Assignment-1

Hiya Sharma
21BCY10078
VIT Bhopal

Perform the top 5 vulnerabilities of OWASP.

1. Cross-Site Scripting-

CWE : CWE-79

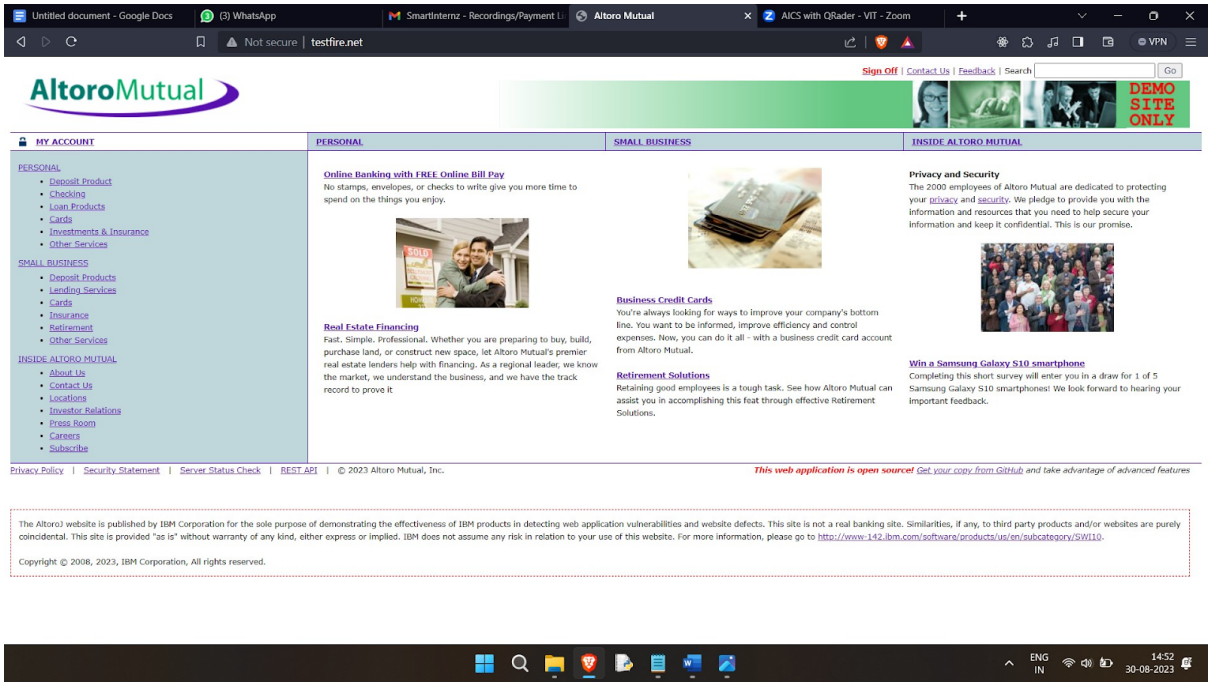
OWASP Category: A03:2021 – Injection

Description: Cross-Site Scripting (XSS) is a security vulnerability where attackers inject malicious code into a website, potentially compromising users' browsers and stealing sensitive data.

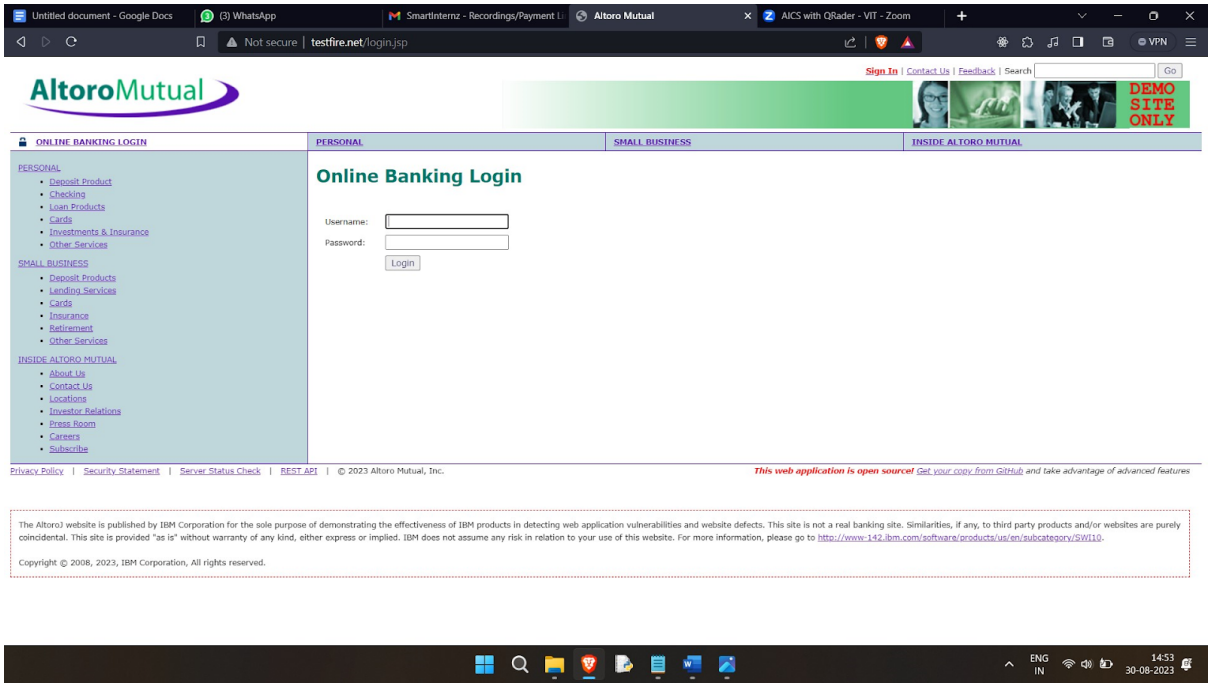
Business Impact: Cross-Site Scripting (XSS) poses a significant risk to businesses due to its potential for severe consequences. One primary concern is the possibility of a data breach, where attackers exploit vulnerabilities to access sensitive user information, resulting in legal troubles and reputational harm. Financial losses can stem from fraudulent activities like unauthorised transactions and subsequent chargebacks, accompanied by the costs of investigating and resolving these incidents. Such security lapses can lead to a damaged reputation, causing a loss of customer trust and impeding growth opportunities. Legal ramifications can also arise, including regulatory fines and compliance complexities. Additionally, successful XSS attacks can disrupt services, causing downtime, reduced sales, and increased customer support demands. Furthermore, there's an SEO impact, as search engines may penalise compromised websites, diminishing their online visibility. Addressing these vulnerabilities requires a diversion of resources from other important projects, potentially creating a competitive disadvantage in industries where security is a key differentiator. Protection of intellectual property is another concern, as attacks could result in the theft of valuable proprietary information. Even after an attack is thwarted, businesses may need prolonged security investments to prevent future incidents and restore customer confidence. To counter these risks, businesses must emphasize security testing, secure coding practices, and employee training while maintaining swift and effective vulnerability response measures.

