



InputValidation_XSS_Group_Finding - Findings Report

Oct 26, 2007 7:07:24 PM

Classification	Type	Severity	File	Line	CWE ID	SmartTrace
Vulnerability	CrossSiteScripting		Z:\jspwiki\JSPWiki_2_4_104\JSPWiki-src\web-root\JSPWiki.war\templates\default\GroupContent.jsp	109	79	
<p>Description: The Group Content JSP(s) contain a variety of different XSS attacks (some potentially stored, while others reflected). Please see below for the specific XSS detected.</p> <ol style="list-style-type: none"> GroupContent.jsp - The "group" parameter is used in various conditions without validation/output encoding. Also, it should be noted that in a certain case (delete confirm) the script tags do not need to be injected by the attacker since the payload would be rendered within existing script tags. * Attack URL: <a href="http://localhost:8080/JSPWiki/Group.jsp?group=<script>alert(document.cookie);</script>&members=<script>alert(document.cookie);</script>">http://localhost:8080/JSPWiki/Group.jsp?group=<script>alert(document.cookie);</script>&members=<script>alert(document.cookie);</script> GroupContent.jsp - Stored XSS if the "modifier", "creator", "modified", "created" contains malicious payload. The likelihood may be reduced since only asserted usernames can contain malicious payload. However, best practices should output encode these values for additional protection. PageActions.jsp - will set and use the group without validation/output encoding. It does so when the current user has edit group or delete permissions. <p>Recommendation: Output Encode the value rendered to the user. Use the "TextUtil.replaceEntities()" method. In cases where the data is already rendered within existing script tags, consider very strong input validation and even removing this exclusion within existing script tags.</p> <pre> </div> </div> <div class="instructions"> <%=modifier%> saved this group on <%=modified%>
 <%=creator%> created it on <%=created%>. </div> </div> <% } %> </pre>						
Vulnerability	CrossSiteScripting		Z:\jspwiki\JSPWiki_2_4_104\JSPWiki-src\web-root\JSPWiki.war\templates\default\GroupContent.jsp	108	79	
<p>Description: The Group Content JSP(s) contain a variety of different XSS attacks (some potentially stored, while others reflected). Please see below for the specific XSS detected.</p> <ol style="list-style-type: none"> GroupContent.jsp - The "group" parameter is used in various conditions without validation/output encoding. Also, it should be noted that in a certain case (delete confirm) the script tags do not need to be injected by the attacker since the payload would be rendered within existing script tags. * Attack URL: <a href="http://localhost:8080/JSPWiki/Group.jsp?group=<script>alert(document.cookie);</script>&members=<script>alert(document.cookie);</script>">http://localhost:8080/JSPWiki/Group.jsp?group=<script>alert(document.cookie);</script>&members=<script>alert(document.cookie);</script> GroupContent.jsp - Stored XSS if the "modifier", "creator", "modified", "created" contains malicious payload. The likelihood may be reduced since only asserted usernames can contain malicious payload. However, best practices should output encode these values for additional protection. PageActions.jsp - will set and use the group without validation/output encoding. It does so when the current user has edit group or delete permissions. 						

Recommendation: Output Encode the value rendered to the user. Use the "StringUtil.replaceEntities()" method. In cases where the data is already rendered within existing script tags, consider very strong input validation and even removing this exclusion within existing script tags.

```
        The group&#8217;s membership.
    </div>
</div>

<div class="instructions">
    <%=modifier%> saved this group on <%=modified%><br/>
    <%=creator%> created it on <%=created%>.
</div>
</div>
<%=
}
```

Vulnerability CrossSiteScripting



Z:\jspwiki\JSPWiki_2_4_104\JSPWiki-src\web-root\JSPWiki.war\templates\default\GroupContent.jsp

108

79

Description: The Group Content JSP(s) contain a variety of different XSS attacks (some potentially stored, while others reflected). Please see below for the specific XSS detected.

1. GroupContent.jsp - The "group" parameter is used in various conditions without validation/output encoding. Also, it should be noted that in a certain case (delete confirm) the script tags do not need to be injected by the attacker since the payload would be rendered within existing script tags.

