TryHackMe Room Design Report.

IT 23 2387 94 – DISSANAYAKE Y Y

Table of Contents

Room Overview.

Topic Title

OWASP Top 10 Web Security

Room Summary

The purpose of this TryHackMe room is to provide students real-world, hands-on practice in comprehending and taking advantage of the OWASP Top 10 web application vulnerabilities. Every assignment in the room walks the participant through locating and taking advantage of a real-world weakness. To prepare students for careers in cybersecurity, penetration testing, and safe web development, the objective is to close the knowledge gap between theory and practice.

Key Concepts Covered

The OWASP Top 10, a list of the top ten security threats for web applications, serves as the basis for the room:

- Broken Access Control
- Cryptographic Failures
- Injection
- Insecure Design
- Security Misconfiguration
- Vulnerable and Outdated Components

- Identification and Authentication Failures
- Software and Data Integrity Failures
- Security Logging and Monitoring Failures
- Server-Side Request Forgery (SSRF)

Learning Objectives.

Broken Access Control.

Learn about the many methods of access control, including forced browsing, privilege escalation, and IDOR (Insecure Direct Object Reference).

Cryptographic Failures.

Discover how online apps use cryptography (e.g., HTTPS, password hashing). Determine the consequences of improper or non-existent use of encryption.

Injection.

Recognize how injection attacks, such as SQL injection, work.

Discover how attackers take advantage of input fields to run unwanted commands.

Insecure Design.

Understand the factors that contribute to an application's architecture being insecure. Acquire knowledge of the significance of threat modelling and security planning.

Security Misconfiguration.

Learn about instances such as default credentials, open ports, or exposed admin panels to gain an understanding of how incorrect server, app, or framework settings can reveal vulnerabilities.

Vulnerable and Outdated Components.

Understand the dangers of using out-of-date plugins, libraries, or dependencies.

Discover why software updates are essential for security.

Identification and Authentication Failures.

Recognize typical login system flaws. Acquire knowledge of the significance of secure session handling, MFA, and strong password policies.

Software and Data Integrity Failures.

Discover how software integrity can be violated in applications.
Recognize the dangers of CI/CD pipeline exposes and untrusted updates.

Security Logging and Monitoring Failures.

Understand the importance of monitoring and recording in identifying attacks.
Recognize the dangers of not having alerting systems or logs in place.

Server-Side Request Forgery (SSRF).

Learn how attackers can force the server to deliver fraudulent requests via server-side request forgery (SSRF).
Acquire knowledge of the possible consequences, such as accessing internal resources.

Room Structure.

The room is divided into 10 sections, each focusing on a specific OWASP vulnerability. Every section includes:

- Overview of the Concept
- Interactive Web App That Is Vulnerable
- Flag-based challenge
- Advice and Tips for Solution

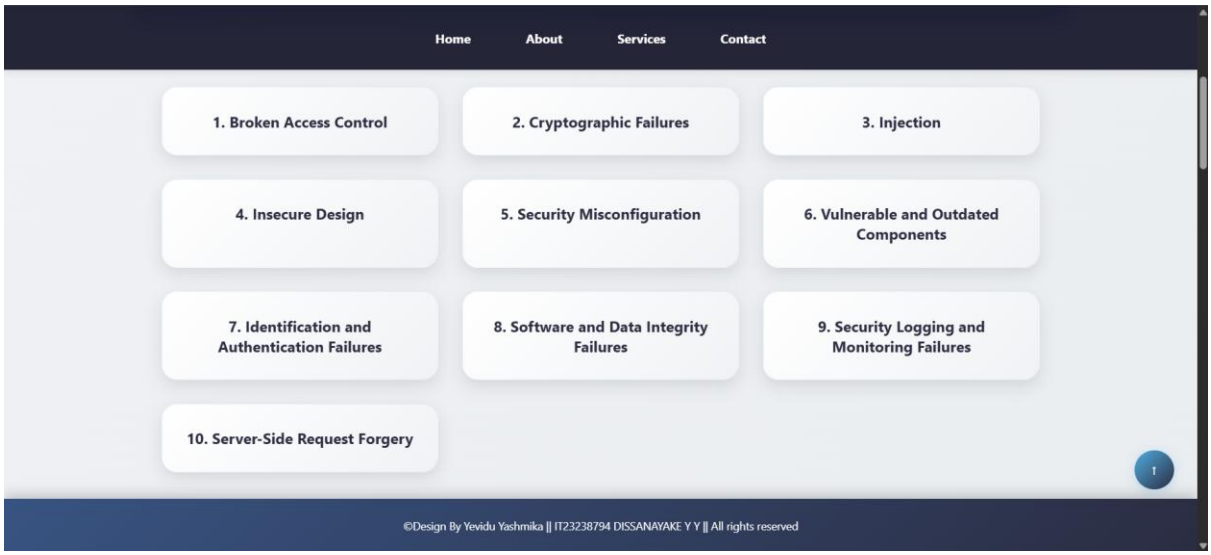
Room Links.

TryHackMe Room:

https://tryhackme.com/jr/owazaptop10websecurity

Project Website (Supplemental Material):

https://it23238794-ws.netlify.app/



Reflection.

Lessons Learned:

This room's construction allowed for a thorough exploration of cybersecurity education's technological and instructional design facets. It was beneficial to me:

- Learn the specifics of how each OWASP vulnerability operates.
- Discover how to securely model exploits in a sandbox setting.
- Develop my ability to write instructional materials and design lab sets that are easy to use.
- TryHackMe, Netlify, and Visual Studio Code are just a few of the platforms, frameworks, and security tools you can practice with.