Assignment 4

Name : Bhargava Sai Jetti
College : Dr.Lankapalli Bullayya College
Regd.No : 721128805292
Date : 15/03/2024

Step 1: OWASP Top 10 Vulnerabilities Overview

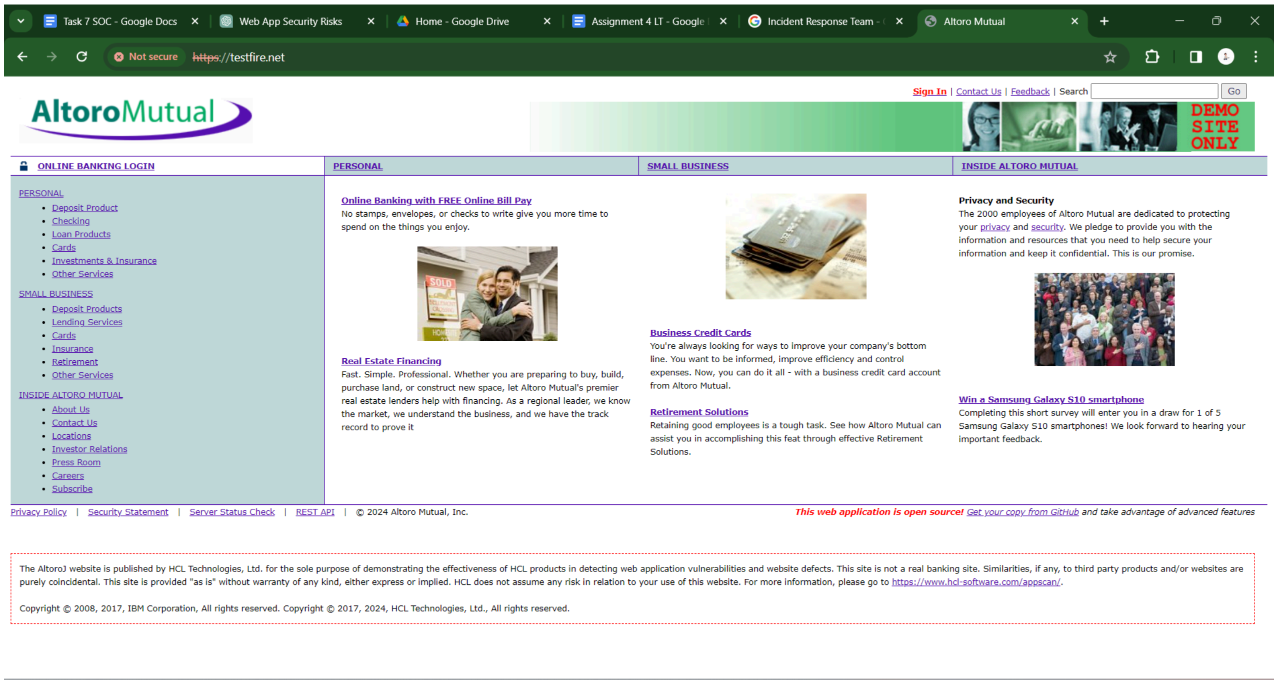
Here's an overview of these vulnerabilities and their potential impact on web application security:

