TASK: 3

Vulnerabilities Report On

“WEB APPLICATION SECURITY”

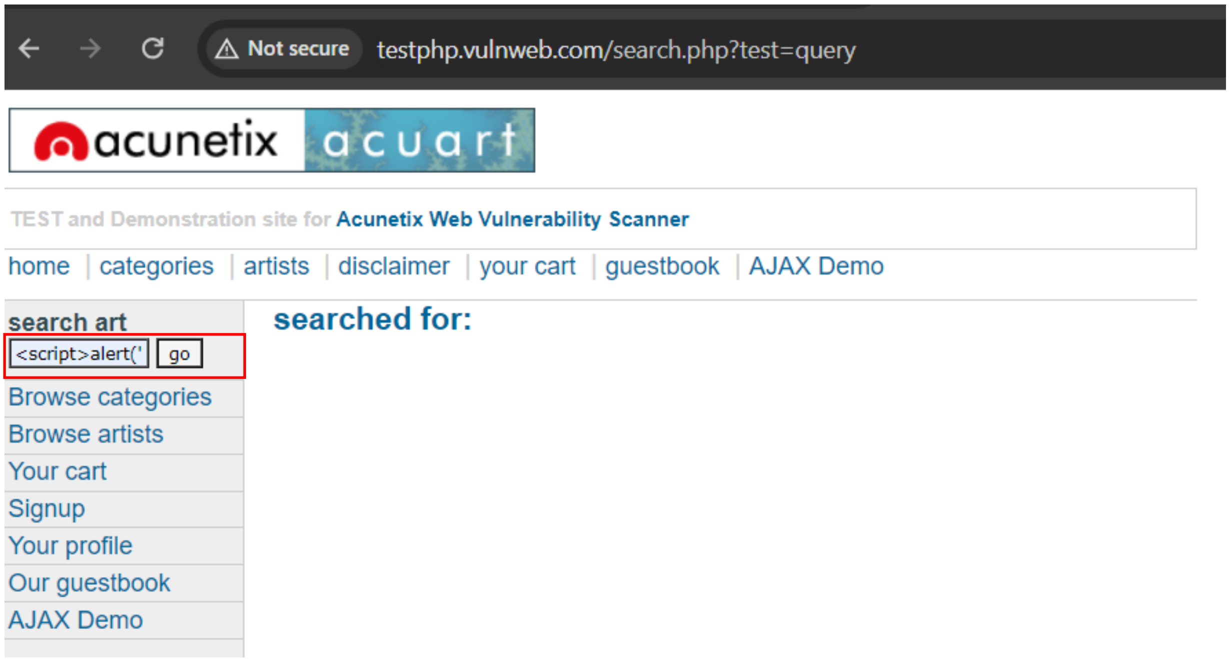
Testing Website 1: http://testphp.vulnweb.com/

Description: The website "http://testphp.vulnweb.com/" is a place where you can practice finding and fixing security problems in web applications. It has purposely made PHP scripts and apps with vulnerabilities like SQL injection, Authentication Bypass and Cross-Site Scripting. It's a safe way for developers and cybersecurity folks to learn about common web threats.

Vulnerability 1: XSS (cross-site scripting)

Description: Cross-Site Scripting (XSS) is a security issue where attackers inject harmful code, like JavaScript, into web pages viewed by others. This lets them steal data or take control of accounts. There are three types: Reflected (in the URL), Stored (in the database), and DOM-based (in the browser).

Payload: <script>alert("You Have Been Hacked!!!")</script>



Not secure testphp.vulnweb.com/search.php?test=query

TEST and Demonstration site for Acunetix Web Vulnerability Scanner

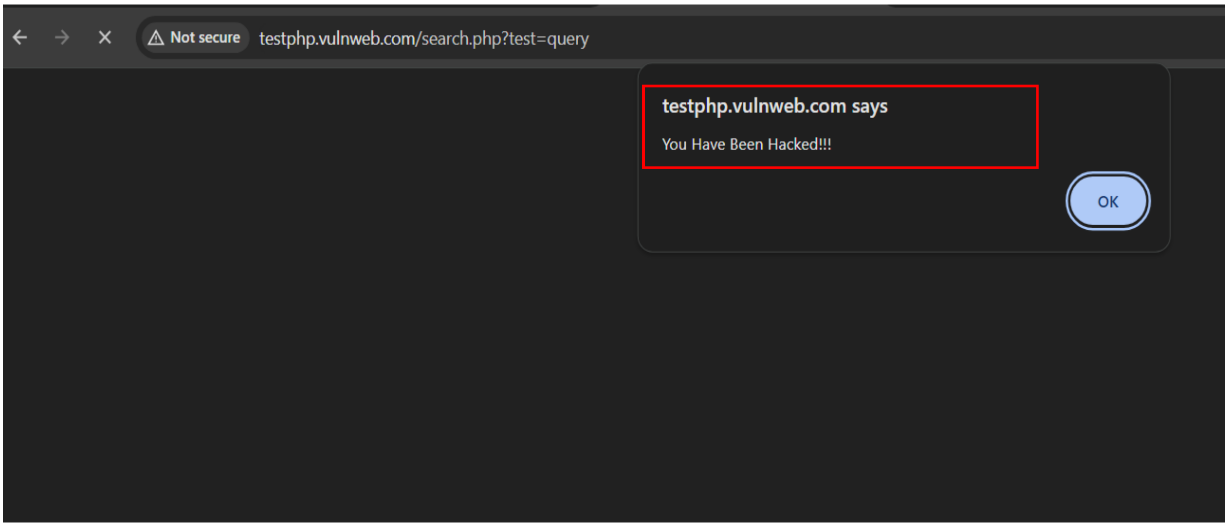
home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo

search art

<script>alert(' go

searched for:

