

Assignment – 1

AI for Cyber Security

Name: Devarakonda Sravani

Reg No: 21BEC7224

List of Vulnerable Parameter

- 1. OWASP CATEGORY: A01 2021 Broken Access Control**
CWE-284: Improper Access Control

Lab: Unprotected admin functionality with unpredictable URL

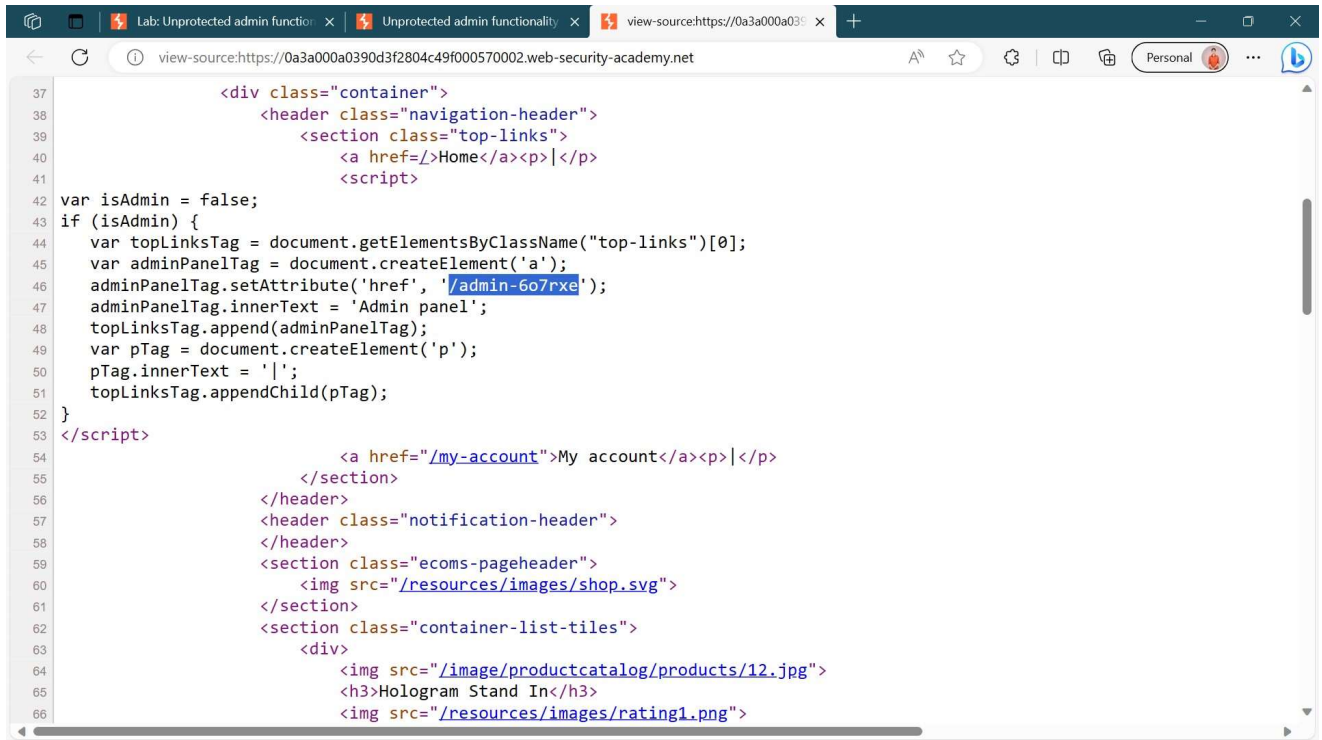
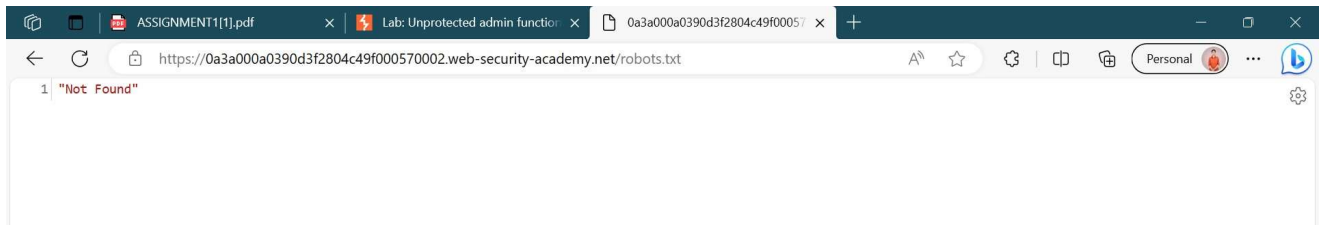
APPRENTICE

