**Secure S/W Development**

**-과제 관리 시스템**

**Instructor :**Prof. Shinhye Park

**Prepared by :**<Team 1>  
임혜선  
조은상  
양경석  
이수훈  
홍명기

# Table of Contents

[Table of Contents 2](#_Toc436901114)

[Figure & Table 2](#_Toc436901115)

[1. Introduction 3](#_Toc436901116)

[1.1 Purpose 3](#_Toc436901117)

[1.2 Specification 3](#_Toc436901118)

[1.3 Function 4](#_Toc436901119)

[1.4 Site Map 4](#_Toc436901120)

[1.5 Database Structure 5](#_Toc436901121)

[2. 보안 요구사항 적용 7](#_Toc436901122)

[2.1 기본 보안 설정 7](#_Toc436901123)

[2.2 OWASP 적용 8](#_Toc436901124)

[3. Secure Coding 9](#_Toc436901125)

[3.1 Securecoding 9](#_Toc436901126)

[4. Tool 9](#_Toc436901127)

[4.1 PMD 9](#_Toc436901128)

[4.2 Acunetix 11](#_Toc436901129)

[5. Project Summary 11](#_Toc436901130)

[5.1 Business Plan 12](#_Toc436901131)

# Figure & Table

[Table 1. 과제관리 시스템 Page 3](#_Toc436664207)

[Table 2. 과제관리 시스템 Specification 3](#_Toc436664208)

[Table 3. Site Map 4](#_Toc436664209)

[Table 4. Site Map - Page 별 address 5](#_Toc436664210)

[Table 5. 기본보안 적용내용 6](#_Toc436664211)

[Table 6. OWASP 적용내용 7](#_Toc436664212)

# Introduction

## Purpose

과제 제출, 점수부여, 점수확인을 하는 AIMS2 E-Class 과제 게시판 기능을 하는 Web Service 형태의 [과제 관리 시스템]을 개발하였다. Secure S/W 설계 과목에서 추구하는 Secure coding을 위하여 우리의 프로젝트에 CIA, OWASP Top10을 적용하고 Manual Test 및 PMD, Acunetix tool을 통하여 지적된 보안요구 사항을 적용함으로써 이미 존재하고 있거나 발생 가능한 위협과 취약점에 대한 우려사항을 해결하고자 하였다.

## Specification

[과제 관리 시스템]은 Web service를 하는 시스템으로 Client에게 Web browser를 통하여 서비스 한다. Web Service로 Apache Tomcat8, DB Server로 My SQL을 사용하였다.

|  |  |
| --- | --- |
| **Home page** | <http://210.107.197.133:8080/secure/> |
| **FTP Server** | <ftp://210.107.197.133:21> |
| **Database** | 210.107.197.133/myadmin/ |

Table . 과제관리 시스템 Page

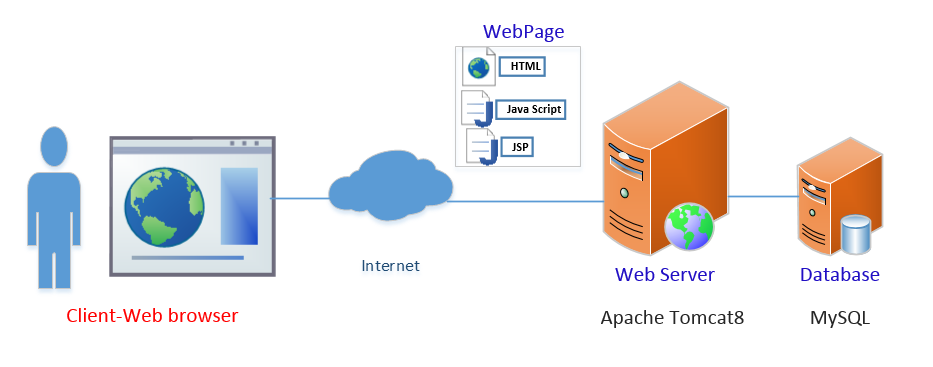


Figure . 과제관리 시스템 Specification

## Function

[과제 관리 시스템]에서 제공하는 서비스에 포함된 기능은 아래와 같다.

1. 회원가입, 회원정보 수정, 회원관리, Log-in, Log-out
2. 게시판: 과제등록, 등록과제 수정, 과제제출, 제출된 과제확인, 제출과제 수정, 점수부여, 점수확인, 점수수정, 과제파일 업로드
3. Access control: 사용자 권한에 따른 page, data 접근통제 (관리자, 교수, 학생)

## Site Map

[과제 관리 시스템]의 Site Map은 아래와 같다.

