

# *Malware Detection and Classification*

## **Team Members:**

Naladala Navya

Sanisetty Hema Sagar

Kurra Naveen Abhiram

Vishnubhatla V L Sruta Keerthi

## **Vulnerabilities Report for Practice Website**

### **1. Vulnerability Name:**

**Protection Mechanism Failure** (Web Application Potentially Vulnerable to Clickjacking)

**CWE:** CWE-693

**OWASP Category:** A04:2021 – Broken Access Control

### **Description:**

This weakness covers three distinct situations. A "missing" protection mechanism occurs when the application does not define any mechanism against a certain class of attack. An "insufficient" protection mechanism might provide some defences - for example, against the most common attacks - but it does not protect against everything that is intended. Finally, an "ignored" mechanism occurs when a mechanism is available and in active use within the product, but the developer has not applied it in some code path.

### **Business Impact:**

Protection Mechanism Failure in web applications potentially vulnerable to clickjacking can result in severe business impacts. Clickjacking is a malicious technique where attackers deceive users into performing unintended actions. When protection mechanisms fail, it can lead to unauthorized data exposure, financial losses, and reputational damage. This, in turn, can erode trust, harm customer relationships, and lead to legal consequences, making robust security measures crucial for preserving a business's integrity and success.

**Vulnerability Path:** <http://testfire.net/>

**Vulnerability Parameter:** <https://demo.testfire.net/bank/main.jsp>

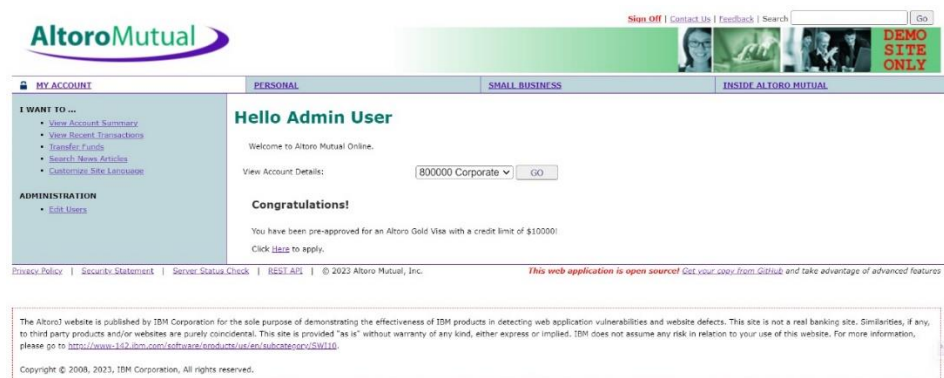
### **Steps to Reproduce:**

Identify a security feature or protection mechanism in the application (e.g., authentication, access control).

Attempt to bypass or defeat the security mechanism.

Observe whether the application allows unauthorized access or circumvention.

If you successfully bypass the protection mechanism, you have reproduced a protection mechanism failure issue.



### **Recommendation:**

Return the X-Frame-Options or Content-Security-Policy (with the 'frame-ancestors' directive) HTTP header with the page's response.

This prevents the page's content from being rendered by another site when using the frame or iframe HTML tags.

## **2. Vulnerability Name:**

**Improper Input Validation (CGI Generic XSS)**

**CWE:** CWE-20

**OWASP Category:** A04:2021 – Injection

### **Description:**

The product receives input or data, but it does not validate or incorrectly validates that the input has the properties that are required to process the data safely and correctly.

### **Business Impact:**

Improper input validation, specifically in the context of CGI scripts, can lead to Cross-Site Scripting (XSS) vulnerabilities. The business impact of this vulnerability includes risks of data

theft, unauthorized access, and damage to a company's reputation, potentially resulting in financial losses, legal liabilities, and loss of customer trust.

**Vulnerability Path:** <http://testfire.net/>

**Vulnerability Parameter:**

<http://testfire.net/search.jsp?query=%27John%27+OR+%271%27%3D%271%27>

### **Steps to Reproduce:**

Identify a user input field in the target application, such as a search box or login form.

