security Shepherd's web applications may be vulnerable to SQL injection or command injection if proper input validation and parameterized queries are not implemented. Attackers could exploit this vulnerability to execute arbitrary SQL commands or system commands.

b. Cross-Site Scripting (XSS)

If Security Shepherd's web applications lack proper input sensitization and output encoding, they could be susceptible to XSS attacks. Attackers could inject malicious scripts that execute in users' browsers, leading to session hijacking, data theft, or defacement.

c. Insecure Direct Object References (IDOR)

If Security Shepherd's applications do not enforce proper access controls or expose sensitive object references, attackers could manipulate parameters to access unauthorized data or functionality. For example, accessing another user's profile or sensitive files.

2.Misconfigurations

a. Weak Authentication and Session Management

Inadequately configured authentication mechanisms in Security Shepherd, such as weak password policies or lack of proper session expiration controls, may create opportunities for attackers to gain unauthorized access to user accounts or hijack active sessions, potentially leading to security breaches and data compromise.

b. Insufficient Security Headers

If Security Shepherd's applications do not implement necessary security headers, such as Content Security Policy (CSP) or X-Frame-Options, they may be vulnerable to various attacks like XSS, click jacking, or data ex filtration.

c. File Upload Misconfigurations

Improperly configured file upload functionality without proper validation and restrictions could lead to security risks such as file inclusion attacks, malware uploads, or denial-of-service (DOS) attacks.

3.Attack Vectors

a. Brute Force Attacks

Attackers could attempt to brute force login credentials for Security Shepherd's administrative interfaces or user accounts, especially if weak or default passwords are used.

b. Session Hijacking

Attackers could intercept or steal session tokens through techniques like session fixation, session sniffing, or XSS attacks, gaining unauthorized access to user accounts or sensitive data.

c. Exploitation of Known Vulnerabilities

Attackers may exploit known vulnerabilities in Security Shepherd's underlying technologies or third-party libraries, especially if patches or updates are not promptly applied.

We have already seen how vulnerable is Security Shepherd now let us analysis it in a deep ways as shown below

❖ Vulnerability analysis

1. Injection Vulnerabilities

- Severity: High
- Likelihood of Exploitation: High
- Impact: Significant
- Explanation: SQL injection and command injection vulnerabilities allow attackers to inject malicious code into input fields, leading to unauthorized access to sensitive data, manipulation of database contents, or complete compromise of the system. These vulnerabilities are highly exploitable due to their prevalence and can result in severe consequences if not addressed promptly, compromising both system integrity and confidentiality.

2.Cross-Site Scripting (XSS):

- Severity: Medium to High
- Likelihood of Exploitation: Medium to High
- Impact: Significant
- Explanation: XSS vulnerabilities enable attackers to inject and execute malicious scripts in users' browsers, potentially leading to theft of session tokens, compromise of user accounts, or defacement of the application. While the impact may vary depending on the context and sensitivity of the data, XSS attacks are commonly used by attackers and pose a considerable risk to both system integrity and user confidentiality.

2. Insecure Direct Object References (IDOR)

- Severity: Medium
- Likelihood of Exploitation: Medium
- Impact: Moderate
- Explanation: IDOR vulnerabilities may allow attackers to access unauthorized data or perform unauthorized actions within the application. While the impact may not be as severe as injection vulnerabilities or XSS, IDOR can still lead to data breaches or unauthorized access to sensitive resources, compromising both system integrity and confidentiality to a moderate extent.

4. Weak Authentication and Session Management

- Severity: Medium
- Likelihood of Exploitation: Medium
- Impact: Moderate
- Explanation: Weak authentication mechanisms and session management practices increase the risk of unauthorized access to user accounts or session hijacking. While the impact may not be immediate or severe, compromised user accounts can lead to further exploitation or data breaches, compromising both system integrity and confidentiality to a moderate extent.

5. Insufficient Security Headers:

- Severity: Low to Medium
- Likelihood of Exploitation: Low to Medium
- Impact: Low to Moderate
- Explanation: Insufficient security headers may expose the application to various attacks such as XSS, click jacking, or data ex filtration. While the impact may vary depending on the specific vulnerability and attacker's goals, implementing proper security headers is essential for mitigating these risks and preventing compromise of system integrity and confidentiality to a low to moderate extent.

6. File Upload Misconfigurations

- Severity: Low to Medium
- Likelihood of Exploitation: Low to Medium
- Impact: Low to Moderate
- Explanation: Improperly configured file upload functionality may lead to file inclusion attacks, malware uploads, or DOS attacks. While the impact may not be as severe as other vulnerabilities, file upload misconfigurations can still result in data loss, service disruption, or compromise of the application, compromising system integrity and confidentiality to a low to moderate extent.

7. Brute Force Attacks

- Severity: Low to Medium
- Likelihood of Exploitation: Low to Medium
- Impact: Low to Moderate
- Explanation: Brute force attacks targeting login credentials pose a risk of unauthorized access to administrative interfaces or user accounts. While the impact may vary depending on the success of the attack and the privileges associated with compromised accounts, implementing proper password policies and rate-limiting mechanisms can help mitigate this risk, preventing compromise of system integrity and confidentiality to a low to moderate extent.

8. Session Hijacking

- Severity: Medium
- Likelihood of Exploitation: Medium
- Impact: Moderate
- Explanation: Session hijacking techniques such as session fixation, session sniffing, or XSS attacks can lead to unauthorized access to user accounts or sensitive data. While the impact may not be immediate or severe, compromised sessions can be leveraged for further exploitation or data theft, compromising both system integrity and confidentiality to a moderate extent.

9. Exploitation of Known Vulnerabilities

- Severity: Medium to High
- Likelihood of Exploitation: Medium to High
- Impact: Significant
- Explanation: Exploiting known vulnerabilities in underlying technologies or third-party libraries can result in unauthorized access, data breaches, or system compromise. While the impact may vary depending on the specific vulnerability and attacker's capabilities, prompt patching and updates are essential for mitigating this risk and preventing compromise of system integrity and confidentiality to a significant extent.