We injected payload <script>alert("You Have Been Hacked!!!")</script> into a test vulnerable web page, when we click on go it causes a pop-up alert to appear with the text "You Have Been Hacked".



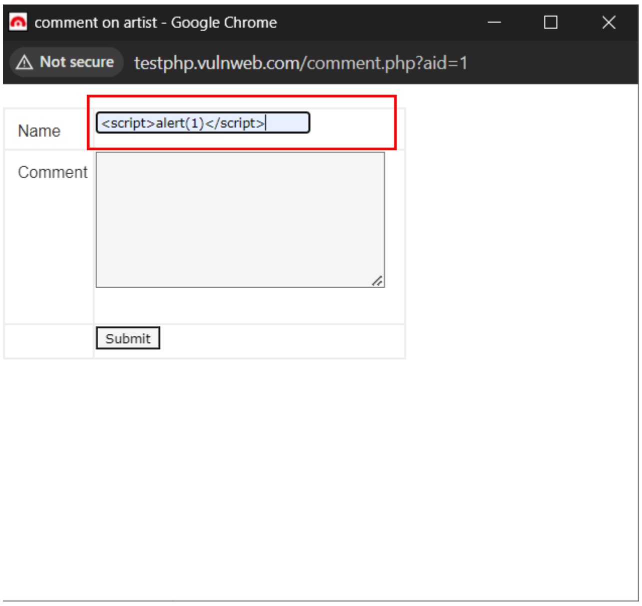
testphp.vulnweb.com says

You Have Been Hacked!!!

Pop-up shown in image

This demonstrates how attackers can insert harmful code into websites to steal information or perform other malicious actions on users' browsers.

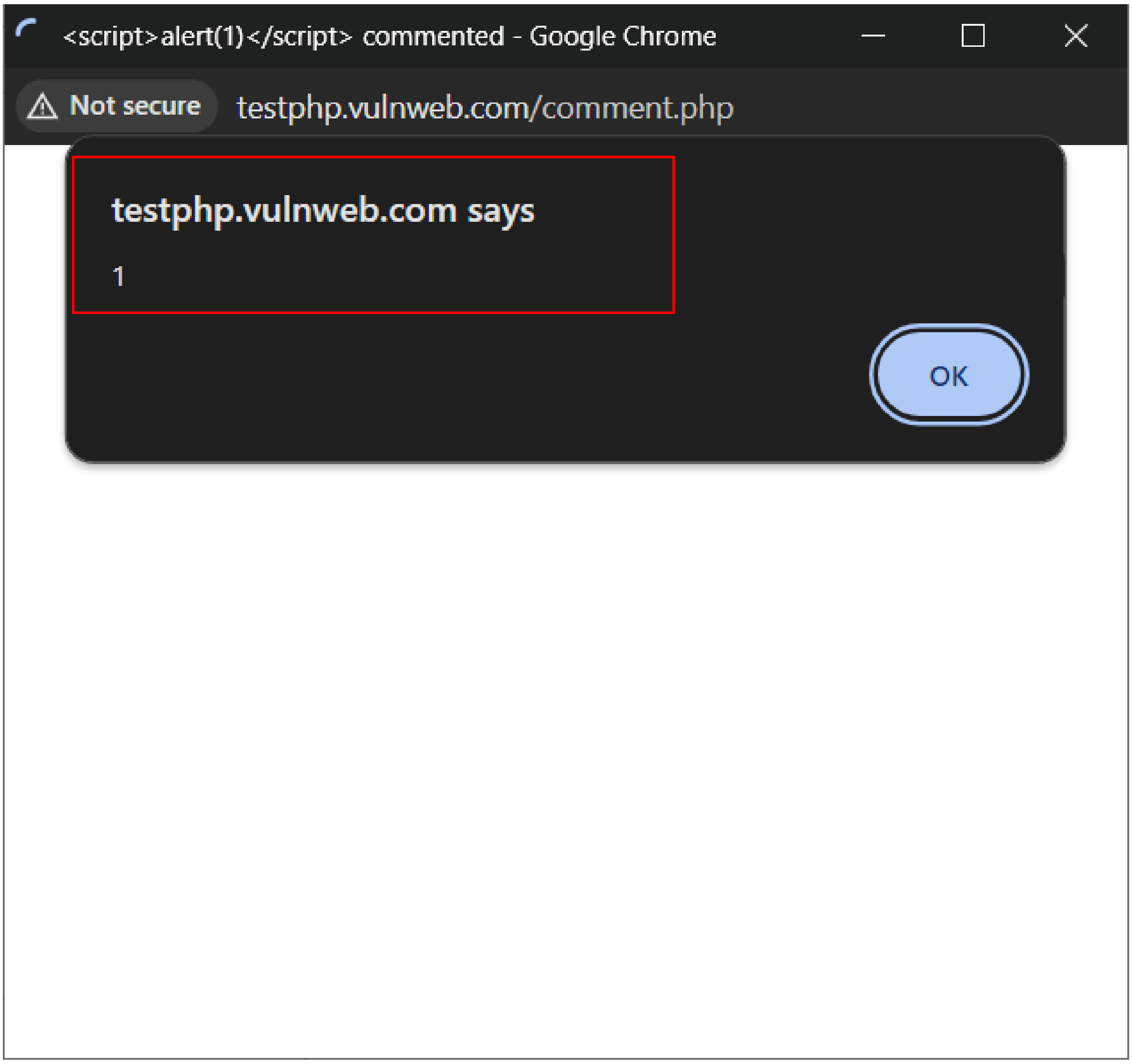
XSS IN COMMENT BOX:In the website, there's a comment box where users can leave comments on posted pictures. I typed in a basic JavaScript code and hit enter. After that, pop-up window appeared on the screen.



