Owasp EJSF Project Plan

Version 1.0 ● 26 July 2013

Owasp EJSF

PROJECT PLAN

[Owasp org]

[Development of Security Framework based on Owasp Esapi for JSF2.0]

|  |  |
| --- | --- |
| VERSION: [1.0] | REVISION DATE: [26 June 2013] |

*Approval of the Project Plan indicates an understanding of the purpose and content described in this deliverable. Approval of the Project Plan constitutes approval of the project planning results and hereby certifies the overall accuracy, viability, and defensibility of the content and estimates. By signing this deliverable, each individual agrees the project has been planned effectively as described herein.*

|  |  |  |
| --- | --- | --- |
| **Project Manager** | | |
| [Prof.Dr Emmanuel Benoist] | [emmanuel.benoist@bfh.ch] | [Telephone] |
| Signature | | Date |

|  |  |  |
| --- | --- | --- |
| **Developer** | | |
| [Rakesh Kachhadiya] | [Rakeshkumar.kachhadiya@owasp.org] | [0049-17631333817] |
| Signature | | Date |

Contents

[Section 1. Project Overview 1](#_Toc249246255)

[1.1 Project Description 1](#_Toc249246256)

[1.2 Project Scope 1](#_Toc249246257)

[1.3 Assumptions 1](#_Toc249246258)

[1.4 Constraints 2](#_Toc249246259)

[Section 2. Project Organization 3](#_Toc249246260)

[2.1 Project Structure 3](#_Toc249246261)

[2.2 Roles and Responsibilities 3](#_Toc249246263)

[Section 3. Project Start-Up 4](#_Toc249246264)

[3.1 Project Life Cycle 4](#_Toc249246265)

[3.2 Methods, Tools, and Techniques 4](#_Toc249246266)

[3.3 Schedule Allocation 5](#_Toc249246269)

[Section 4. Quality Management 8](#_Toc249246276)

[4.1 Quality Management Approach 8](#_Toc249246277)

[4.2 Quality Objectives and Standards Identification 8](#_Toc249246278)

[4.3 Project Reviews and Assessments 8](#_Toc249246279)

[4.4 Deliverables Acceptance Criteria 9](#_Toc249246280)

[4.5 Process Improvement Activities 9](#_Toc249246281)

[Section 5. Configuration Management 11](#_Toc249246286)

[5.1 Configuration Management Approach 11](#_Toc249246287)

[5.2 Configuration Management Tools, Environment, and Infrastructure 11](#_Toc249246288)

[5.3 Configuration Identification 11](#_Toc249246289)

[5.4 Configuration Control 11](#_Toc249246290)

[5.5 Status Accounting and Reporting 11](#_Toc249246291)

[5.6 Audits and Reviews 11](#_Toc249246292)

[5.7 Interface Control 12](#_Toc249246293)

[5.8 Vendor Control 12](#_Toc249246294)

[Section 6. Risk Management 14](#_Toc249246298)

[6.1 Risk Management Approach 14](#_Toc249246299)

[6.2 Risk Assessment 14](#_Toc249246300)

[6.3. Risk Monitoring and Control 15](#_Toc249246301)

[Section 7. Project Transition 16](#_Toc249246302)

[7.1 Closeout Plan 16](#_Toc249246303)

[7.2 Phase Closeout 16](#_Toc249246304)

[Section 8. References 17](#_Toc249246305)

[Section 9. Glossary 18](#_Toc249246306)

[Section 10. Revision History 19](#_Toc249246307)

[Section 11. Appendices 20](#_Toc249246308)

# Section 1. Project Overview

## Project Description

Owasp EJSF is a software security framework based on JSF2.0 to provide additional security features to the existing or new JSF2.0 based web application without much development efforts. Application will be automatically configured with the security frame work by configuring few existing xml files which is indirectly using Owasp Esapi in the backend with additional security programs. The main purpose of the project to deliver the JSF2.0 based web application which is minimum affected with the security breaches such as cross-site scripting (XSS), cross-site request forgery (CSRF), File based authorization and many more.

## 1.2 Project Scope

| **Project Includes** |
| --- |
| 1. Cross-site scripting. |
| 1. Cross-site request forgery. |
| 1. File based authorization. |

| **Project Excludes** |
| --- |
|  |
|  |
|  |

## 1.3 Assumptions

| **Assumptions** |
| --- |
| This security framework is based on Apache my faces, as of now, it is not compatible with all other available library such as sun java faces. |
|  |
|  |

## 1.4 Constraints

| **Constraints** |
| --- |
| The size of team is the main constraints as of now. |
|  |
|  |

# Section 2. Project Organization

## 2.1 Project Structure

## 

## 2.2 Roles and Responsibilities

| **Role** | **Responsibility** |
| --- | --- |
| Project Manager | Remove project obstacles, understand project requirement and documentation |
| Developer | Development of core security features. |
|  |  |

# Section 3. Project Start-Up

## 3.1 Project Life Cycle

It follows the incremental development approach in which all the requirements are modeled in the beginning, however design and implementation occurs at every iteration of the projects. Moreover, in each iteration of the project add new features.

## 3.2 Methods, Tools, and Techniques

The main programming language is Java, Moreover Java Server Faces 2.0 framework used for development of the security framework which is calling Owasp Esapi library in the backend. Apache tomcat 5.0 deploys the applications.