Attempt to enter malicious input, like SQL code, HTML tags, or special characters, in the input field.

Submit the input and observe if the application processes it without validation or sanitization.

If the application accepts and processes the input without any validation or sanitization, you have successfully reproduced the issue.



### **Recommendation:**

The recommendation for addressing Improper Input Validation (CGI Generic XSS) is to implement robust input validation and filtering mechanisms in web applications to prevent the execution of malicious cross-site scripting (XSS) attacks, typically by sanitizing and escaping user input data before rendering it on web pages.

### **3. Vulnerability Name:**

**Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') (CGI Generic XSS)**

**CWE:** CWE-79

**OWASP Category:** A04:2021 – Injection

### **Description:**

The product does not neutralize or incorrectly neutralizes user-controllable input before it is placed in output that is used as a web page that is served to other users.

### **Business Impact:**

The business impact of improper neutralization of input during web page generation, commonly known as 'Cross-site Scripting' (XSS), can be significant. It can result in compromised user data, loss of customer trust, reputational damage, and potential legal and regulatory consequences. XSS attacks can enable malicious actors to inject malicious code into web pages, leading to data theft, account hijacking, and other security breaches, which can harm a company's bottom line and credibility.

**Vulnerability Path:** <http://testfire.net/>

### **Vulnerability Parameter:**

<http://testfire.net/search.jsp?query=%27+%3Cscript%3Ealert%28%22XSS+Attack%21%22%29%3B%3C%2Fscript%3E%27>

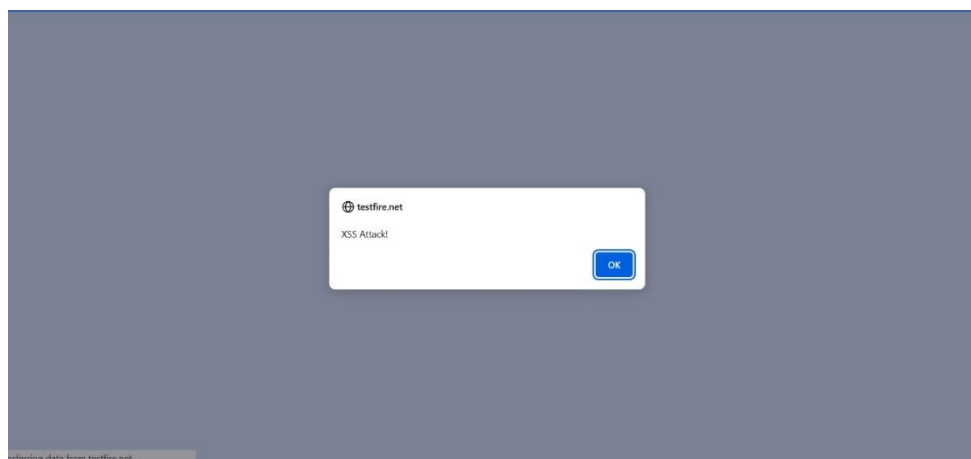
### **Steps to Reproduce:**

Locate a web page that includes user-generated content, such as comments or input fields.

Insert malicious script code, such as JavaScript, into the input field.

Submit the input, and view the page to see if the script executes in the context of another user's session.

If the script runs when viewed by another user, you have successfully reproduced a Cross-site Scripting (XSS) issue.



### **Recommendation:**

The recommended solution for Improper Neutralization of Input During Web Page Generation, commonly known as Cross-site Scripting (XSS), is to implement proper input validation and output encoding to prevent malicious scripts from executing on a website. This involves validating and sanitizing user inputs and encoding output data to ensure that potentially harmful code cannot be injected or executed on the web page.

#### 4. **Vulnerability Name:**

**Improper Neutralization of Script-Related HTML Tags in a Web Page (Basic XSS)** (CGI Generic XSS)

**CWE:** CWE-80

**OWASP Category:** A04:2021 – Injection

#### **Description:**

The product receives input from an upstream component, but it does not neutralize or incorrectly neutralizes special characters such as "<", ">", and "&" that could be interpreted as web-scripting elements when they are sent to a downstream component that processes web pages.

