Penetration Testing Report

Executive Summary

This penetration testing report outlines the assessment conducted on the HackThisSite web application, focusing on the Basic challenges from Level 1 to 11. The assessment aimed to identify vulnerabilities within the application and provide recommendations for improving its security posture. Through comprehensive testing, several vulnerabilities were discovered across various levels, highlighting the importance of implementing robust security measures to safeguard against potential threats.

Scope of Web Application Tested

The scope of the assessment included the HackThisSite web application, specifically targeting the Basic challenges from Level 1 to 11. The assessment primarily focused on identifying vulnerabilities within the application's functionalities and security controls, aiming to assess its resilience against common attack vectors.

Vulnerability Description and Key Findings

1. Basic Level 1

- Vulnerability Description: Lack of input validation in the login form.
- Key Findings: Exploiting the lack of input validation allowed unauthorized access to the application.
- Recommendation: Implement proper input validation mechanisms to mitigate the risk of unauthorized access through injection attacks.

2. Basic Level 2

- Vulnerability Description: Insecure password storage.
- Key Findings: Passwords were stored in plaintext, posing a significant security risk.
- Recommendation: Utilize strong hashing algorithms (e.g., bcrypt) to securely store passwords and prevent unauthorized access to sensitive information.

3. Basic Level 3

- Vulnerability Description: Cross-Site Scripting (XSS) vulnerability.
- Key Findings: The application lacked proper input sanitization, enabling attackers to execute malicious scripts.
- Recommendation: Implement input sanitization techniques (e.g., encoding/escaping user input) to mitigate XSS attacks and enhance overall security.

4. Basic Level 4

- Vulnerability Description: Insecure Direct Object Reference (IDOR).
- Key Findings: Access controls were not adequately enforced, allowing unauthorized access to restricted resources.
- Recommendation: Implement proper access controls and enforce authorization mechanisms to prevent IDOR attacks and unauthorized access to sensitive data.

5. Basic Level 5

- Vulnerability Description: SQL Injection vulnerability.
- Key Findings: Lack of parameterized queries in database interactions led to SQL Injection attacks.
- Recommendation: Utilize parameterized queries or ORM frameworks to prevent SQL Injection attacks and enhance database security.

6. Basic Level 6

- Vulnerability Description: Insecure file upload functionality.
- Key Findings: Lack of file type validation allowed attackers to upload malicious files.
- Recommendation: Implement strict file type validation and content filtering to prevent malicious file uploads and mitigate associated risks.

7. Basic Level 7

- Vulnerability Description: Insecure session management.
- Key Findings: Weak session tokens and insufficient session expiration policies posed a risk of session hijacking.
- Recommendation: Implement secure session management practices, including the use of strong, random session tokens and proper session expiration policies to mitigate session hijacking attacks.

8. Basic Level 8

- Vulnerability Description: Insufficient authentication controls.
- Key Findings: Lack of multi-factor authentication (MFA) and weak password policies.
- Recommendation: Implement MFA for user authentication and enforce strong password policies (e.g., minimum length, complexity requirements) to enhance authentication security.

9. Basic Level 9

- Vulnerability Description: Cross-Site Request Forgery (CSRF) vulnerability.
- Key Findings: Lack of CSRF tokens in form submissions made the application susceptible to CSRF attacks.
- Recommendation: Implement CSRF tokens and enforce their inclusion in form submissions to mitigate CSRF attacks and protect against unauthorized actions.

10. Basic Level 10

- Vulnerability Description: Insecure deserialization.
- Key Findings: Lack of proper input validation in deserialization processes led to potential remote code execution.
- Recommendation: Implement input validation and integrity checks during deserialization to prevent deserialization vulnerabilities and mitigate the risk of remote code execution.

11. Basic Level 11

- Vulnerability Description: Insecure API endpoints.
- Key Findings: Lack of proper authentication and authorization controls in API endpoints exposed sensitive data.
- Recommendation: Implement robust authentication and authorization mechanisms for API endpoints, such as OAuth 2.0 or JWT-based authentication, to secure sensitive data and prevent unauthorized access.

Recommendations for Securing the Web Application

1. Implement comprehensive input validation mechanisms to prevent injection attacks.
2. Utilize strong encryption and hashing algorithms to securely store sensitive data, such as passwords.
3. Implement proper input sanitization techniques to mitigate XSS vulnerabilities.

4. Enforce strict access controls and authorization mechanisms to prevent unauthorized access to resources.
5. Utilize parameterized queries and ORM frameworks to prevent SQL Injection attacks.
6. Implement strict file upload validation to prevent malicious file uploads.
7. Employ secure session management practices to mitigate session hijacking attacks.
8. Implement multi-factor authentication (MFA) and enforce strong password policies for user authentication.
9. Implement CSRF tokens and enforce their inclusion in form submissions to mitigate CSRF attacks.
10. Implement input validation and integrity checks during deserialization to prevent deserialization vulnerabilities.
11. Secure API endpoints with robust authentication and authorization mechanisms to protect sensitive data.

Additional Resources:

- OWASP Top Ten: https://owasp.org/www-project-top-ten/
- PortSwigger Web Security Academy: https://portswigger.net/web-security
- HackThisSite forums and community resources for further guidance and support.

This report highlights the critical vulnerabilities discovered during the assessment of the HackThisSite web application and provides recommendations for mitigating these risks to enhance overall security. Implementing the suggested measures will strengthen the application's security posture and safeguard against potential threats.