- Injection Attacks: Injection flaws, such as SQL injection and NoSQL injection, occur when untrusted data is sent to an interpreter as part of a command or query. Attackers can exploit these vulnerabilities to execute malicious commands, access unauthorized data, or manipulate the application's behavior.
- Broken Authentication: Weaknesses in authentication mechanisms can lead to various attacks, such as credential stuffing, session hijacking, and brute force attacks. Attackers can exploit these vulnerabilities to gain unauthorized access to user accounts, compromise sensitive data, or impersonate legitimate users.
- Sensitive Data Exposure: Failure to adequately protect sensitive data, such as passwords, credit card numbers, or personal information, can result in data breaches and privacy violations. Attackers can exploit these vulnerabilities to steal confidential information, commit identity theft, or conduct financial fraud.
- XML External Entities (XXE): XML external entity attacks occur when an application parses XML input containing external entity references in an unsafe manner. Attackers can exploit these vulnerabilities to read sensitive files, perform SSRF (Server-Side Request Forgery) attacks, or launch Denial of Service (DoS) attacks.
- Broken Access Control: Inadequate access control mechanisms can allow unauthorized users to access restricted functionality or sensitive data. Attackers can exploit these vulnerabilities to escalate privileges, bypass authorization checks, or manipulate access controls to gain unauthorized access.
- Security Misconfigurations: Improperly configured security settings, such as default passwords, unnecessary services, or excessive permissions, can create opportunities for attackers to exploit vulnerabilities and compromise the application or underlying infrastructure.
- Cross-Site Scripting (XSS): XSS vulnerabilities occur when an application includes untrusted data in a web page without proper validation or escaping. Attackers can exploit these vulnerabilities to execute malicious scripts in the context of legitimate users, steal session cookies, deface websites, or redirect users to malicious sites.
- Insecure Deserialization: Insecure deserialization vulnerabilities arise when an application deserializes untrusted data without proper validation or integrity checks. Attackers can exploit these vulnerabilities to execute arbitrary code, perform object injection attacks, or tamper with application logic.

- Using Components with Known Vulnerabilities: Integrating third-party libraries, frameworks, or components with known security vulnerabilities can expose the application to exploitation. Attackers can exploit these vulnerabilities to compromise the application, escalate privileges, or gain unauthorized access to the underlying system.
- Insufficient Logging and Monitoring: Inadequate logging and monitoring mechanisms can hinder the detection and response to security incidents. Attackers can exploit these deficiencies to conduct stealthy attacks, evade detection, or maintain persistence within the environment.

Step 2: Altoro Mutual Website Analysis

Below is the Altoro Mutual website, this website a vulnerable website specially used for demo testing and learning purposes.
Altro Mutual is a financial institution's website that offers various services such as banking, investment, and insurance products to its customers. It provides features like user registration, login, payment portal, contact forms, and other interactive functionalities. The website allows users to manage their accounts, make transactions, and interact with customer support.



Sign In | Contact Us | Feedback | Search Go

ONLINE BANKING LOGIN | PERSONAL | SMALL BUSINESS | INSIDE ALTORO MUTUAL

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

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 Mutual's premier real estate lenders help with financing. As a regional leader, we know the market, we understand the business, and we have the track record to prove it

Business Credit Cards
You're always looking for ways to improve your company's bottom line. You want to be informed, improve efficiency and control expenses. Now, you can do it all - with a business credit card account from Altoro Mutual.

Retirement Solutions
Retaining good employees is a tough task. See how Altoro Mutual can assist you in accomplishing this feat through effective Retirement Solutions.

Privacy and Security
The 2000 employees of Altoro Mutual are dedicated to protecting your privacy and security. We pledge to provide you with the information and resources that you need to help secure your information and keep it confidential. This is our promise.

Win a Samsung Galaxy S10 smartphone
Completing this short survey will enter you in a draw for 1 of 5 Samsung Galaxy S10 smartphones! We look forward to hearing your important feedback.

