**보안결함 방지를 위한 웹코딩 상세 구현 가이드**

내용

[개요 2](#_Toc458028142)

[URL 파라미터, form 파라미터의 처리 3](#_Toc458028143)

[SQL Injection 3](#_Toc458028144)

[myBatis 3](#_Toc458028145)

[XSS(Cross Site Scripting) 3](#_Toc458028146)

[jQuery의 .html(), .text() 3](#_Toc458028147)

[JSTL 4](#_Toc458028148)

[참조링크 4](#_Toc458028149)

[CSRF 4](#_Toc458028150)

[Tomcat에서의 구현 4](#_Toc458028151)

[참조링크 4](#_Toc458028152)

[Cross-Domain 4](#_Toc458028153)

[Same Orgin 4](#_Toc458028154)

[postMessage() 4](#_Toc458028155)

[File Download 4](#_Toc458028156)

[암호화 4](#_Toc458028157)

[Spring Framework 4](#_Toc458028158)

[별첨 5](#_Toc458028159)

[행정기관 및 공공기관 정보시스템 구축·운영 지침 5](#_Toc458028160)

[시큐어 코딩 가이드 (행정자치부-2012) 5](#_Toc458028161)

[개발보안 점검 가이드 (행정자치부-2016) 5](#_Toc458028162)

## 개요

### 문서의 목적

기존 공공기관에서 배포되는 시큐어코딩 가이드에는 최근에 사용된 자바프레임웍, 자바스크립트 라이브러리를 고려한 내용은 포함되지 않는다.

이에 현실적이고 효율적인 웹개발을 위해 범용적으로 사용되는 웹 구성 환경에 적용할 수 있는 코딩가이드를 제공하기 위해 본 문서를 작성한다.

또한, 보안을 위해 사용이 허용되지 않는 기능을 우회적으로 구현하기 위한 기법도 가이드한다.

### 문서의 포함 범위

**개별보안결함 방지 코딩 내용 추가**

## URL 파라미터, form 파라미터의 처리

## SQL Injection

### myBatis

매퍼에서의 파라미터 처리는 #{key} 엘리먼트로 처리하지만, SQL 구문에 대한 처리를 위해 ${sql} 엘리먼트를 사용할 수도 있으며 이때 SQL Injection 결함이 발생할 수도 있다.  
이 엘리먼트를 사용할 경우 사용자가 입력한 값을 이용하지 않고, 검증된 white list를 이용한 방식을 사용하도록 한다.

## XSS(Cross Site Scripting)

### jQuery의 .html(), .text()

|  |
| --- |
| jQuery('div').text('<test>a&f"#</test>');​  jQuery('div').html('&lt;test&gt;a&f"#&lt;/test&gt;');​ |

위의 text, .html 함수는 동일한 결과를 출력, text 함수는 파라미터를 unescape하여 처리하므로 XSS 방지 효과를 기대할 수 있다.

예를 들어 XSS 안전성이 검증되지 않고 DB에 저장된 사용자 입력 게시판의 내용을 화면에 출력할 경우 사용할 수 있다.

|  |
| --- |
| $(function() {  var untrusted = "<script>alert('xss');</script>";  $("#jaquery").html(  $("<div/>").text(untrusted).html()  );  }); |

html과 text 함수를 위의 예와 같이 혼합하여 사용할 수 있다

### JSTL

<c:out> 커스텀 태그의 사용법은 <c:out value="value" [escapeXml="{true|false}"] /> 이다.

경우에 따라 간략하게 ${변수명} 형식으로 사용할 수 있으나, 이 경우 escapeXml이 false로 설정된 결과가 출력된다.

c:out 태그를 사용하면 escapeXml의 기본설정값은 true이므로, <c:out value="value"/> 형식으로만 사용하더라도 XSS 방지 효과가 있다.

### 참조링크

https://www.owasp.org/index.php/XSS\_(Cross\_Site\_Scripting)\_Prevention\_Cheat\_Sheet

## CSRF

### Tomcat에서의 구현

Tomcat에서는 기본적으로 CSRF 방지를 위한 필터를 제공한다.

구현을 위하여 다음의 절차를 따른다.

### 참조링크

https://tomcat.apache.org/tomcat-7.0-doc/config/filter.html

## Cross-Domain

## Same Orgin

### postMessage()

## File Download

## 암호화

### Spring Framework

## 별첨

### 행정기관 및 공공기관 정보시스템 구축·운영 지침

* 행정기관 및 공공기관 정보시스템 구축·운영 지침.hwp
* 행정기관 및 공공기관 정보시스템 구축·운영 지침-별첨- [별표 3] 소프트웨어 보안약점 기준(제52조 관련).hwp
* 행정기관 및 공공기관 정보시스템 구축·운영 지침-[별표 4] 소프트웨어 보안약점 진단원의 자격기준(제54조 관련).hwp

### 시큐어 코딩 가이드 (행정자치부-2012)

* JAVA\_시큐어코딩\_가이드-201209.pdf

### 개발보안 점검 가이드 (행정자치부-2016)

* 공개SW를\_활용한\_소프트웨어\_개발보안\_점검가이드.pdf