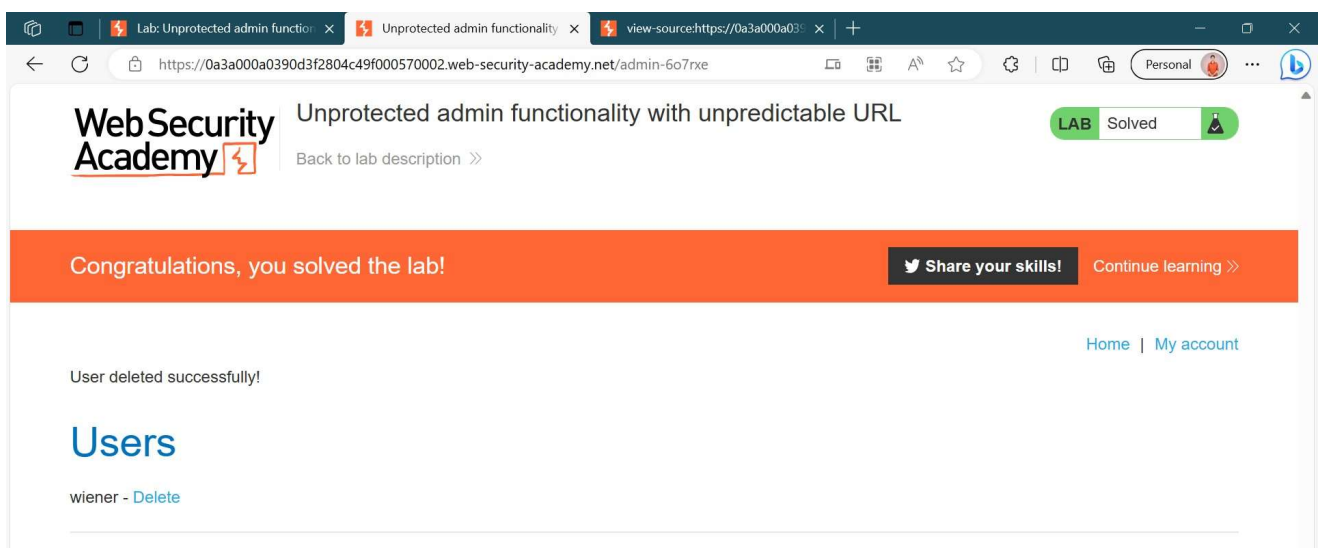
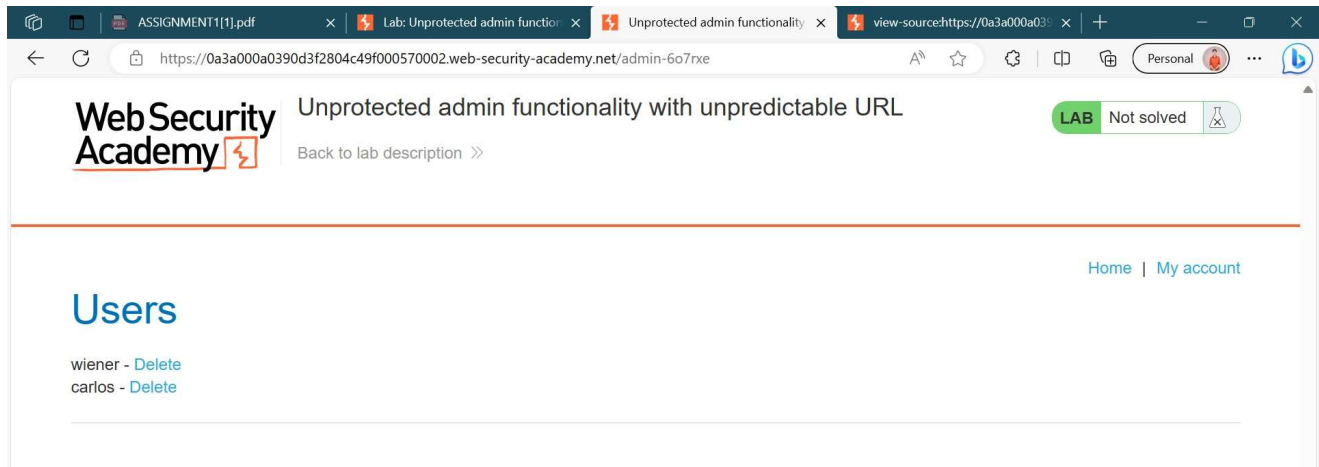


This lab has an unprotected admin panel. It's located at an unpredictable location, but the location is disclosed somewhere in the application.

Solve the lab by accessing the admin panel, and using it to delete the user `carlos`.

Solving the Lab:





Description:

The product does not restrict or incorrectly restricts access to a resource from an unauthorized actor.

Business Impact:

"Improper Access Control," represents a significant threat to businesses. This vulnerability arises when systems fail to enforce proper restrictions on user access to resources. Exploiting this flaw can lead to unauthorized data exposure, alterations, or even complete system compromise. Business consequences include breaches of sensitive information, loss of intellectual property, regulatory penalties, legal actions, erosion of customer trust, and reputational damage. Additionally, addressing this issue demands substantial investments in code review, security architecture, and access management solutions, diverting resources

from core business activities and hindering innovation while striving to ensure robust access controls.

In the above experiment we have clearly seen that we have gained access to the admin panel by just seeing the java script code and using the URL we can be able to gain unauthorized access and deleted the user.

