**TASK: 3**

**Vulnerabilities Report On**

**“WEB APPLICATION SECURITY”**

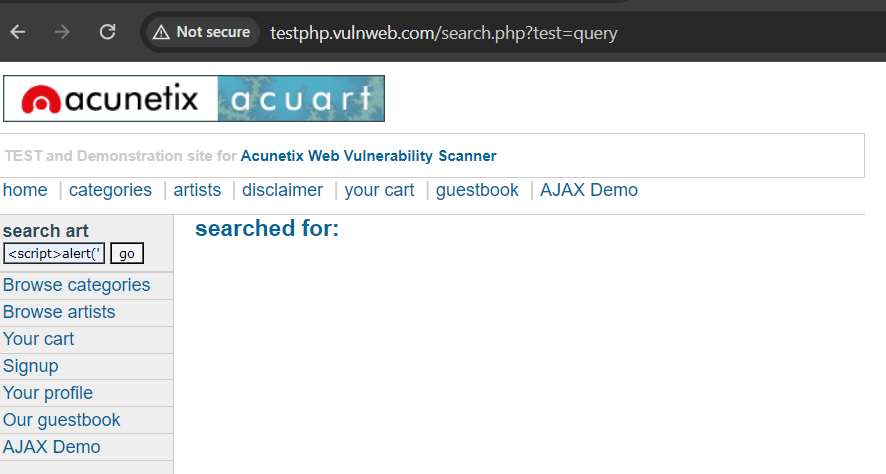
**Testing Website 1:** [**http://testphp.vulnweb.com/**](http://testphp.vulnweb.com/)

**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.

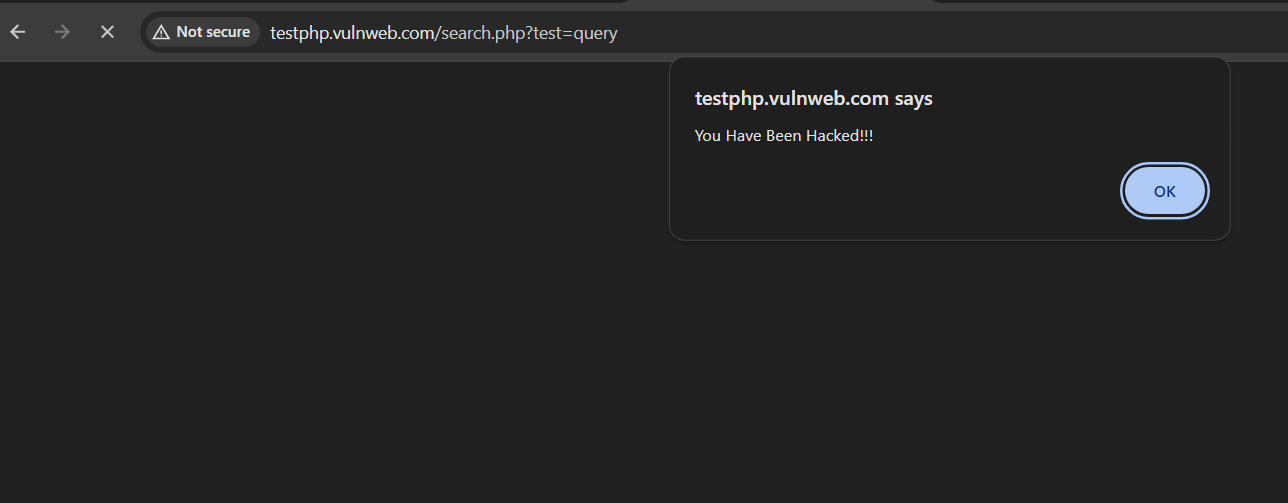
**Vulnerability 1: XSS (cross-site scripting)**

