**SCENARIO**

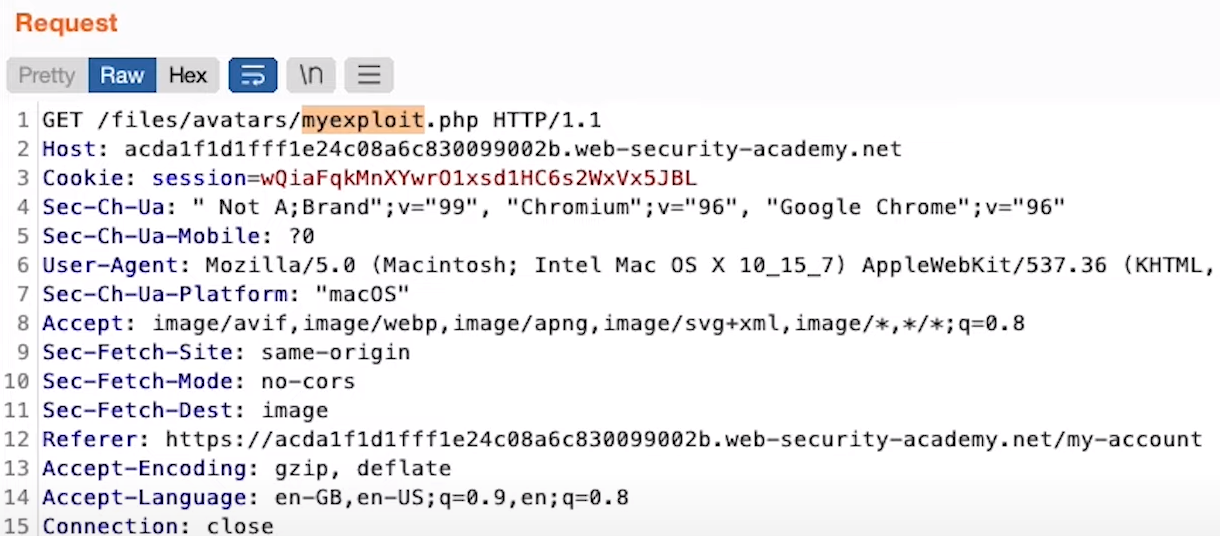
The lab displays a fragile image uploading function which fails to validate the files users upload before saving them onto the server's filesystem. This oversight allows potential attackers to upload and execute arbitrary code on the server.

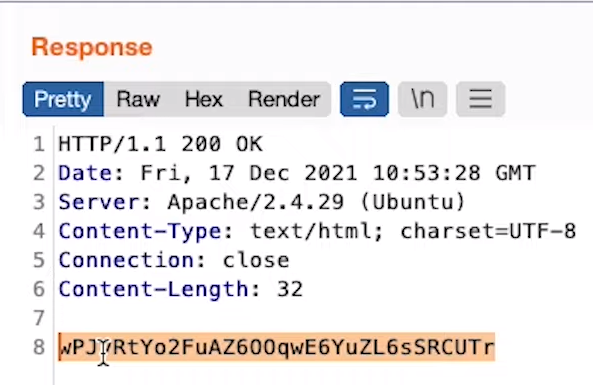
**PROCEDURE**

1. Using Burp Suite, we logged into the provided account and noticed an option to upload an avatar image.
2. After uploading an arbitrary image and revisiting the account page, an avatar preview appeared.
3. Filtering for images in Burp's HTTP history revealed the path /files/avatars/<OUR-IMAGE>.
4. We created an exploit.php file on our system, which contained PHP code to fetch Carlos's secret file.
5. Using the avatar upload function, we uploaded the malicious PHP file.
6. In Burp Repeater, the path of the GET request was modified to target our PHP file, which, when executed, disclosed Carlos's secret.

**PAYLOAD**

<?php echo file\_get\_contents('/home/carlos/secret'); ?>

**PROOF OF CONCEPT**

**  
REMEDIATION**

1. **File Type Verification:** Implement a strict whitelist of allowed file types and validate using server-side logic.
2. **File Content Analysis:** Analyze uploaded file content to ensure they match expected file types.
3. **Limit Execution:** Store uploaded files in a non-executable directory or outside the web root.