HANDLING OF SECURITY REQUIREMENTS IN SOFTWARE DEVELOPMENT LIFECYCLE

DANIEL KEFER
ISSUES
REPEATING MISTAKES
SECURITY DOCUMENTATION
SECURITY BEHIND DEV PROCESSES AND TOOLING
APPROACH
ALIGN THE PROCESS
SCALE
KISS
USE CASES

New assets

Production assets
INTERNALS
Based on JHipster
## Requirement Skeletons

### Secure Architecture

<table>
<thead>
<tr>
<th>Short Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SA-01</strong></td>
<td>3rd party code is identified, checked for security vulnerabilities and its update process is defined.</td>
</tr>
<tr>
<td></td>
<td>Implementation of automated tooling can support this task:</td>
</tr>
<tr>
<td></td>
<td>- <a href="https://www.owasp.org/index.php/OWASP_Dependency_Check">https://www.owasp.org/index.php/OWASP_Dependency_Check</a> (mapping of dependencies to CVEs)</td>
</tr>
<tr>
<td></td>
<td>- <a href="https://nodesecurity.io/tools">https://nodesecurity.io/tools</a> (evaluation of vulnerable packages for npm)</td>
</tr>
<tr>
<td></td>
<td>- <a href="http://retirejs.github.io/retire.js/">http://retirejs.github.io/retire.js/</a> (JavaScript libraries with known vulnerabilities)</td>
</tr>
<tr>
<td></td>
<td>Decrease the security risk being introduced by using vulnerable libraries. Be able to find out quickly if we’re affected when new vulnerabilities are published.</td>
</tr>
</tbody>
</table>

| **SA-02**  | No fundamentally different roles are present in the same application. |
|            | Example:  |
|            |   - internal employees and external customers should work on completely separated systems so that the privilege escalation probability and impact in case |

---

[Task]
### Optional Columns

<table>
<thead>
<tr>
<th>Short Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA-01</td>
<td>3rd party code is identified, checked for security vulnerabilities and its update process is defined.</td>
</tr>
<tr>
<td></td>
<td>Implementation of automated tooling can support this task:</td>
</tr>
<tr>
<td></td>
<td>- <a href="https://www.owasp.org/index.php/OWASP_Dependency_Check">https://www.owasp.org/index.php/OWASP_Dependency_Check</a> (mapping of dependencies to CVEs)</td>
</tr>
<tr>
<td></td>
<td>- <a href="https://nodesecurity.io/tools">https://nodesecurity.io/tools</a> (evaluation of vulnerable packages for npm)</td>
</tr>
<tr>
<td></td>
<td>- <a href="http://retirejs.github.io/retire.js/">http://retirejs.github.io/retire.js/</a> (JavaScript libraries with known vulnerabilities)</td>
</tr>
<tr>
<td></td>
<td>Decrease the security risk being introduced by using vulnerable libraries. Be able to find out quickly if we’re affected when new vulnerabilities are published.</td>
</tr>
</tbody>
</table>

| SA-02      | No fundamentally different roles are present in the same application. |
|            | Example: |
|            | - internal employees and external customers should work on completely separated systems so that the privilege escalation probability and impact in case |
## Alternatives to Option Columns

### JAVA Application

**Example of a prepared statement for SQL queries:**

```java
String selectSQL = "SELECT USER_ID, USERNAME FROM DBUSER
WHERE USER_ID = ?";
PreparedStatement preparedStatement = dbConnection.prepareStatement(selectSQL);
preparedStatement.setInt(1, 1001);
ResultSet rs = preparedStatement.executeQuery(selectSQL);
while (rs.next()) {
    String userid = rs.getString("USER_ID");
}
```
### Secure Architecture

