**Élaboration de démonstrateurs d’attaques Web : OWASP Top 10 Vulnerabilities (2021)**

**Development of Web Attack Demonstrators : OWASP Top 10 Vulnerabilities (2021)**

|  |  |
| --- | --- |
| **Project supervisors:** | **Project team** |
| **Christophe Kiennert, Gregory Blanc** | **Antoine Pennamen, Salma El Bougrini** |

|  |  |
| --- | --- |
| **Mars 2025** | **Telecom Sud-Paris**  **MS-Cybersecurity** |

# Project abstract

* A brief summary of the project, including objectives, methodology, and key findings (typically 150-250 words)

**Table of contents**

[Project abstract 2](#_Toc179916283)

[1 Introduction 6](#_Toc179916284)

[2 Literature Review 6](#_Toc179916285)

[3 Methodology 6](#_Toc179916286)

[4 Vulnerabilities study 6](#_Toc179916287)

[4.1 Broken Access Control 6](#_Toc179916288)

[4.1.1 Description 6](#_Toc179916289)

[4.1.2 Occurrence 6](#_Toc179916290)

[4.1.3 Risk Measure 6](#_Toc179916291)

[4.1.4 Reproduction Methods 7](#_Toc179916292)

[4.2 Cryptographic Failures 7](#_Toc179916293)

[4.2.1 Description 7](#_Toc179916294)

[4.2.2 Occurrence 7](#_Toc179916295)

[4.2.3 Risk Measure 7](#_Toc179916296)

[4.2.4 Reproduction Methods 7](#_Toc179916297)

[4.3 Injection 8](#_Toc179916298)

[4.3.1 Description 8](#_Toc179916299)

[4.3.2 Occurrence 8](#_Toc179916300)

[4.3.3 Risk Measure 8](#_Toc179916301)

[4.3.4 Reproduction Methods 8](#_Toc179916302)

[**4.4** Insecure Design 9](#_Toc179916303)

[4.4.1 Description 9](#_Toc179916304)

[4.4.2 Occurrence 9](#_Toc179916305)

[4.4.3 Risk Measure 9](#_Toc179916306)

[4.4.4 Reproduction Methods 9](#_Toc179916307)

[4.5 Security Misconfiguration 9](#_Toc179916308)

[4.5.1 Description 9](#_Toc179916309)

[4.5.2 Occurrence 9](#_Toc179916310)

[4.5.3 Risk Measure 9](#_Toc179916311)

[4.5.4 Reproduction Methods 10](#_Toc179916312)

[4.6 Vulnerable and Outdated Components 10](#_Toc179916313)

[4.6.1 Description 10](#_Toc179916314)

[4.6.2 Occurrence 10](#_Toc179916315)

[4.6.3 Risk Measure 10](#_Toc179916316)

[4.6.4 Reproduction Methods 10](#_Toc179916317)

[4.7 Identification and Authentication Failures 11](#_Toc179916318)

[4.7.1 Description 11](#_Toc179916319)

[4.7.2 Occurrence 11](#_Toc179916320)

[4.7.3 Risk Measure 11](#_Toc179916321)

[4.7.4 Reproduction Methods 11](#_Toc179916322)

[4.8 Software and Data Integrity Failures 12](#_Toc179916323)

[4.8.1 Description 12](#_Toc179916324)

[4.8.2 Occurrence 12](#_Toc179916325)

[4.8.3 Risk Measure 12](#_Toc179916326)

[4.8.4 Reproduction Methods 12](#_Toc179916327)

[4.9 Security Logging and Monitoring Failures 12](#_Toc179916328)

[4.9.1 Description 12](#_Toc179916329)

[4.9.2 Occurrence 12](#_Toc179916330)

[4.9.3 Risk Measure 13](#_Toc179916331)

[4.9.4 Reproduction Methods 13](#_Toc179916332)

[4.10 Server-Side Request Forgery (SSRF) 13](#_Toc179916333)

[4.10.1 Description 13](#_Toc179916334)

[4.10.2 Occurrence 13](#_Toc179916335)

[4.10.3 Risk Measure 13](#_Toc179916336)

[4.10.4 Reproduction Methods 13](#_Toc179916337)

[5 Discussion 15](#_Toc179916338)

[6 Conclusion 15](#_Toc179916339)

[7 References 15](#_Toc179916340)

[8 Appendices 16](#_Toc179916341)

[9 Acknowledgments 16](#_Toc179916342)

[10 Executive Summary 17](#_Toc179916343)

# Introduction

* Background information on the topic.
* Purpose and objectives of the project.
* Scope of the report.
* Significance of the study.

