Team 7.5

Practice Website Vulnerabilities Report

Team Members:

HARSHIT RAJ

Pabbisetty Pranavi

Shivanshu Tiwari

LAKSHMI

1. ## Vulnerability Name: Apache Tomcat Insecure Default Administrative Password

CWE: CWE- 693

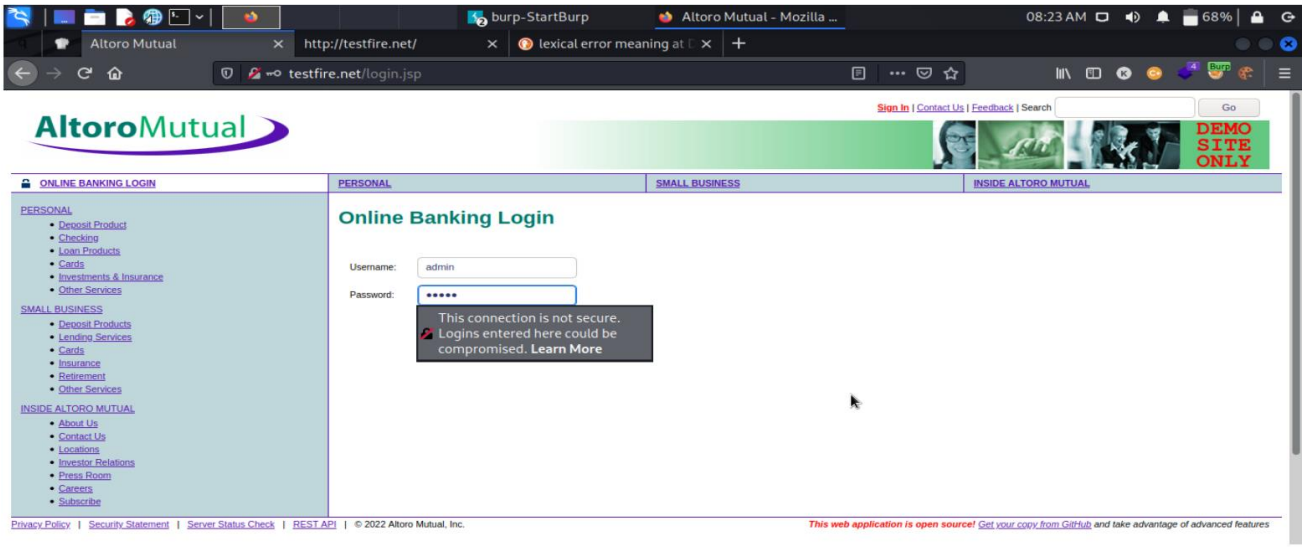
Description: Test fire allows the use of default admin password being used on the login page due to which any unauthenticated user knowing the default password available on the internet can gain access to the admin account and have admin privileges.

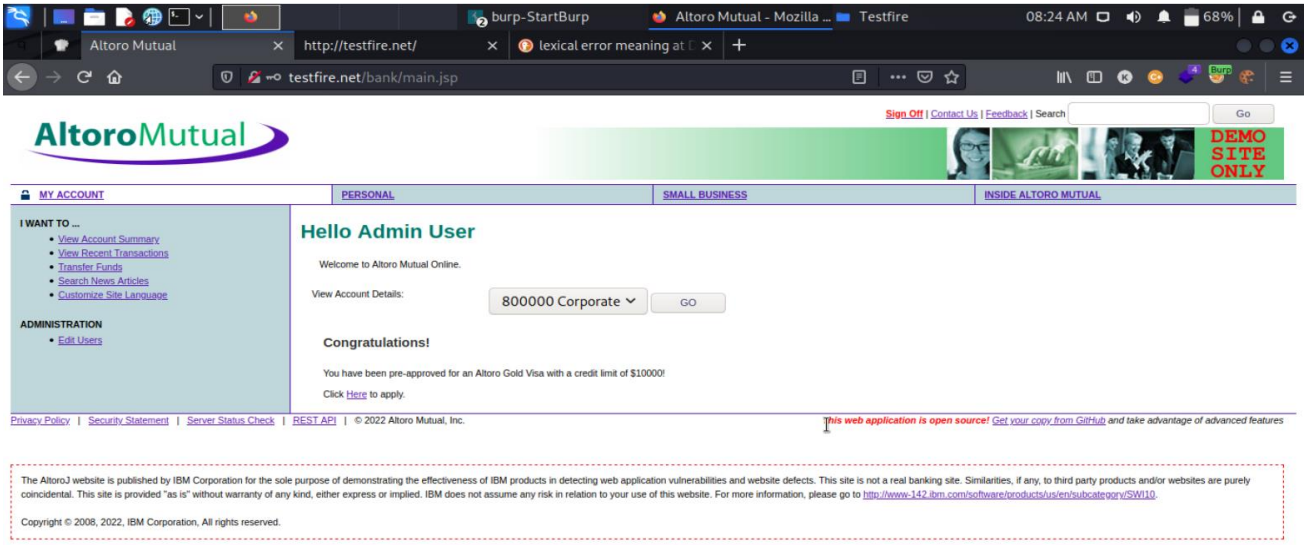
Solution: Change the default password

Business Impact:

Likelihood: High – This attack is effective on web app and have major consequences to it.

Impact: Very High – This attack gives admin privilege to a user who can make any changes on the web application.