#### **Business Impact:**

The business impact of Improper Neutralization of Script-Related HTML Tags (Basic XSS) in a web page can be severe, leading to compromised user data, damaged reputation, legal consequences, and financial losses. This vulnerability allows attackers to inject malicious scripts into web pages, potentially stealing sensitive information, disrupting services, and undermining customer trust.

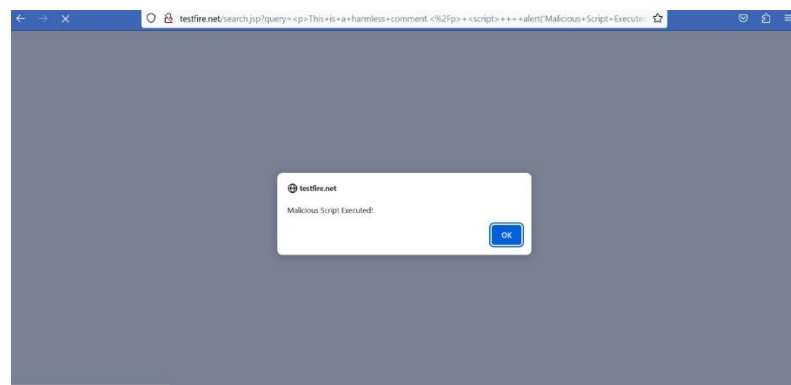
**Vulnerability Path:** <http://testfire.net/>

#### **Vulnerability Parameter:**

<http://testfire.net/search.jsp?query=%3Cp%3EThis+is+a+harmless+comment.%3C%2Fp%3E+%3Cscript%3E++++alert%28%27Malicious+Script+Executed%21%27%29%3B+%3C%2Fscript%3E>

#### **Steps to Reproduce:**

Find a web page with user-generated content, like comments or profile descriptions.  
Inject a malicious script by adding HTML tags containing JavaScript into the content.  
Save or post the content and view the page to check if the script is executed.  
If the script runs in the context of the page, you have successfully reproduced a Basic XSS issue



**Recommendation:**

To mitigate Improper Neutralization of Script-Related HTML Tags in a Web Page (Basic XSS) (CGI Generic XSS) vulnerabilities, ensure proper input validation and output encoding to prevent malicious script injection. Utilize web application firewalls (WAFs) and security best practices, like using Content Security Policy (CSP) headers, to protect against cross-site scripting attacks.

**5. Vulnerability Name:**

**Improper Neutralization of Special Elements in Output Used by a Downstream Component ('Injection')** (CGI Generic XSS)

**CWE:** CWE-74

**OWASP Category:** A04:2021 – Injection

**Description:**

The product constructs all or part of a command, data structure, or record using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify how it is parsed or interpreted when it is sent to a downstream component.

**Business Impact:**

The business impact of "Improper Neutralization of Special Elements in Output Used by a Downstream Component" (CGI Generic XSS) includes potential security breaches, data theft, and reputation damage, leading to financial losses, legal liabilities, and customer trust erosion.

**Vulnerability Path:** <http://testfire.net/>

**Vulnerability Parameter:** <http://testfire.net/login.jsp>

**Steps to Reproduce:**

Identify a user input field that influences a downstream component, such as a search bar or a form.

Attempt to inject malicious code, like SQL, command injection, or other malicious payloads, into the input field.

Observe if the malicious code is executed by the downstream component.

If the malicious code is executed without proper neutralization, you have successfully reproduced the issue.



### **Recommendation:**

The recommendation for addressing Improper Neutralization of Special Elements in Output Used by a Downstream Component ('Injection'), often related to CGI Generic Cross-Site Scripting (XSS), is to implement proper input validation, output encoding, and security controls in your web application to prevent untrusted data from being executed as code. This helps safeguard against XSS attacks, ensuring the integrity and security of the application.