**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>

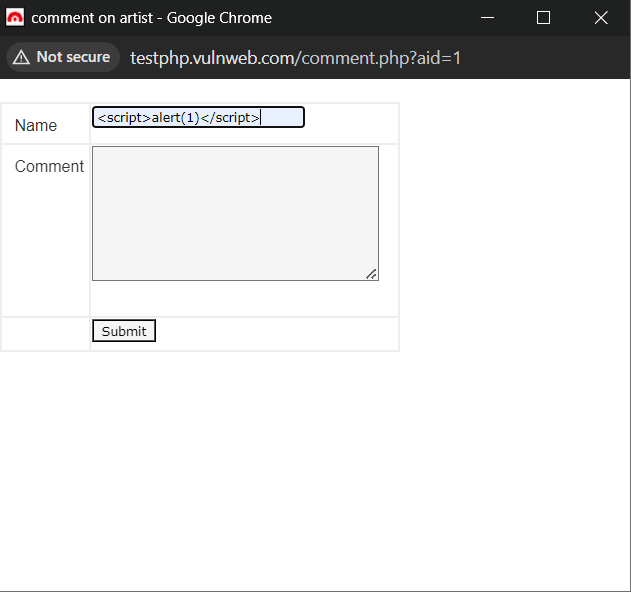


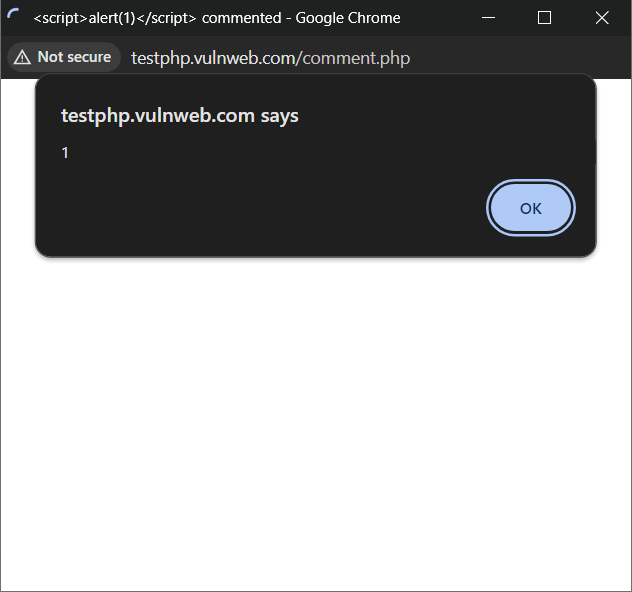
We injected payload <script>alert("You Have Been Hacked!!!")</script> into a test vulnerable web page, when we click on go it causes a pop-up alert to appear with the text "You Have Been Hacked".



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.