2. Vulnerability Name: Insecure Direct object Reference CVE-2022-29627

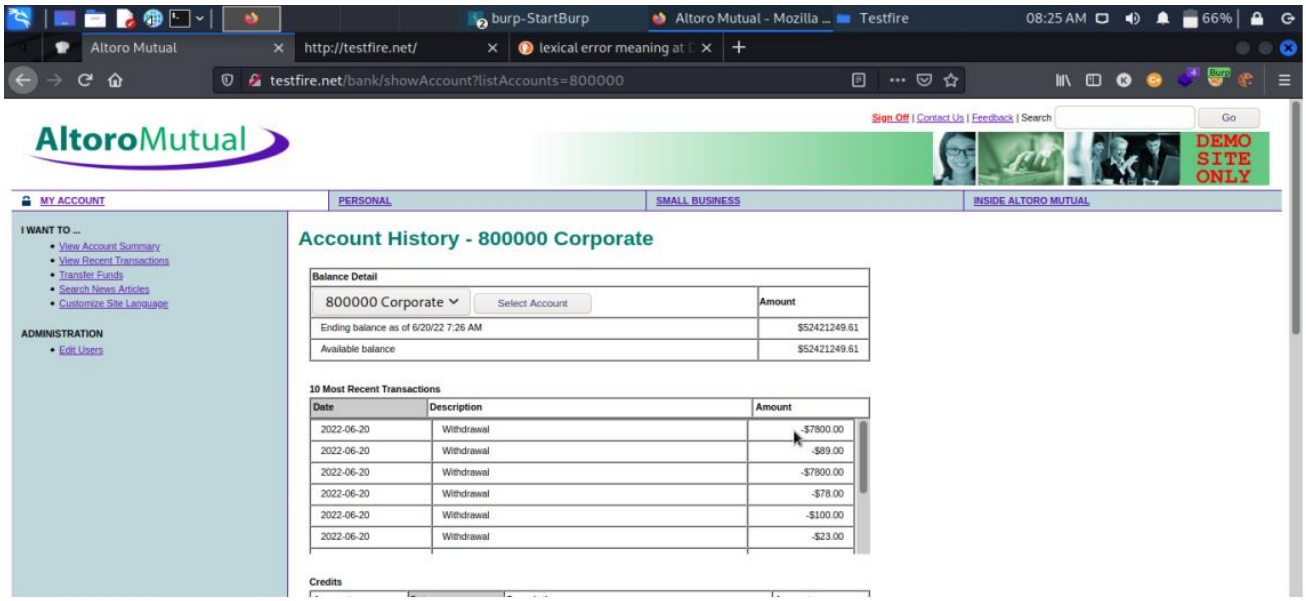
Description: This vulnerability allows any user to view all account info of different user without authentication. The user just has to change the account number of the get request and he/she will be able to view sensitive account information about a diiferent user.

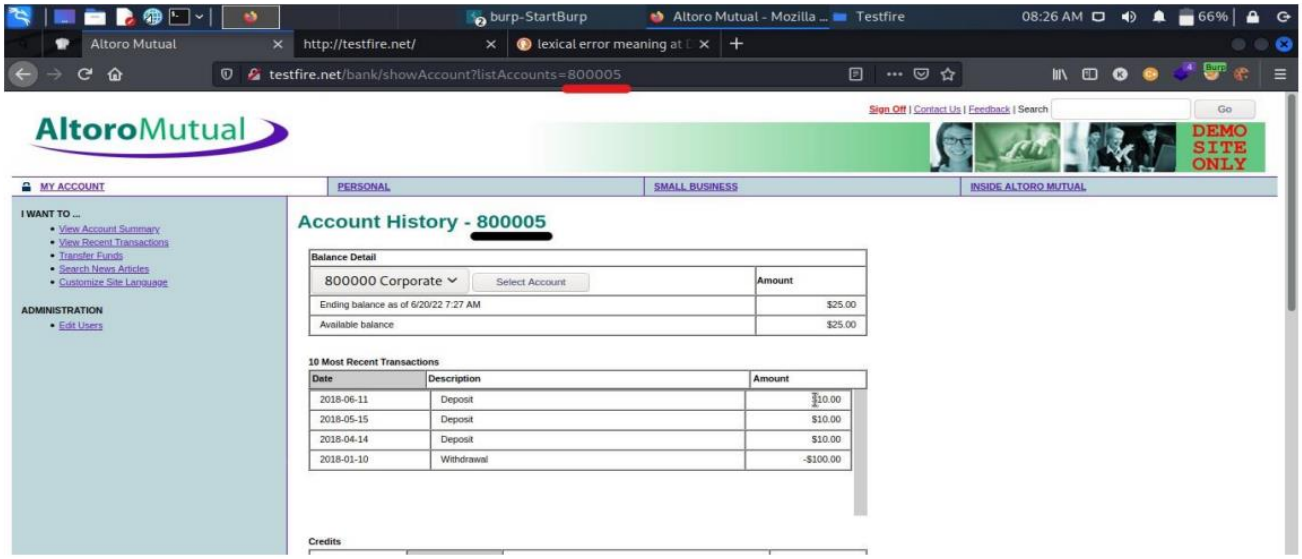
Solution: Proper implementation of security standards on the get and past request

Business Impact:

Likelihood: High – This attack is effective and can display sensitive account information about a different user

Impact: Very High – Changing get request directly from the allows user to view account infromation about different user.





3. **Vulnerability Name: SQL Injection Vulnerability allowing login bypass (Critical).**

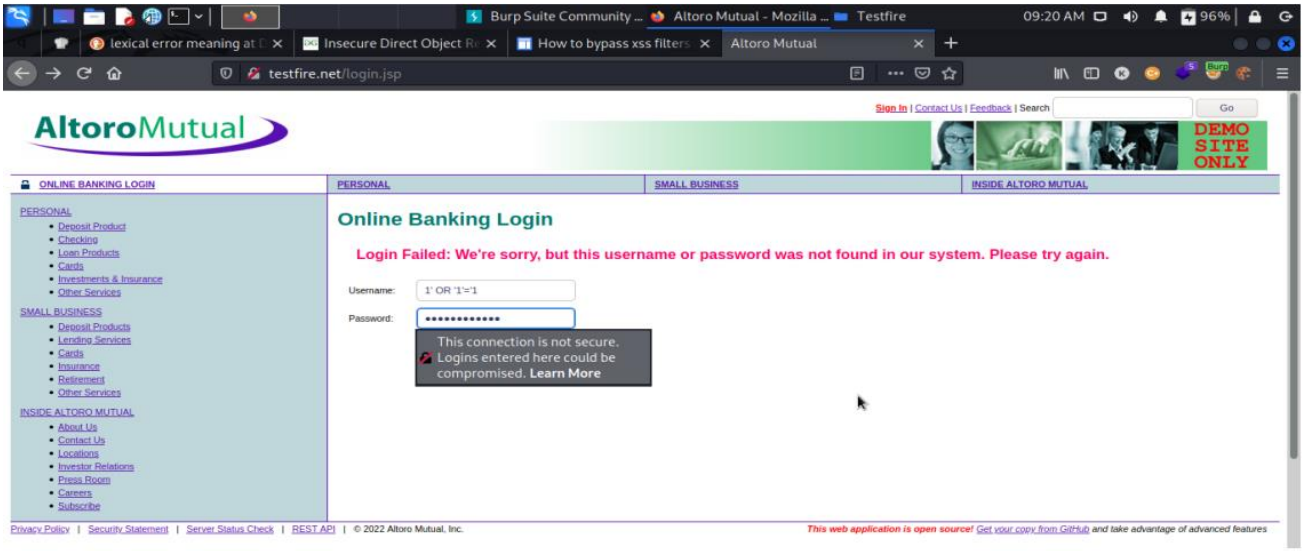
Description: TestFire allowed a successful SQL injection attack can result in unauthorized access to admin account and user accounts as well using the same method the login security if the web app was fully compromised.