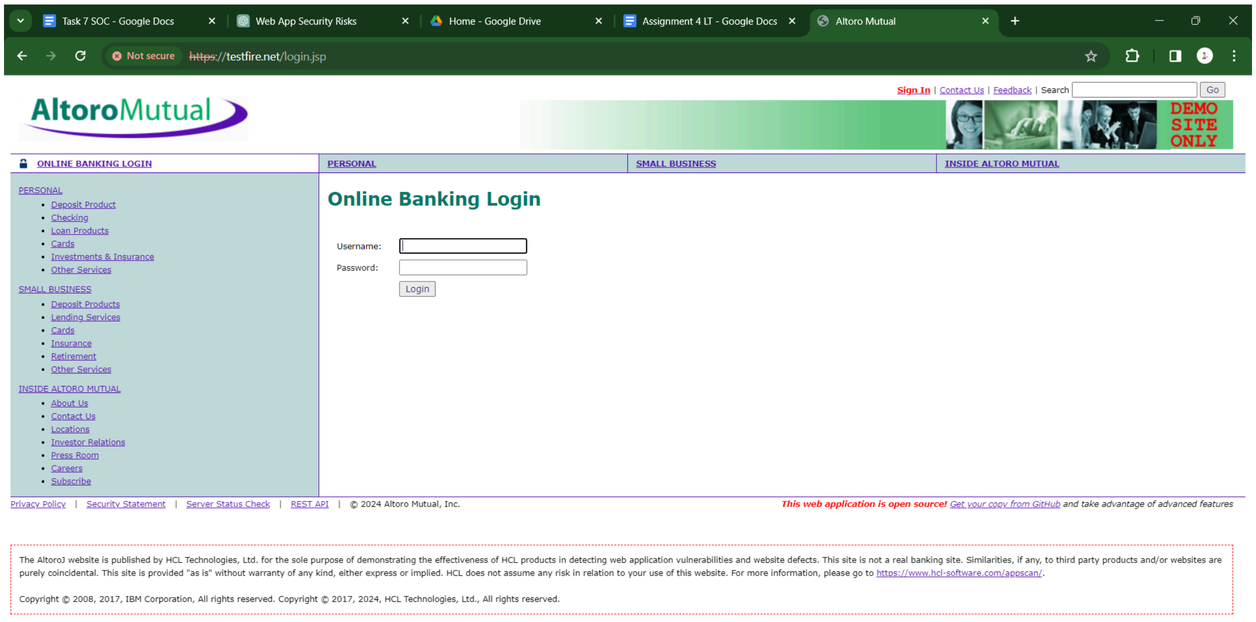
Privacy Policy | Security Statement | Server Status Check | REST API | © 2024 Altoro Mutual, Inc.

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

The AltoroJ website is published by 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.

This is the Login Page where we can try logging in later



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

Online Banking Login

Username:

Password:

Login

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

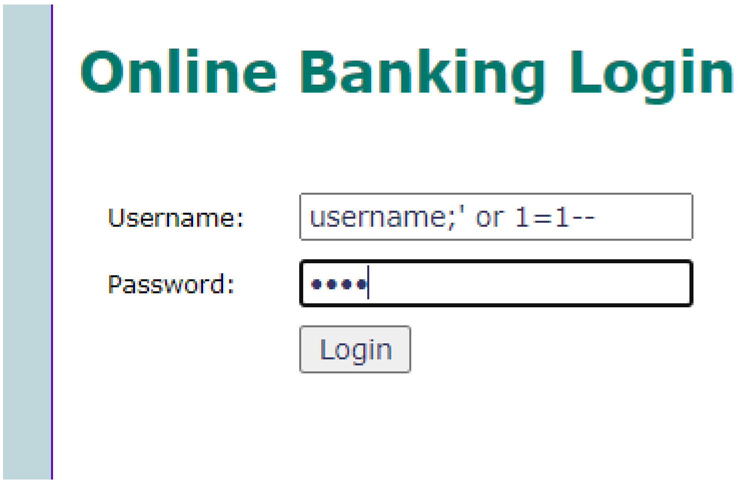
This web application is open source! Get your copy from GitHub and take advantage of advanced features

The AltoroJ website is published by 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.

Step 3: Vulnerability Identification Report

We will now try using "username;' or 1=1–" as username and password as "1234".