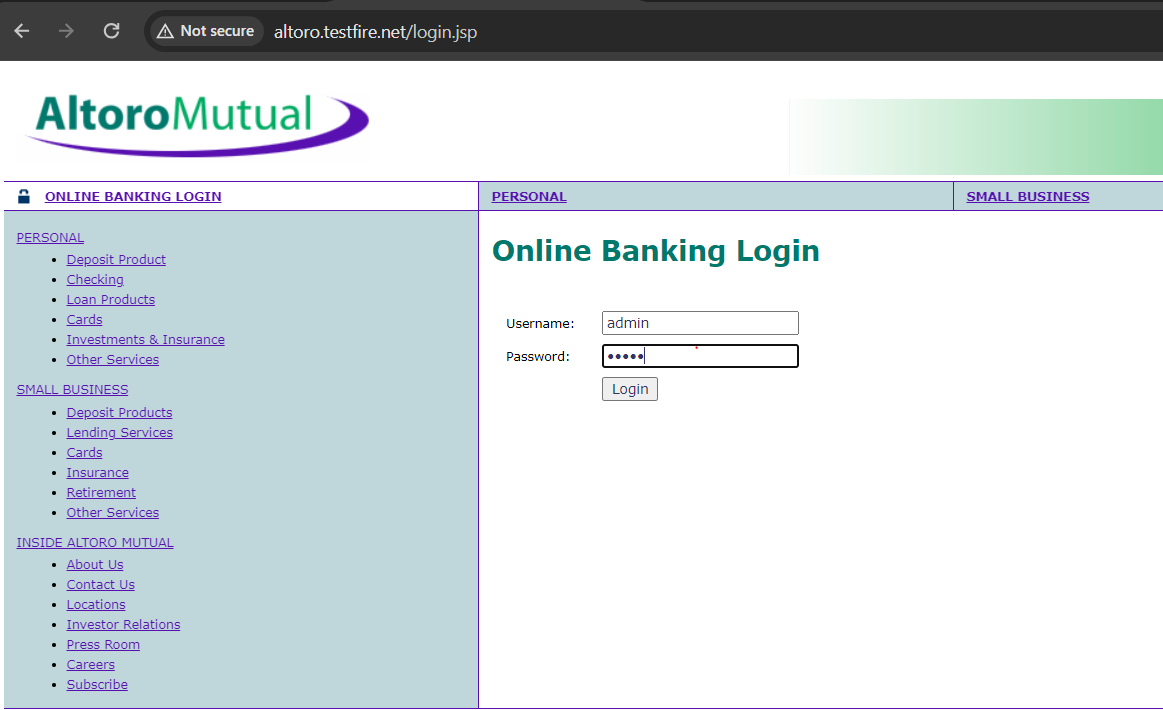
After execution of the code the show the pop-up on the screen.

**Testing Website 2:** [**http://altoro.testfire.net/**](http://altoro.testfire.net/)

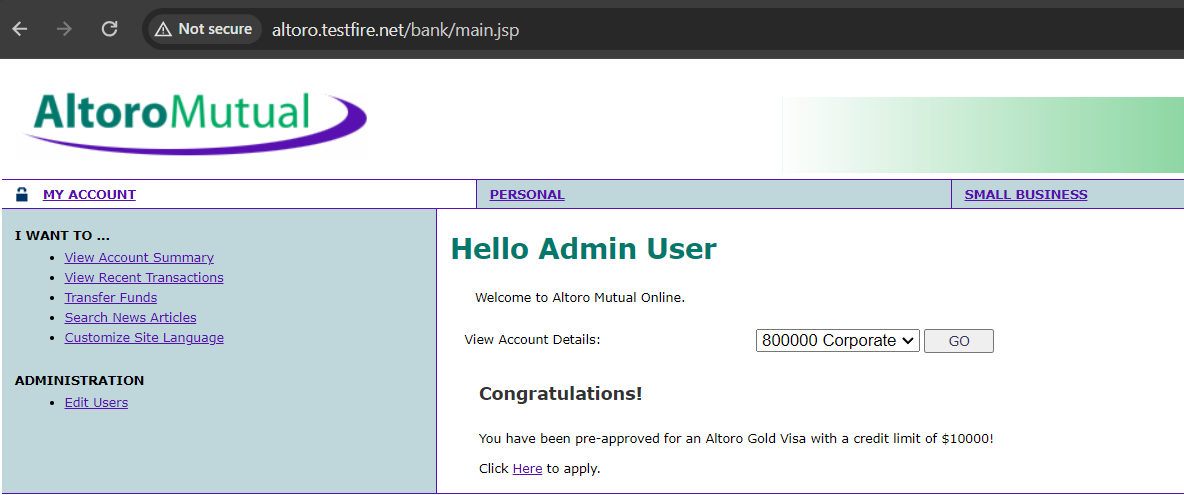
**Vulnerability 2: Bypass by Default credentials**