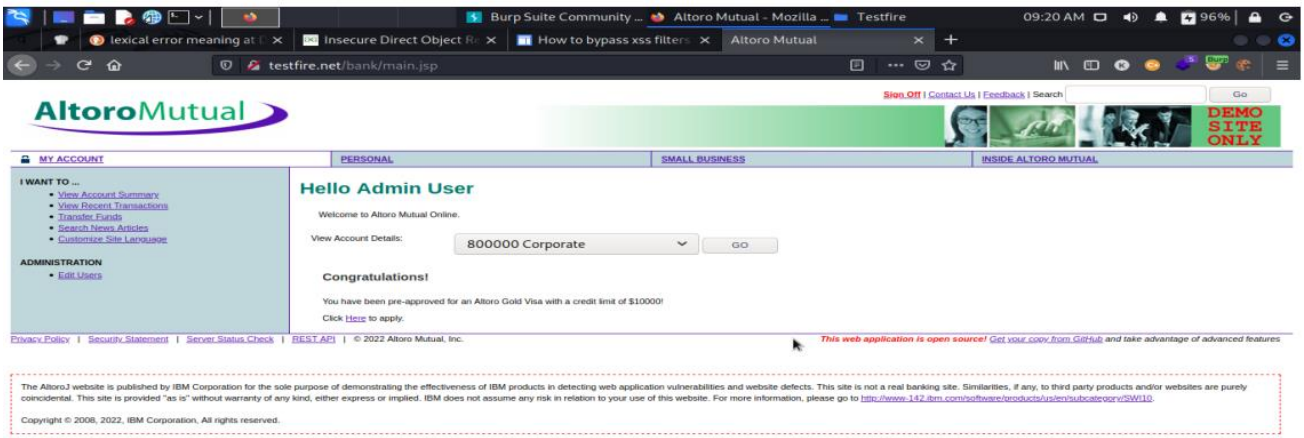
Solution: Proper Input Validation And filtering of username and password.

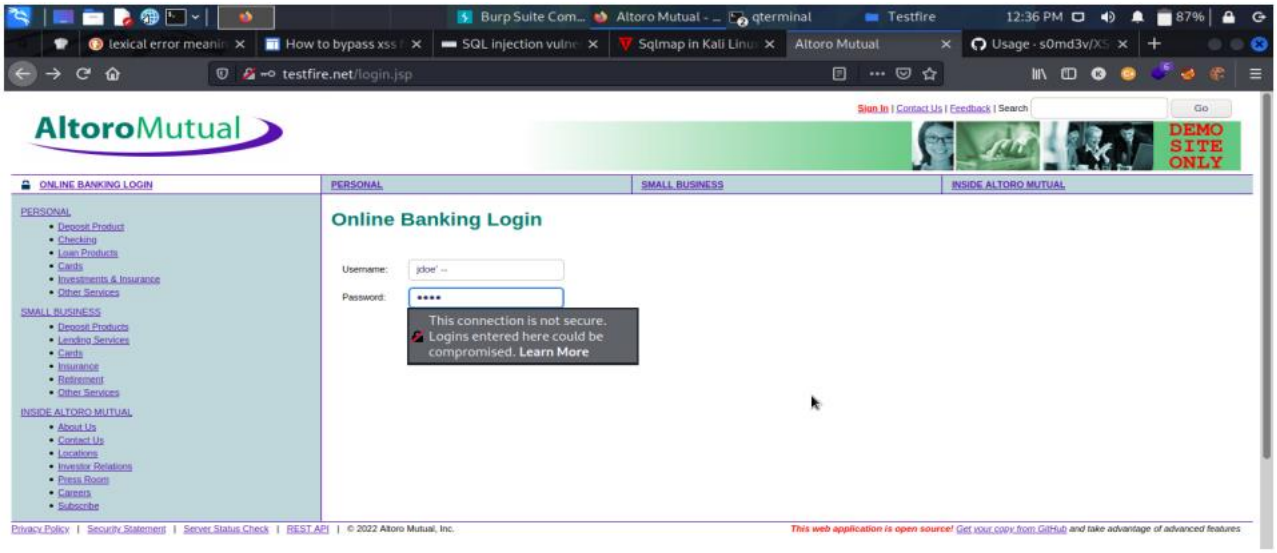
Business Impact:

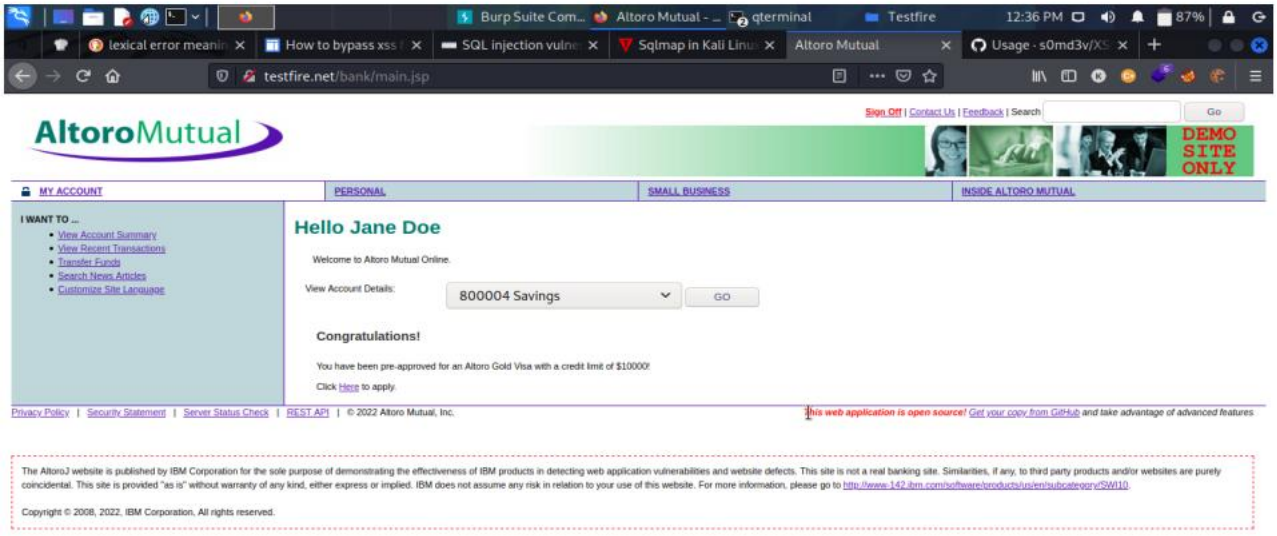
Likelihood: Critical – This attack allowed admin as well as user login acces to the web application.

Impact: Critical – After gaining admin privilege the user has all acces to the backend of the system









4. Vulnerability Name: Login Brute force of username and password

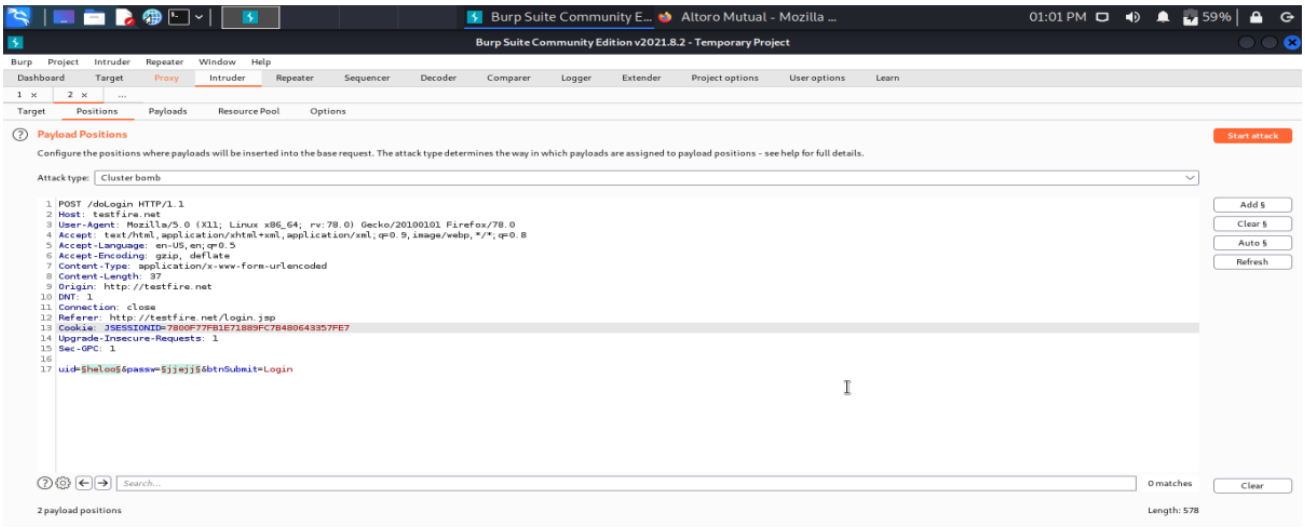
Description: Testfire allowed multiple spraying of username and password on the web app without any restriction.

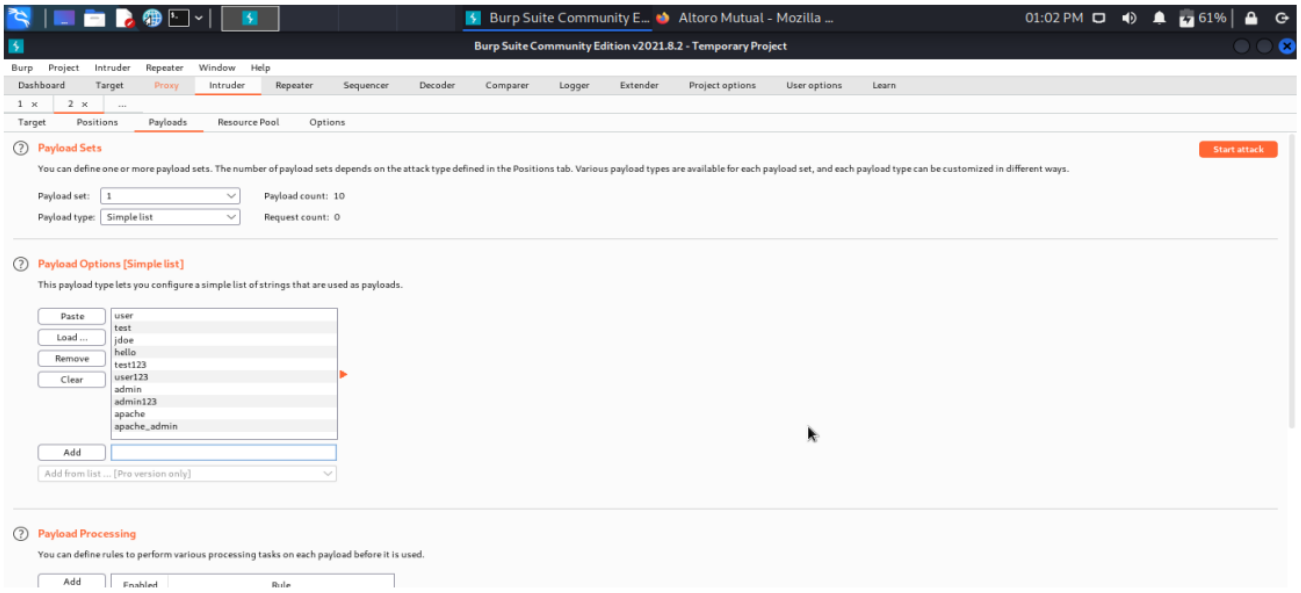
Solution: Most instances of SQL injection can be prevented by using parameterized queries (also known as prepared statements) instead of string concatenation within the query.