## **6. Vulnerability Name:**

**Insufficiently Protected Credentials** (Web Server Transmits Cleartext Credentials)

**CWE:** CWE-522

**OWASP Category:** A04:2021 – Identification and Authentication Failure

### **Description:**

The product transmits or stores authentication credentials, but it uses an insecure method that is susceptible to unauthorized interception and/or retrieval.

### **Business Impact:**

The business impact of Insufficiently Protected Credentials, such as a web server transmitting cleartext credentials, can result in severe security breaches, data theft, and reputation damage. This vulnerability can lead to unauthorized access to sensitive information and compromise customer trust, potentially causing financial losses and legal consequences.

**Vulnerability Path:** <http://testfire.net/>

**Vulnerability Parameter:** <http://testfire.net/login.jsp>

### **Steps to Reproduce:**

Identify a login or authentication form in the target application.

Attempt to log in using an incorrect username and password.

Capture the network traffic or observe the response.

If the application's response provides clear information indicating which part of the login credentials was incorrect (e.g., "Invalid username"), you may have found insufficiently protected credentials.

**Recommendation:**

The recommendation for "Insufficiently Protected Credentials (Web Server Transmits Cleartext Credentials)" is to implement secure communication protocols like HTTPS and employ encryption to safeguard sensitive login credentials transmitted between the client and the web server.

**7. Vulnerability Name:**

**Improper Handling of an Exceptional Condition** (Web Server Transmits Cleartext Credentials)

**CWE:** CWE-928

**OWASP Category:** A04:2021 – Vulnerable and Outdated Components

**Description:**

software weakness where exceptional conditions, such as errors or unexpected events, are not handled or managed appropriately within the application's code. This weakness can manifest in various ways depending on the specific application and programming language being used.

**Business Impact:**

The business impact of improper handling of an exceptional condition where a web server transmits cleartext credentials can lead to significant security vulnerabilities, potential data breaches, damage to the organization's reputation, legal liabilities, and financial losses due to remediation and compensation.

**Vulnerability Path:** <http://testfire.net/>

**Steps to Reproduce:**

This CWE does not correspond to a specific security issue that can be easily reproduced in the same way as the others. It relates to improper error handling and exception management, which can manifest in various ways depending on the application. To identify such issues, you would need to analyse the application's code and error-handling mechanisms.

**Recommendation:**

The recommended solution for improper handling of an exceptional condition where a web server transmits cleartext credentials is to implement secure communication protocols, such



as HTTPS, to encrypt the transmission of sensitive information and prevent unauthorized access.

## 8. **Vulnerability Name:**

**Improper Handling of an Exceptional Condition** (Web Server Transmits Cleartext Credentials)

**CWE:** CWE-930

**OWASP Category:** A02:2021 – Cryptographic Failures

### **Description:**

software weakness where an application makes the incorrect assumption that certain data is immutable (i.e., it cannot be modified or changed) when, in reality, this data can be altered by external entities, including attackers, or even by the program itself.

### **Business Impact:**

The business impact of improper handling of an exceptional condition, such as a web server transmitting cleartext credentials, can result in severe security breaches, damage to the organization's reputation, potential legal repercussions, and financial losses due to compromised user data and trust.

**Vulnerability Path:** <http://testfire.net/>

### **Steps to Reproduce:**

This CWE relates to verifying cryptographic signatures properly and securely. Reproducing this CWE would involve analysing an application's cryptographic signature verification process to determine if it's performed incorrectly or inadequately. This may require code analysis and testing within the specific context of the application.

### **Recommendation:**

The recommended solution for improper handling of an exceptional condition where a web server transmits cleartext credentials is to implement secure encryption protocols (e.g., HTTPS) to protect sensitive data during transmission and to handle exceptions gracefully, providing informative error messages without exposing sensitive information.

## 9. **Vulnerability Name:**

**Insecure Direct Object Reference (IDOR)**

**CWE:** cwe-639

**OWASP Category:** A01:2021 – Broken Access Control

## Description:

The system's authorization functionality does not prevent one user from gaining access to another user's data or record by modifying the key value identifying the data.