**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…

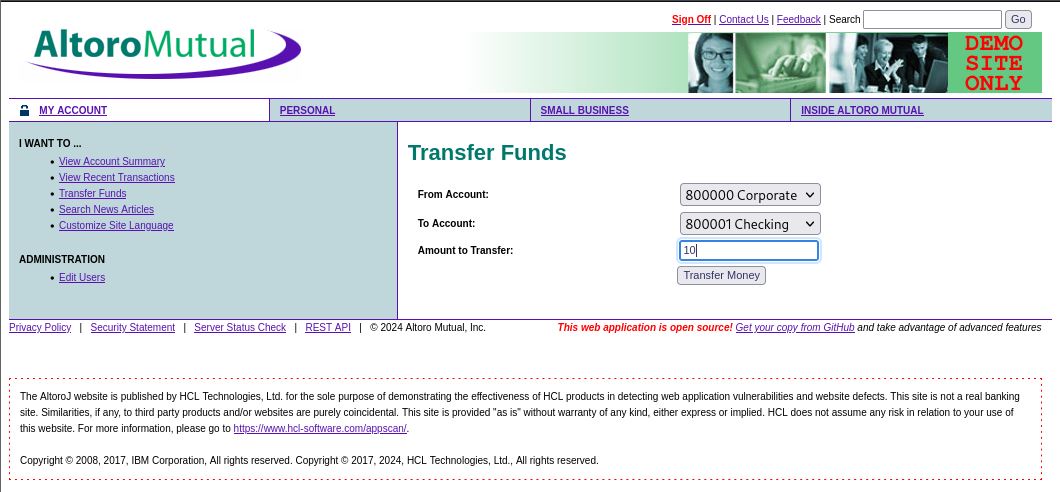


After click on login button the server sends reply Congratulation!. It means we are login successfully.

