SecurITY design Patterns lab

1. Preparation

Suggested readings

1. OWASP Application Security Verification Standard (ASVS): https://www.owasp.org/index.php/Category:OWASP_Application_Security_Verification_Standard_Project

Project

1. Each trainee should create a list of security design patterns utilized in previous projects, noting which ASVS requirements were fulfilled by applying those patterns.

2. Introduction

When integrating security into a software system there are two aspects that need to be addressed. First, we need to identify a need for a type of security control, to recognize that a security design pattern is required. This is the goal of design-level threat modeling or security design analysis. Once the correct pattern is selected, it is just as important to correctly implement it. This boils down to security control configuration, where we need to select a reliable provider, examine if the version of the implementation has vulnerabilities, and investigate what are the best practice configurations for that control. With poor configuration, a security control can introduce just as many (if not more[1]) vulnerabilities to the system as if there is no control.

3. Assignments

The following assignments are focused around the activity of research, where trainees are expected to explore public resources and documents to discover information, analyze it, and compound it to produce requirements for a secure implementation of a security design pattern.

a. Password Hashing

It is often the case that passwords used for user authentication do not need to be read. Rarely is there a use case for the query “SELECT passwords FROM Users”. Therefore, storing passwords in plaintext or even encrypted is not necessary. Instead, the contemporary approach to storing passwords entails the use of a hash function. When the user registers, the password is hashed and stored. When the user wants to log into the system, the supplied password is hashed, and the resulting hash is compared with the one in the database.

Assignment

Design a hashing mechanism with the goal of protecting the confidentiality of user passwords.

- Research different algorithms and select the most secure one;
- Examine the algorithm’s configuration parameters and discover the recommended best practice configuration;

[1] OpenSSL Heartbleed vulnerability, http://heartbleed.com/

- Choose a reliable provider;
- Investigate if the latest version of the implementation contains severe vulnerabilities;
- Specify requirements for a secure implementation of the hashing mechanism, using all the above as input.

B. Auditing Mechanism

Log files play an important role in software maintenance, as they provide information for debugging issues in running software. On the other hand, log files contribute towards the security of a system, as they provide non-repudiation and event entries which can be used by monitoring tools to detect malicious actors and suspicious behavior.

Assignment

Design an event logging mechanism which answers the following requirements:

- Log files need to provide debug information to support issue resolution;
- All events requiring non-repudiation need to be logged, enough information needs to be supplied to prove non-repudiation, and these events should be easily extracted;
- Log entries should not store sensitive data;
- The log mechanism needs to be reliable, ensuring the availability of the mechanism and integrity of the log files;
- Log entries across the system need to accurately state the creation time;
- The log mechanism should strive to minimize cluttering the log files.

The assignment is to research how each of the requirements can be fulfilled and to specify concrete implementation tasks for the designed mechanism. Using a complete solution is allowed, if it either fulfills all the requirements, or can be extended to fulfill them.

C. Additional Security Controls

Following the approaches used to solve the previous two assignments, it is possible to analyze any security design pattern and specify requirements for its secure implementation.

Assignment

Analyze the security controls implemented as part of previous projects and determine the extent to which the implemented configuration of the controls differs from a recommended secure configuration.