comment on artist - Google Chrome
Not secure
testphp.vulnweb.com/comment.php?aid=1
Submit

Name <script>alert(1)</script>

Comment



<script>alert(1)</script> commented - Google Chrome

testphp.vulnweb.com/comment.php

testphp.vulnweb.com says

1

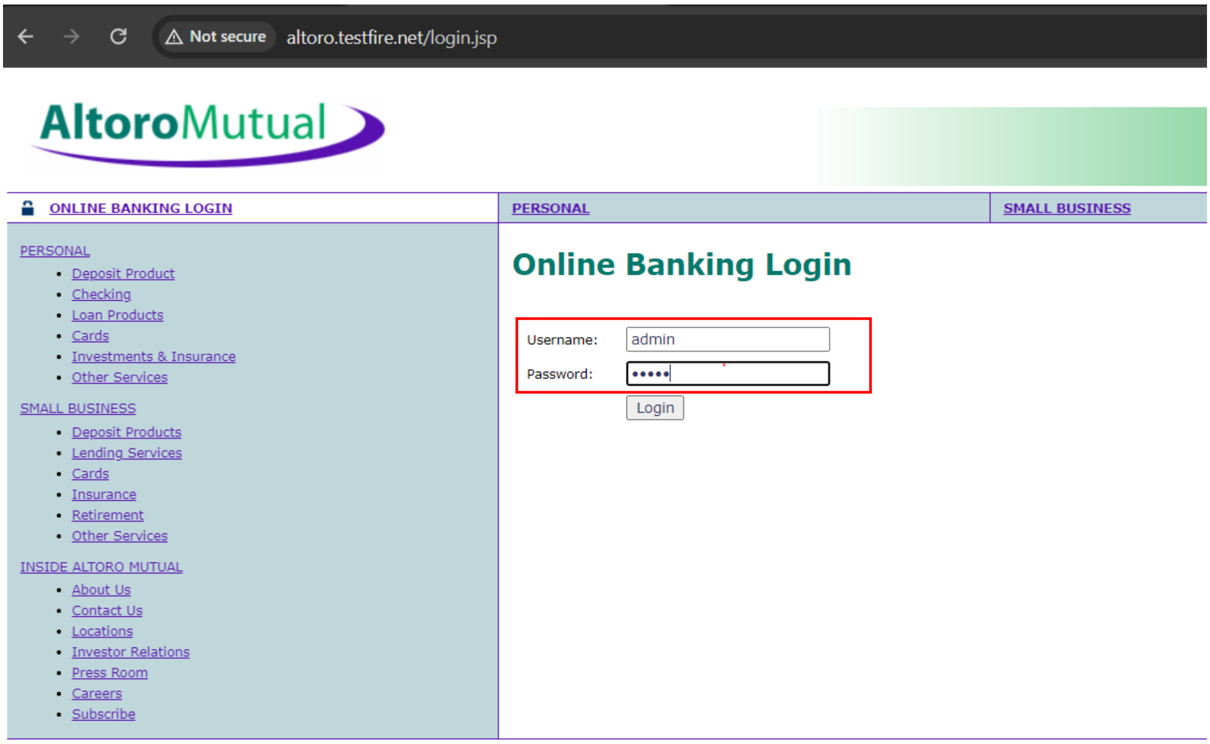
After execution of the code the show the pop-up on the screen.

Testing Website 2: http://altoro.testfire.net/

Vulnerability 2: Bypass by Default credentials

Description: Bypassing authentication using default credentials means accessing a system or website without using the required login information because the default usernames and passwords haven't been changed. For eg. user-user, admin-admin, admin-admin123, user-password.