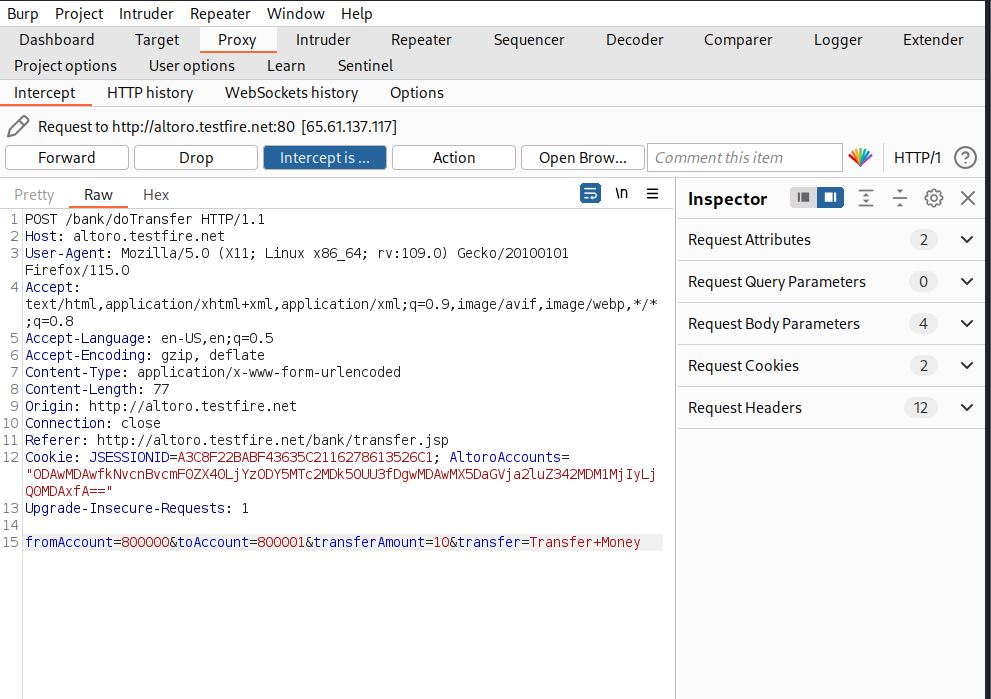


**Vulnerability 3: Request Manipulation**

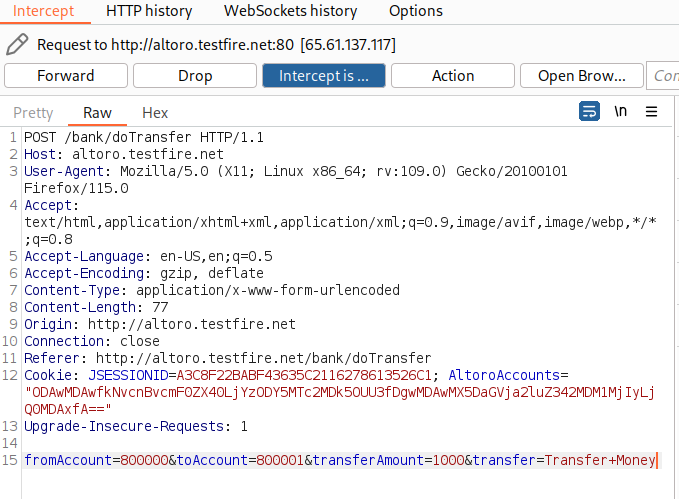
**Description:** In this vulnerability, the attacker uses a tool like Burp Suite to intercept the requests sent from the user's browser to the server. They then modify these requests to carry out malicious actions, like bypassing authentication or injecting harmful code into the application.



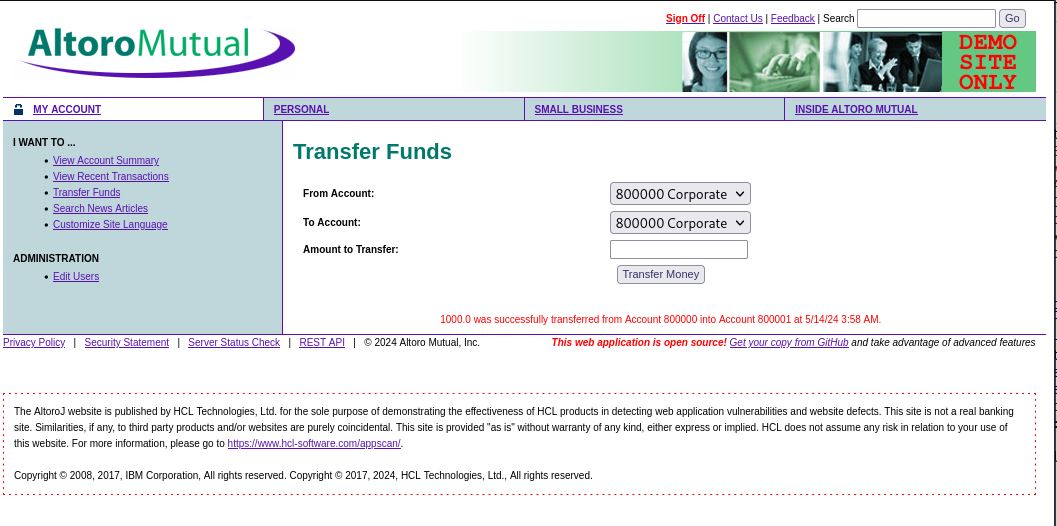
In the upper image we are sending some (10) money from one account to another account.



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



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.

