Risk Management

for

OpenRead Web-Application

Prepared by

Anthony Gilrandy Theo- 2201734814
Dewi Puspita Tanurezal - 2201740546
Devita - 2201755131
Steven - 2201852132

Bina Nusantara University

2020

Risk Assessment

Potential Risk

> SQL injection vulnerability
> Broken Authentication & Session Management
> Broken Access Control
> Data Loss
> Data Corruption
> Poor quality of end product

Risk Analysis

> SQL injection vulnerability

SQL injection is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database. It generally allows an attacker to view data that they are not normally able to retrieve. This might include data belonging to other users, or any other data that the application itself is able to access. In many cases, an attacker can modify or delete this data, causing persistent changes to the application's content or behavior. In some situations, an attacker can escalate an SQL injection attack to compromise the underlying server or other back-end infrastructure, or perform a denial-of-service attack.

> Broken Authentication & Session Management

These types of weaknesses can allow an attacker to either capture or bypass the authentication methods that are used by a web application.

- User authentication credentials are not protected when stored.
- Predictable login credentials.
- Session IDs are exposed in the URL (e.g., URL rewriting).
- Session IDs are vulnerable to session fixation attacks.
- Session value does not timeout or does not get invalidated after logout.
- Session IDs are not rotated after successful login.
- Passwords, session IDs, and other credentials are sent over unencrypted connections.

The goal of an attack is to take over one or more accounts and for the attacker to get the same privileges as the attacked user.

> Broken Access Control

Access control enforces policy such that users cannot act outside of their intended permissions. Failures typically lead to unauthorized information disclosure, modification or destruction of all data, or performing a business function outside of the limits of the user.

> Data Loss

Data loss is an error condition in information systems in which information is destroyed by failures (like failed spindle motors or head crashes on hard drives) or neglect (like mishandling, careless handling or storage under unsuitable conditions) in storage, transmission, or processing.

> Data Corruption

Data corruption refers to errors in computer data that occur during writing, reading, storage, transmission, or processing, which introduce unintended changes to the original data. Computer, transmission, and storage systems use a number of measures to provide end-to-end data integrity, or lack of errors.

> Poor quality of end product
Poor quality is not an inevitable attribute of software. It results from known causes. It can be predicted and controlled, but only if its causes are understood and addressed.

With more critical business processes being implemented in software, quality problems are a primary business risk.

Primary causes of poor software quality :

- Exceeding Timeframes
- Unmet requirements
- The lack of skilled web developers

Risk Prioritization

Top to Down (Most threatening to least threatening)

> Data Loss
> Data Corruption
> SQL injection vulnerability
> Broken Access Control
> Broken Authentication & Session Management
> Poor quality of end product

Risk Control

Risk Management Plan

>Data Loss

Multiple backup will be done daily, including local drives, virtualization, and cloud backup systems. This is to ensure redundancy and viability. The backups will be conducted into a testing as well to ensure the backup storage, quality, and the performance .

>Data Corruption

Same with data loss, multiple backup will be done daily, including local drives, virtualization, and cloud backup systems. This is to ensure redundancy and viability. The backups will be conducted into a testing as well to ensure the backup storage, quality, and the performance .

>SQL injection vulnerability

Parameterized Queries measured will be done. Parameterized queries are a means of pre-compiling an SQL statement so that we can then supply the parameters in order for the statement to be executed. This method makes it possible for the database to recognize the code and distinguish it from input data.

>Broken Access Control

Exception and errors code will be added. Also the following measures might be conducted :

- Implement access control mechanism in accordance with your business needs and re-using them throughout the application, including minimizing CORS usage
- Use Access control lists and also role-based authentication mechanisms
- Deny access to functionality by default with exception of public resources
- Disable web server directory listing and ensure file metadata and backup files are not present in webroot
- JWT tokens should be invalidated on the server-side after logout
- Rate limit API and controller access to minimize automated attacks
- Creating multi-layered login-in processes and workflow accessibility
- Monitoring activity for unauthorized personal-use web sites, telephone usage, and software installation, also log access control failures and alert admins when appropriate(like repeated failures)

>Broken Authentication & Session Management

- Password length must be at least 8 characters long.
- Session IDs will have timeout. User sessions or authentication tokens should be properly invalidated during logout.
- User credentials are stored using hashing and encryption.
- Session ID's will not be exposed in the URL to prevent url rewriting.

>Poor Quality of end product

Throughout the whole project, there will be Project Manager and people who will do Software Quality Assurance accordingly to assure the quality of our application. Robust testing environment will be conducted, criteria for the product's release will be selected very carefully. Hire qualified quality assurance tester to verify that the software meets its requirement and standards.