







InputValidation_XSS_TagsWhichRequireOutputEncoding_Finding - Findings Report

Oct 26, 2007 7:07:24 PM

Classification	Type	Severity	File	Line	CWE ID	SmartTrace
Vulnerability	Validation.EncodingRequired		Z:\jspwiki\JSPWiki_2_4_104\JSPWiki-src\src\com\ecyrd\jspwiki\tags\CalendarTag.java	288	116	
<div> <div>Description:</div> <div>The following tags are observed to render contents directly to the pageContext without Output Encoding. It may be possible for XSS to occur in each of these tags.</div> <div>Recommendation:</div> <div>Output Encode the value rendered to the user. Use the "TextUtil.replaceEntities()" method.</div> </div> <div> <pre> out.write("<table class=\"calendar\">\n"); HttpServletRequest httpRequest = m_wikiContext.getHttpRequest(); String queryString = engine.safeGetQueryString(httpRequest); out.write("<tr>"+ getMonthNaviLink(prevCal,"&lt;&lt;", queryString)+ "<td colspan=5 class=\"month\">"+ getMonthLink(cal)+ "</td>"+ getMonthNaviLink(nextCal,"&gt;&gt;", queryString)+ </pre> </div>						
Vulnerability	CrossSiteScripting		Z:\jspwiki\JSPWiki_2_4_104\JSPWiki-src\src\com\ecyrd\jspwiki\tags\CookieTag.java	181	79	
<div> <div>Description:</div> <div>The following tags are observed to render contents directly to the pageContext without Output Encoding. It may be possible for XSS to occur in each of these tags.</div> <div>Recommendation:</div> <div>Output Encode the value rendered to the user. Use the "TextUtil.replaceEntities()" method.</div> </div> <div> <pre> if(m_var != null) { int scope = getScope(m_scope); pageContext.setAttribute(m_var, out, scope); } else { try { pageContext.getOut().print(out); } catch(IOException ioe) { log.warn("Failed to write to JSP page: " + ioe.getMessage(), ioe); } } </pre> </div>						
Vulnerability	CrossSiteScripting		Z:\jspwiki\JSPWiki_2_4_104\JSPWiki-src\src\com\ecyrd\jspwiki\tags\EditorItratorTag.java	112	79	
<div>Description:</div>						

The following tags are observed to render contents directly to the pageContext without Output Encoding. It may be possible for XSS to occur in each of these tags.

Recommendation:

Output Encode the value rendered to the user. Use the "StringUtil.replaceEntities()" method.

```
if( bodyContent != null )
{
    try
    {
        JspWriter out = getPreviousOut();
        out.print(bodyContent.getString());
        bodyContent.clearBody();
    }
    catch( IOException e )
    {
        log.error("Unable to get inner tag text", e);
    }
}
```

Vulnerability CrossSiteScripting



Z:\jspwiki\JSPWiki_2_4_104\JSPWiki-src\src\com\ecyrd\jspwiki\tags\AttachmentsIteratorTag.java

127

79



Description:

The following tags are observed to render contents directly to the pageContext without Output Encoding. It may be possible for XSS to occur in each of these tags.

Recommendation:

Output Encode the value rendered to the user. Use the "StringUtil.replaceEntities()" method.

```
if( bodyContent != null )
{
    try
    {
        JspWriter out = getPreviousOut();
        out.print(bodyContent.getString());
        bodyContent.clearBody();
    }
    catch( IOException e )
    {
        log.error("Unable to get inner tag text", e);
    }
}
```

Vulnerability CrossSiteScripting



Z:\jspwiki\JSPWiki_2_4_104\JSPWiki-src\src\com\ecyrd\jspwiki\tags\SearchResultIteratorTag.java

133

79



Description:

The following tags are observed to render contents directly to the pageContext without Output Encoding. It may be possible for XSS to occur in each of these tags.

Recommendation:

Output Encode the value rendered to the user. Use the "StringUtil.replaceEntities()" method.

```
if( bodyContent != null )
{
    try
    {
        JspWriter out = getPreviousOut();
        out.print(bodyContent.getString());
        bodyContent.clearBody();
    }
    catch( IOException e )
    {
        log.error("Unable to get inner tag text", e);
    }
}
```

**Description:**

The following tags are observed to render contents directly to the pageContext without Output Encoding. It may be possible for XSS to occur in each of these tags.

Recommendation:

Output Encode the value rendered to the user. Use the "TextUtil.replaceEntities()" method.

```
if( bodyContent != null )
{
    try
    {
        JspWriter out = getPreviousOut();
        out.print(bodyContent.getString());
        bodyContent.clearBody();
    }
    catch( IOException e )
    {
        log.error("Unable to get inner tag text", e);
    }
}
```

**Description:**

The following tags are observed to render contents directly to the pageContext without Output Encoding. It may be possible for XSS to occur in each of these tags.

Recommendation:

Output Encode the value rendered to the user. Use the "TextUtil.replaceEntities()" method.

```
if( bodyContent != null )
{
    try
    {
        JspWriter out = getPreviousOut();
        out.print(bodyContent.getString());
        bodyContent.clearBody();
    }
    catch( IOException e )
    {
        log.error("Unable to get inner tag text", e);
    }
}
```

**Description:**

The following tags are observed to render contents directly to the pageContext without Output Encoding. It may be possible for XSS to occur in each of these tags.

Recommendation:

Output Encode the value rendered to the user. Use the "TextUtil.replaceEntities()" method.

```
// Add any explicit body content. This is not the intended use
// of LinkTag, but happens to be the way it has worked previously.
if( m_bodyContent != null )
{
    String linktext = m_bodyContent.getString().trim();
    out.write( linktext );
}
```

```
    // Finish off by closing opened anchor  
    if( m_format == ANCHOR ) out.print("</a>");  
}
```