Steps to perform:

Step-1- Access the URL

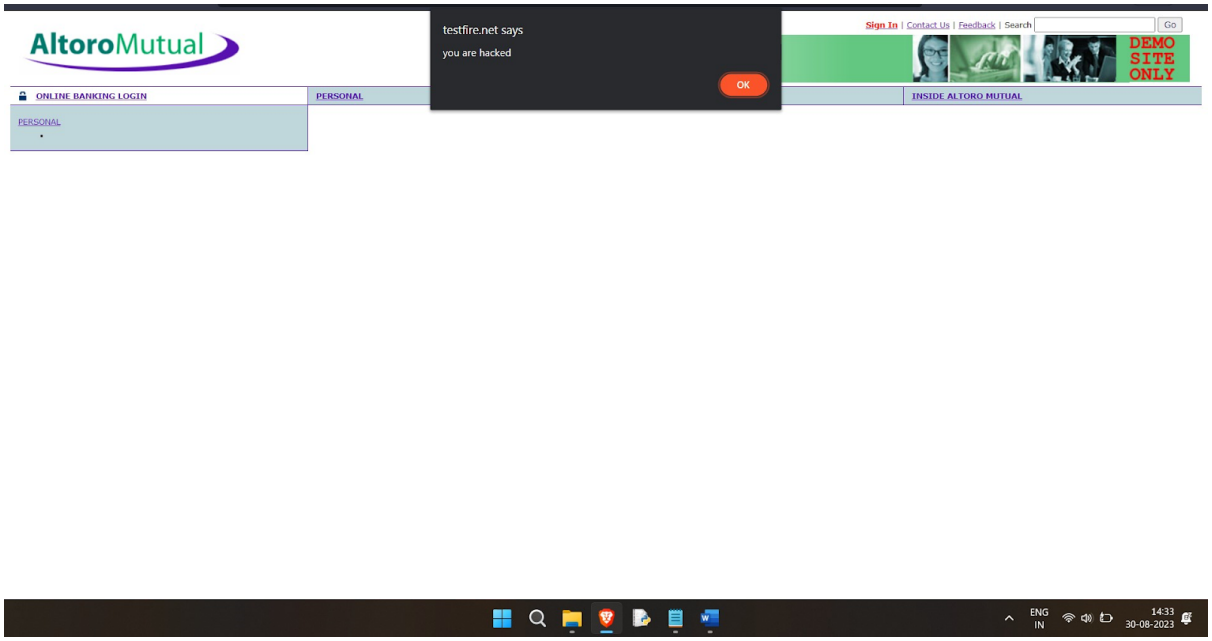


Step 2: Go to the login page and enter credentials



Step 3: We will enter the script in the search bar.

Step 4:- This is the pop up you get after you successfully inject the script in the contact page.



Recommendation-

"Mitigate Cross-Site Scripting (XSS) vulnerabilities by implementing strict input validation, output encoding, security training, and robust web application firewalls."

2. SQL Injection-

CWE : CWE-284

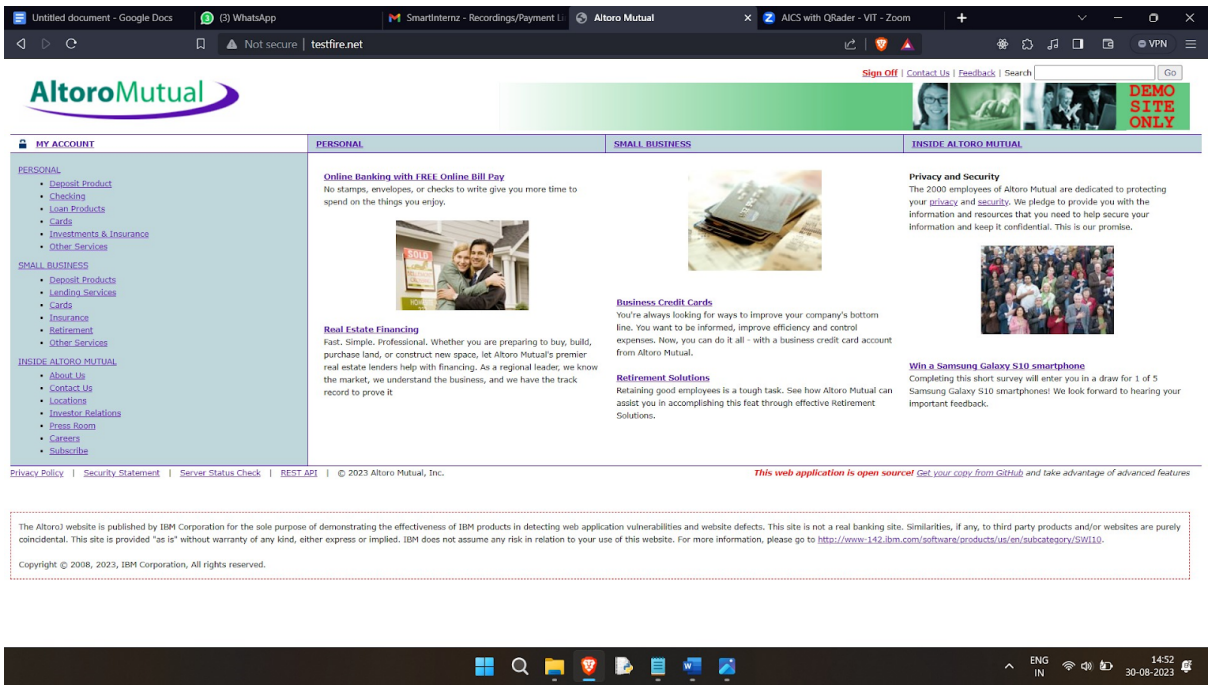
OWASP Category:A03:2021-Injections

Description: SQL Injection is a malicious technique where attackers exploit vulnerabilities in a web application's input fields to manipulate SQL queries executed on a database. By injecting malicious SQL code, they can gain unauthorized access to sensitive data, modify or delete records, and potentially take control of the database, leading to data breaches, unauthorized actions, and security risks. Preventing SQL Injection involves input validation, parameterized queries, and security best practices to ensure the integrity and security of database interactions.