Here we can try to login with default username and password which is admin-admin lets see what will happened…

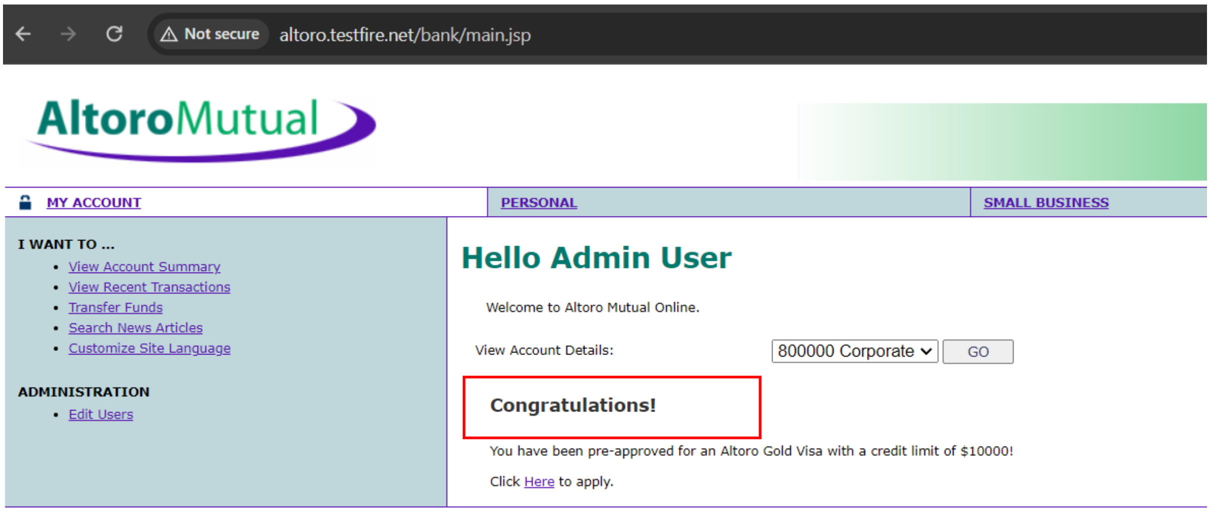


Online Banking Login

Username: admin

Password: •••••

After click on login button the server sends reply Congratulation!. It means we are login successfully.



Hello Admin User

Welcome to Altoro Mutual Online.

View Account Details: 800000 Corporate

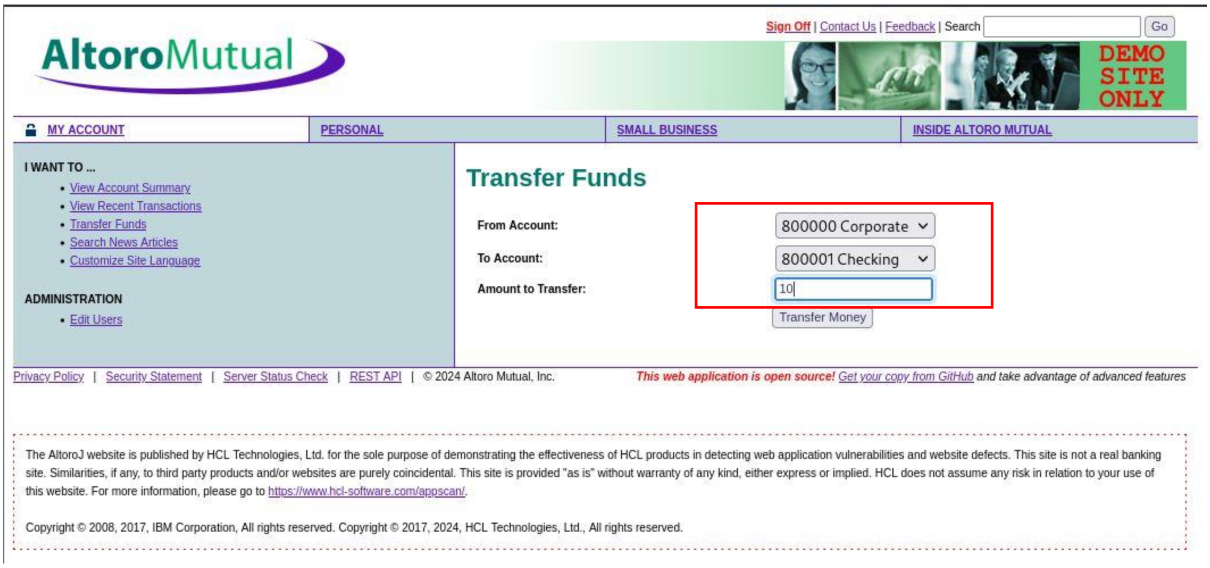
Congratulations!

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

Click Here to apply.

Vulnerability 3: Request Manipulation

Description: In this vulnerability, the attacker uses a tool like Burp Suite to intercept the requests sent from the user's browser to the server. They then modify these requests to carry out malicious actions, like bypassing authentication or injecting harmful code into the application.