2. OWASP CATEGORY: A02 2021 Cryptographic Failures CWE-259: Use of Hard-coded Password

MD5 Hash Generator

Use this generator to create an MD5 hash of a string:

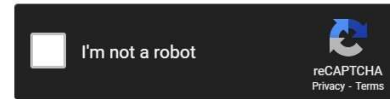
pranavi

Generate →

Your String	pranavi
MD5 Hash	da4ce4d4ab4ffaab037bc1ba64eaec2c <button>Copy</button>
SHA1 Hash	392dcd7e078a7a730c3967e5967c1451702a5862 <button>Copy</button>

Enter up to 20 non-salted hashes, one per line:

da4ce4d4ab4ffaab037bc1ba64eae2c



Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1 sha1_bin), QubesV3.1BackupDefaults

Hash	Type	Result
da4ce4d4ab4ffaab037bc1ba64eae2c	md5	pranavi

Color Codes: Green Exact match, Yellow Partial match, Red Not found.

Description:

The product contains a hard-coded password, which it uses for its own inbound authentication or for outbound communication to external components.

Business Impact:

"Use of Hard-coded Password," poses a substantial business risk by embedding passwords directly into code or configuration files. This practice can lead to unauthorized access, data breaches, and compromised systems. If exploited, attackers can gain control over critical assets, manipulate sensitive data, disrupt operations, and compromise customer trust. Business impact includes financial losses due to data loss or theft, regulatory fines for noncompliance, legal liabilities, reputational damage, and operational downtime. Mitigation efforts require resources for code review, password management, and system updates, impacting development timelines and diverting focus from innovation to security remediation.

In the above experiment we have cracked the password by using md5 hash of that password. This comes under one of the cryptographic failure.

3. OWASP CATEGORY: A03 2021 Injection CWE-564: SQL Injection: Hibernate

Lab: SQL injection vulnerability in WHERE clause allowing retrieval of hidden data