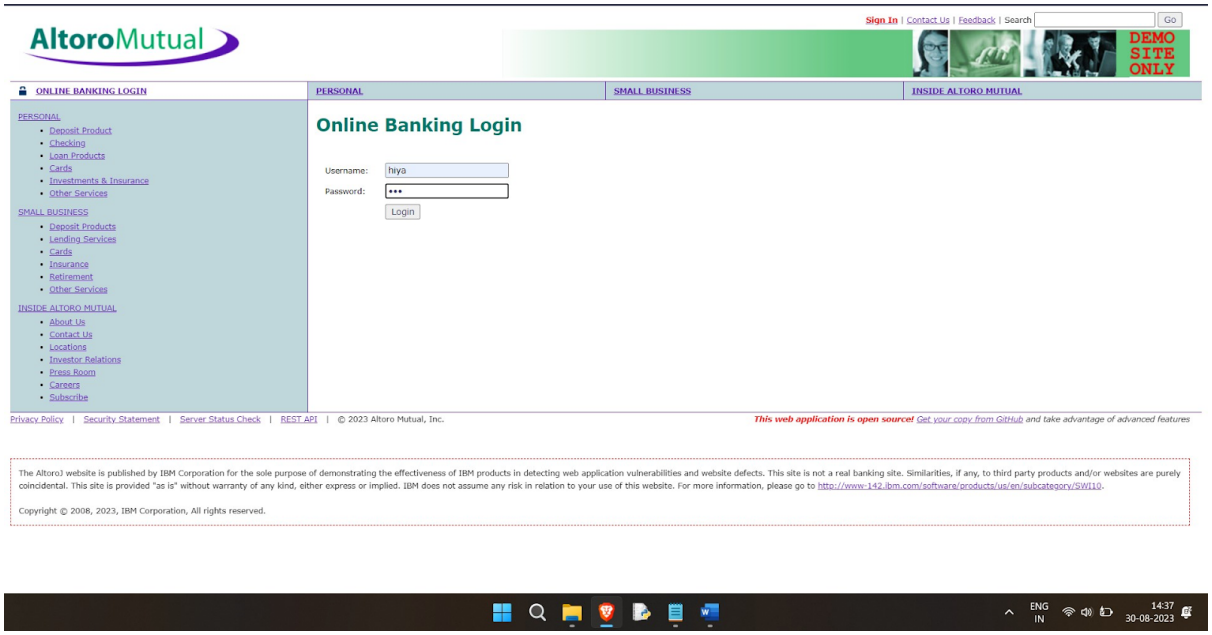
Business Impact: SQL Injection vulnerabilities can have dire consequences for businesses. They expose critical data to unauthorized access, leading to breaches that trigger legal actions, regulatory fines, and reputational harm. Financial losses are incurred through fraudulent transactions and customer support costs. The resulting damaged reputation undermines customer trust, hindering growth and competitiveness. Service disruptions and potential intellectual property theft compound these issues. Preventative measures such as input validation, parameterized queries, and security best practices are essential to mitigate these risks and their far-reaching business impacts.

Steps to perform:

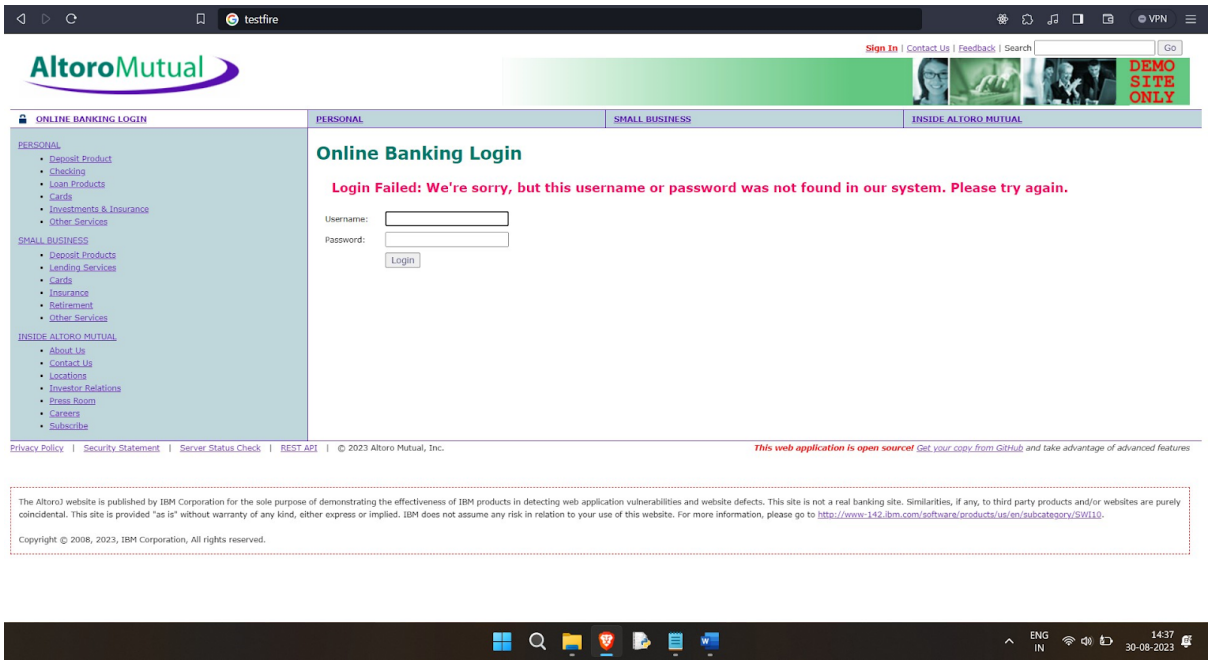
Step-1: Access the URL



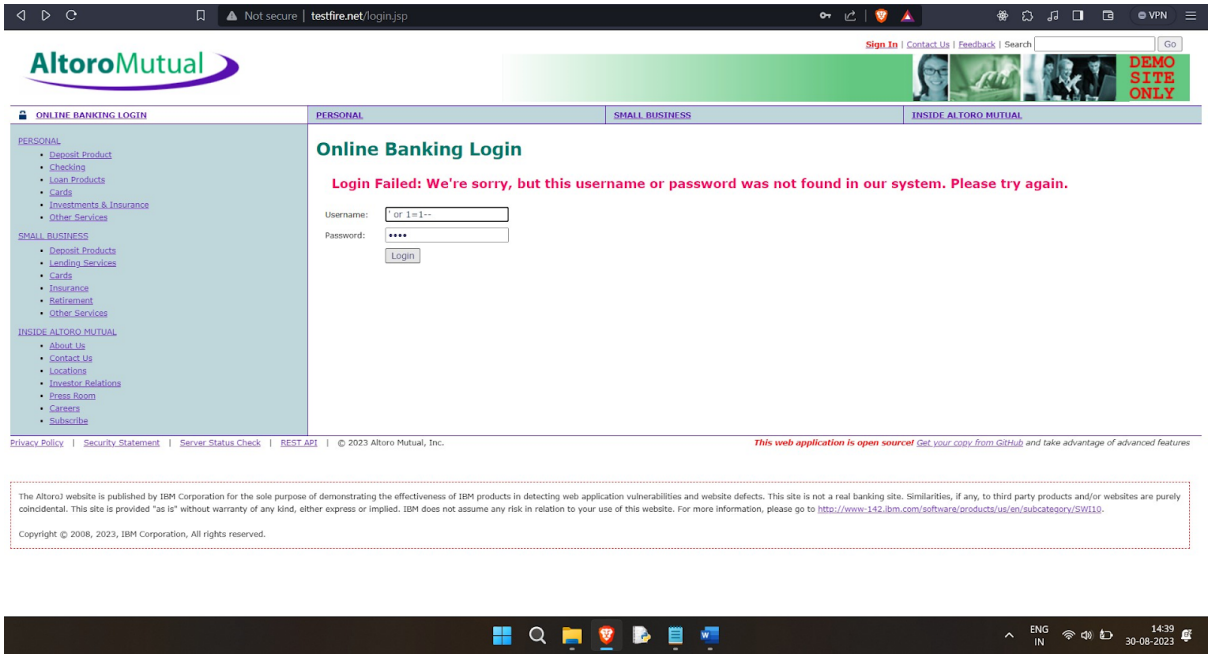
Step-2: Enter the login credentials in and try to validate as shown below.



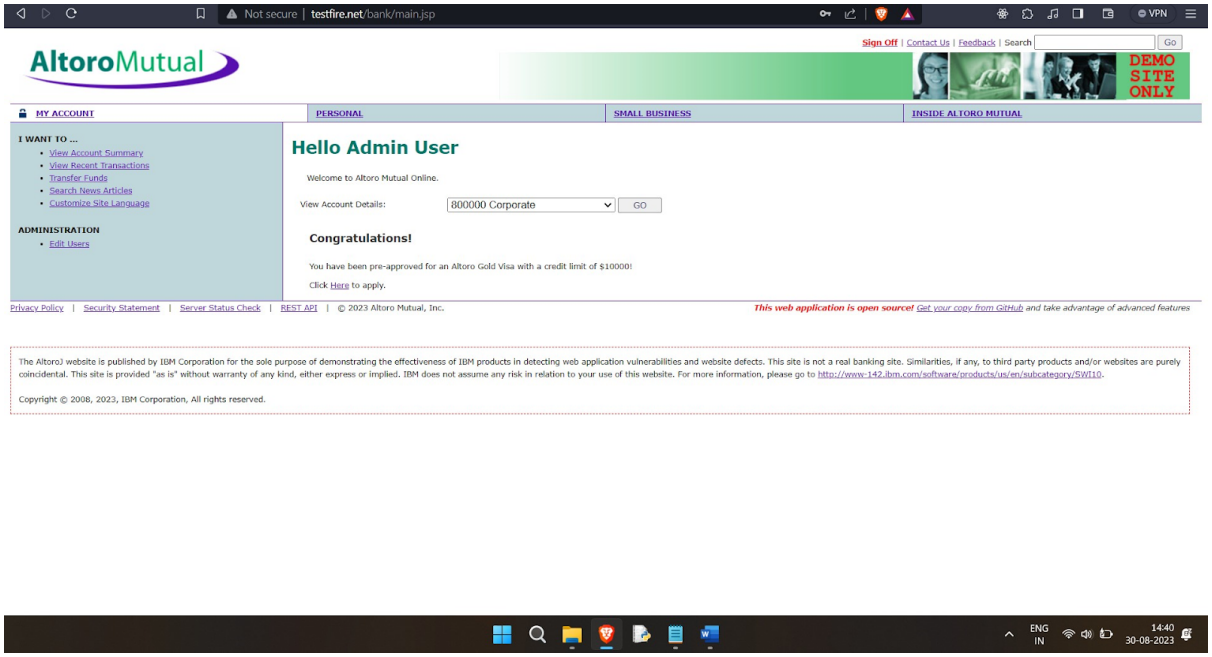
It will show login fails.



Step-3: We will put the SQL statement.



Now we are logged into the account.



Recommendations: "Prevent SQL Injection by using parameterized queries, input validation, and security testing."