Transfer Funds

From Account: 800000 Corporate

To Account: 800001 Checking

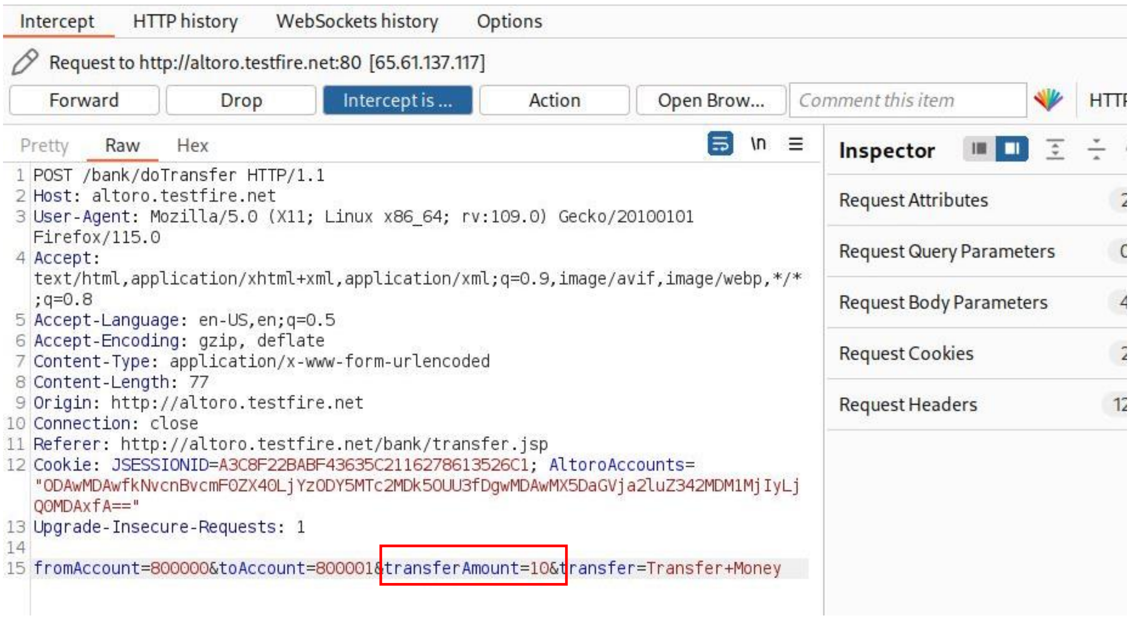
Amount to Transfer: 10

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

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. 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. 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/appscan/.

Copyright © 2008, 2017, IBM Corporation, All rights reserved. Copyright © 2017, 2024, HCL Technologies, Ltd., All rights reserved.

In the upper image we are sending some (10) money from one account to another account.