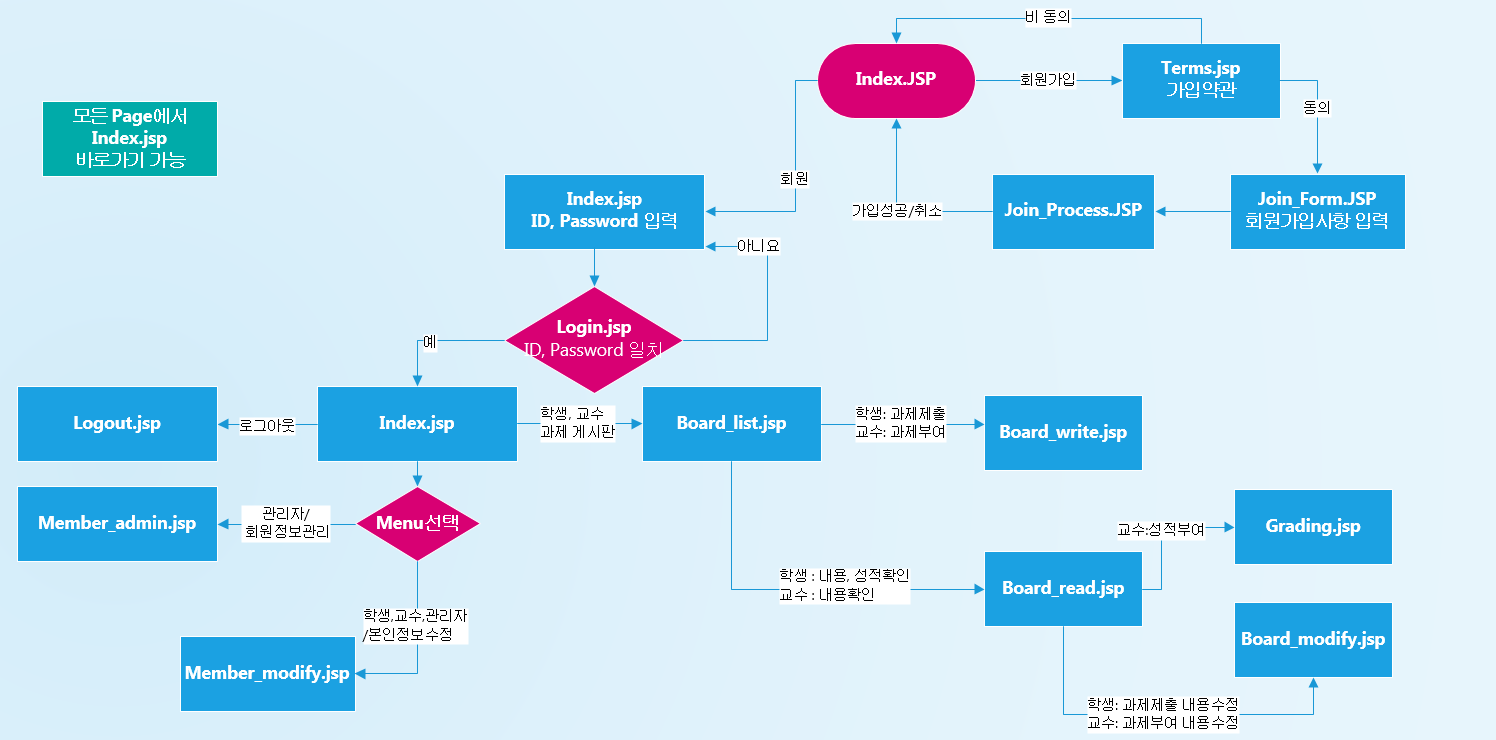


Figure . Sitemap

|  |  |
| --- | --- |
| **Main Page** | <http://210.107.197.133:8080/secure/> |
| **회원가입 약관동의** | <http://210.107.197.133:8080/secure/register_notice.jsp> |
| **회원가입 Form** | <http://210.107.197.133:8080/secure/register_form.jsp> |
| **회원정보 수정** | <http://210.107.197.133:8080/secure/member_modify.jsp> |
| **사용자 권한변경** | <http://210.107.197.133:8080/secure/user_info.jsp> |
| **과제목록** | <http://210.107.197.133:8080/secure/board_list.jsp> |
| **과제 상세보기** | <http://210.107.197.133:8080/secure/board_read.jsp> |
| **게시글 작성, 과제등록** | <http://210.107.197.133:8080/secure/write_index.jsp> |
| **과제 내용 수정** | <http://210.107.197.133:8080/secure/board_modify.jsp> |
| **점수부여** | <http://210.107.197.133:8080/secure/grading.jsp>  <http://210.107.197.133:8080/secure/sub_grading.jsp> |
| **회원정보관리(관리자)** | <http://210.107.197.133:8080/secure/member_admin.jsp> |