Online Banking Login

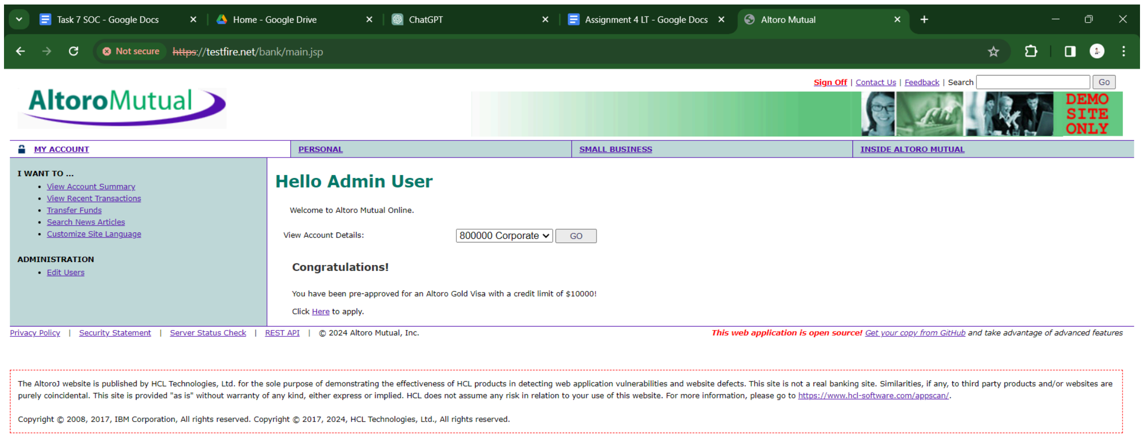
Username: username;' or 1=1--

Password: ••••

Login

There we can see below that we have logged in successfully into an unknown wrong user account and we can access their data, and this is a vulnerability. This vulnerability is known as "Sensitive data exposure".
Sensitive Data Exposure occurs when an application fails to adequately protect sensitive information such as passwords, credit card numbers, or personal identifiable information (PII). If an attacker gains access to this data, they could potentially display users' credentials or other

sensitive information. This vulnerability often arises due to inadequate encryption, improper storage of credentials, or insufficient access controls.



Task 7 SOC - Google Docs | Home - Google Drive | ChatGPT | Assignment 4 LT - Google Docs | Altoro Mutual

Not secure ~~https~~://testfire.net/bank/main.jsp

Sign Off | Contact Us | Feedback | Search Go

AltoroMutual

DEMO SITE ONLY

MY ACCOUNT | PERSONAL | SMALL BUSINESS | INSIDE ALTORO MUTUAL

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.

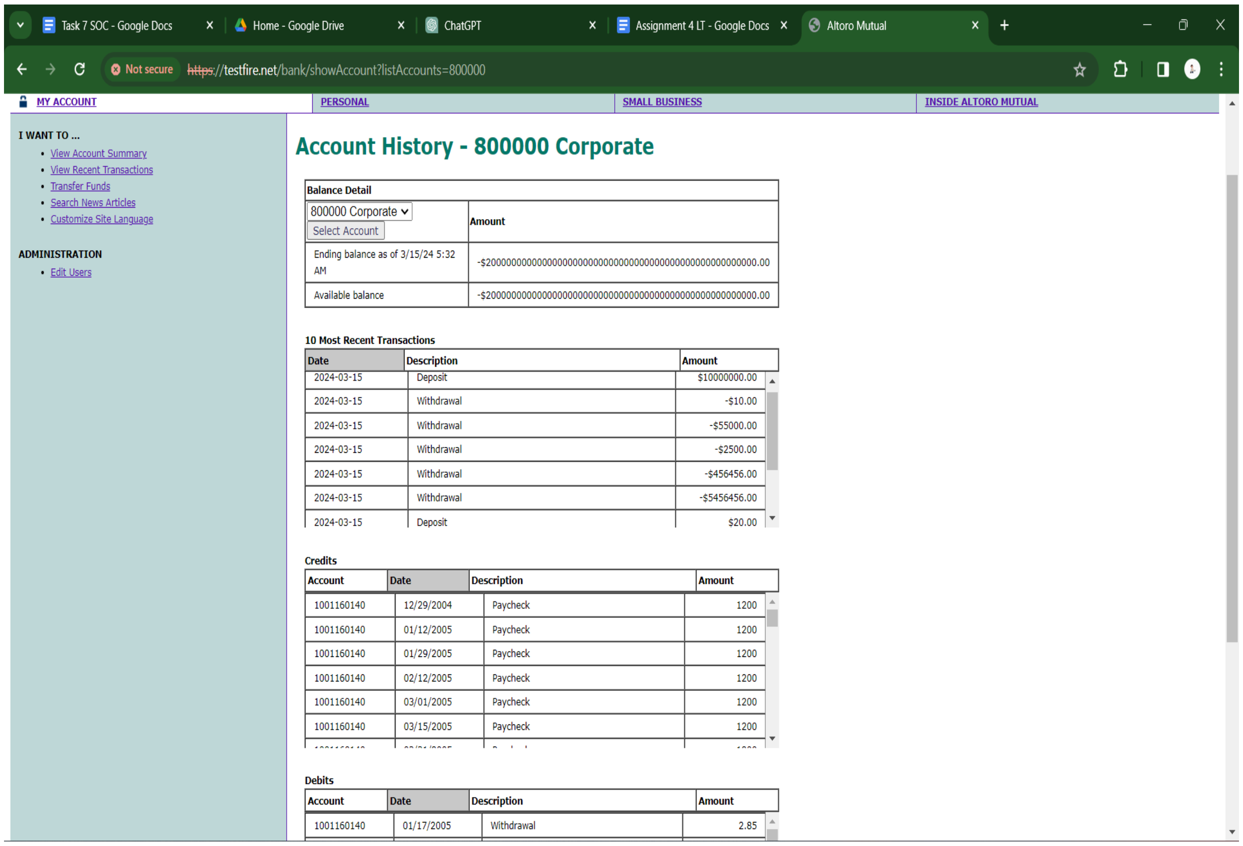
Privacy Policy | Security Statement | Server Status Check | REST API | © 2024 Altoro Mutual, Inc.