3. **Cross-Site request forgery-**

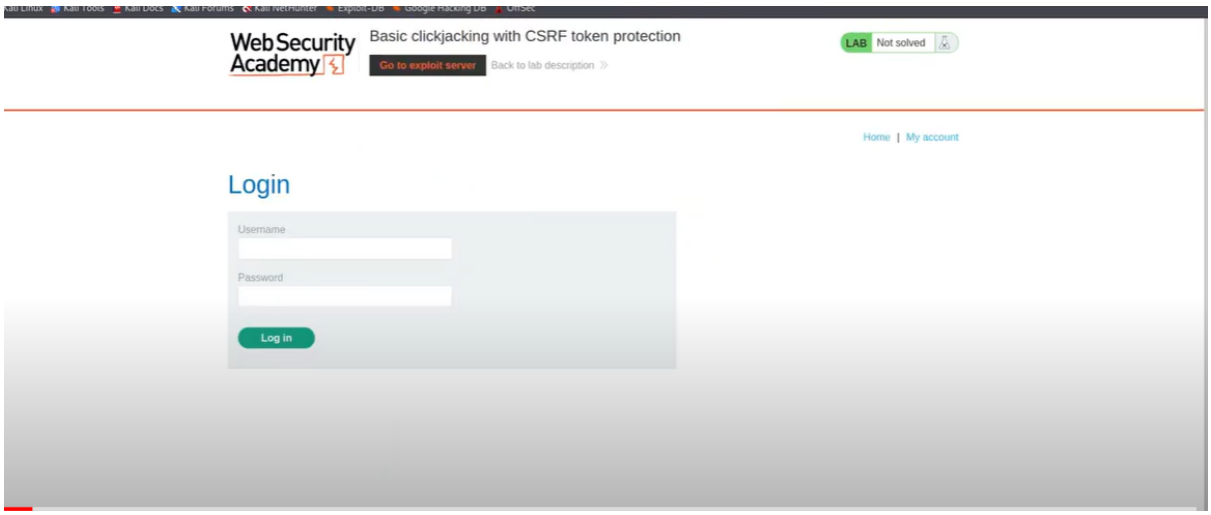
CWE : CWE-353

OWASP Category: A05:2021-Injections

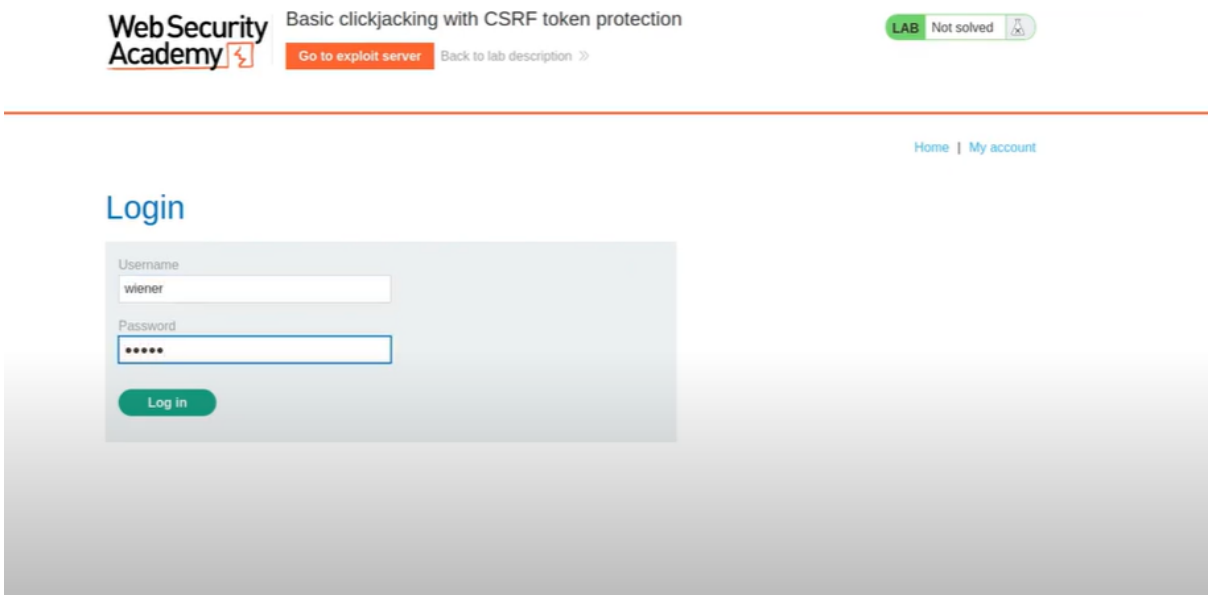
Description: Cross-Site Request Forgery (CSRF) is a web security flaw where attackers trick users into performing unintended actions on a different site, potentially leading to unauthorized activities and data breaches. Preventive measures include anti-CSRF tokens and validating request origins.

Steps to perform:

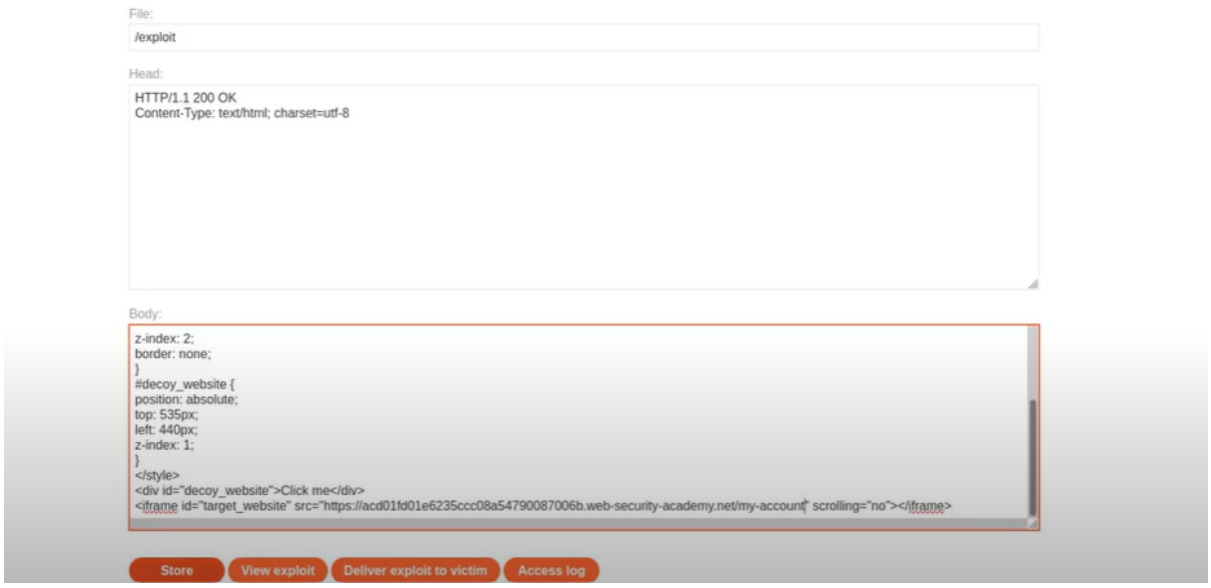
Step-1: Access the url and come to login page.



