SCENARIO

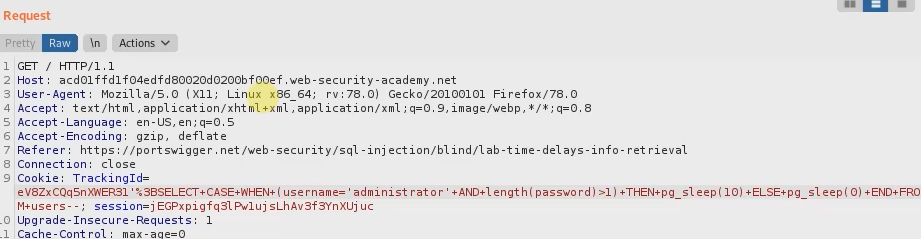
The online shopping platform appears to harbor a blind SQL injection vulnerability within its analytics. Similar to Scenario 14, the results of the SQL query are concealed, yet we suspect that we can induce conditional time delays to extract key data.

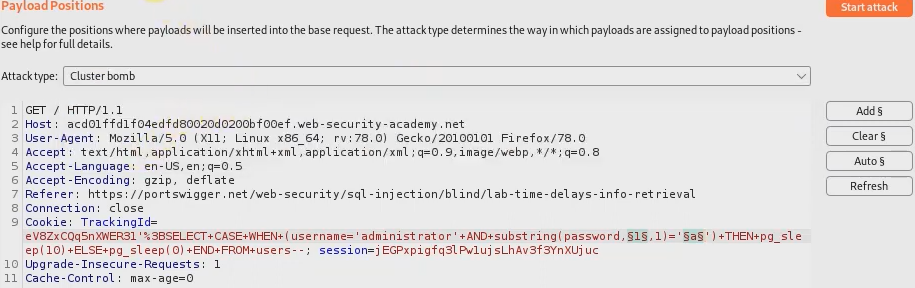
**PROCEDURE**

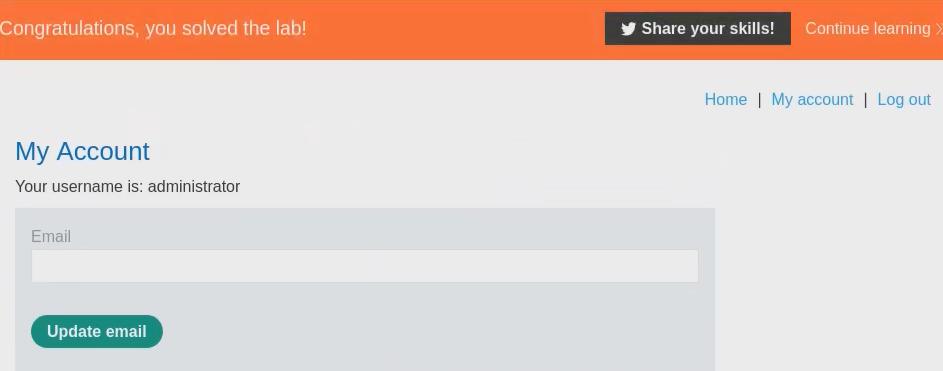
1. Start by accessing the main page and intercept the TrackingId cookie request using Burp Suite.
2. Modify the TrackingId value to see if a delay can be initiated: TrackingId=x'%3BSELECT+CASE+WHEN+(1=1)+THEN+pg\_sleep(10)+ELSE+pg\_sleep(0)+END--.
3. If the application takes 10 seconds to respond, the injected delay was successful.
4. Continue refining the SQL payload to extract the username and password for the administrator user.
5. Use the extracted password to log in as the administrator.

**PAYLOAD**

TrackingId=x'%3BSELECT+CASE+WHEN+(1=1)+THEN+pg\_sleep(10)+ELSE+pg\_sleep(0)+END--

**PROOF OF CONCEPT**

****

****

**REMEDIATION**

1. Employ parameterized SQL queries to prevent direct execution from user inputs.
2. Ensure robust input validation mechanisms.
3. Limit the database permissions for user accounts.
4. Implement a web application firewall (WAF) to monitor and restrict suspicious patterns.
5. Regularly conduct security reviews and patch systems to stay updated against known vulnerabilities.