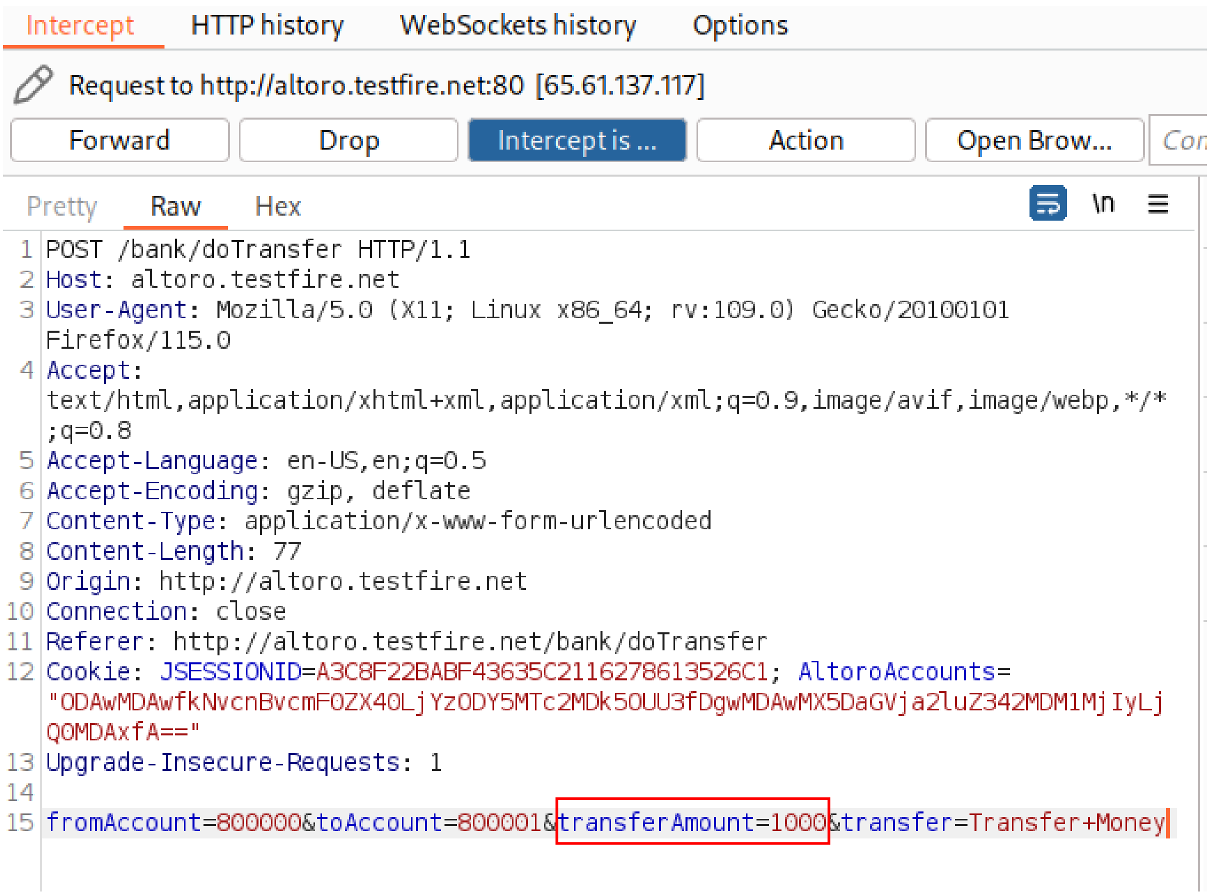


Request to http://altoro.testfire.net:80 [65.61.137.117]

```
POST /bank/doTransfer HTTP/1.1
Host: altoro.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
Content-Type: application/x-www-form-urlencoded
Content-Length: 77
Origin: http://altoro.testfire.net
Connection: close
Referer: http://altoro.testfire.net/bank/transfer.jsp
Cookie: JSESSIONID=A3C8F22BABF43635C2116278613526C1; AltoroAccounts=
"ODAwMDAwfkNvcnBvcmF0ZX40LjYzODY5MTc2MDk5OUU3fDgwMDAwMX5DaGVja2luZ342MDM1MjIyLj
Q0MDAxfA=="
Upgrade-Insecure-Requests: 1

fromAccount=800000&toAccount=800001&transferAmount=10&transfer=Transfer+Money
```

In image user sending $10 and attacker caputes the request using burp suite and change the transaction amount.

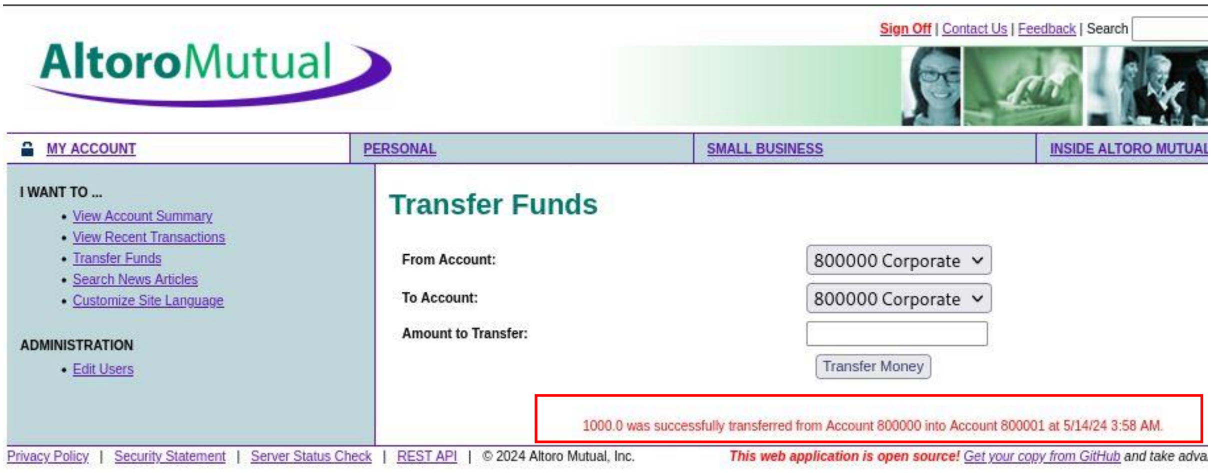


Request to http://altoro.testfire.net:80 [65.61.137.117]

```
POST /bank/doTransfer HTTP/1.1
Host: altoro.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
Content-Type: application/x-www-form-urlencoded
Content-Length: 77
Origin: http://altoro.testfire.net
Connection: close
Referer: http://altoro.testfire.net/bank/doTransfer
Cookie: JSESSIONID=A3C8F22BABF43635C2116278613526C1; AltoroAccounts=
"ODAwMDAwfkNvcnBvcmF0ZX40LjYzODY5MTc2MDk5OUU3fDgwMDAwMX5DaGVja2luZ342MDM1MjIyLj
QOMDAxfA=="
Upgrade-Insecure-Requests: 1

fromAccount=800000&toAccount=800001&transferAmount=1000&transfer=Transfer+Money
```

After changing the amount $1000 was debited from the users account, we were seen the red line was highlighted that 1000$ was successfully transferred Image are given below.