Table . Site Map - Page 별 address

## Database Structure

* Table 1 : member

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **value** | **column** | **value** |
| No | 회원가입번호 | Prof | 교수인지 확인 |
| Id | ID | Email | E-mail |
| Pw | 비밀번호 | Phone | 전화번호 |
| Name | 이름 | Register\_date | 회원가입일 |
| Student\_no | 학번 | Password\_changed\_date | 비밀번호 변경일 |
| Fail\_flag | 로그인 실패횟수 |  |  |

Table . Database\_member

* Table 2 : assignment

|  |  |  |  |
| --- | --- | --- | --- |
| **column** | **value** | **column** | **value** |
| submit\_no | 제출번호 | student\_no | 학번 |
| assignment\_no | 과제번호 | Title | 제목 |
| content | 내용 | grade | 학점 |
| prof | 교수인지 확인 | Submit\_yesno | 제출여부 |
| Assignment\_register\_date | 제출일 |  |  |

Table . Database\_assignment

* Table 3 : data\_list

|  |  |  |  |
| --- | --- | --- | --- |
| **column** | **value** | **column** | **value** |
| Data\_no | 제출번호 | Board\_no | 과제번호 |
| File\_name | 과제첨부파일 | User\_id | 제출자 ID |
| New\_file\_name | 첨부파일명 변환 | User\_no | 제출자 학번 |
| publisher | 제출자 이름 |  |  |

Table . Databse\_data\_list

# 보안 요구사항 적용

## 기본 보안설정

|  |  |
| --- | --- |
| **내 용** | **적용사항** |
| 회원가입 필드,  유효한 value만 입력 | 회원가입 시, 사용자 입력 value를 유요한 value만 입력하도록 적용. [포맷스트링 삽입- S/W47] |
| Password 입력 시, \* 로 표시 | 입력 한 password대로 보이지 않도록 처리 |
| Password 30일 마다 변경 알림 | 사용자 password 설정 후, 30일 경과 시, 아래 message box 출력 (password\_changed\_date) |
| Password 전송 시, 평문으로 전송  -> 암호화 전송 | 암호화된 정보를 데이터베이스에 저장, 데이터베이스에서 암호화된 정보를 가져와 복호화하여 사용 |
| Password 설정 시,  영문+숫자+특수문자 5-20자 조건 | 5~20자 영문대소문자, 숫자, 특수문자 혼합하여 사용, 그 외 회원가입 필드 조건에 대해 조건에 맞지 않을 경우, 아래 Message box 출력 [취약한 비밀번호 허용 – S/W47] |
| Log-In Password 5회 잘못 입력 시, 해당 정보로 접근 막고, 관리자 연락처와 조치사항 알림 | 로그인 실패 카운터 (fail\_flag)를 db에 갱신하여 5회 이상 실패 시 로그인 불가, 관리자 연락 메시지 출력    [반복된 인증시도 제한 기능부재 – S/W47] |
| ID 중복체크 | 회원가입 시, ID 중복체크, 중복된 ID 입력 시, 아래 Message 출력 |
| DB 접근 : 아이디/비밀번호 하드코딩 -> 별도 로그인 기능 추가 | DB 접근: 아이디/비밀번호 하드코딩 -> 별도 로그인 기능 추가 |