Step-2 : Try to enter the login credentials



Step-3: Make changes to the HTML code.



Business Impact:

Cross-Site Request Forgery (CSRF) attacks can have detrimental consequences for businesses. Attackers exploit vulnerabilities to manipulate user accounts, initiate unauthorized transactions, and potentially cause financial losses. These attacks erode customer trust, increase support costs, and damage the company's reputation due to perceived security weaknesses. Effective prevention measures and user awareness are crucial to mitigate these business risks.

Recommendations:

"Prevent CSRF attacks by using anti-CSRF tokens, enforcing strict referer headers, and implementing same-origin policies."

4. **ClickJacking**

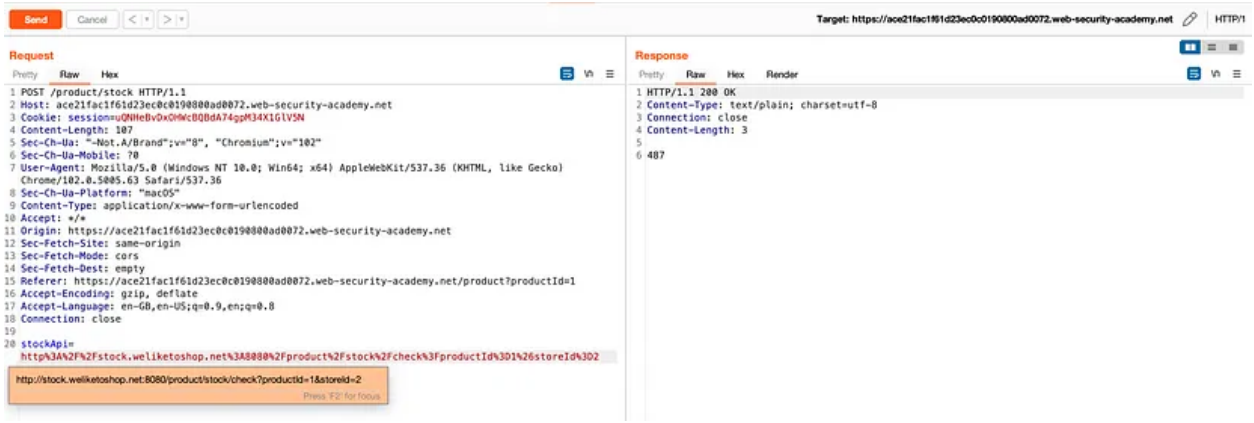
CWE : CWE-451

OWASP Category : UI redress attack

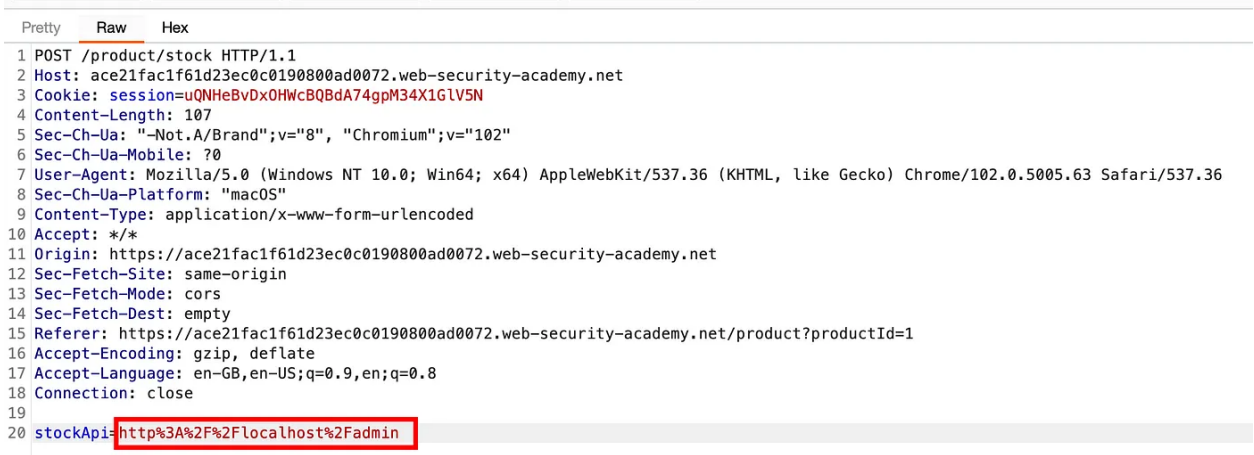
Description: Clickjacking, also known as a “UI redress attack”, is when an attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on another page when they were intending to click on the top level page. Thus, the attacker is “hijacking” clicks meant for their page and routing them to another page, most likely owned by another application, domain, or both.

Steps to perform:

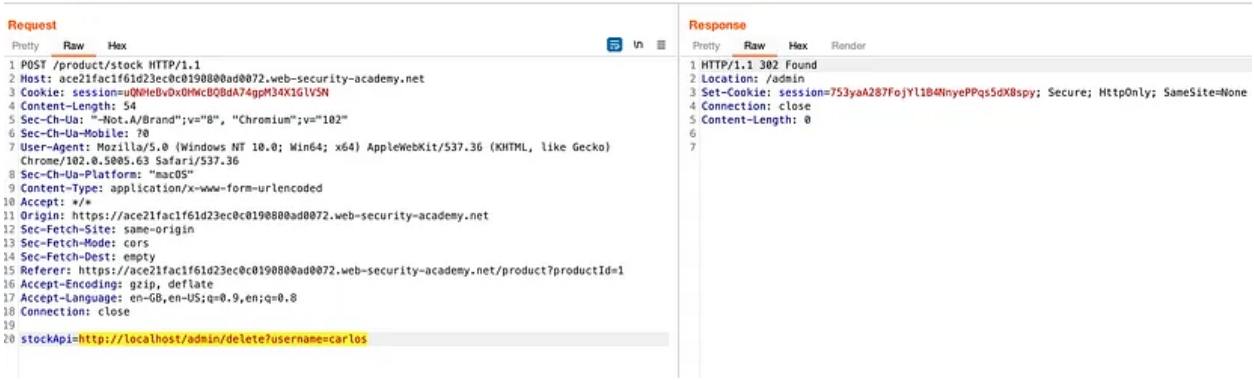
Step 1: Begin by accessing the lab, clicking on a product, and intercepting the Check Stock functionality using Burp Suite. The stockApi was accessing an internal system to check the stock of the product.



