**SCENARIO**

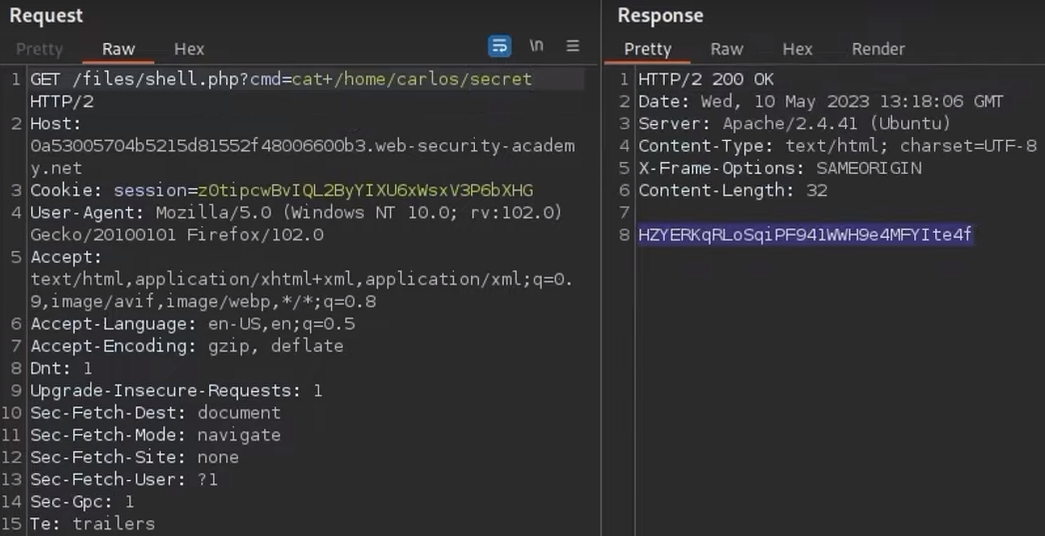
The lab features an image upload system that refrains from executing user-provided files. However, a secondary vulnerability allows us to bypass this protection.

**PROCEDURE**

1. We uploaded an avatar image and reviewed its retrieval path in Burp.
2. Preparing our malicious exploit.php script, we tried to upload it. The server stored it but refused to execute the PHP.
3. In Burp, we manipulated the upload request to include a directory traversal sequence in the filename, thus changing its storage location.
4. Revisiting our account page and inspecting the executed file path, we retrieved Carlos's secret.

**PAYLOAD**

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

**  
PROOF OF CONCEPT**

**  
REMEDIATION**

1. **Limit File Types:** Ensure only specific file types are allowed for upload.
2. **Directory Traversal Prevention:** Implement filters to catch and block directory traversal attempts.
3. **Set Correct File Permissions:** Uploaded files should not have executed permissions by default.