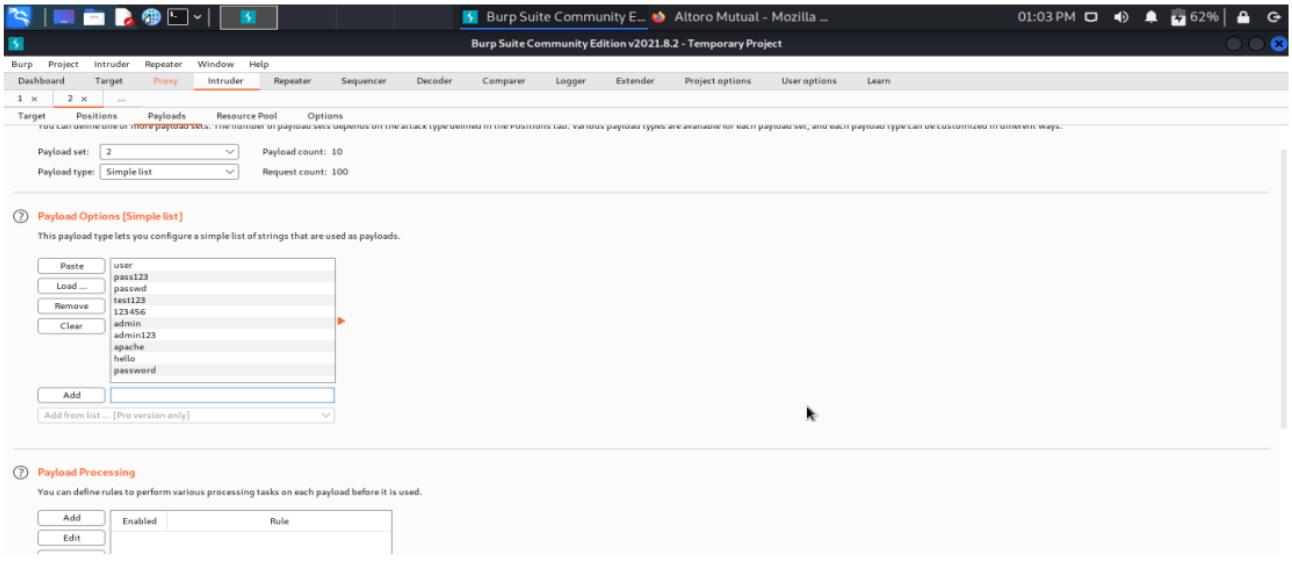
Business Impact:

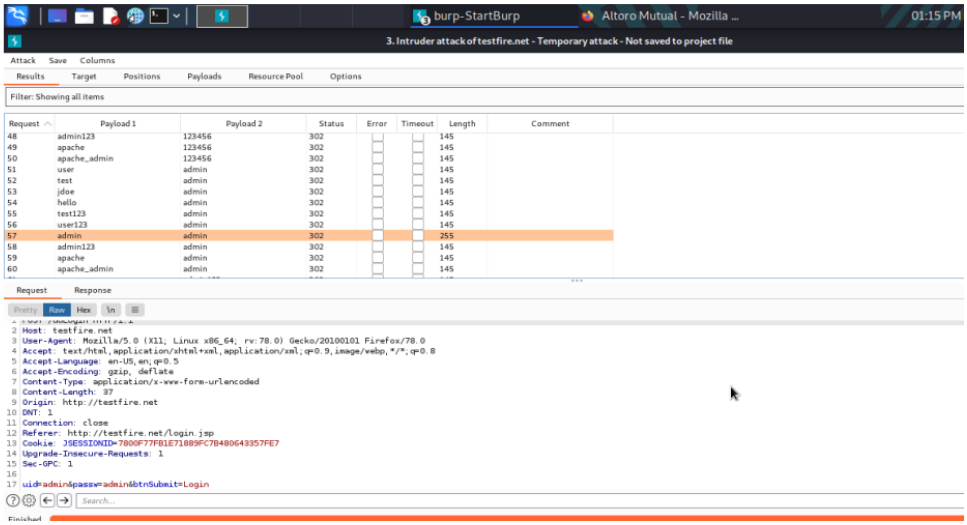
Likelihood: High – The penetration tester sprayed hundreds of username and password on the web application.

Impact: Very High – The attacker can get the username and password of the admin and users.









5. Vulnerability Name: Improper input validation

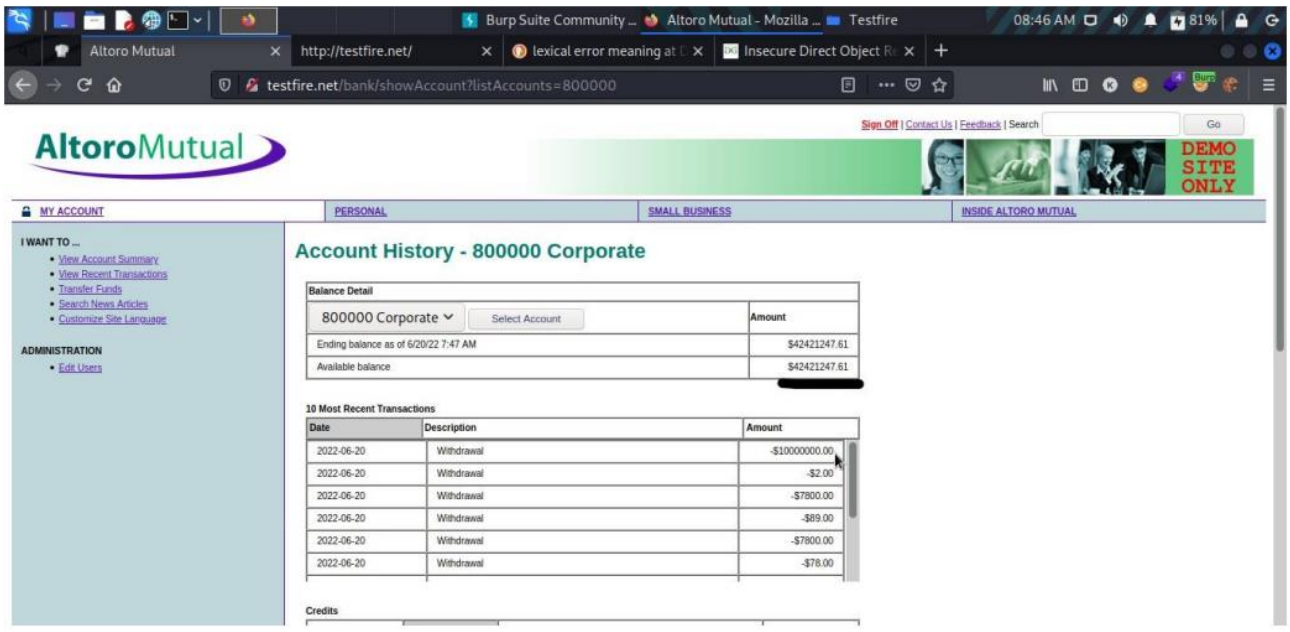
Description: The pen tester was able to deposit a huge amount of money from his own bank account inspite of having low account balance because the amount input field was not properly validated.