* Attack URL: [http://localhost:8080/JSPWiki/Group.jsp?group=<script>alert\(document.cookie\);</script>&members=<script>alert\(document.cookie\);</script>](http://localhost:8080/JSPWiki/Group.jsp?group=<script>alert(document.cookie);</script>&members=<script>alert(document.cookie);</script>)

2. GroupContent.jsp - Stored XSS if the "modifier", "creator", "modified", "created" contains malicious payload. The likelihood may be reduced since only asserted usernames can contain malicious payload. However, best practices should output encode these values for additional protection.

3. PageActions.jsp - will set and use the group without validation/output encoding. It does so when the current user has edit group or delete permissions.

Recommendation: Output Encode the value rendered to the user. Use the "StringUtil.replaceEntities()" method. In cases where the data is already rendered within existing script tags, consider very strong input validation and even removing this exclusion within existing script tags.

```
        The group&#8217;s membership.
    </div>
</div>

<div class="instructions">
    <%=modifier%> saved this group on <%=modified%><br/>
    <%=creator%> created it on <%=created%>.
</div>
</div>
<%=
}
```

Vulnerability CrossSiteScripting



Z:\jspwiki\JSPWiki_2_4_104\JSPWiki-src\web-root\JSPWiki.war\templates\default\GroupContent.jsp

46

79



Description: The Group Content JSP(s) contain a variety of different XSS attacks (some potentially stored, while others reflected). Please see below for the specific XSS detected.

1. GroupContent.jsp - The "group" parameter is used in various conditions without validation/output encoding. Also, it should be noted that in a certain case (delete confirm) the script tags do not need to be injected by the attacker since the payload would be rendered within existing script tags.

* Attack URL: [http://localhost:8080/JSPWiki/Group.jsp?group=<script>alert\(document.cookie\);</script>&members=<script>alert\(document.cookie\);</script>](http://localhost:8080/JSPWiki/Group.jsp?group=<script>alert(document.cookie);</script>&members=<script>alert(document.cookie);</script>)

2. GroupContent.jsp - Stored XSS if the "modifier", "creator", "modified", "created" contains malicious payload. The likelihood may be reduced since only asserted usernames can contain malicious payload. However, best practices should output encode these values for additional protection.

3. PageActions.jsp - will set and use the group without validation/output encoding. It does so when the current user has edit group or delete permissions.

Recommendation: Output Encode the value rendered to the user. Use the "TextUtil.replaceEntities()" method. In cases where the data is already rendered within existing script tags, consider very strong input validation and even removing this exclusion within existing script tags.

```
<script language="javascript" type="text/javascript">
function confirmDelete()
{
    var reallydelete = confirm("Are you sure you want to permanently delete group '<%=name%>'? Users might not be able to access pages whose
    ACLS contain this group. \n\nIf you click OK, the group will be removed immediately.");
    return reallydelete;
}
</script>
```

Vulnerability CrossSiteScripting



Z:\jspwiki\JSPWiki_2_4_104\JSPWiki-src\web-root\JSPWiki.war\templates\default\GroupContent.jsp

109

79

Description: The Group Content JSP(s) contain a variety of different XSS attacks (some potentially stored, while others reflected). Please see below for the specific XSS detected.

1. GroupContent.jsp - The "group" parameter is used in various conditions without validation/output encoding. Also, it should be noted that in a certain case (delete confirm) the script tags do not need to be injected by the attacker since the payload would be rendered within existing script tags.

* Attack URL: [http://localhost:8080/JSPWiki/Group.jsp?group=<script>alert\(document.cookie\);</script>&members=<script>alert\(document.cookie\);</script>](http://localhost:8080/JSPWiki/Group.jsp?group=<script>alert(document.cookie);</script>&members=<script>alert(document.cookie);</script>)

2. GroupContent.jsp - Stored XSS if the "modifier", "creator", "modified", "created" contains malicious payload. The likelihood may be reduced since only asserted usernames can contain malicious payload. However, best practices should output encode these values for additional protection.

3. PageActions.jsp - will set and use the group without validation/output encoding. It does so when the current user has edit group or delete permissions.

Recommendation: Output Encode the value rendered to the user. Use the "TextUtil.replaceEntities()" method. In cases where the data is already rendered within existing script tags, consider very strong input validation and even removing this exclusion within existing script tags.

```
</div>
</div>

<div class="instructions">
    <%=modifier%> saved this group on <%=modified%><br/>
    <%=creator%> created it on <%=created%>.
</div>
</div>
<%
}
%>
```