APPRENTICE



LAB



Solved

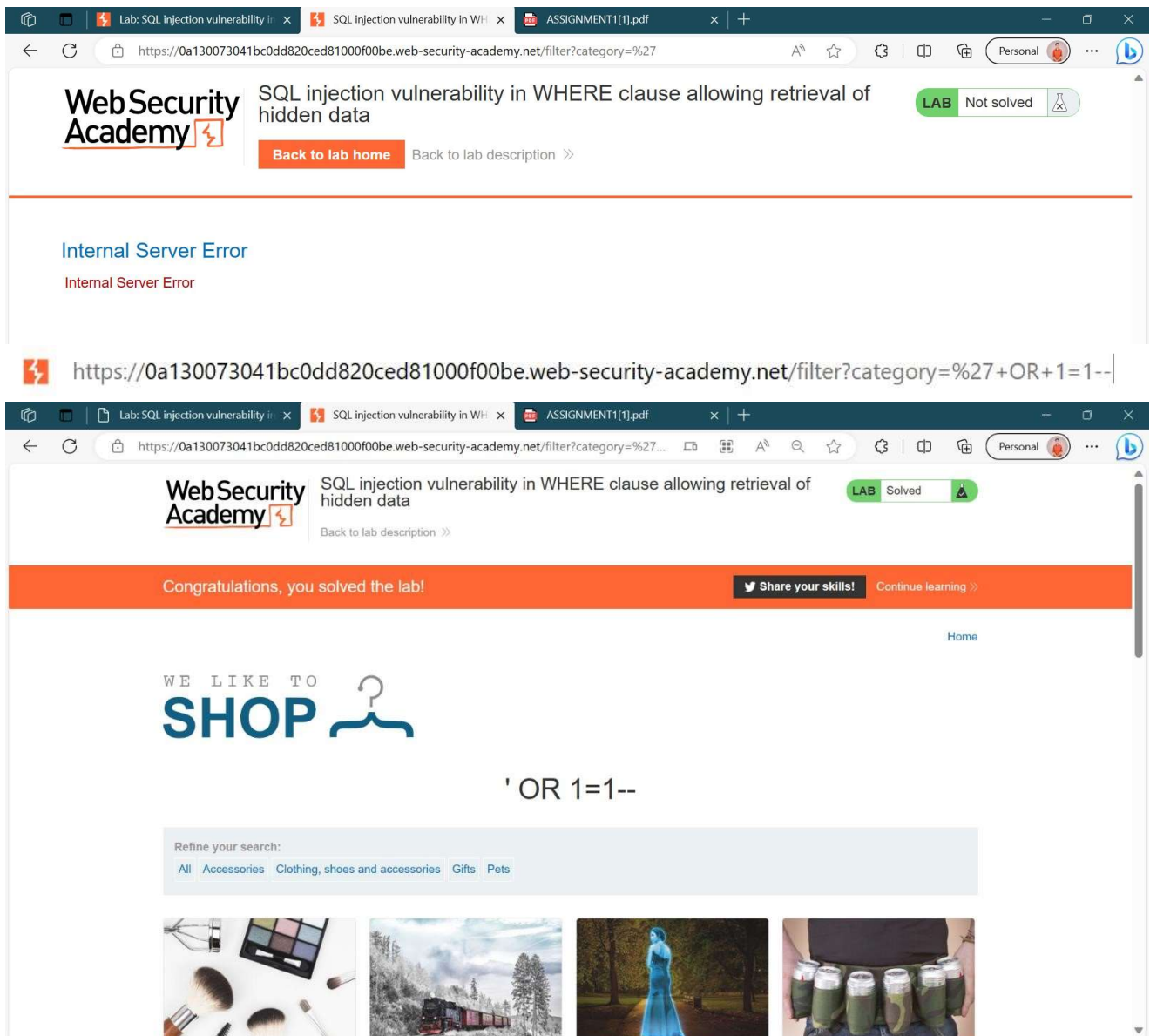


This lab contains a SQL injection vulnerability in the product category filter. When the user selects a category, the application carries out a SQL query like the following:

```
SELECT * FROM products WHERE category = 'Gifts' AND rel
```

To solve the lab, perform a SQL injection attack that causes the application to display one or more unreleased products.

Solving the Lab:



SELECT * FROM Products WHERE category = ' ' OR 1 = 1 - - ' Description:

Using Hibernate to execute a dynamic SQL statement built with user-controlled input can allow an attacker to modify the statement's meaning or to execute arbitrary SQL commands.

Business Impact:

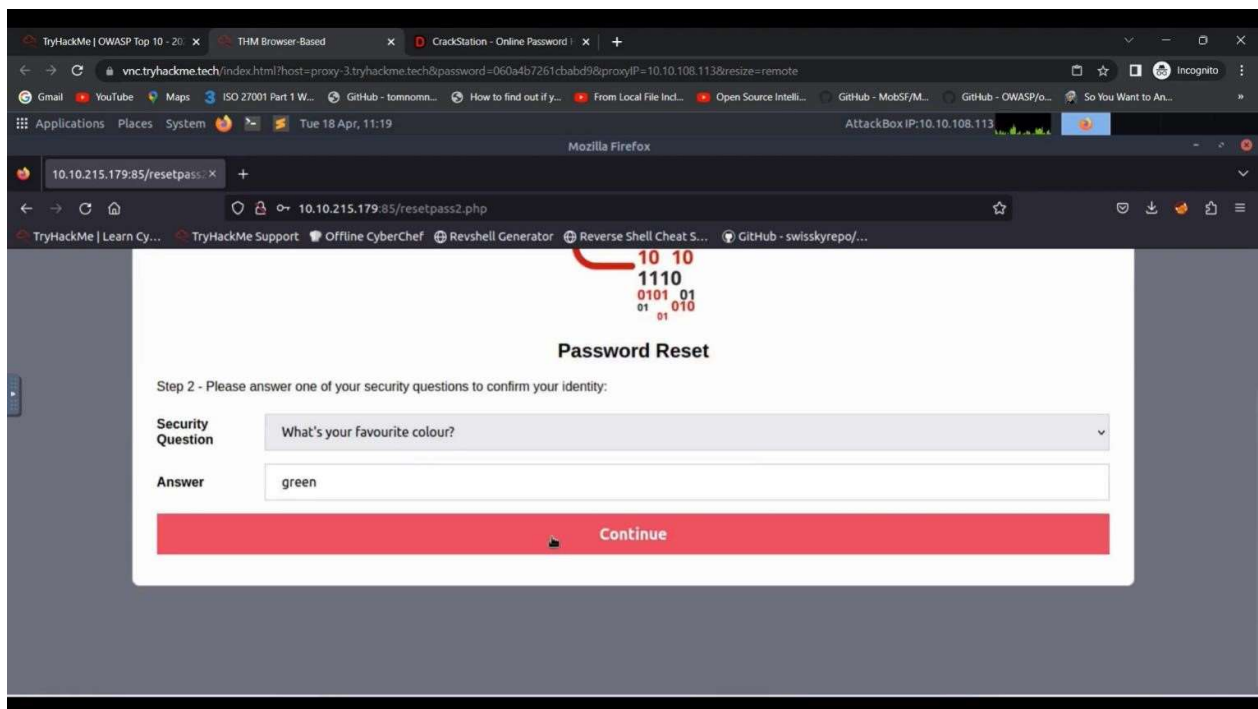
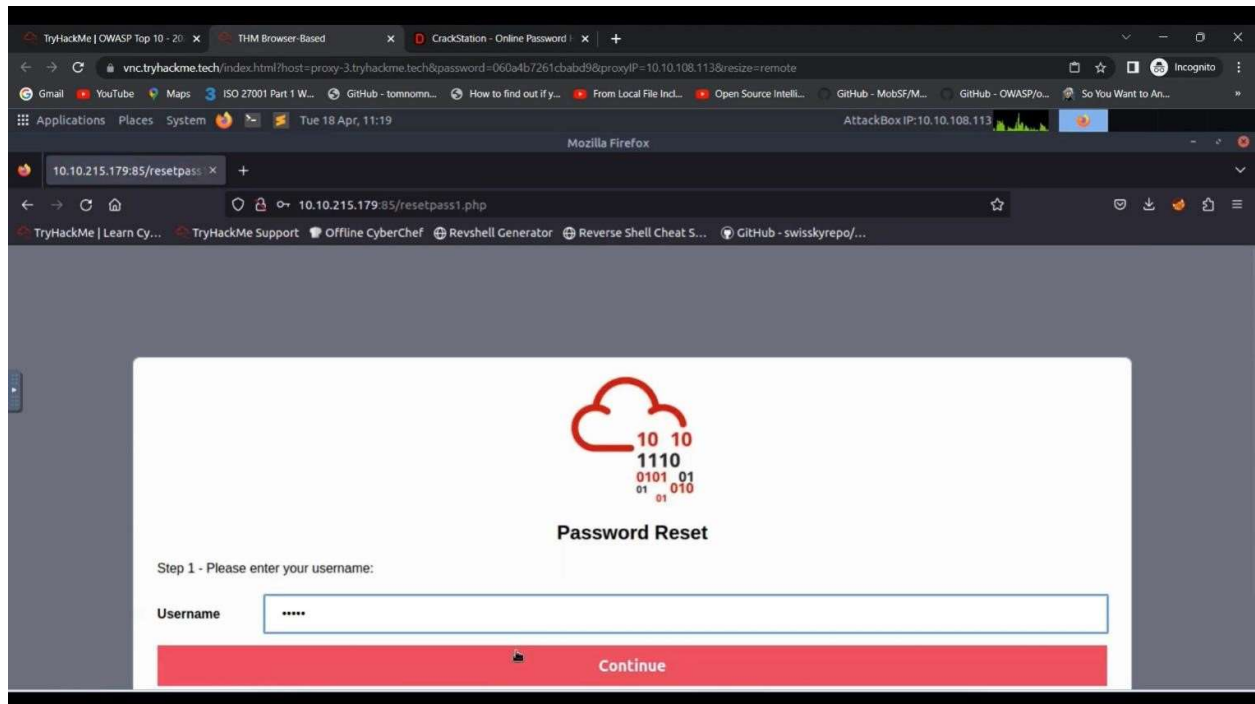
SQL Injection in Hibernate can have severe business impact. Attackers exploit vulnerabilities in Hibernate ORM framework to inject malicious SQL queries, potentially leading to unauthorized access, data leakage, or manipulation of sensitive data stored in the database. This can compromise data integrity, breach user privacy, and result in financial losses, legal