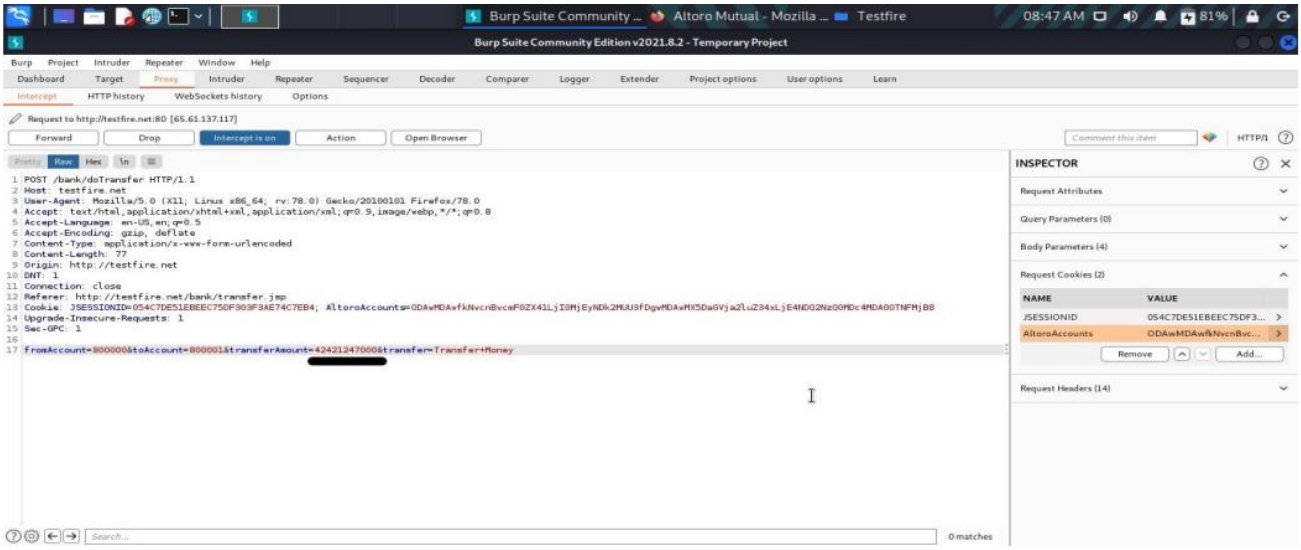
Solution: Proper Input validation mechanism to be implemented on the amount field with some blocking filters.

Business Impact:

Likelihood: High – Was able to deposit huge amount in his account.

Impact: Very High – can cause a huge financial loss to the organization.





6. Vulnerability Name: Reflected XSS CVE-2022-27926

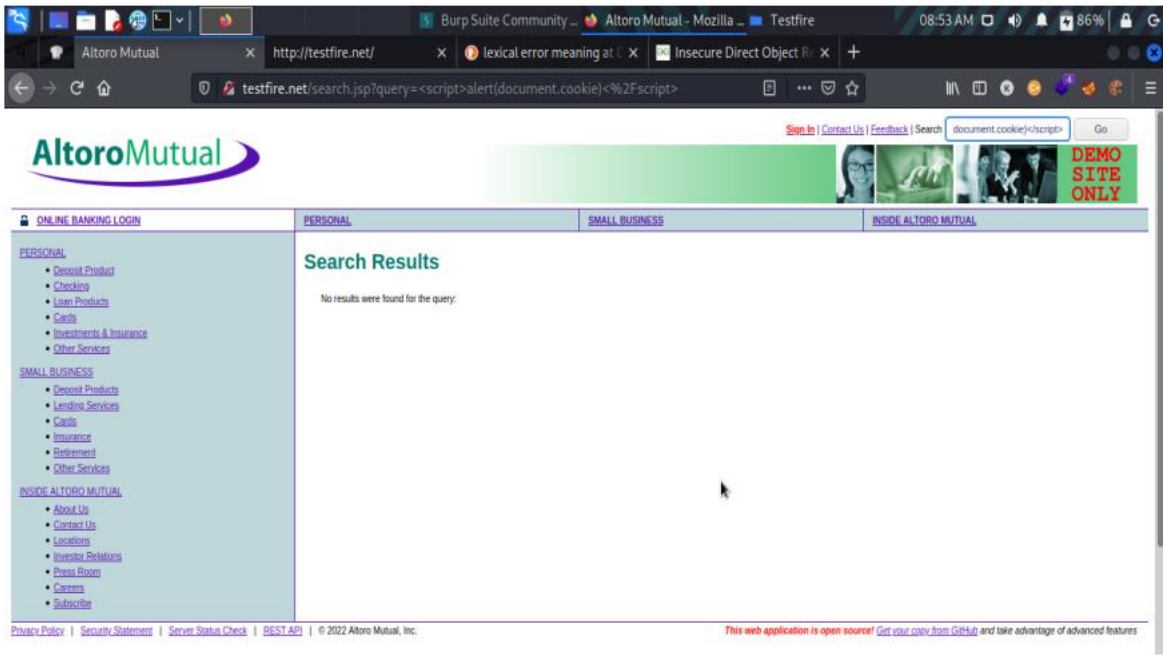
Description: The pen tester was able to send a crafted input on the search field which in result lead to a pop up alert displaying the session cookie.