# Literature Review

* Summary of existing research and literature related to the project.
* Identification of gaps in knowledge that your project addresses.

# Methodology

* Description of methods and approaches used in the project.
* Explanation of data collection techniques (e.g., surveys, experiments, interviews).
* Discussion of analysis methods.

# Vulnerabilities study

## Broken Access Control

### Description

Broken access control occurs when users can act outside their intended permissions, allowing unauthorized actions.

### Occurrence

Commonly found in web applications where user roles are poorly defined or enforced.

### Risk Measure

Risk Level: High. Exploitation can lead to data breaches, unauthorized data manipulation, and complete system compromise.

### Reproduction Methods

#### Easy Detection:

* + Attempt to access a restricted resource (e.g., admin dashboard) by manipulating the URL (e.g., /admin).
  + **Mitigation**: Implement strict role-based access controls and server-side validation of permissions.

#### Medium Difficulty Detection:

* + Use a testing tool to enumerate endpoints and check for unauthorized access (e.g., Postman).
  + **Mitigation**: Use security headers and enforce access control checks at all endpoints.

#### High Difficulty Detection:

* + Conduct a manual review of business logic in the application to find subtle access control flaws.
  + **Mitigation**: Perform regular code reviews and audits, including testing for business logic vulnerabilities.

## Cryptographic Failures

### Description

Cryptographic failures involve insecure cryptographic practices, leading to data exposure or manipulation.

### Occurrence

Occurs when sensitive data is improperly encrypted or when weak algorithms are used.

### Risk Measure

Risk Level: High. Can result in sensitive data exposure, leading to identity theft or fraud.

### Reproduction Methods

#### Easy Detection:

* + Inspect traffic with tools like Wireshark to find unencrypted sensitive data (e.g., passwords).
  + **Mitigation**: Enforce encryption (e.g., TLS) for data in transit and at rest.

#### Medium Difficulty Detection:

* + Analyze code for hard-coded keys or weak encryption algorithms.
  + **Mitigation**: Use strong encryption standards (e.g., AES) and secure key management practices.

#### High Difficulty Detection:

* + Conduct a security audit to uncover vulnerabilities in custom cryptographic implementations.
  + **Mitigation**: Regularly update cryptographic libraries and use vetted libraries instead of custom solutions.

## Injection

### Description

Injection vulnerabilities occur when untrusted data is sent to an interpreter, leading to execution of unintended commands.

### Occurrence

Common in SQL, NoSQL, and command injection scenarios.

### Risk Measure

Risk Level: High. Can lead to data loss, data corruption, or unauthorized access to systems.

### Reproduction Methods

#### Easy Detection:

* + Use SQL injection payloads (e.g., 1' OR '1'='1) in input fields.
  + **Mitigation**: Use parameterized queries and prepared statements.

#### Medium Difficulty Detection:

* + Test APIs with various payloads using tools like Burp Suite to detect injection points.
  + **Mitigation**: Implement input validation and sanitize user inputs.

#### High Difficulty Detection:

* + Analyze source code for vulnerable query construction, especially in complex applications.
  + **Mitigation**: Conduct regular code reviews and security testing, including fuzzing.

## Insecure Design

### Description

Insecure design refers to flaws in application design that fail to consider security risks.

### Occurrence

Often seen in applications lacking a security-focused design phase.

### Risk Measure

Risk Level: Medium to High. Can lead to multiple security vulnerabilities if not addressed during the design phase.

### Reproduction Methods

#### Easy Detection:

* + Review application features for unnecessary data exposure (e.g., error messages revealing stack traces).
  + **Mitigation**: Adopt secure design principles and perform threat modeling.