<table>
<thead>
<tr>
<th>Short Name</th>
<th>Description</th>
<th>Task</th>
</tr>
</thead>
</table>
| SA-01      | 3rd party code is identified, checked for security vulnerabilities and its update process is defined. Implementation of automated tooling can support this task:  
- https://www.owasp.org/index.php/OWASP_Dependency_Check (mapping of dependencies to CVEs)  
- https://nodesecurity.io/tools (evaluation of vulnerable packages for npm)  
- http://retirejs.github.io/retire.js/ (JavaScript libraries with known vulnerabilities)  
Decrease the security risk being introduced by using vulnerable libraries. Be able to find out quickly if we’re affected when new vulnerabilities are published. | |
| SA-02      | No fundamentally different roles are present in the same application. Example:  
- internal employees and external customers should work on completely separated systems so that the privilege escalation probability and impact in case | |

### Status Columns

<table>
<thead>
<tr>
<th>Short Name</th>
<th>Description</th>
<th>More Information</th>
<th>Motivation</th>
<th>Strategy</th>
<th>Comment</th>
<th>Select</th>
</tr>
</thead>
</table>
Implementation Type

Artifact Properties:

- Criticality
- System Type
- Authentication
- Session Management
- Reachability

Implementation:

Implementation Type
Collections

Artifact Properties:

- Criticality
- System Type
- Authentication
- Session Management
- Reachability

Implementation:

- Implementation Type
## Tags

### Artifact Settings

#### Tags

<table>
<thead>
<tr>
<th>Requirement Owner</th>
<th>Product Manager</th>
<th>Security Mentor</th>
<th>Project Manager</th>
<th>SCRUM Master</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase relevance</td>
<td>Initiation</td>
<td>Design</td>
<td>Coding</td>
<td>QA</td>
</tr>
<tr>
<td>QA</td>
<td>BlackBox</td>
<td>Functional Test</td>
<td>White box</td>
<td></td>
</tr>
<tr>
<td>Documentation</td>
<td>Design</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
AUTHENTICATION

Own authentication scheme

CAS (Central Authentication Service)
ROLES

Frontend User

User

Admin
JIRA INTEGRATION

Cross Origin Request Sharing

SecurityRAT inherits user’s rights in JIRA
Test requirements

Please make sure that the selected requirements are testable. Depending on how a requirement is tested, make sure to fill the necessary fields.

You have selected 9 requirements.

Application URL: http://example.com

SCM URL: https://gitlab_url/example_com

Sonarqube Key: com.example
Test results

Alternatively is the result available at https:///serviceapi/resource/46 for a week as from now.

<table>
<thead>
<tr>
<th>Short Name</th>
<th>Description</th>
<th>Result</th>
<th>Confidence level</th>
<th>Message</th>
<th>Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>IV-08</td>
<td>Buffer overflow attacks are mitigated.</td>
<td>✔️</td>
<td>90%</td>
<td>The test was successful.</td>
<td>sonarMS</td>
</tr>
<tr>
<td>EHL-01</td>
<td>The system does not output error messages data that could assist an attacker.</td>
<td>✗️</td>
<td>70%</td>
<td>The test was unsuccessful. Check for the sonarqube vulnerabilities to your projects with tag(s) error-handling.</td>
<td>sonarMS</td>
</tr>
<tr>
<td>SA-01</td>
<td>3rd party code is identified, checked for security vulnerabilities and its update process is defined.</td>
<td>✔️</td>
<td>70%</td>
<td>The test was successful.</td>
<td>sonarMS</td>
</tr>
<tr>
<td>IV-04</td>
<td>Cross-Site Request Forgery attacks are mitigated.</td>
<td>✔️</td>
<td>50%</td>
<td>The test was successful.</td>
<td>sonarMS</td>
</tr>
<tr>
<td>OE-01</td>
<td>All untrusted data outputted to any interface are properly escaped for the particular context using a common and standardized approach.</td>
<td>✔️</td>
<td>90%</td>
<td>The test was successful.</td>
<td>sonarMS</td>
</tr>
</tbody>
</table>
FUTURE PLANS
SECURITYRAT 2.0

COMMUNITY

Issues
Pull requests
Derived projects
THANK YOU FOR YOUR ATTENTION!

https://securityrat.github.io

dan.kefer@gmail.com