Table . 기본보안 적용내용

## OWASP 10, S/W 개발보안 47

|  |  |  |
| --- | --- | --- |
| **OWASP** | **내 용** | **적용사항** |
| A1. SQL Injection | ID, Password 입력 창에서 query 문 입력 통제 (입력한 데이터 Filtering) | 정규식을 이용하여 가용된 문자 이외 사용하지 못하도록 구현 |
| 페이지 오류 시 Error Message 차단 [오류 메시지를 통합 정보노출-SW47]  [오류상황대응부재 – S/W47]  [부적절한 예외처리 – S/W47] | error.html로 이동하도록 조치. web.xml에 코드 삽입 |
| A2. 인증 및 세션관리 취약점  [잘못된 세션에 의한 정보노출 – S/W47] | 로그인 과정 암호화(SSL: Secure Socket Layer) | 로그인이나 회원가입시 ID는 256bit의 키(AES256) Password는 160bit(SHA256)의 Hash로 암호화하여 사용 (암호화, 복호화) |
| 로그인 후, 10분간 활동 없을 시에 강제 로그아웃 | 페이지 이동 시 로그인한 시간부터 체크해서 10분 경과 시 Message box 출력 후,  로그아웃(세션삭제) |
| IE 창 닫으면 자동 로그아웃 | 브라우저 자체적으로 지원됨. 다만, 크롬은 세션삭제가 안됨. 브라우저 문제로 알려짐. |
| A3. XSS  [크로스사이트 스크립팅 S/W47]  [신뢰되지 않은 URL 주소로 자연접속 연결 –SW47]  A8. CSRF | 입력한 데이터 Filtering (스크립트 실행 금지) | 게시판 내용-CKEDITOR의 소스 버튼 제거 |
| 입력 값 검증 및 치환(게시판) | 제목-<. >, ', " 등 xss에 사용될 수 있는 특수문자 &lt 등으로 치환 => 태그실행이 되지 않고, 스크립트 자체가 텍스트로 출력. |
| 게시판 file upload 시 upload file 형식규제  [위험한 형식 파일 업로드 – S/W47] | 파일 확장자가 .zip이 아닌 경우에는 경고메시지 출력하고 전송하지 않음(글 작성/수정 시) |
| A4. 취약한 직접 객체 참조 | 오류 Page, Message -> Redirection | "페이지 오류 시 Error Message 차단"에서 완료 |
| 웹 페이지 강제 입력 시 세션 체크하여 권한 확인, 주소 입력 접속차단 [경로조작 및 자원삽입 –S/W47] | userid 세션 없고 , assignment\_no 없으면 index.jsp로 강제 리다이렉트 or error message 출력 |
| A6. 민감 데이터노출 | 중요 Data 암호화  [중요정보 평문저장, 중요정보 평문전송- S/W47] | AES256 암호화(양방향 암호화)를 사용하여 회원의 중요 Data를 암호화 |
| Web Page에서 DB Connection 시 ID, PW 암호화 | AES256 Cipher를 dbUser, dbPass로 사용 후 connection시 디코딩하여 사용 |
| A7. 기능 수준의  접근통제 누락 | 사용자 권한에 따른 페이지 접속 여부  [적적한 인증 없는 중요기능 허용 –S/W47]  [부적절한 인가 – S/W47]  [중요한 자원에 대한 잘못된 권한 설정-S/W47] | 관리자, 교수, 학생 권한으로 나뉘어 해당하는 기능만 사용할 수 있도록 함 |

Table . OWASP 10 적용내용

래는 과제관리 시스템에서 구현한 보안 요구사항 중, 중요한 항목에 대해 구현 전과 후를 나타낸다.

### SQL Injection

SQL Injection 구현 전, 회원로그인 창에서 ID에 ' or 1=1 # 입력하면, 아래 그림처럼, 로그인이 되었다.



Figure . SQL Injection 구현 전

보안상 취약점이 발견되어 SQL Injection 구현 후, 로그인이 되지 않고 error message를 출력함으로써 보안취약 사항을 개선하였다.

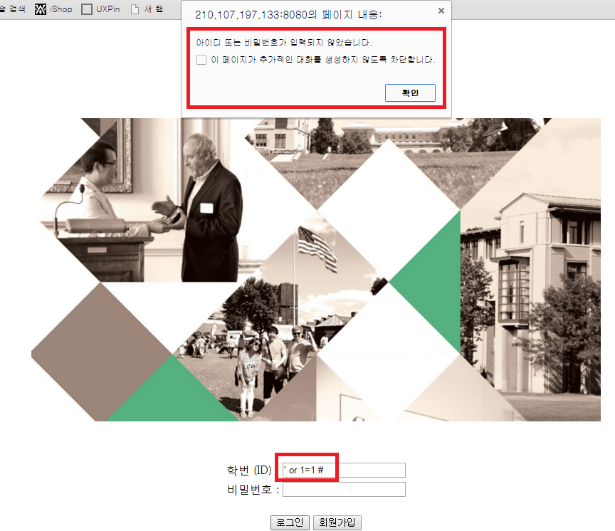


Figure . SQL Injection 구현 후

### Cross-Site Scripting, Cross-Site Request Forgery

1. Cross-Site Scripting, Cross-Site Request Forgery 취약점을 개선하기 위해, 스크립트를 입력할 수 있는 게시판에서 태그실행이 되지 않고, 스크립트 자체가 텍스트로 출력되도록 처리하였다.  
   게시판에서 제목, 본문에 각각 Script를 삽입하면,

