Description

There are two websites: Site A and Site B.
Site A is closed so members of Site A should migrate their information into Site B by themselves.
Once a member signs in Site A and requests migration into Site B, Site A inserts a new record into `member` table in Site B's DB.
DBs of Site A and Site B are separated.

Site A

Site A has only two pages: sign in page and migration page.

Sign in page

At first, user has to sign in.
When a user inputs his ID and password, Site A finds out the record matched with the input ID.
And it says one of "No matched ID", "No matched ID & PW", and "OK."
"OK" means user succeed in signing in.
Now he can go to migration page.

Migration page

The page says "The information to migrate is user ID, encrypted password, etc."
The `member` table of Site A has columns of `idx`, `is_admin`, `user_id` and `password`.

Trying migration, the user can input new ID for Site B or just use the ID used in Site A.
If the user pushes Migrate button, Site A copies the user's record into Site B by accessing the secret page of Site B, which excutes insert query in the Site B's DB.
If there is some error for query such as duplicated ID or syntax error, Site A says that there is an error, without the detail error information.

Site B

There is no sign up page in Site B.
Site B has nothing special page when a normal user signs in.
However, admin can see the admin page.

Admin Page of Site B

There are two menus: See Log and See Information.

See Log

It shows log about Site B and the admin can search log by its `idx`.
`log` table is empty.

See Information

It shows information about Site B such as FTP access information.
There is a record like the following in the `information` table:
protocol | host | id | pw
FTP | 127.0.0.1 | flag | FLAG{flag}

However, the `password` column is not shown in the information page.
In this case, "FLAG{flag}" is hidden in the page.

Vulnerability

Site A

- Player can do blind SQL injection in the user ID field in the sign in page.
- Player can do SQL injection in the new user ID field in the migration page, and he can know if error occurs during doing it.
- Player can NOT do any SQL injection in the password field.

Site B

- Player can do error-based blind SQL injection in the log_idx field in the See Log page. (quote is escaped and some keywords like "union" is filtered.)
- Player can NOT do any SQL injection in the sign in page.
- Player can NOT do any SQL injection in the See Information page.
- It is NOT IMPORTANT fact that there is a secret page to execute SQL in Site B though the SQL injection is executed in this page.

Designed hints

Password hash type

In the challenge description, we offers 1000 records of `member` table as "member.txt", which is breached.
It contains `user_id`, `password` fields.
And the password is MD5ed `idx.` See the following.
blah | md5("1")
blah | md5("2")
...
blah | md5("1000")
It is easy to guess that password hash type is MD5. Because one can know it easily by googling hash of md5("1").
This hint is used to generate password of the additional inserted record.

Guide to solve

There is a record of 'admin' in the member table of Site A.
But its password value is 32 f's (ffffffffffffffffffffffffffffffff).
It implies that breaking the password hash of 'admin' is improper trial.

SQL injection

Sign in page

It may be used to know `member` table schema.

Intended solution

1. The player picks an arbitrary record in "member.txt"
2. He finds out that MD5 is used to hash by googling its hash, and knows the plain password of the record.
3. At first, he signs in Site A and migrate its naive information into Site B, and he finds out that there is nothing special in Site B.
4. He infers that `is_admin` column in `member` table of Site A by just guessing or seeing `information_schema` by SQLi.
5. He finds out that `is_admin` column is also copied into Site B and he can do SQLi in this procedure.
6. He does how to attack Site A.

- sign in Site A with ID : blah / PW : 1
- he goes to migration page and chooses malicious new ID:
  blah', 'C4CA4238A0B923820DCC509A6F75849B'), (NULL, 1, 'i_am_also_admin

7. He signs in Site B with ID : i_am_also_admin / PW : 1 (because the original password of 'blah' is copied)
8. Now he can see and go to admin page.
9. He guesses that the target is password of ftp://flag@127.0.0.1
10. He checks that See Log page is SQL-injectable.
11. He does error-based blind SQL injection in See Log page's search idx field.

- Because the table `log` is empty, most expressions are never evaluated.
- Some expressions can make error conditionally such as "pow(~0, ~0 * EXPRESSION_TO_EVALUATE)"

12. He can get the password by doing blind SQL injection serveral times.