Step 2: Replace this URL with localhost: http://localhost/admin to see if the internal admin interface was accessible. It worked and the application returned the delete user endpoint /delete?username=carlos in the response.



Step 3: Append it in the stockApi POST request body and the user gets deleted, which completed the lab.



Business Impact:

Clickjacking poses significant business risks, as malicious actors manipulate users into unknowingly interacting with hidden elements, leading to unauthorized actions, data breaches, financial loss, and reputational damage due to customer distrust and potential legal liabilities.

Recommendations:

Mitigate clickjacking risks by implementing frame-busting scripts, utilizing X-Frame-Options headers, employing Content Security Policy (CSP), and staying updated on emerging clickjacking techniques to ensure robust protection against unauthorized framing of your website's content.

5. Broken Access Control-

CWE : CWE-285

OWASP Category : A5

Description:

Broken Access Control refers to security vulnerabilities where improper authorization mechanisms enable unauthorized users to access restricted resources or perform actions they shouldn't have permission for, leading to data breaches, unauthorized operations, and compromised system integrity.

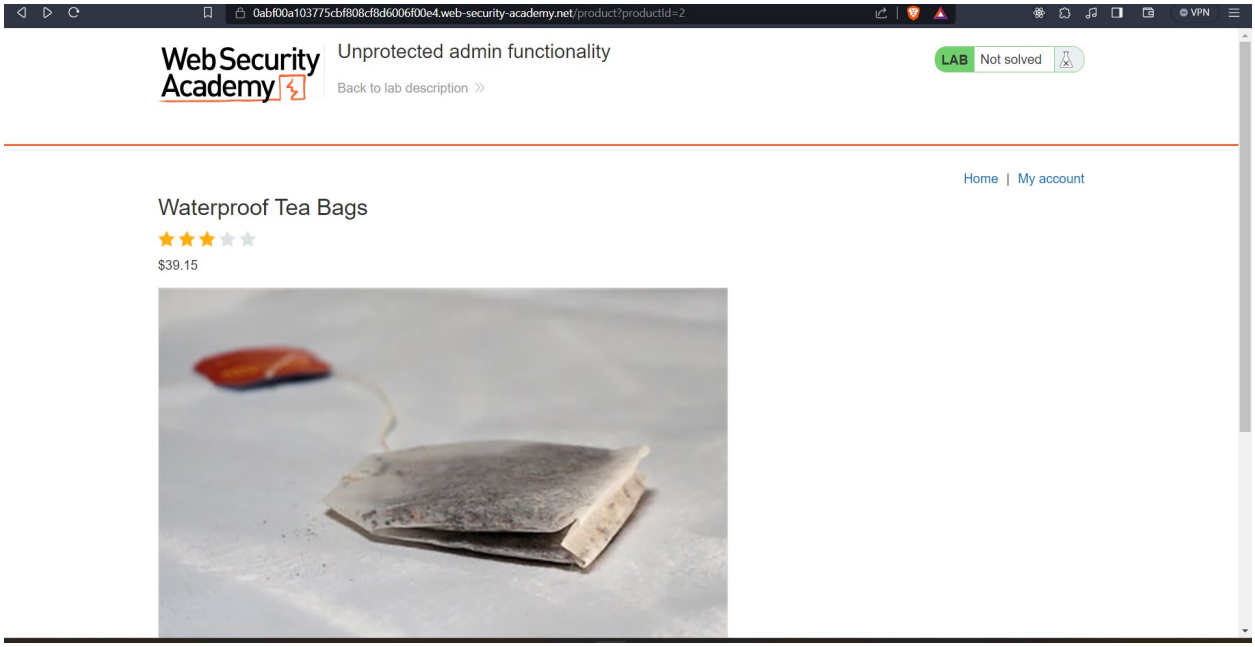
Business Impact:

Broken Access Control vulnerabilities can have severe business consequences by enabling unauthorized users to access sensitive data or perform restricted actions, leading to data

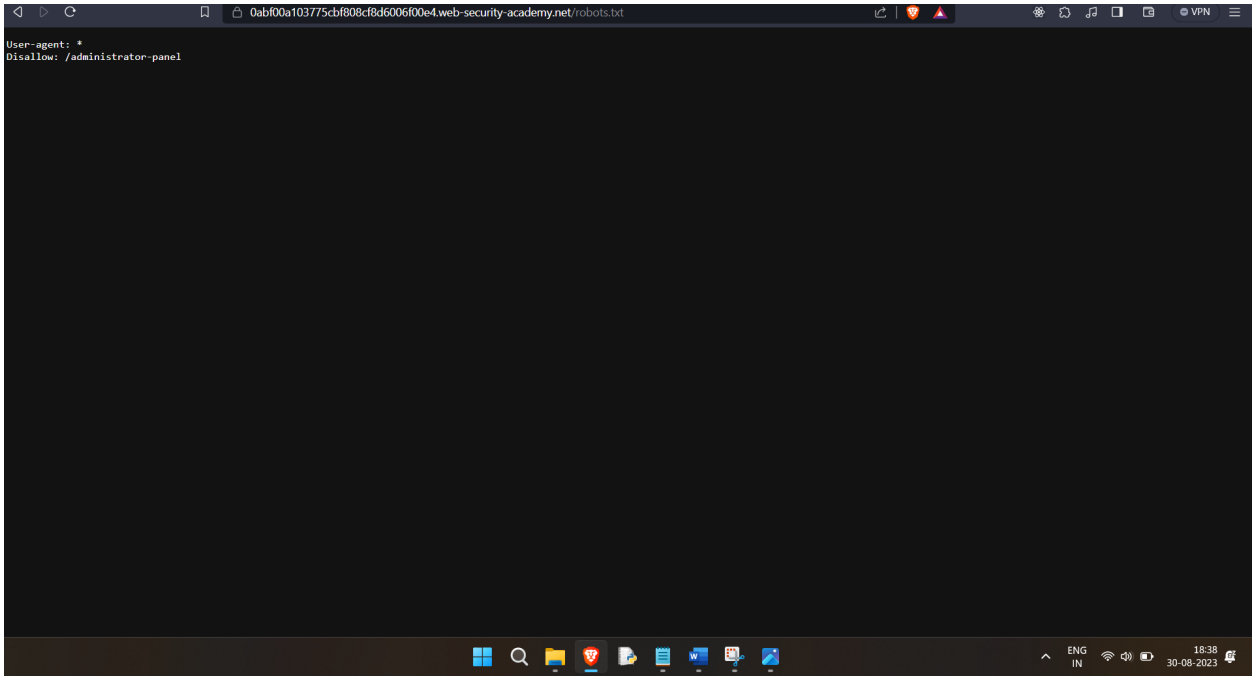
breaches, compliance violations, legal liabilities, reputational damage, loss of customer trust, and potential regulatory fines.