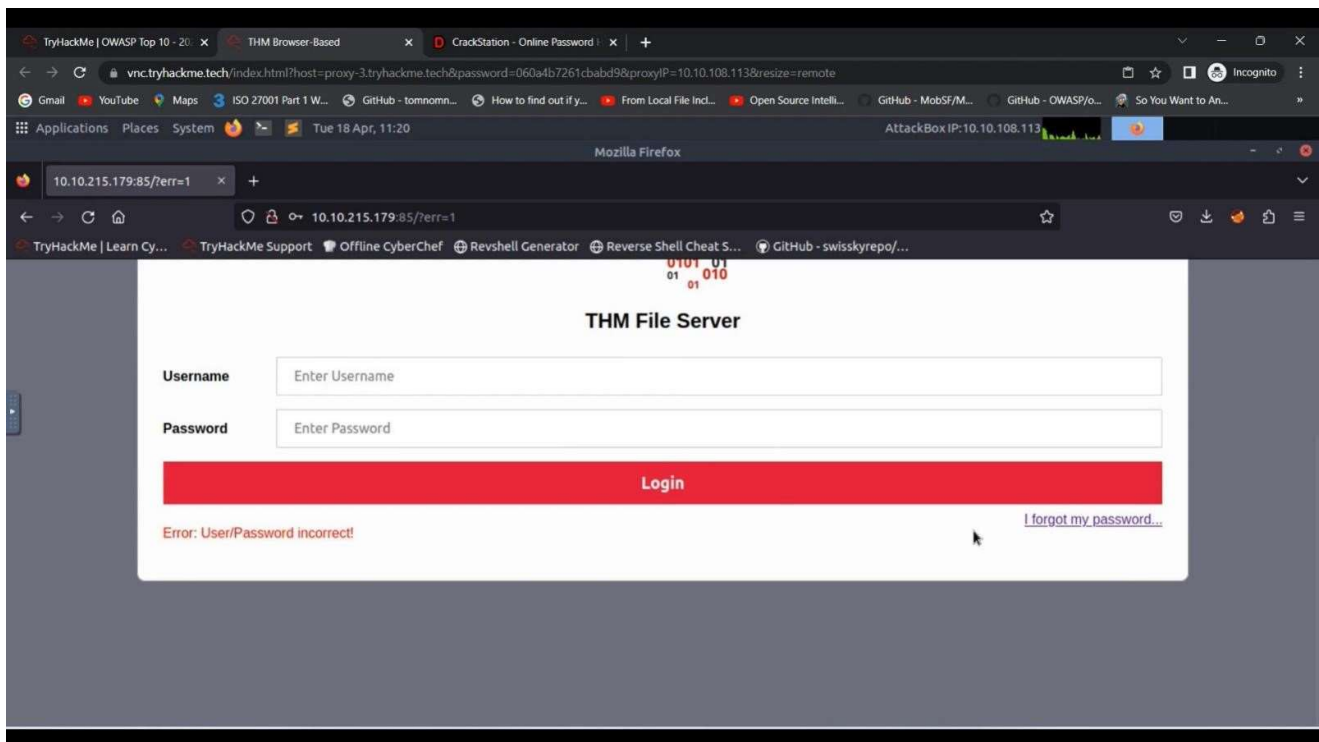
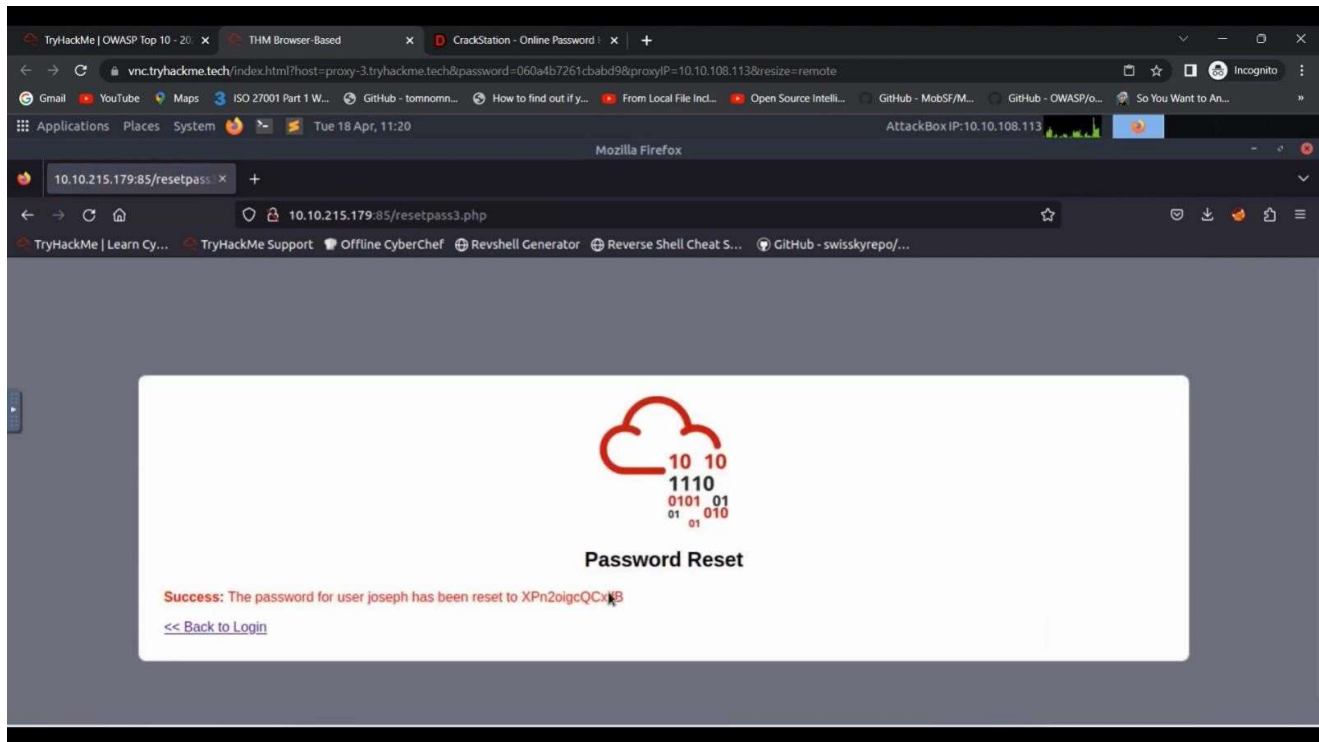
liabilities, and reputational damage. Promptly addressing this vulnerability is crucial to safeguarding the application's security and maintaining trust among users and stakeholders.