Transfer Funds

From Account: 800000 Corporate

To Account: 800000 Corporate

Amount to Transfer:

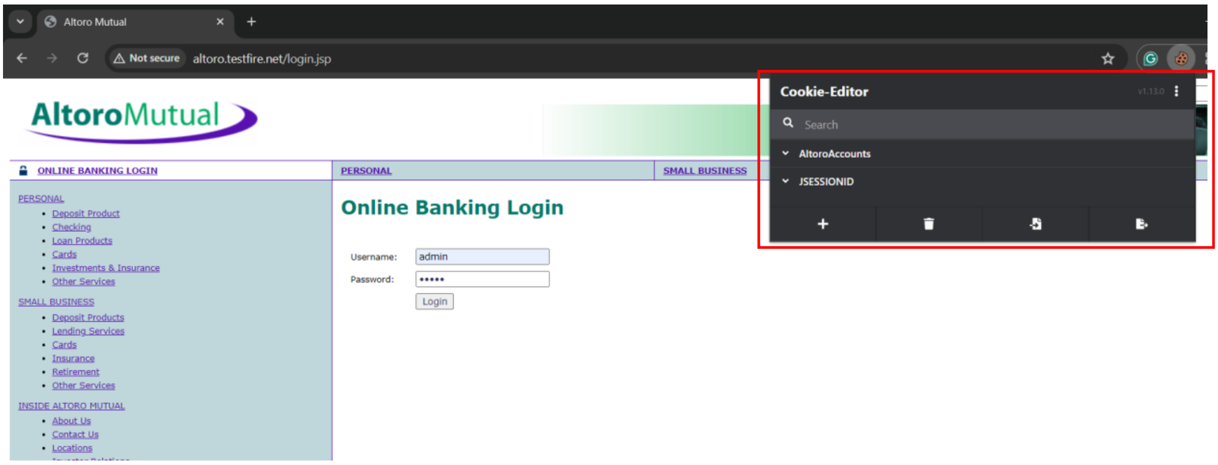
1000.0 was successfully transferred from Account 800000 into Account 800001 at 5/14/24 3:58 AM.

Vulnerability 4: Session Hijacking

Description: Session hijacking occurs when an unauthorized person takes control of an ongoing online session, gaining access to sensitive information or performing actions as if they were the legitimate user. It's like someone stealing the keys to your car while you're driving, allowing them to take over control.

We use Cookie Editor Extension & testfire.net Website

In following image we capture the login cookies in cookie editor (Chrome Browser) extension and after that export that cookie in JSON file.



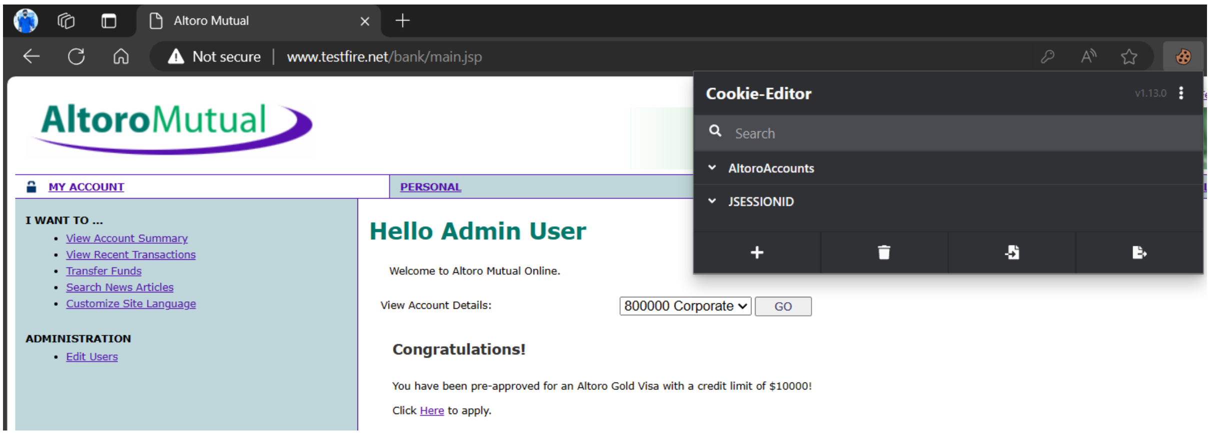
Online Banking Login

Username: admin

Password: •••••

Login

After that we going in other browser (Microsoft Edge) and import that JSON file in cookie editor and refresh the page then they automatically login as Admin user. There is no need to type username and password in browser they automatically login from (Chrome Browser) cookies.



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!

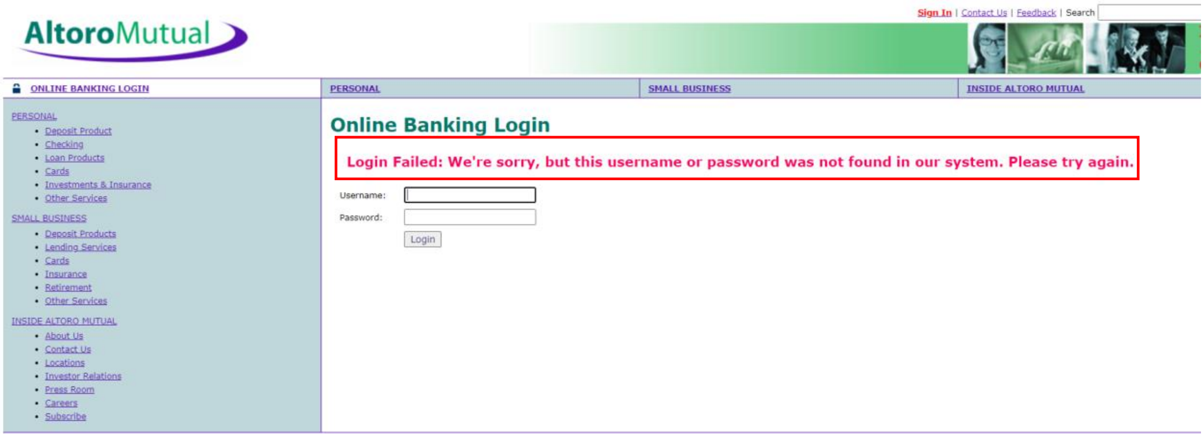
Click Here to apply.