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.

Also we could also access the payment portal of that user as follows and this vulnerability, This is typically known as "Insecure Direct Object References (IDOR)" according to OWASP Top 10 Vulnerabilities. This occurs when an application exposes internal implementation objects (e.g., files, directories, database records) to users without proper authorization checks. In the context of accessing other users' credentials, an IDOR vulnerability would allow an attacker to manipulate a parameter or URL to access another user's authentication credentials or sensitive data directly, bypassing the intended access controls.



Task 7 SOC - Google Docs | Home - Google Drive | ChatGPT | Assignment 4 LT - Google Docs | Altoro Mutual

Not secure ~~https~~://testfire.net/bank/showAccount?listAccounts=800000

MY ACCOUNT | PERSONAL | SMALL BUSINESS | INSIDE ALTORO MUTUAL

I WANT TO ...

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

ADMINISTRATION

- Edit Users

Account History - 800000 Corporate

Balance Detail

800000 Corporate Select Account	Amount
Ending balance as of 3/15/24 5:32 AM	-$2000000000000000000000000000000000000000000000000000000.00
Available balance	-$2000000000000000000000000000000000000000000000000000000.00

10 Most Recent Transactions

Date	Description	Amount
2024-03-15	Deposit	$10000000.00
2024-03-15	Withdrawal	-$10.00
2024-03-15	Withdrawal	-$55000.00
2024-03-15	Withdrawal	-$2500.00
2024-03-15	Withdrawal	-$456456.00
2024-03-15	Withdrawal	-$5456456.00
2024-03-15	Deposit	$20.00