#### Medium Difficulty Detection:

* + Analyze application workflows for security gaps during user interactions.
  + **Mitigation**: Conduct security reviews during the design phase, involving security experts.

#### High Difficulty Detection:

* + Perform a comprehensive architecture review to identify inherent security flaws.
  + **Mitigation**: Utilize frameworks and best practices for secure design.

## Security Misconfiguration

### Description

Security misconfiguration occurs when default settings are not changed or when security controls are misconfigured.

### Occurrence

Common in cloud services, web servers, and application settings.

### Risk Measure

Risk Level: Medium. May lead to unauthorized access and data exposure.

### Reproduction Methods

#### Easy Detection:

* + Use security scanning tools (e.g., Nmap) to check for open ports or default credentials.
  + **Mitigation**: Regularly review configurations and change default settings.

#### Medium Difficulty Detection:

* + Conduct manual reviews of application settings and server configurations.
  + **Mitigation**: Implement automated configuration management tools.

#### High Difficulty Detection:

* + Analyze cloud infrastructure configurations for compliance with security policies.
  + **Mitigation**: Utilize Infrastructure as Code (IaC) practices to enforce secure configurations.

## Vulnerable and Outdated Components

### Description

Using outdated libraries or components that have known vulnerabilities can expose applications to attacks.

### Occurrence

Common in projects that rely on third-party libraries or frameworks.

### Risk Measure

Risk Level: Medium to High. Can lead to exploitation if vulnerabilities in components are publicly known.

### Reproduction Methods

#### Easy Detection:

* + Scan the application with tools like Snyk to find outdated dependencies.
  + **Mitigation**: Regularly update dependencies and monitor for vulnerabilities.

#### Medium Difficulty Detection:

* + Review the dependency tree for known vulnerabilities using tools like npm audit.
  + **Mitigation**: Establish a regular update schedule for dependencies.

#### High Difficulty Detection:

* + Conduct a manual code review to identify indirect dependencies and their vulnerabilities.
  + **Mitigation**: Adopt a policy for evaluating and vetting third-party components.

## Identification and Authentication Failures

### Description

Failures in user authentication and session management can allow attackers to gain unauthorized access.

### Occurrence

Often found in applications with weak password policies or improper session handling.

### Risk Measure

Risk Level: High. Can lead to account takeovers and unauthorized actions.

### Reproduction Methods

#### Easy Detection:

* + Test for weak passwords or lack of account lockout mechanisms.
  + **Mitigation**: Implement strong password policies and multi-factor authentication (MFA).

#### Medium Difficulty Detection:

* + Analyze session management logic for vulnerabilities (e.g., session fixation).
  + **Mitigation**: Use secure session management practices, including token expiration.

#### High Difficulty Detection:

* + Review authentication code for insecure implementations or patterns.
  + **Mitigation**: Conduct regular security assessments and code reviews focused on authentication.

## Software and Data Integrity Failures

### Description

Failures related to the integrity of software and data can allow unauthorized modifications.

### Occurrence

Common in applications that lack proper integrity checks for their software and data.

### Risk Measure

Risk Level: Medium. Can lead to tampering with application logic or data.

### Reproduction Methods

#### Easy Detection:

* + Check if application updates are performed without validation (e.g., file downloads without checks).
  + **Mitigation**: Implement cryptographic checksums to verify software integrity.

#### Medium Difficulty Detection:

* + Analyze the update mechanism for security flaws (e.g., lack of secure channels).
  + **Mitigation**: Use secure transmission methods and validate all data inputs.

#### High Difficulty Detection:

* + Conduct a thorough review of the software update processes and integrity verification methods.
  + **Mitigation**: Implement comprehensive integrity checks and monitoring systems.

## Security Logging and Monitoring Failures

### Description

Inadequate logging and monitoring can prevent timely detection of security breaches.

### Occurrence

Common in applications that do not maintain detailed logs of user activity or security events.

### Risk Measure

Risk Level: Medium. Delays in breach detection can lead to more extensive damage.