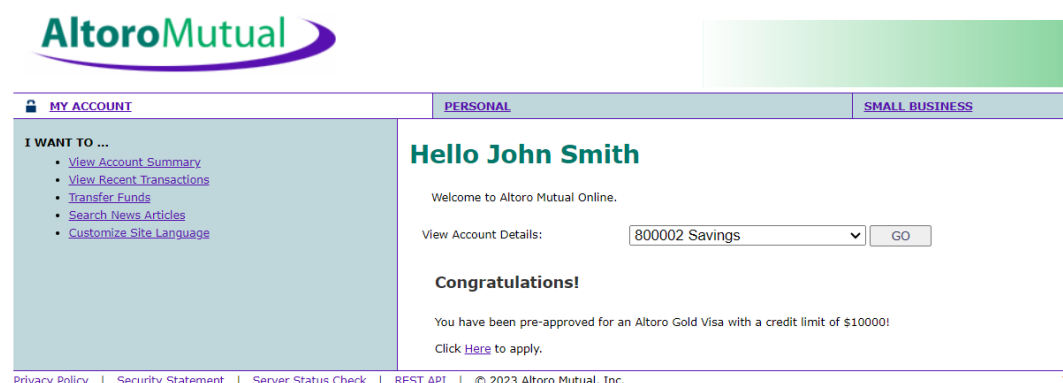
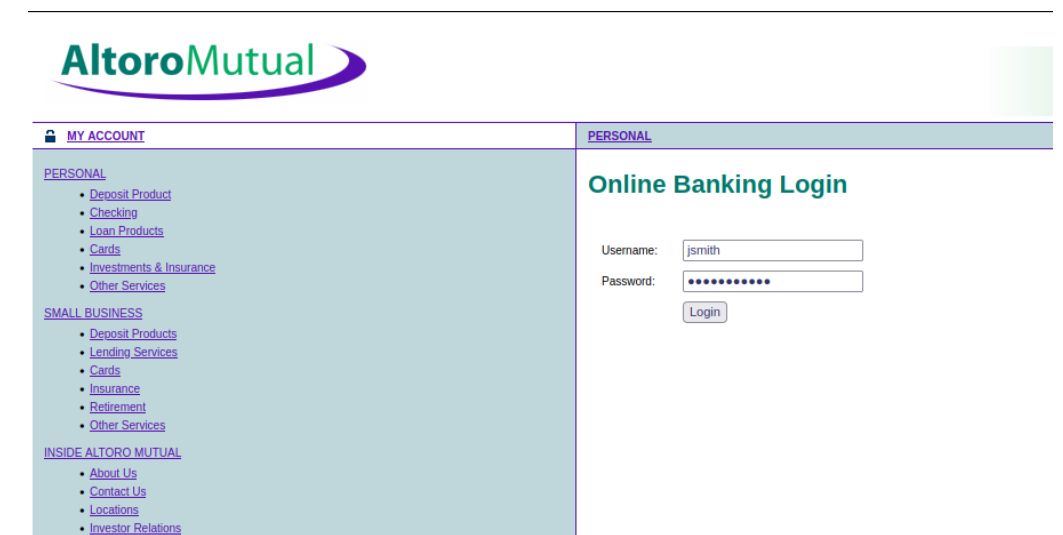
## Business Impact:

IDOR can lead to unauthorized access to sensitive data or resources, potentially resulting in data breaches, privacy violations, financial losses, and damage to an organization's reputation. It can also lead to legal and regulatory consequences, impacting the overall trust and confidence in the business.

## Vulnerability Path: <http://testfire.net/>

## Steps to Reproduce:

Navigate into the given URL and login using john smith credentials. Click on “Go” to view John Smith’s savings. Change the listAccount=800002 to 800003 to view account history of other customers.



demo.testfire.net/bank/showAccount?listAccounts=800002

**AltoroMutual**

MY ACCOUNT PERSONAL SMALL BUSINESS

I WANT TO ...