Steps to perform:

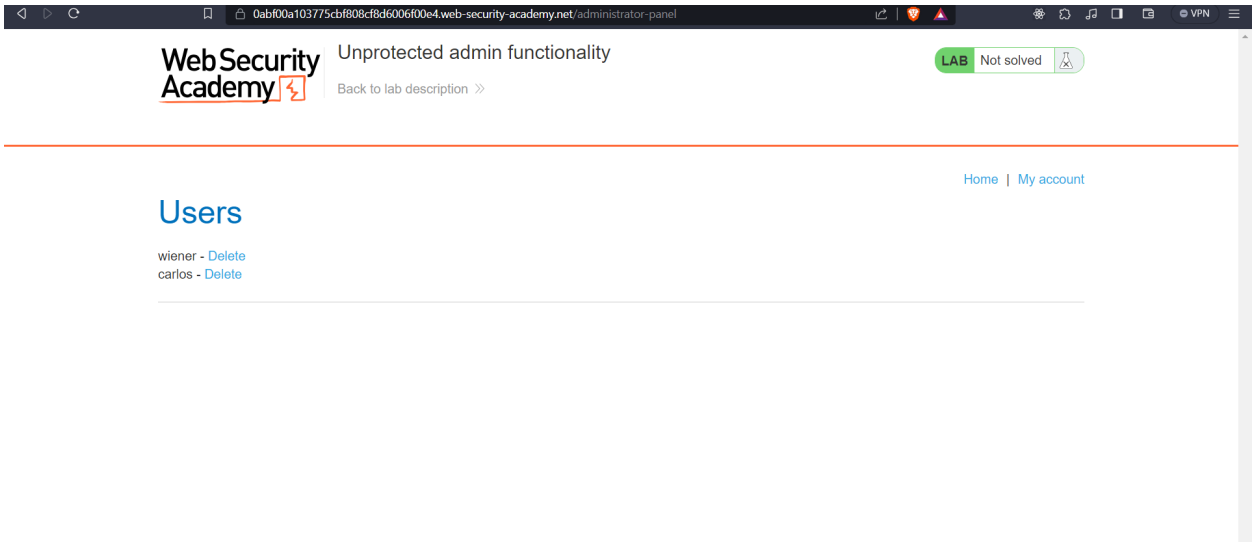
Step-1: Begin by accessing the lab, view any product from the lab.



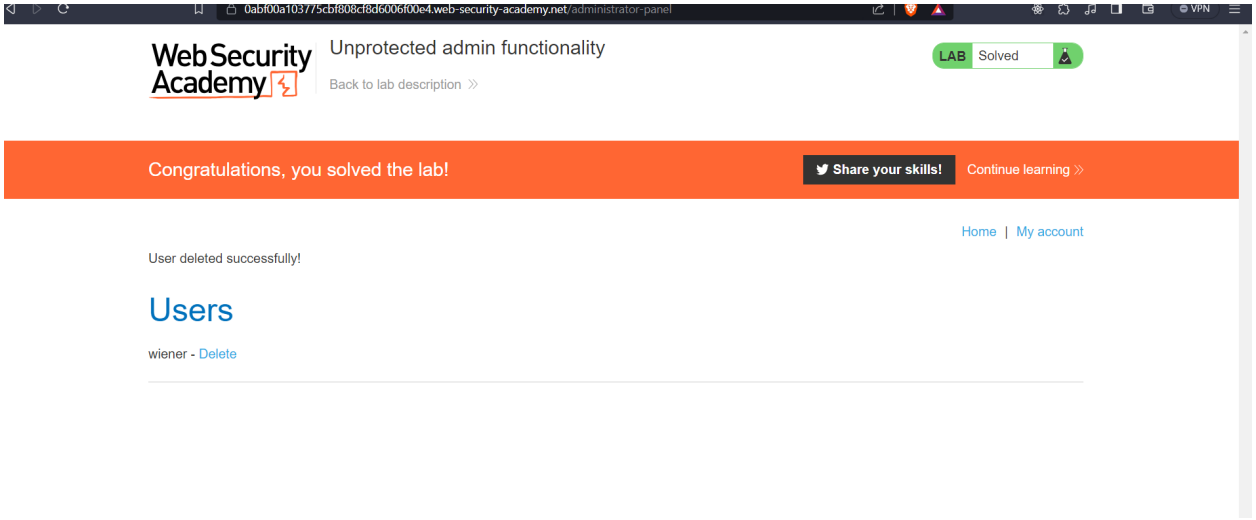
Step-2: view `robots.txt` by appending `/robots.txt` to the lab URL.



Step-3: Copy administrator-panel and paste it to the URL.



Step-4: Delete `carlos`



Recommendations:

"Prevent Broken Access Control vulnerabilities by implementing proper authorization checks, role-based access controls, and continuous security testing."