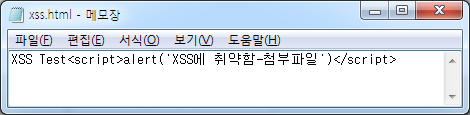


Figure . XSS, CSRF Test Script

Script가 실행되어 취약함을 확인할 수 있다.

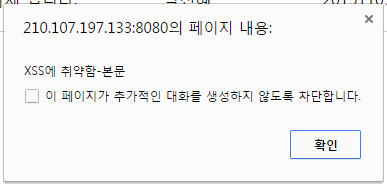
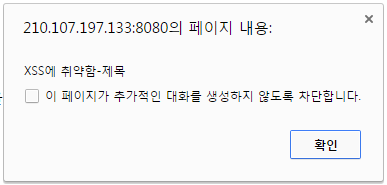


Figure . XSS, CSRF – 구현 전



Figure . XSS, CSRF – 구현 후

1. 게시판 file upload 시 upload 파일의 형식을 규제하여 위험한 파일의 실행을 차단하였다. 파일 확장자가 .zip이 아닌 경우에는 경고 메시지를 출력하고 파일을 전송하지 못하도록 하였다.

본 취약점 수정 전, 게시판에 figure5의 스크립트가 추가된 xss.html 파일을 업로드하고 파일을 다운로드 받을 경우, 해당 파일이 실행되어 보안에 취약점이 드러났다.

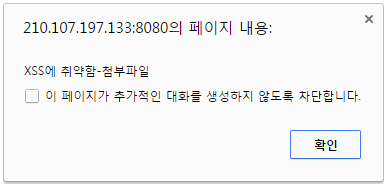


Figure . XSS, CSRF - 파일형식 규제-구현 전

아래와 같이 취약점을 개선하였다.

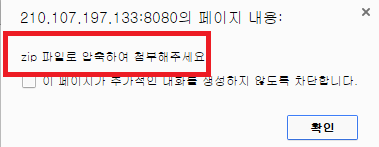


Figure . XSS, CSRF - 파일형식 규제-구현 후

### Away from Home

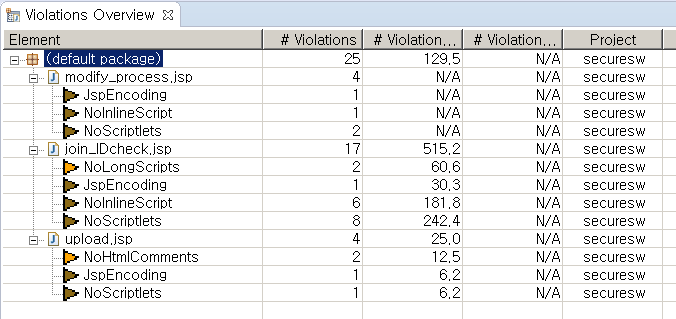
# Secure Coding

## Securecoding

# Tool

## PMD

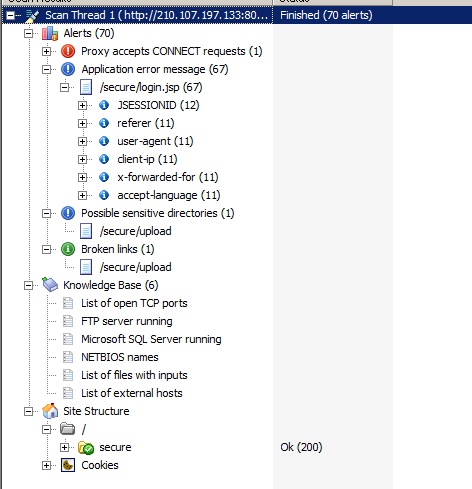
1. **2015년 11월 20일 PMD 체크결과**



1. **2015년 12월 1일 PMD 체크결과**
2. **항목별 수정사항**

## Acunetix

1. 2015년 11월 24일 결과



# Project Summary

## Project WBS