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

ADMINISTRATION

- Edit Users

### Account History - 800002

Balance Detail		Amount
800000 Corporate	Select Account	
Ending balance as of 10/16/23 10:05 AM		-\$1999543407407875070.00
Available balance		-\$1999543407407875070.00

#### 10 Most Recent Transactions

Date	Description	Amount
2023-10-16	Withdrawal	-\$200.00
2023-10-16	Withdrawal	-\$200.00
2023-10-16	Deposit	\$200.00
2023-10-16	Withdrawal	-\$1234.00
2023-10-16	Withdrawal	-\$1000.00
2023-10-16	Withdrawal	-\$821029.00

#### Credits

Account	Date	Description	Amount
1001160140	12/29/2004	Paycheck	1200
1001160140	01/12/2005	Paycheck	1200
1001160140	01/29/2005	Paycheck	1200
1001160140	02/12/2005	Paycheck	1200
1001160140	03/01/2005	Paycheck	1200
1001160140	03/15/2005	Paycheck	1200

demo.testfire.net/bank/showAccount?listAccounts=800003

**AltoroMutual**

MY ACCOUNT PERSONAL SMALL BUSINESS

I WANT TO ...

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

ADMINISTRATION

- Edit Users

### Account History - 800003

Balance Detail		Amount
800000 Corporate	Select Account	
Ending balance as of 10/16/23 10:07 AM		\$1066020696702566500000.00
Available balance		\$1066020696702566500000.00

#### 10 Most Recent Transactions

Date	Description	Amount
2023-10-16	Deposit	\$1234.00
2023-10-16	Withdrawal	-\$1234.00
2023-10-16	Deposit	\$18446744073709552000.00
2023-10-16	Withdrawal	-\$18446744073709552000.00
2023-10-16	Deposit	\$4294967297.00
2023-10-16	Withdrawal	-\$4294967297.00

#### Credits

Account	Date	Description	Amount
1001160140	12/29/2004	Paycheck	1200
1001160140	01/12/2005	Paycheck	1200
1001160140	01/29/2005	Paycheck	1200
1001160140	02/12/2005	Paycheck	1200
1001160140	03/01/2005	Paycheck	1200
1001160140	03/15/2005	Paycheck	1200

### Recommendation:

The recommendation or solution for Insecure Direct Object Reference (IDOR) is to implement proper access controls and validation mechanisms to ensure that users can only access or modify the data and resources they are authorized to, and not directly reference internal

objects. This can be achieved through role-based access control, session management, and thorough input validation.

## 10. Vulnerability Name: Information Disclosure

**CWE:** cwe-200

**OWASP Category:** A03:2021 - Cryptographic Failures

### **Description:**

The product exposes sensitive information to an actor that is not explicitly authorized to have access to that information.

### **Business Impact:**

Information disclosure jeopardizes privacy, competitive standing, and trust, potentially resulting in legal actions, financial losses, and reputational harm, undermining an organization's security, prosperity, and image.

### **Vulnerability Path:**

[https://demo.testfire.net/index.jsp?content=inside\\_jobs.html](https://demo.testfire.net/index.jsp?content=inside_jobs.html)

### **Steps to Reproduce:**

Navigate to the URL and then the details are clearly visible.

The screenshot shows a web browser displaying the AltoroMutual website. The URL in the address bar is [demo.testfire.net/index.jsp?content=inside\\_jobs.html](https://demo.testfire.net/index.jsp?content=inside_jobs.html). The website has a navigation bar with links: Sign Off, Contact Us, Feedback, Search, and a Go button. Below the navigation bar is a banner with the AltoroMutual logo and a 'DEMO SITE ONLY' watermark. The main content area is titled 'Current Job Openings' and includes a table of job listings. The table has columns for Group, Date Posted, and Title. The job listings are as follows:

Group	Date Posted	Title
Administration	Oct-23-2006	<a href="#">Executive Assistant</a>
Consumer Banking	Oct-19-2006	<a href="#">Teller</a>
Customer Service	Oct-26-2006	<a href="#">Customer Service Representative</a>
Marketing	Oct-25-2006	<a href="#">Loyalty Marketing Program Manager</a>
Risk Management	Oct-17-2006	<a href="#">Operational Risk Manager</a>
Sales	Oct-24-2006	<a href="#">Mortgage Lending Account Executive</a>

Below the table, there is a disclaimer: 'Altoro Mutual and its affiliates recruit and hire qualified candidates without regard to race, religion, color sex, sexual orientation, age, national origin, ancestry, citizenship, veteran or disability status or any factor prohibited by law, and as such affirms in policy and practice to support and promote the concept of equal employment opportunity and affirmative action, in accordance with all applicable federal, state and municipal laws. Candidates must possess the right to work in the United States, as it is not the practice of Altoro Mutual to sponsor individuals for work visas.'