In the above experiment we have modified the SQL statement and displayed all the products in the category whether they are released or not. This is known as SQL Injection.

4. OWASP CATEGORY: A04 2021 Insecure Design CWE-657:

Violation of Secure Design Principles





Description:

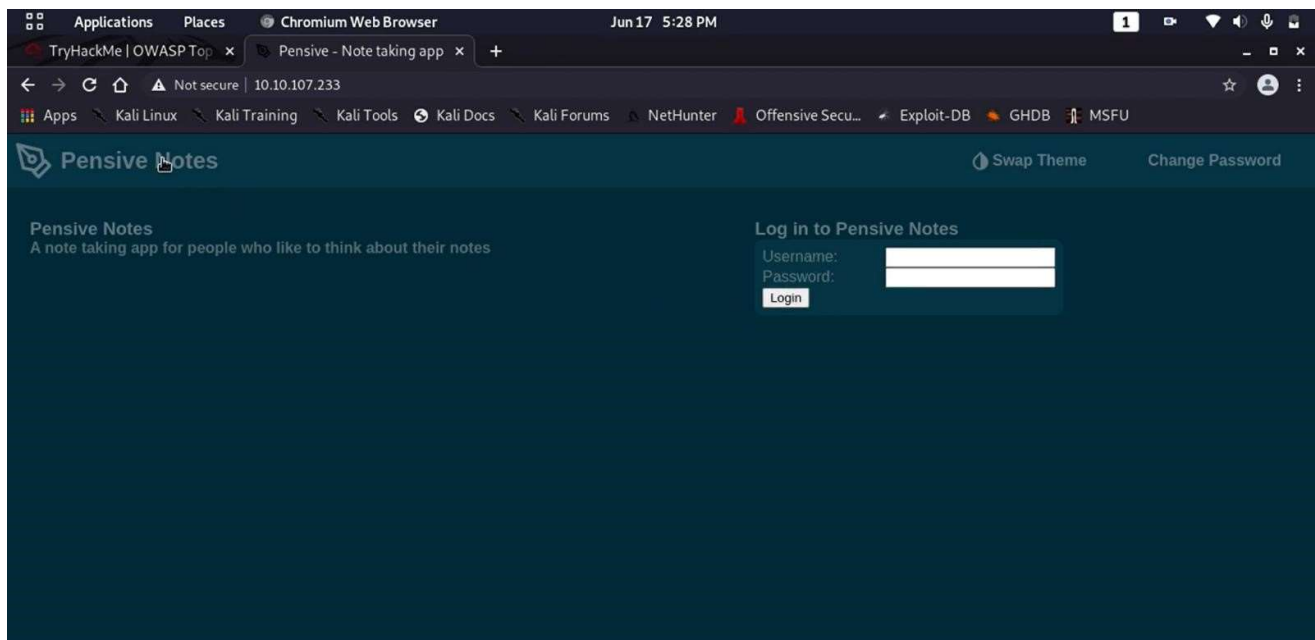
The product violates well-established principles for secure design.

