Test Website Vulnerabilities

TEAM 7.6

SALONI GHULE- 21BCE1967

SHREYA SINGH-21BPS1435

KREET ROUT-21BCE1482

NITIN KUMAR-21BCE1792

1. SQL Injection vulnerability allowing login bypass

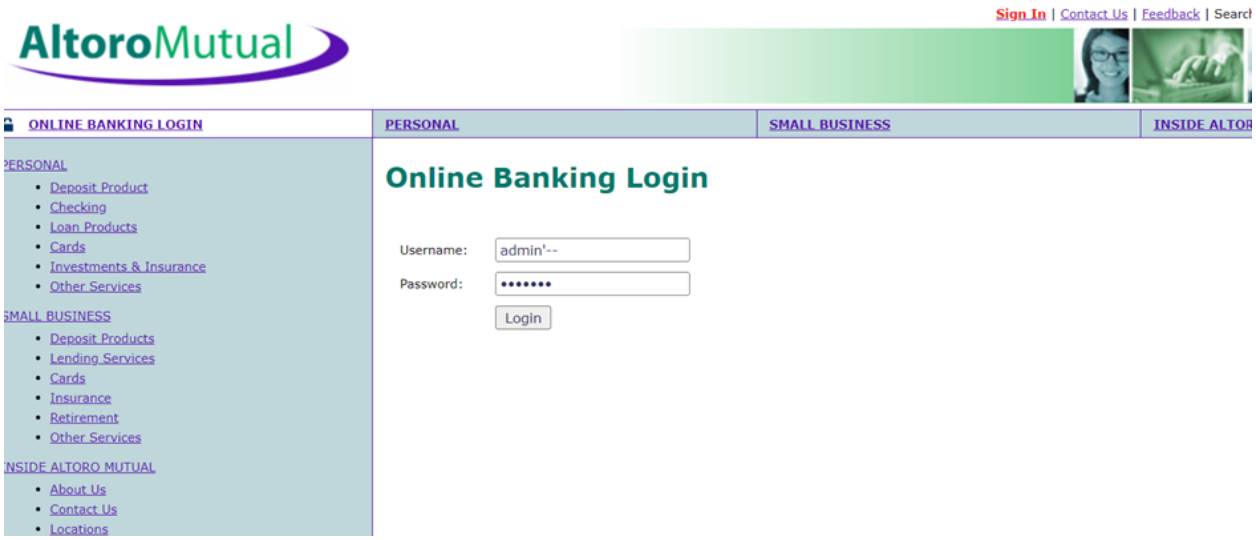
CWE-288: Authentication Bypass Using an Alternate Path or Channel

Description: A product requires authentication, but the product has an alternate path or channel that does not require authentication.

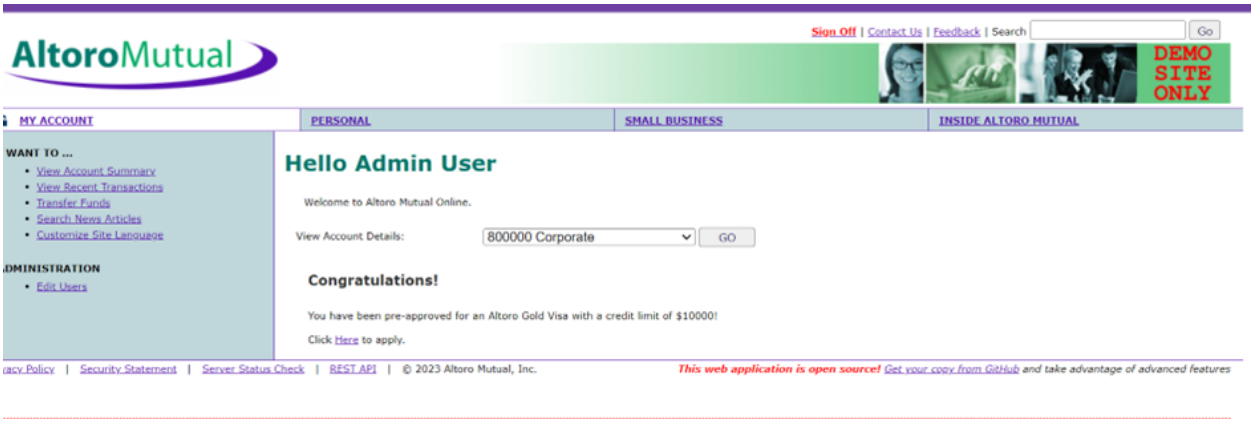
Business Impact:

SQL injection attacks represent an extreme security danger to associations. A successful SQL injection assault can bring about confidential and important information being erased, edited or taken out for malicious uses. Other risks are sites being ruined, defaced or unapproved access to frameworks or accounts and, eventually, compromised machines or whole systems.

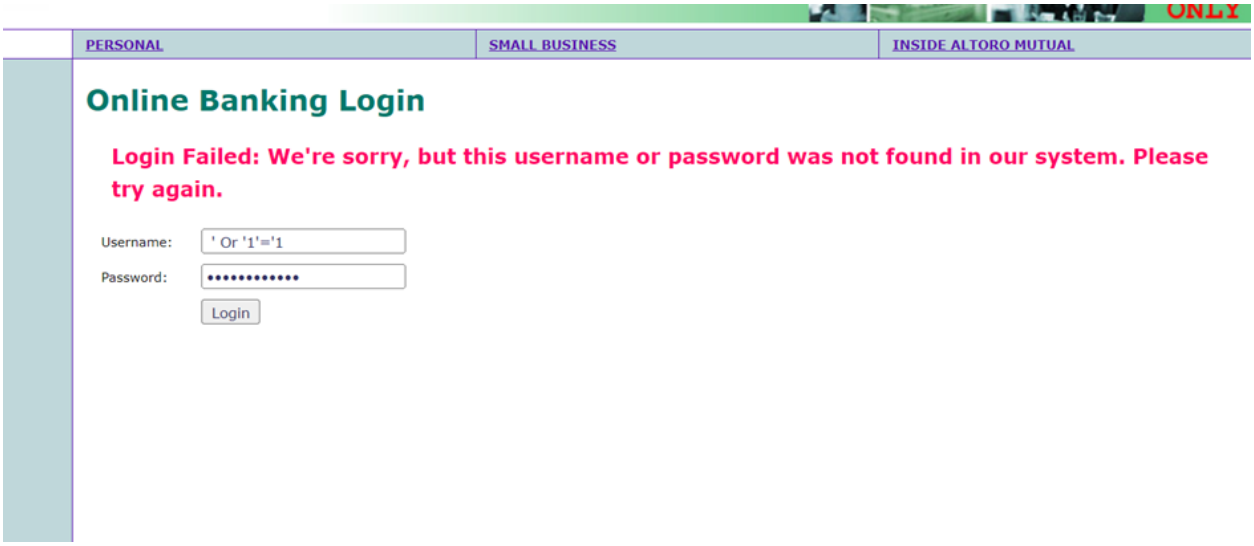
Testing:

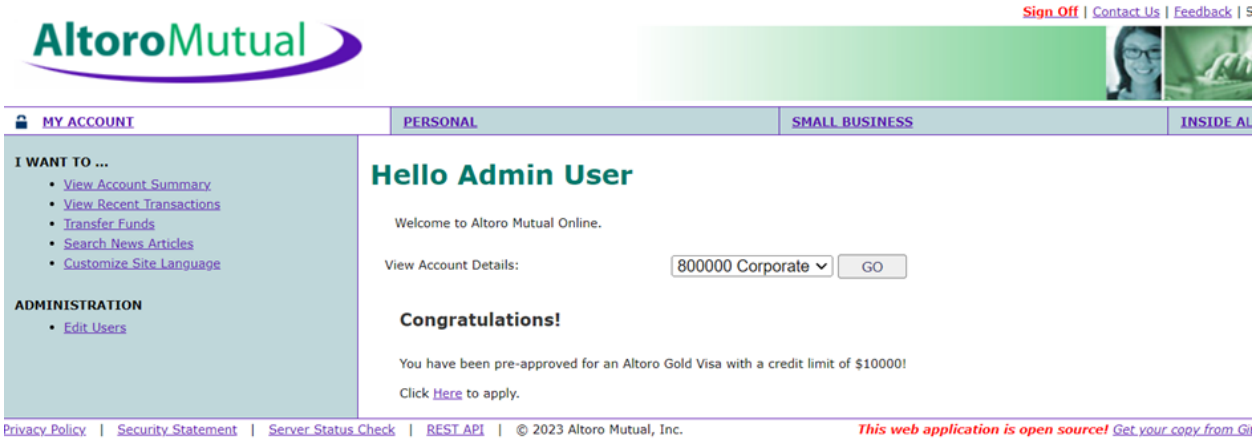


We used the command admin'-- as the username what it signifies is that for the username administrator login into the application without checking the password because -- is used which comments out whatever is written ahead of it thus commenting out the password matching SQL query.



Some other SQL Injections that we can use can be the basic ' Or '1'='1.





2.Brute Force Attack

CWE-1391: Use of Weak Credentials

Description: The product uses weak credentials (such as a default key or hard-coded password) that can be calculated, derived, reused, or guessed by an attacker.

CWE-307: Improper Restriction of Excessive Authentication Attempts

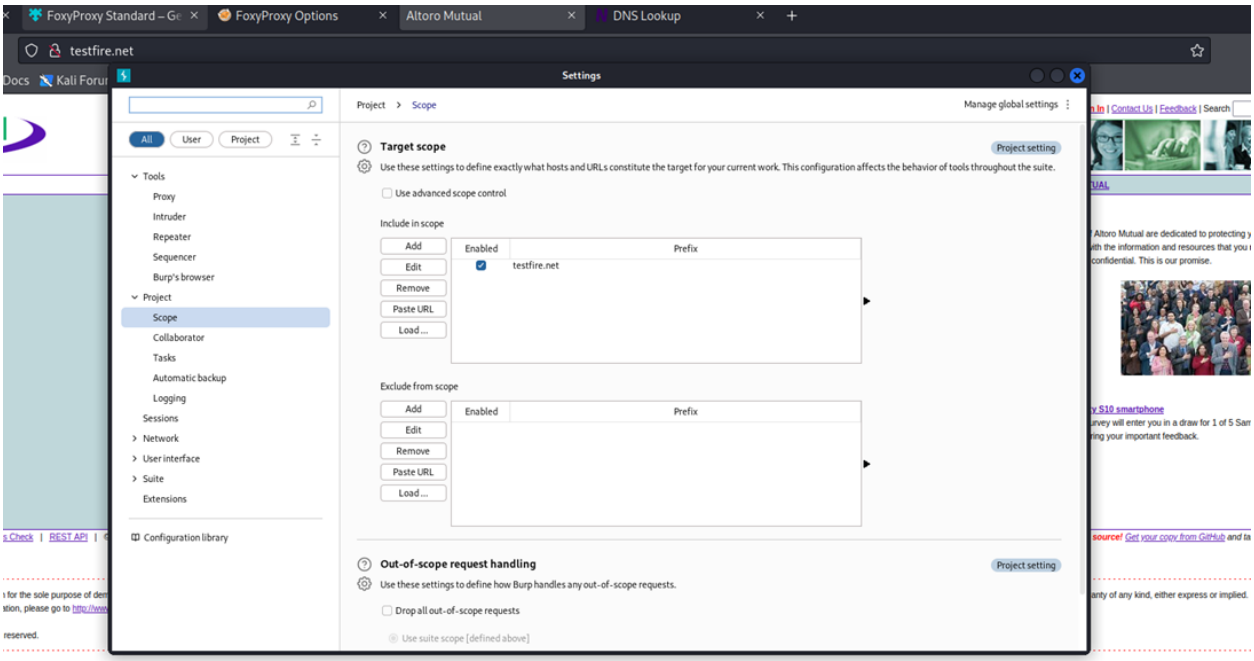
Description: The product does not implement sufficient measures to prevent multiple failed authentication attempts within a short time frame, making it more susceptible to brute force attacks.

