TASK 5: SQL INJECTION - #3 (flag8 - 7.5 pts / flag9 - 7.5 pts)

***NOTE - Task 5 has 2 flags (flag8, flag9)**

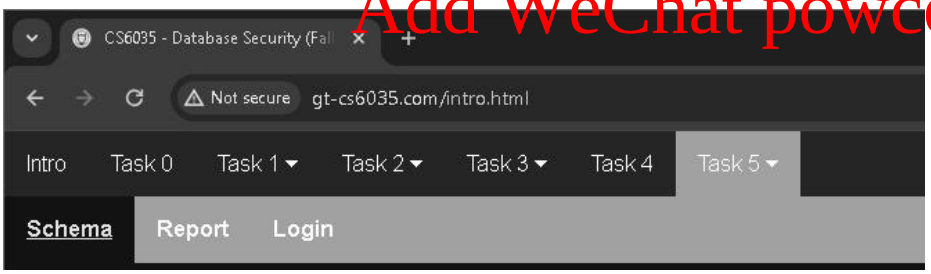
In this case, you will be looking at a search engine for a database of music albums for a music store. You have discovered a page called "schema," which offers the user a view of the underlying metadata of the table used to populate the main report. By now, if you have been paying attention, the designers of these sites have followed a similar pattern for how their sites work, especially if you look at the hyperlinks that lead to the pages. You know, therefore, that there is probably a way to get admin access to that schema by leveraging that knowledge of how they use links. That means table definitions might be available to users who poke around and try to find things they should not.

You have learned that the Schema page uses database user Schema to access the database and that this user only has select/read access to the Music table. You will need to inject to provide the Schema user access to all tables, so when you find out how to display the database table schemas, more than the Music table will be displayed. You do not know the table name to which you are trying to add access or the database schema to which the table belongs. This is important in discovering how to craft your attack's first part.

So, your task is first to figure out how to modify access for the Schema user, then see if you can find any additional information about tables in the database. Finally, you must learn how to leverage your discovered information to construct the SQL injection. If you successfully craft a SQL injection, you will find an actual database account appearing on the report that you can use to log into the system using the "Login" screen on task 3. If you have the correct login information you can only obtain it by SQL injection. You can enter the login information into that screen and access the management console. Of course, for our purposes, the "management console" is just the hash you need to get credit for the attack. You will need to enter the newly visible hash in your JSON file.

Both attacks will need to be accomplished using the Report page!

- Hover over the Task 5 Menu, which will display four links: Schema, Report, and Login.



- Click on the link you want to view (i.e., Schema). The link will open in a new tab.
- You can use the Schema link to see the schema for the table used to build the report.
- The report search is a simple "SELECT.....FROM....WHERE" query using the schema you can see on the initial schema screen. Once you know how to execute the injection, type your attack into the search field and click "Search."

To earn your hash for flag8, you must perform the following actions.

- Once you have figured out how to execute the injection to provide access, type your attack into the search field on the Report page and click "Search."
- Return to the Schema page once you have completed the injection correctly (the report page's results may be misleading).
- On the Schema page, consider how you might find more information than in this table based on how the links have been designed. You might need admin access to see information.
- When you see the new table (not Music), the Hash in the table Schema header is your flag8 hash. Record the hash into your JSON file and submit it to Gradescope.

To earn your hash for flag9, you must perform the following actions.

- Once you can obtain the other table's schema, you can again use the search input to try to return rows from other tables in the database other than what is intended. Once you have figured out how to execute the injection, type your attack into the search field on the Report page and click "Search."
- Once you have done the injection correctly, a username and a password will appear in the report data. Use these values to log into the management console on the "Login" screen. When you have the correct login, you will get into the system, and your hash will be displayed. Record the hash into your JSON file as flag9 and submit it to Gradescope.

Hints:

- You notice that the pages and urls for the different clients look very similar, with similar structure and conventions, can a slight manipulation lead to provide more information?
- How can you add contents from another table to an existing SQL query to return one data set in a report?
- You have heard that this company has realized that client-side scripting is easily bypassable, offers a server-side scripting template, and has started removing the client-side sanitization login. However, you are EXTREMELY confident that they are still using the same server-side sanitization logic/code they have previously used for other sites.
- There is a difference between encoding, escaping, and sanitization. Understand the difference and the validity of using one or the other, remembering there is no client-side and only server-side sanitization.
- Null works great for an SQLi when you don't know the schema and when the developers aren't expecting it. The developers are looking for it!
- Think about the results received from your successful injection in flag8. What is that telling you if you receive a similar output in flag9?
- The login bypass logic you used from Task 3/4 will not work on Task 5, although the choice to attempt is up to you. We explicitly prevented SQL injection on the login screen for this exercise.
- Improperly messing with access can cause issues. You will likely need to re-import your VM Image if you cannot access pages.

Include your flag8 hash into the JSON file and submit it to Gradescope.

See Submission Details for more information.