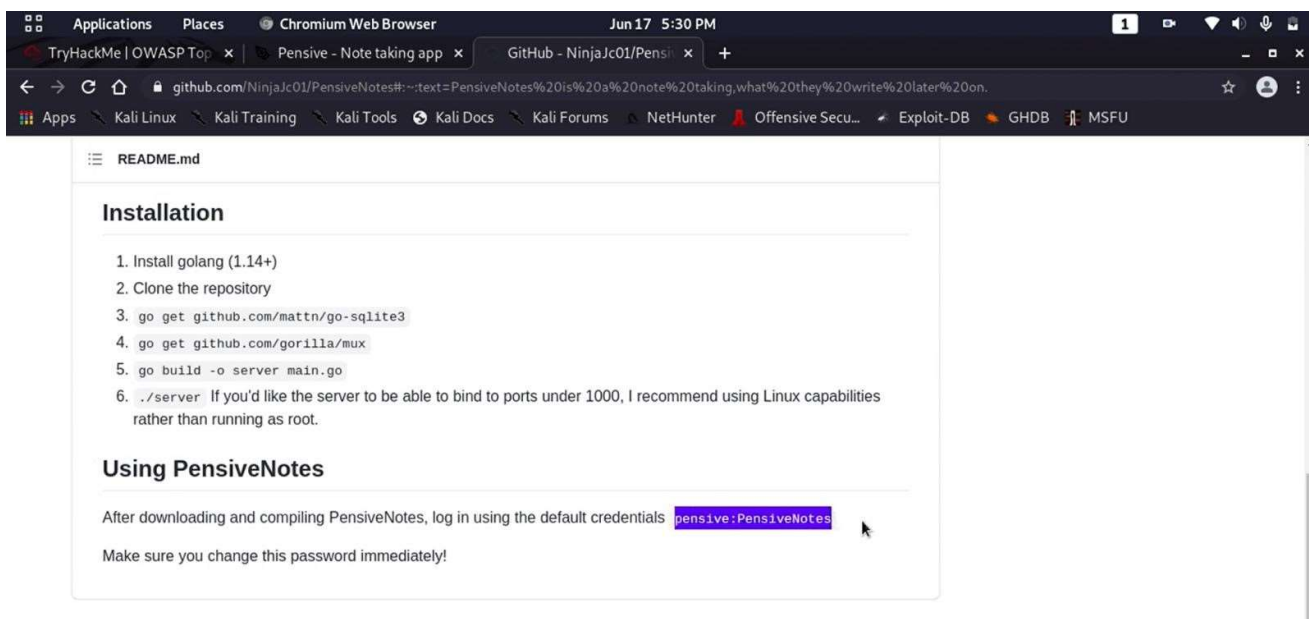
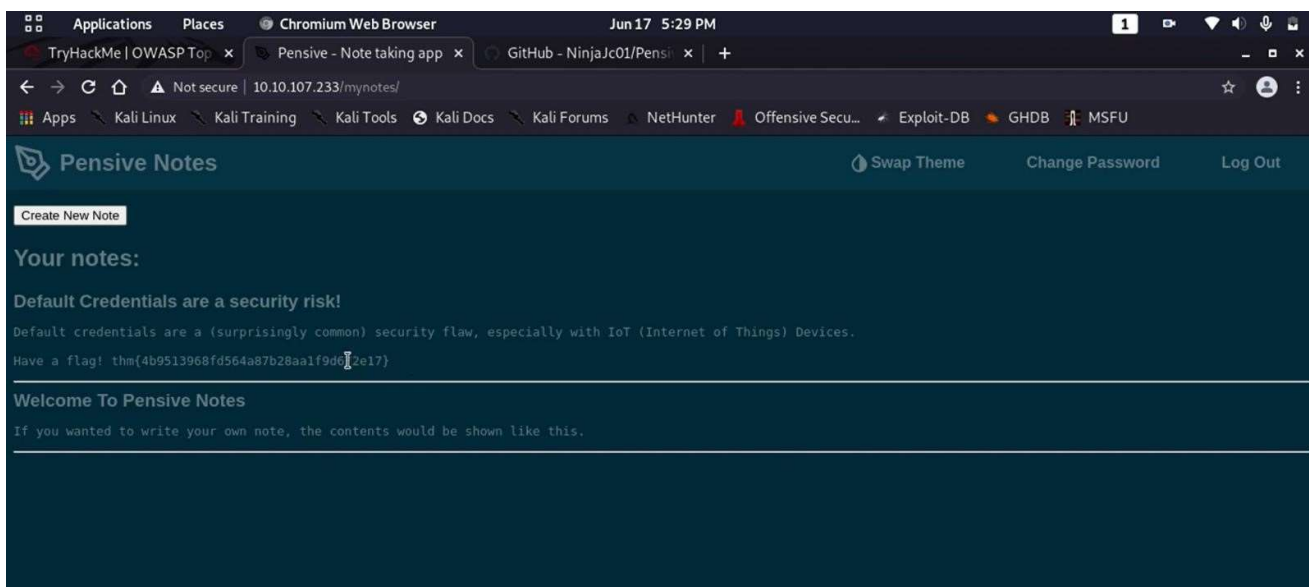
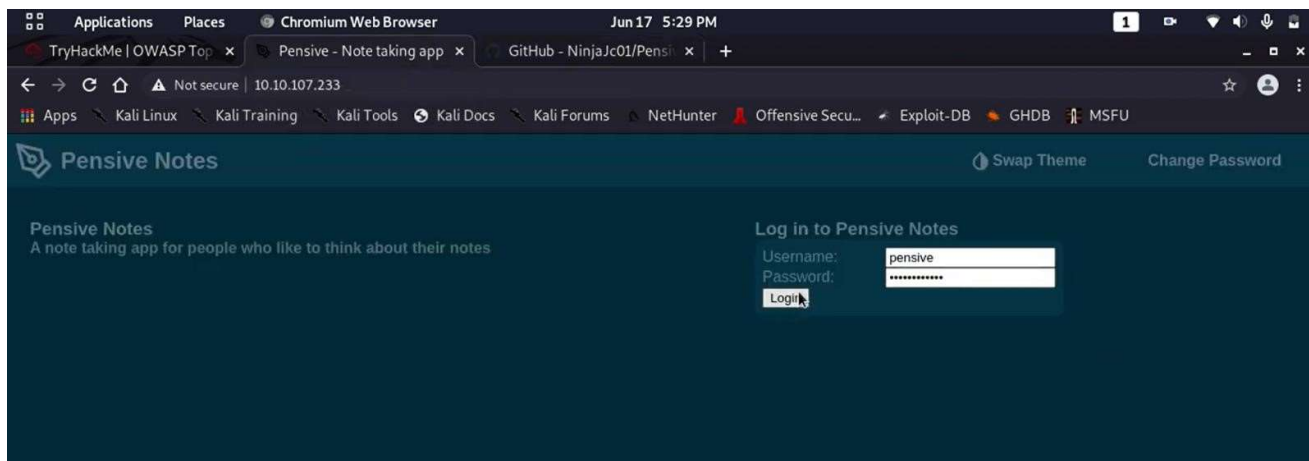
Business Impact:

"Violation of Secure Design Principles," presents a critical risk to businesses. This vulnerability emerges when software systems are developed without following established security best practices and design principles. As a result, applications become susceptible to various attacks and breaches. Exploiting this flaw can lead to unauthorized access, data breaches, system compromise, and operational disruptions. The business impact encompasses financial losses, regulatory penalties, legal actions, reputational damage, erosion of customer trust, and increased security-related development costs. Addressing this issue necessitates reevaluating and retrofitting the software's design, diverting resources from innovation and development efforts, potentially delaying project timelines and undermining competitiveness.

5. OWASP CATEGORY: A05 2021 Security Misconfiguration

CWE-11: ASP.NET Misconfiguration: Creating Debug Binary





Description:

The product transmits sensitive or security critical data in cleartext in a communication channel that can be sniffed by unauthorized actors.

Business Impact:

Security Misconfiguration can harm businesses by exposing confidential data during transmission, leading to data breaches, compromised customer trust, regulatory violations, and potential legal consequences.

In the above experiment we have noticed that credentials are available online for the website and anyone can have access to it. All the confidential details are exposed to everyone by which anyone can have access to data.