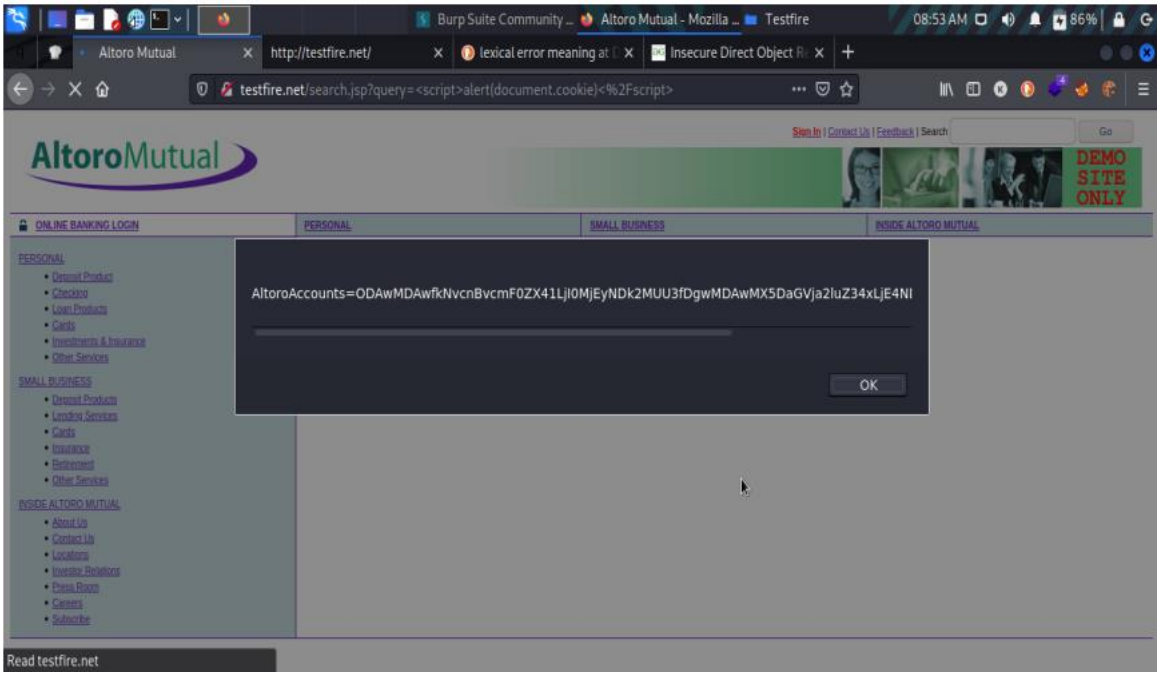
Solution: Proper implementation of input filtering on various character on the search field.

Business Impact:

Likelihood: High – Attacker can send crafted input to users and can steal the cookie.

Impact: Moderate - If exploited, an attacker can send crafted input to other user and can aquire their session cookie





7. Displaying user on web app

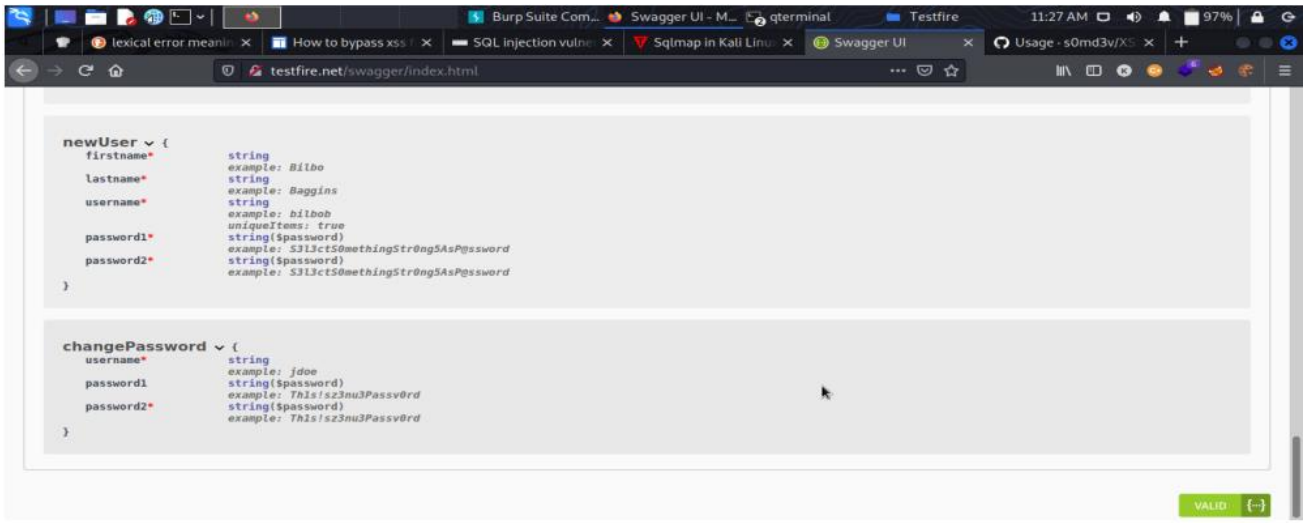
Description: Testfire displayed username and user account number on their web page which can help the attacker in crafting their attack