## 3.3 Schedule Allocation

| **Project Schedule Location** | **N/A** |
| --- | --- |

| **Major Milestone/Deliverable** | **Planned Completion Date** |
| --- | --- |
| N/A | N/A |
|  |  |
|  |  |
|  |  |
|  |  |

## 

# Section 4. Quality Management

## 4.1 Quality Management Approach

*N/A*

## 4.2 Quality Objectives and Standards Identification

*N/A*

## 4.3 Project Reviews and Assessments

*N/A*

| **Review Type** | **Frequency** | **Tools** | **Reviewer** | **Reports** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## 4.4 Deliverables Acceptance Criteria

*N/A*

| **Deliverable** | **Final Approval Process** | **Stakeholder Acceptance Criteria** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## 4.5 Process Improvement Activities

*Describe the activities that will be performed periodically to assess the project’s processes, identify areas for improvement, and implement improvement plans.*

⇒

# Section 5. Configuration Management

## 5.1 Configuration Management Approach

*Describe the overall, high-level approach to configuration management (CM) for the project.* *Summarize how configuration management activities will be accomplished collectively. If formal configuration management policies and procedures have been established at the organization or agency level, refer to the agency and/or organizational practices. In the description, include tailoring of any practices to accommodate specific project needs if applicable.*

⇒

## 5.2 Configuration Management Tools, Environment, and Infrastructure

## 

## 5.3 Configuration Identification

## 5.4 Configuration Control

## 5.5 Status Accounting and Reporting

## 5.6 Audits and Reviews

## 5.7 Interface Control

## 5.8 Vendor Control

# Section 6. Risk Management

## 6.1 Risk Management Approach

*N/A*

## 6.2 Risk Assessment

### 6.2.1 Risk Identification

*N/A*

### 6.2.2 Risk Analysis

*N/A*

|  |  |
| --- | --- |
| **Risk Analysis Description** |  |
| **Scales Description** |  |
| **Risk Threshold Values Description** |  |

### 6.2.3 Risk Response Strategies

*N/A*

## 6.3. Risk Monitoring and Control

### 6.3.1 Risk Tracking

*N/A*

### 6.3.2 Risk Reporting

*N/A*

# Section 7. Project Transition

## 7.1 Closeout Plan

## 7.2 Phase Closeout

N/A

# Section 8. References

| **Document No.** | **Document Title** | **Date** | **Author** |
| --- | --- | --- | --- |
| 001 | Development of the Security  Framework based on  OWASP ESAPI for JSF2.0 | 14th May 2013 | Matthey Samuel |
| 002 | Development of the Security  Framework based on  OWASP ESAPI for JSF2.0 | 3rd May 2012 | Rakesh Kachhadiya |

# Section 9. Glossary

Modern web application frameworks have made it easy to develop high quality web applications, but developing a secure application still requires programmers to possess a deep understanding of security vulnerabilities and attacks. Sometimes it is difficult for an experienced developer to find and eliminate all the vulnerabilities. This demo represents the JSF-ESAPI framework based on JSF2.0 and OWASP ESAPI. It helps developers write a secure and lower-risk JSF based on a web application with minimal configuration and without extensive prior knowledge of the web security. The figure below shows a complete integration of the JSF-ESAPI framework with JSF2.0 and ESAPI. It works as a middleware and consists of four important modules. ESAPI Validation is the first module which verifies the user input as given in the XSS prevention cheat sheet provided by OWASP. It consists of many user-defined validator tags and generates appropriate error messages if the user input is not valid. In this way it performs a strong validation.

There are also some tags available in the correspondence validators in ESAPI, and they also filter the XSS relevant code from the input. The file-based authorization module simplifies the user’s role, such as admin, user, etc. and gives the permission to visualize certain components at the presentation layer based on assigned roles. ESAPI Filtering layer compares the valid form token with tokens stored in the session for that user. If the token is mitigated or changed by man in the middle during the process of a request-response exchange, it will give the appropriate exception.

The last module is a render module, which renders the output after filtering the XSS content and encodes the vulnerable characters such as <,>,”,’ etc. as given in the XSS prevention cheat sheet provided by OWASP.

This framework will help developers to prevent a myriad of security problems including cross-site scripting, cross-site request forgery, automatic input validation, and automatic output validation with escaped “true” or without this parameter, authorization. All the features are included in one framework.

Advantages:- (1) It requires minimal configuration to use the framework. (2) It ensures retrofit security in the existing application. (3) It provides the same performance as JSF framework. (4) Automatic filtering of the XSS vulnerable code from output takes place when escape equals true” or “false”. (5) The input validation is easy and no additional coding is required. (6) It has a layered architecture. It uses what you need and leaves what you don’t need at the moment. (7) One framework includes the most secure features."

# Section 10. Revision History

*Identify changes to the Project Plan.*

| **Version** | **Date** | **Name** | **Description** |
| --- | --- | --- | --- |
| N/A | N/A | N/A | N/A |
|  |  |  |  |

# Section 11. Appendices

1. <http://lamagiepourlesenfants.com/downloads/doc/Security_framework_OWASP_esapi_JSF_14.05.2013.pdf>
2. <https://www.owasp.org/images/e/e6/AppSecEU2012_Benoist.pdf>
3. <http://lamagiepourlesenfants.com/downloads/doc/Master-Thesis.pdf>
4. <http://lamagiepourlesenfants.com/>