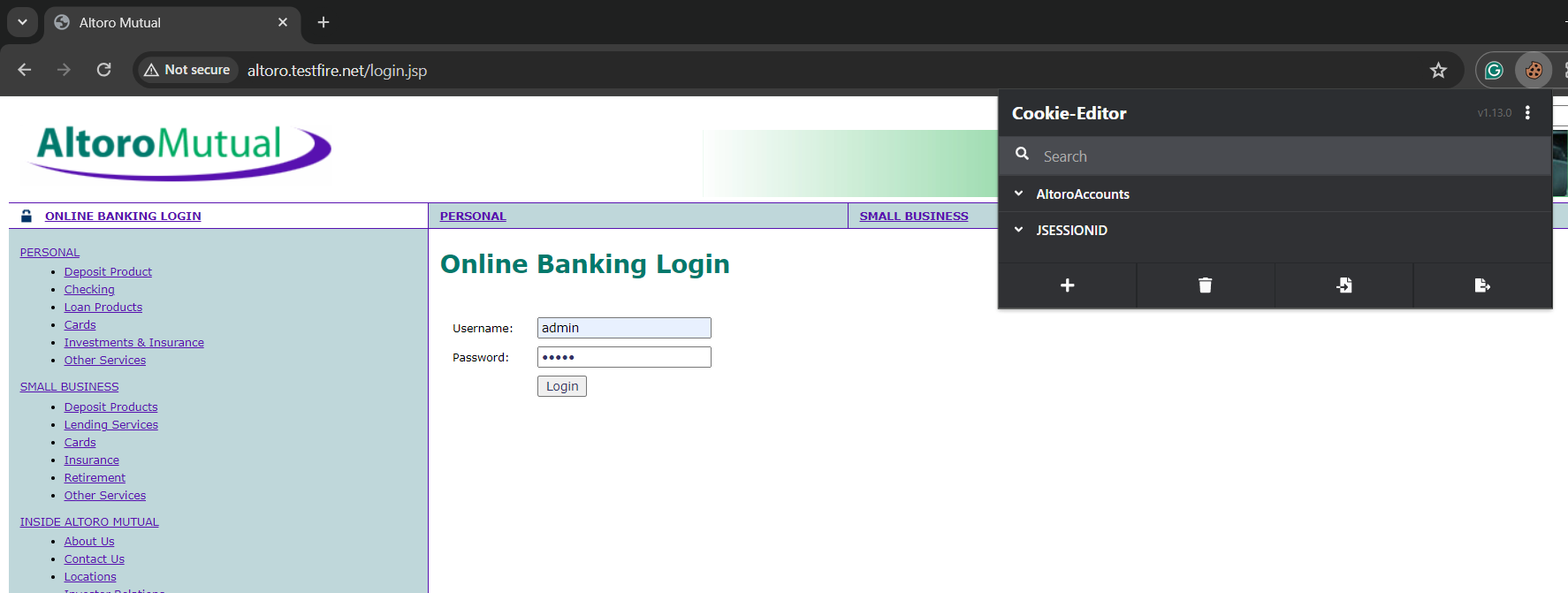


**Vulnerability 4: Session Hijacking**

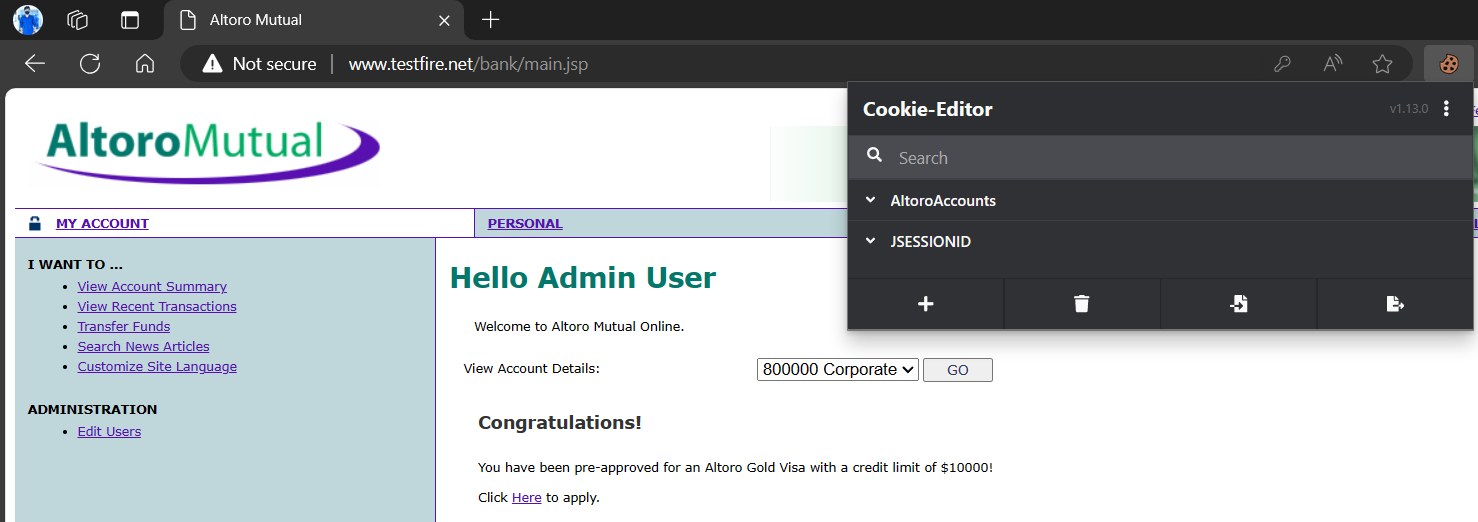
**Description:** Session hijacking occurs when an unauthorized person takes control of an ongoing online session, gaining access to sensitive information or performing actions as if they were the legitimate user. It's like someone stealing the keys to your car while you're driving, allowing them to take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