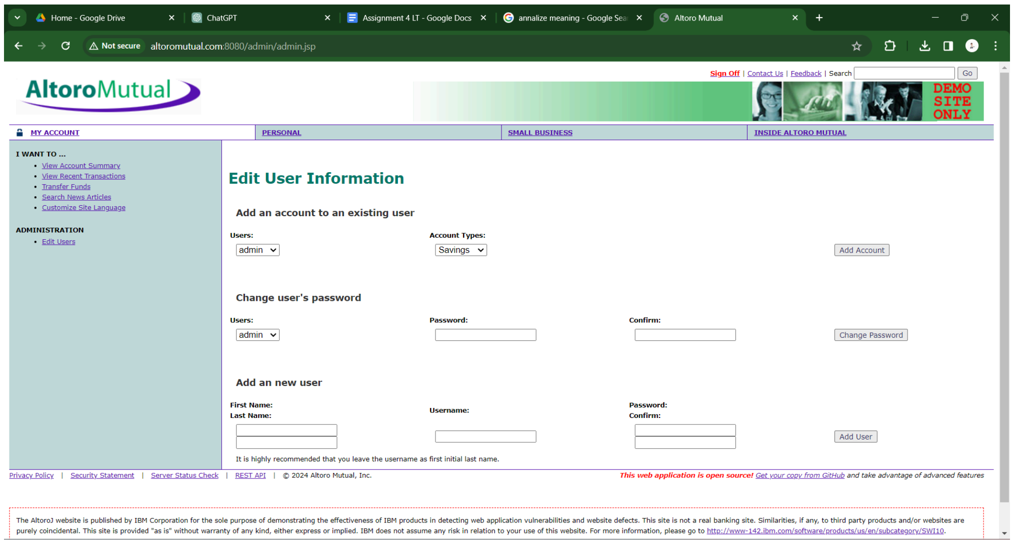
Credits

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

Debits

Account	Date	Description	Amount
1001160140	01/17/2005	Withdrawal	2.85

And we can now also edit the user details without their concern, which is a vulnerability.
This is known as broken access control vulnerability
Broken Authentication refers to vulnerabilities in authentication mechanisms that allow attackers to compromise user accounts, session tokens, or passwords. This can lead to unauthorized access to accounts, manipulation of user data, or impersonation of legitimate users.



altoromutual.com:8080/admin/admin.jsp

Sign Off | Contact Us | Feedback | Search Go

AltoroMutual

DEMO SITE ONLY

MY ACCOUNT | PERSONAL | SMALL BUSINESS | INSIDE ALTORO MUTUAL

I WANT TO ...
- View Account Summary
- View Recent Transactions
- Transfer Funds
- Search News Articles
- Customize Site Language

ADMINISTRATION
- Edit Users

Edit User Information

Add an account to an existing user

Users: admin
Account Types: Savings
Add Account

Change user's password

Users: admin
Password:
Confirm:
Change Password

Add an new user

First Name:
Last Name:
Username:
Password:
Confirm:
Add User

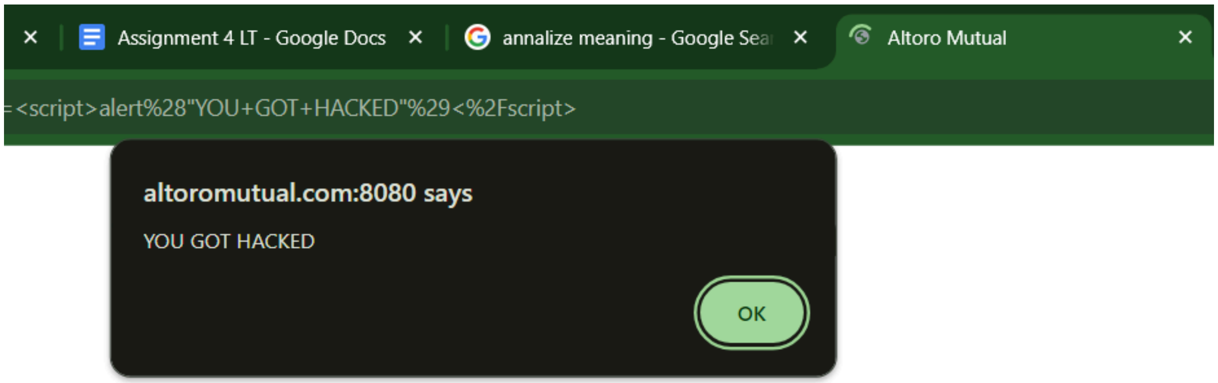
It is highly recommended that you leave the username as first initial last name.

Privacy Policy | Security Statement | Server Status Check | REST API | © 2024 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.

Now we will write a small script as follows to inject it into the site **<script>alert("YOU GOT HACKED")</script>** in the search bar. There as we could see we have injected the code. So this is known as cross site scripting .
Cross-Site Scripting (XSS) is a vulnerability that allows attackers to inject malicious scripts into web pages viewed by other users. These scripts can execute in the context of the user's browser, allowing the attacker to steal session cookies, perform actions on behalf of the user, or deface the website.



=<script>alert%28"YOU+GOT+HACKED"%29<%2Fscript>

altoromutual.com:8080 says

YOU GOT HACKED

Mitigation for the vulnerabilities encountered -

Sensitive Data Exposure:

- Encrypt sensitive data at rest and in transit using strong cryptographic algorithms.
- Implement access controls to ensure that only authorized users can access sensitive data.
- Minimize data retention by storing only the necessary information for the required duration.
- Regularly audit and monitor systems for any potential exposure of sensitive data.