### Reproduction Methods

#### Easy Detection:

* + Review application logs for missing events or anomalies.
  + **Mitigation**: Implement comprehensive logging for all security-related events.

#### Medium Difficulty Detection:

* + Analyze log configurations for security and compliance.
  + **Mitigation**: Regularly review and test logging configurations.

#### High Difficulty Detection:

* + Perform a security audit to evaluate the effectiveness of logging and monitoring systems.
  + **Mitigation**: Establish incident response plans and regular monitoring reviews.

## Server-Side Request Forgery (SSRF)

### Description

SSRF vulnerabilities allow attackers to send unauthorized requests from a server to internal resources.

### Occurrence

Often found in web applications that fetch resources without proper validation of input URLs.

### Risk Measure

Risk Level: High. Can lead to access of sensitive data or internal services.

### Reproduction Methods

#### Easy Detection:

* + Attempt to submit a malicious URL that points to internal resources (e.g., http://localhost:8080).
  + **Mitigation**: Validate and sanitize all input URLs before processing.

#### Medium Difficulty Detection:

* + Use testing tools to send crafted requests to the application.
  + **Mitigation**: Implement network segmentation to limit server access.

#### High Difficulty Detection:

* + Conduct a thorough review of the codebase to find vulnerable request handling.
  + **Mitigation**: Apply strict access controls and regular security

# Discussion

* Analysis of the results in the context of the objectives.
* Comparison with findings from the literature review.
* Implications of the findings for theory, practice, or policy.

# Conclusion

* Summary of key findings.
* Reiteration of the significance of the study.
* Suggestions for future research or recommendations based on findings.

# References

1. **OWASP Foundation.** (2021). *OWASP Top Ten 2021: The Ten Most Critical Web Application Security Risks*. Available at: https://owasp.org/www-project-top-ten/
2. **OWASP Foundation.** (2021). *OWASP Cheat Sheet Series*. Available at: https://cheatsheetseries.owasp.org/

**Cybersecurity Literature**

1. **Shostack, A.** (2014). *Threat Modeling: Designing for Security*. Wiley.
   * This book discusses various vulnerabilities and how to design systems with security in mind.
2. **Hawkins, J.** (2021). *Web Application Security: A Beginner's Guide*. McGraw-Hill Education.
   * This resource provides insights into web application security, vulnerabilities, and mitigation strategies.

**Security Blogs and Whitepapers**

1. **Krebs, B.** (2021). *Krebs on Security*. Available at: <https://krebsonsecurity.com/>
   * A blog that provides insights and analyses on security incidents and vulnerabilities.
2. **OWASP Foundation.** (2021). *OWASP Security Blog*. Available at: https://owasp.org/www/blog/
   * Regular updates on vulnerabilities and security best practices from the OWASP community.

**Security Testing Tools**

1. **PortSwigger.** (2021). *Burp Suite Documentation*. Available at: https://portswigger.net/burp/documentation
   * Documentation for using Burp Suite, a popular tool for web application security testing.
2. **Snyk.** (2021). *Snyk Documentation*. Available at: https://docs.snyk.io/
   * Guidance on using Snyk for identifying vulnerabilities in open-source dependencies.

**Cybersecurity Training Resources**

1. **Coursera.** (2021). *Web Application Security Courses*. Available at: <https://www.coursera.org/courses?query=web%20application%20security>
   * A variety of courses focusing on web application security principles and practices.
2. **Udemy.** (2021). *Web Application Security: The Complete Guide*. Available at: https://www.udemy.com/course/web-application-security-the-complete-guide/
   * An online course that covers a comprehensive range of web security topics.
3. **Cybrary.** (2021). *Web Application Security Courses*. Available at: https://www.cybrary.it/course/web-application-security/
   * Training resources focused on web application security and the OWASP Top Ten.

# Appendices

* + Additional materials that support the report (e.g., raw data, questionnaires, detailed calculations).

# Acknowledgments

* + Recognition of individuals or organizations that contributed to the project.

# Executive Summary

* + A condensed version of the report, summarizing key points for decision-makers (if the report is for an audience that requires quick insights).