Vulnerability 3: SQL injection

Description: SQL injection is a common web security vulnerability that occurs when an attacker inserts malicious SQL code into input fields on a website, exploiting vulnerabilities in the website's code to execute unauthorized SQL commands. These commands can manipulate the database, steal data, or even delete entire tables.

For e.g. **' OR '1**

Website name: testfire.net



Online Banking Login

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

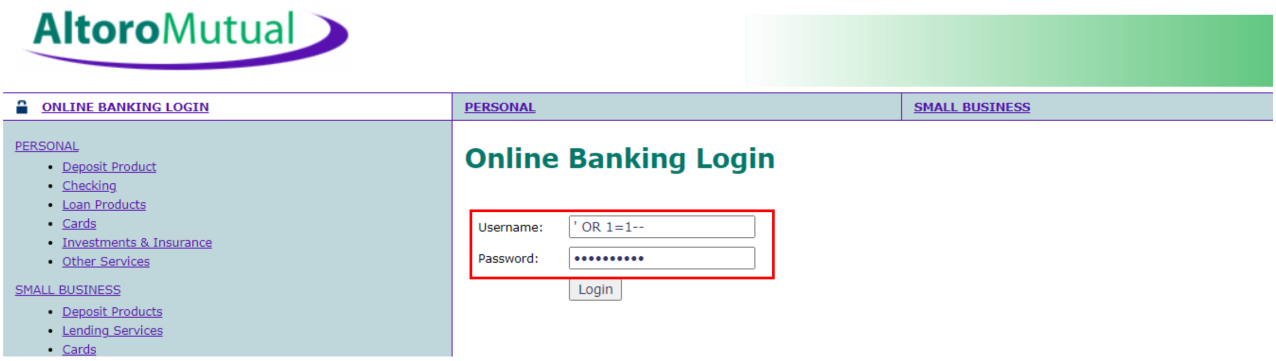
Username:

Password:

We trying Username = hello Password = hello but they show login failed

In a SQL injection attack on a website's login page, an attacker can enter a malicious SQL query into the input field instead of a username and password. If the website is vulnerable, the query is executed, granting the attacker access without needing valid credentials.

We are trying SQL Injection = **' OR 1=1--**

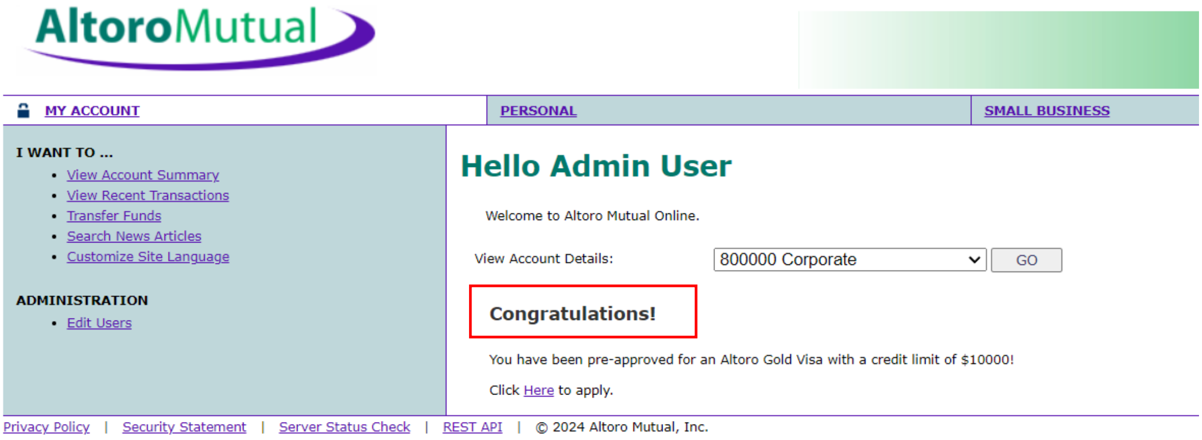


Online Banking Login

Username: ' OR 1=1--

Password: ••••••••••

After the injecting sql query they show the login successful (Congratulations!). We login as Admin user through the sql injection it means this is vulnerable for sql injection.



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.

Privacy Policy | Security Statement | Server Status Check | REST API | © 2024 Altoro Mutual, Inc.