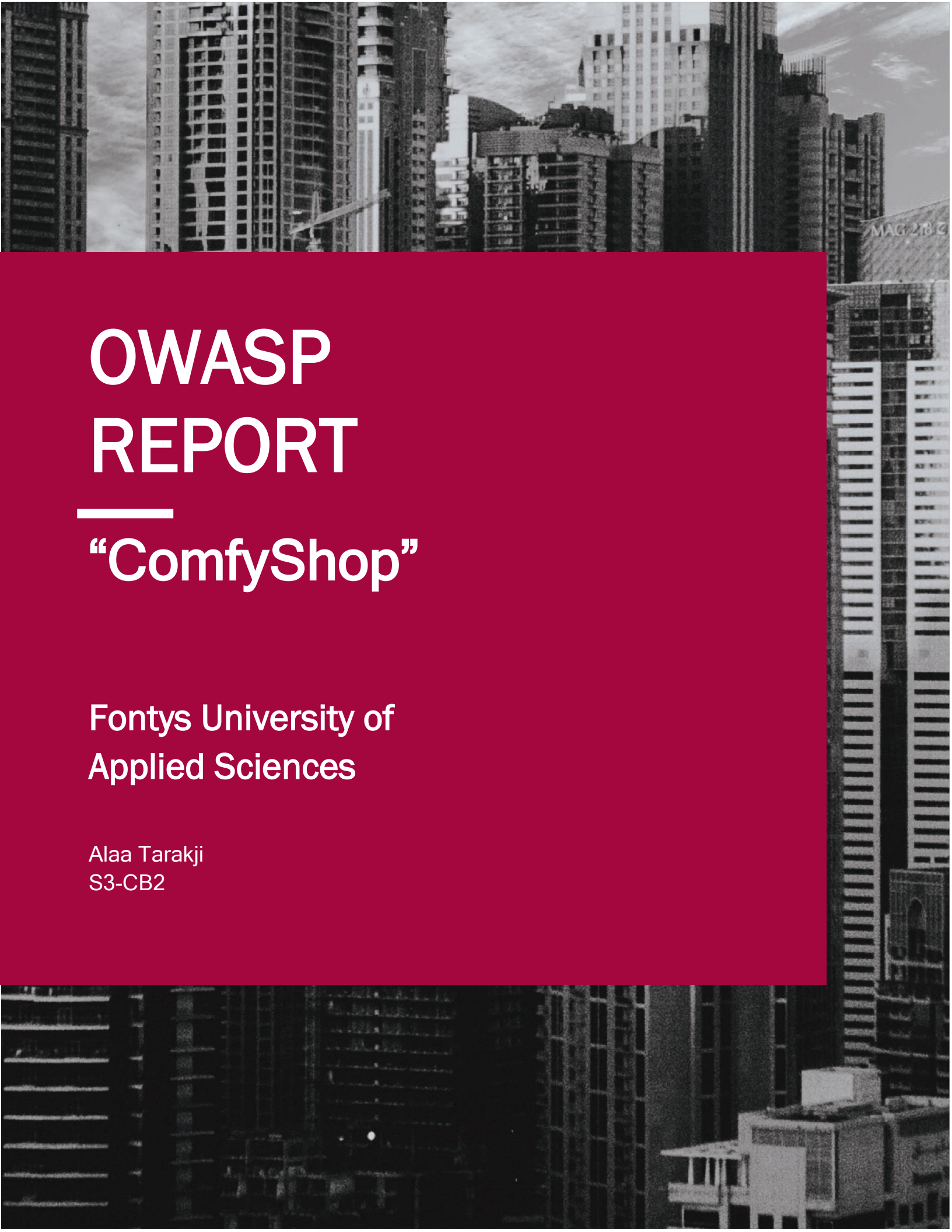

# OWASP REPORT

## “ComfyShop”

Fontys University of Applied Sciences

Alaa Tarakji
S3-CB2

# OWASP report

| | Likelihood | Impact | Risk | Actions | Planned |
|---|---|---|---|---|---|
| A1: broken access control | Low | Severe | Low | 1- Implement Role-Based Access Control (RBAC) using JWT and annotations.<br>2- Apply the @IsAuthenticated annotation to ensure that only authenticated users can access protected resources.<br>3- Use the @RolesAllowed({"ROLE_ADMIN"}) annotation to restrict access to users with the "ROLE_ADMIN" OR "ROLE_CUSTOMER" role. | Yes |
| A2: Cryptographic failure | Low | Severe | Low | 1- Use strong secret keys securely.<br>2- Implement secure cryptographic algorithms and protocols. | Yes |
| A3: Injection | unLikely | Severe | Low | 1- Implement input validation.<br>2- JPA (Java Persistence API) provides support for parameterized queries. | Yes |
| A4: Insecure Design | Low | Moderate | Low | Write unit and integration tests to validate all critical flows. | Unit tests, Integration tests |
| A5: Security Misconfiguration | Very likely | Severe | High | Error handling reveals stack traces or other overly informative error messages to users. | N/A |
| A6: Vulnerable and Outdated Components | Very unlikely | Moderate | Low | 1- Remove unused dependencies, because unused components can introduce unnecessary risks and potential vulnerabilities.<br>2- continuously inventory versions of application. | fixed |
| A7: Identification and Authentication Failures | Likely | Moderate | High | Multi-factor authentication. | Possibly switching to OAuth2 |
| A8: Software and Data Integrity Failures | Likely | Moderate | Moderate | Ensure that CI/CD pipeline has proper segregation, configuration, and access control to ensure integrity of the code. | N/A |
| A9: Security Logging and Monitoring Failures | Likely | Severe | High | | N/A |
| A10: Server-Side Request Forgery | Very likely | Severe | Moderate | 1- In the application layer, all client-supplied data should be sanitized and validated.<br>2- The URL schema, port and destination should be enforced. | N/A |

A1: Broken Access Control

- Ensure each user has appropriate privileges within the application.

Store sensitive keys securely or update them regularly. Avoid storing tokens in vulnerable locations.

A2: Cryptographic Failures
Protect sensitive data with encryption.

A3: Injection
Injection attacks occur when user input is not properly validated, allowing malicious code to be inserted. This can lead to unauthorized access or data manipulation. By implementing input validation and using parameterized queries, we reduce the chance of injection vulnerabilities, minimizing the impact and risk to our application.

A4: Insecure Design

- Test the application regularly to eliminate design errors.

A5: Security Misconfiguration

- Handle errors carefully to prevent exposing vulnerabilities.
- Minimize error details shared with users, especially potential attackers.

A6: Vulnerable and Outdated Components

- Ensure all software components are up-to-date and not vulnerable.

A7: Identification and Authentication Failures

- Use strong passwords and token refresh mechanisms.
- Safeguard tokens stored in local storage to prevent bypassing authentication.

A8: Software and Data Integrity Failures

- Maintain a secure CI/CD pipeline to prevent introducing malicious code.
-

A9: Security Logging and Monitoring Failures

A10: Server-Side Request Forgery

- Prevent unauthorized HTTP requests that may compromise security.
- Be cautious with the storage and alteration of JWT tokens.

# In Conclusion

Our application is secure thanks to the implementation of various security measures. We have addressed risks like access control, encryption, injection attacks, design flaws, configuration errors, outdated components, authentication issues, data integrity, logging and monitoring, and server-side request forgery. By enforcing user privileges, validating inputs, staying updated, and monitoring activities, we have reduced the risk of breaches. However, ongoing evaluation and improvement are essential to keep up with emerging threats and maintain a secure application.