Insecure Direct Object References (IDOR):

- Implement proper access controls and authorization mechanisms to ensure that users can only access resources they are authorized to access.
- Use indirect object references (e.g., mapping database primary keys to unique identifiers) to prevent direct access to internal objects.
- Validate and sanitize user inputs to prevent manipulation of object references through parameters or URLs.
- Implement server-side checks to verify that users are authorized to access requested resources.

Broken Authentication:

- Enforce strong password policies, including requirements for length, complexity, and expiration.
- Implement multi-factor authentication (MFA) to add an extra layer of security to user authentication.
- Use secure session management practices, including random session IDs, session expiration, and secure cookie attributes.
- Protect against common attacks like brute force, credential stuffing, and session fixation by implementing account lockout mechanisms, rate limiting, and session rotation.

Cross-Site Scripting (XSS):

- Implement proper input validation and output encoding to sanitize user inputs and prevent injection of malicious scripts.
- Use Content Security Policy (CSP) headers to restrict the sources from which content, including scripts, can be loaded.
- Employ web application firewalls (WAFs) to detect and block XSS attacks in real-time.
- Educate developers on secure coding practices and conduct regular security reviews and penetration testing to identify and mitigate XSS vulnerabilities.

Step 4: Vulnerability Exploitation Demonstration

Let's demonstrate how SQL injection attacks can be used to extract sensitive information from a database and how cross-site scripting (XSS) attacks can be used to execute malicious scripts in users' browsers.

SQL Injection Attack -
Suppose we have a simple login form on a website where users enter their username and password :

```
<form action="/login" method="post">
  Username: <input type="text" name="username"><br>
  Password: <input type="password" name="password"><br>
  <input type="submit" value="Login">
</form>
```

Now, let's assume that the backend code handling the login process has a SQL query that looks like this,

```
query = "SELECT * FROM users WHERE username='" + user_input_username + "' AND
password='" + user_input_password + "'";
```

If the input is not properly validated or sanitized, an attacker can exploit this vulnerability by entering a malicious SQL statement as the username, such as;

```
username: ' OR '1'='1';--
```

The resulting query becomes,

```
SELECT * FROM users WHERE username='' OR '1'='1';--' AND
password='user_input_password'
```

This effectively bypasses the password check because '1'='1' always evaluates to true. The attacker can now log in without a valid password.

Step 5: Mitigation Strategy Proposal

Here are the mitigation strategies prioritizing high-risk vulnerabilities for each of the mentioned categories:
Sensitive Data Exposure:

- Encryption: Implement strong encryption mechanisms for sensitive data both in transit and at rest.
- Access Controls: Enforce strict access controls to ensure that only authorized users have access to sensitive data.

- Data Masking: Use techniques such as data masking to limit exposure of sensitive information, especially in non-production environments.

Insecure Direct Object References:

- Access Control Checks: Implement proper access control checks to ensure that users can only access resources they are authorized to access.
- Indirect Object References: Use indirect object references (such as a mapping table) to access sensitive data rather than directly exposing internal identifiers.

Broken Authentication:

- Multi-Factor Authentication (MFA): Implement MFA to add an extra layer of security and mitigate the risk of unauthorized access.
- Secure Password Policies: Enforce strong password policies, including requirements for length, complexity, and regular password changes.
- Session Management: Implement secure session management practices, including proper session timeouts and secure session token handling.

Cross-Site Scripting (XSS):

- Input Validation and Output Encoding: Implement proper input validation and output encoding to sanitize user inputs and prevent XSS attacks.
- Content Security Policy (CSP): Utilize CSP headers to restrict the sources from which content, including scripts, can be loaded, thereby mitigating the impact of XSS attacks.
- Regular Code Reviews and Security Audits: Conduct regular code reviews and security audits to identify and fix potential XSS vulnerabilities in the application code.

These strategies provide a prioritized approach to mitigating high-risk vulnerabilities associated with sensitive data exposure, insecure direct object references, broken authentication, and cross-site scripting. Implementing these measures can significantly enhance the security posture of the application and protect against potential exploitation of these vulnerabilities.