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.

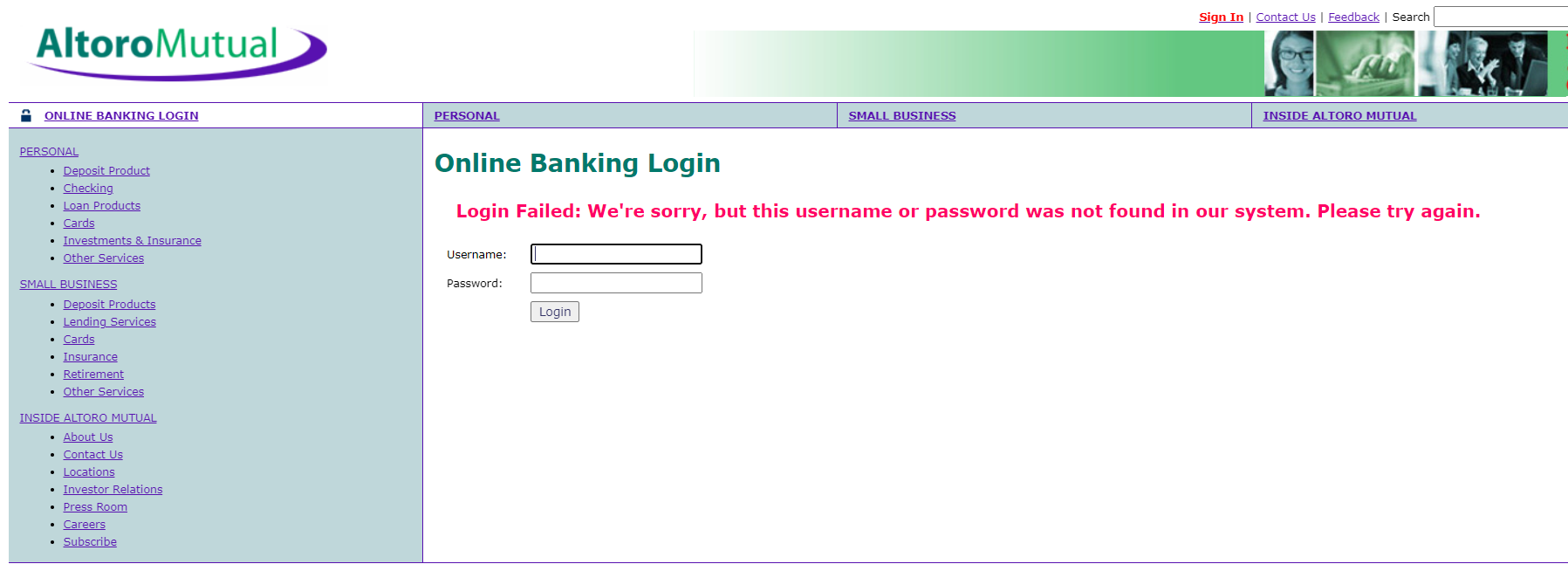


**Vulnerability 3: SQL injection**

**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**

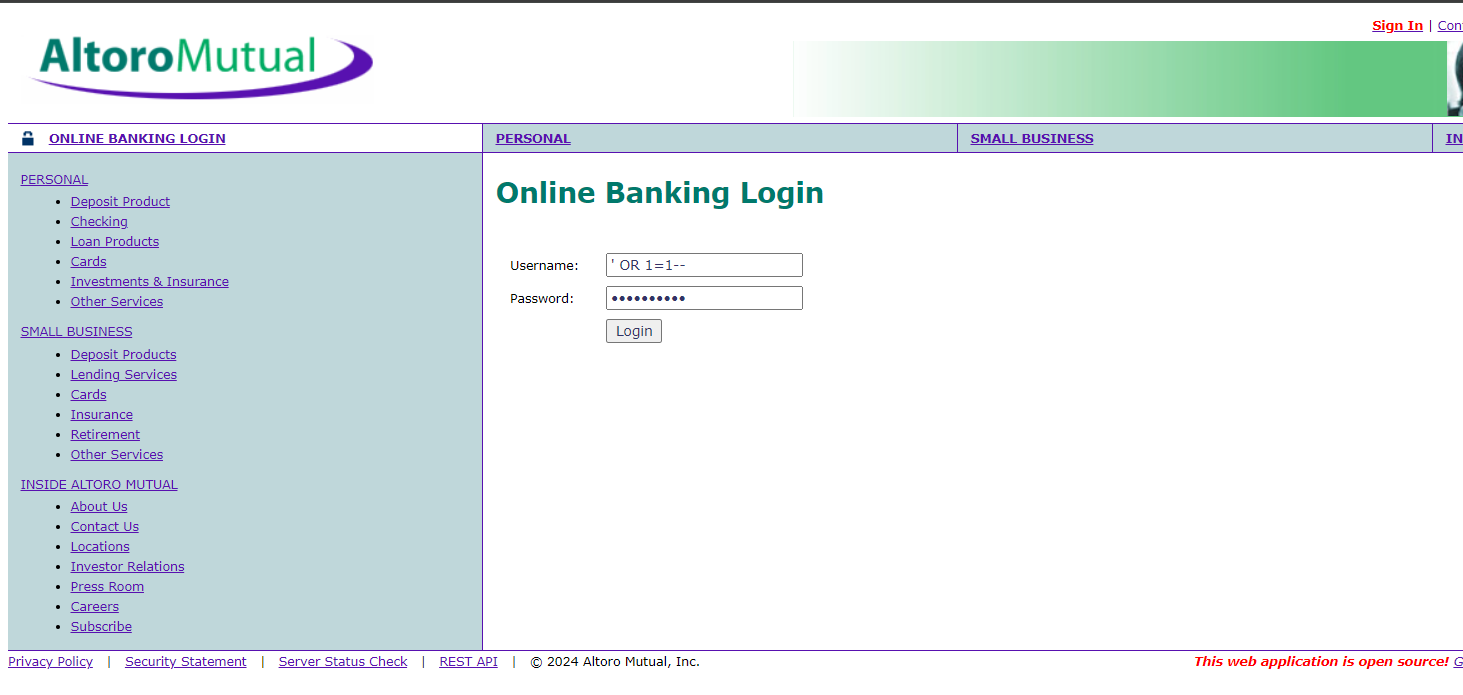
**Website name:** [testfire.net](http://altoro.testfire.net/index.jsp)



We trying Username =helloPassword = 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--**



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.