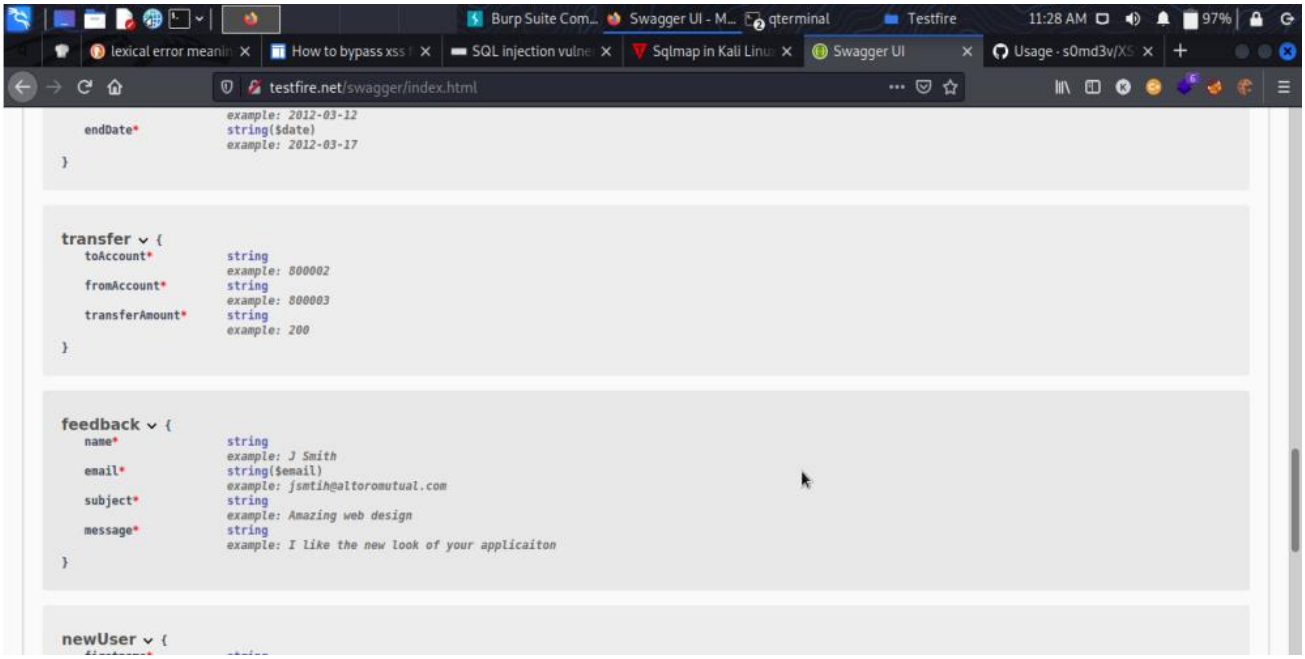
Solution: Developers Should do proper review of the web application before hosting it

Business Impact:

Likelihood: Low

Impact: High – If Sensitive information like username and account is displayed it help in attacking the web app





8. Vulnerability Name: Displaying internal server error

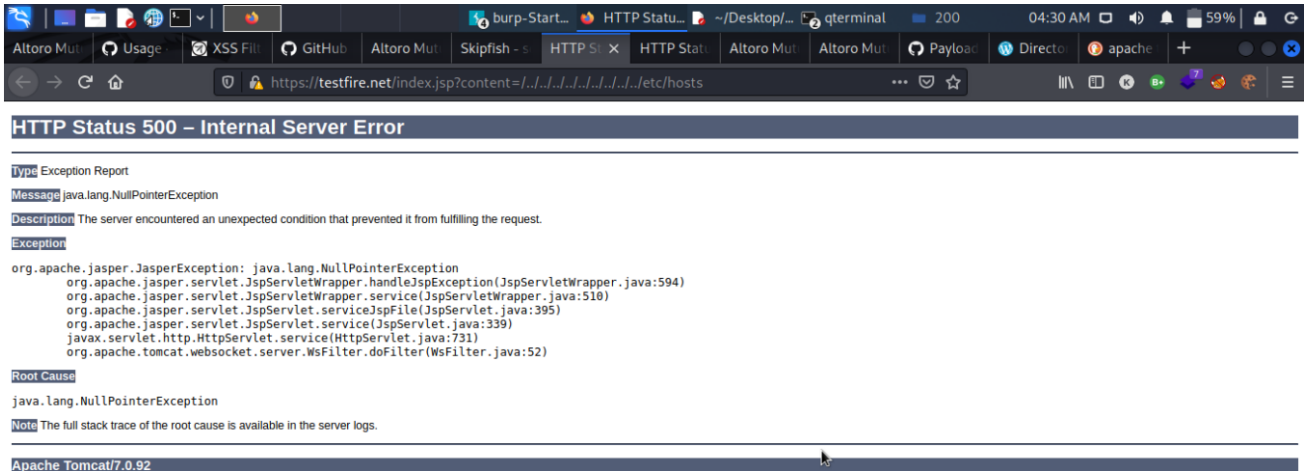
Description: Testfire displayed the error message displaying the apache tomcat version being used in the backend which can help attacker to frame the attack accordingly.

Solution: Developer should do proper review of error messages and error log before hosting the web app.

Business Impact:

Likelihood: High – An attacker can discover the version of the server being used

Impact: Very High – Attacker can exploit according to the version of apache being used and frame attack accordingly.



9.Web Server Transmits Cleartext Credentials

CWE: CWE 522

Description: The remote web server contains several HTML form fields containing an input of type 'password' which transmit their information to a remote web server in cleartext. An attacker eavesdropping the traffic between web browser and server may obtain logins and passwords of valid users

Solution: Make sure that every sensitive form transmits content over HTTPS.

Business Impact: When a web server transmits cleartext credentials, it exposes sensitive user information to potential interception, leading to severe security breaches. This can result in compromised customer trust, legal ramifications, financial losses, and reputational damage for the business, ultimately undermining its credibility and success.

10.Web Server Allows Password Auto-Completion

Description: The remote web server contains at least one HTML form field that has an input of type 'password' where 'autocomplete' is not set to 'off'. While this does not represent a risk to this web server per se, it does mean that users who use the affected forms may have their credentials saved in their browsers, which could in turn lead to a loss of confidentiality if any of them use a shared host or if their machine is compromised at some point.

Solution: Add the attribute 'autocomplete=off' to these fields to prevent browsers from caching credentials.

Business Impact: Allowing password auto-completion on a web server can lead to significant security risks. It increases the chances of unauthorized access and data breaches, potentially compromising sensitive customer information. This could result

in legal liabilities, reputation damage, loss of customer trust, and financial consequences for the business.