SCENARIO

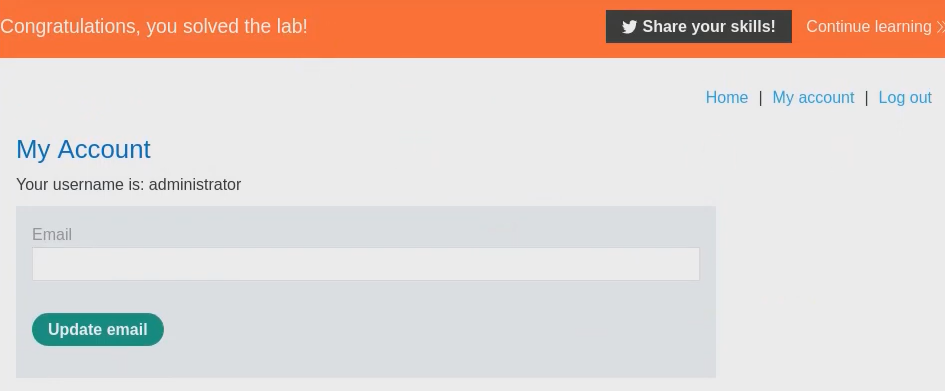
Our digital storefront seems to have a blind SQL injection vulnerability. It uses a tracking cookie for analytics, forming a SQL query from this cookie value. This flaw might be exploited to extract confidential user data. The database used is Oracle, which has unique SQL characteristics.

**PROCEDURE**

1. Access the main page and use Burp Suite to catch the request containing the TrackingId cookie.
2. Attempt an injection by appending a single quotation to the TrackingId: TrackingId=xyz'
3. If an error message appears, test with two quotation marks: TrackingId=xyz''.
4. Formulate a valid SQL subquery for Oracle: TrackingId=xyz'||(SELECT '' FROM dual)||'
5. Once a valid subquery structure is ascertained, exploit it to extract user data.
6. Adapt queries to identify the administrator user and extract their password using Oracle's string functions.
7. Finally, use the acquired password to gain administrator access.

**PAYLOAD**

TrackingId=xyz'||(SELECT '' FROM dual)||'

**PROOF OF CONCEPT**

**REMEDIATION**

1. Use parameterized SQL queries or stored procedures.
2. Adopt strong input validation procedures.
3. Regularly audit and update the database system.
4. Implement web application firewalls (WAFs) to detect and thwart attack patterns.
5. Educate developers about the risks of SQL injection and best practices to prevent them.