The footer of the website includes links to Privacy Policy, Security Statement, Server Status Check, and REST API, along with a copyright notice for 2008, 2023, IBM Corporation. A red banner at the bottom states: 'This web application is open source! Get your copy from GitHub and take advantage of advanced features'.

## **Recommendation:**

Information disclosure refers to the unintentional release of sensitive or confidential data. To mitigate this risk, implement robust data security measures, including encryption, access controls, and employee training, and regularly audit and update security protocols to prevent unauthorized access and data leaks.

## **11.Vulnerability Name:**

### **Web Application Potentially Vulnerable to Clickjacking**

**CWE:** cwe-451

**OWASP Category:** A04:2021 – Insecure Design

## **Description:**

The user interface (UI) does not properly represent critical information to the user, allowing the information - or its source - to be obscured or spoofed. This is often a component in phishing attacks.

## **Business Impact:**

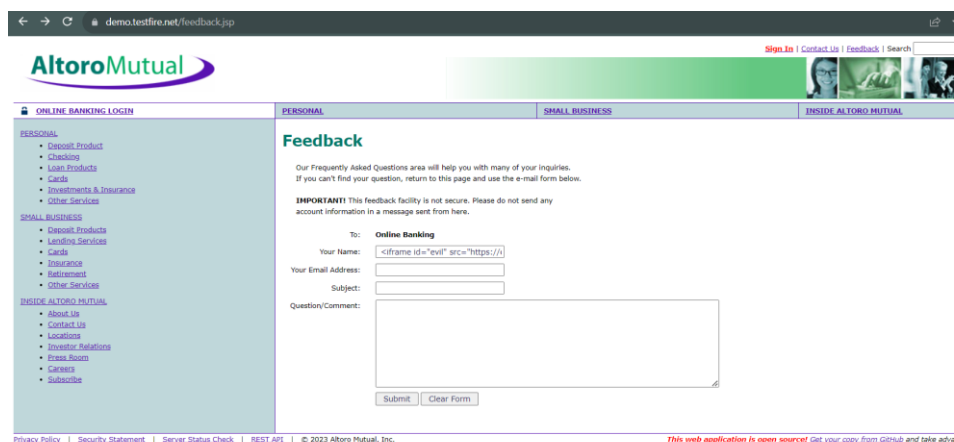
The Impact of Clickjacking The hacker has several ways they can use the redirected clicks for their own gain. A common form of clickjacking involves mirroring a login and password form on a website.

## **Vulnerability Path:**

<https://demo.testfire.net/feedback.jsp>

## **Steps to Reproduce:**

Navigate to the above-mentioned URL. Enter the payload “<iframe id="evil" src=<https://evil.com> sandbox="allow-forms"></iframe>.”



The screenshot shows a web browser window with the address bar displaying `demo.testfire.net/feedback.jsp`. The page is the AltoroMutual feedback form. The 'Your Name' field contains the payload: `<iframe id="evil" src="https://A`. The 'Your Email Address' and 'Subject' fields are empty. The 'Question/Comment' field is a large text area. The 'Submit' and 'Clear Form' buttons are at the bottom of the form. The footer of the page includes links for Privacy Policy, Security Statement, Server Status Check, and REST API, along with a copyright notice for Altoro Mutual, Inc. and a statement that the web application is open source.



## **Recommendation:**

Information disclosure refers to the unintentional release of sensitive or confidential data. To mitigate this risk, implement robust data security measures, including encryption, access controls, and employee training, and regularly audit and update security protocols to prevent unauthorized access and data leaks.