Business Impact:

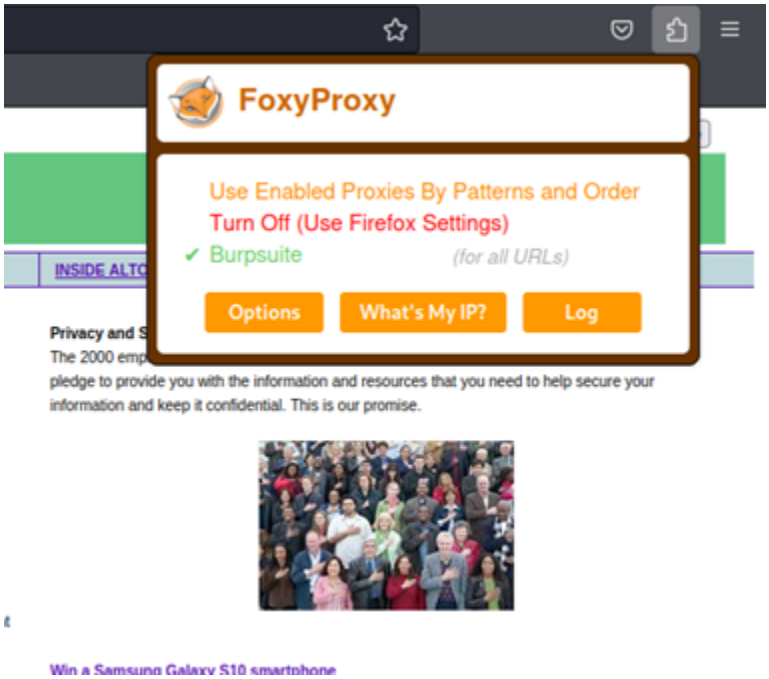
A successful brute force attack on a business can result in unauthorized access, data loss or theft, financial losses, reputational damage, legal consequences, operational disruptions, increased security costs, loss of competitive advantage, damage to trust with customers and partners, and compliance issues, depending on the industry.

Testing:

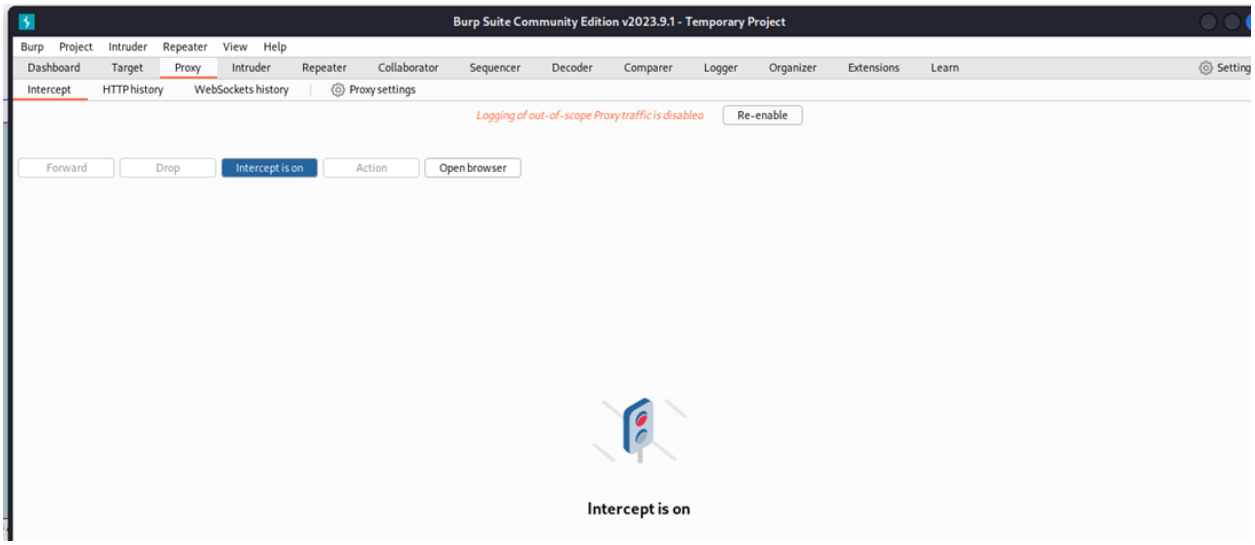
- · First we add the website by going to target tab -> add

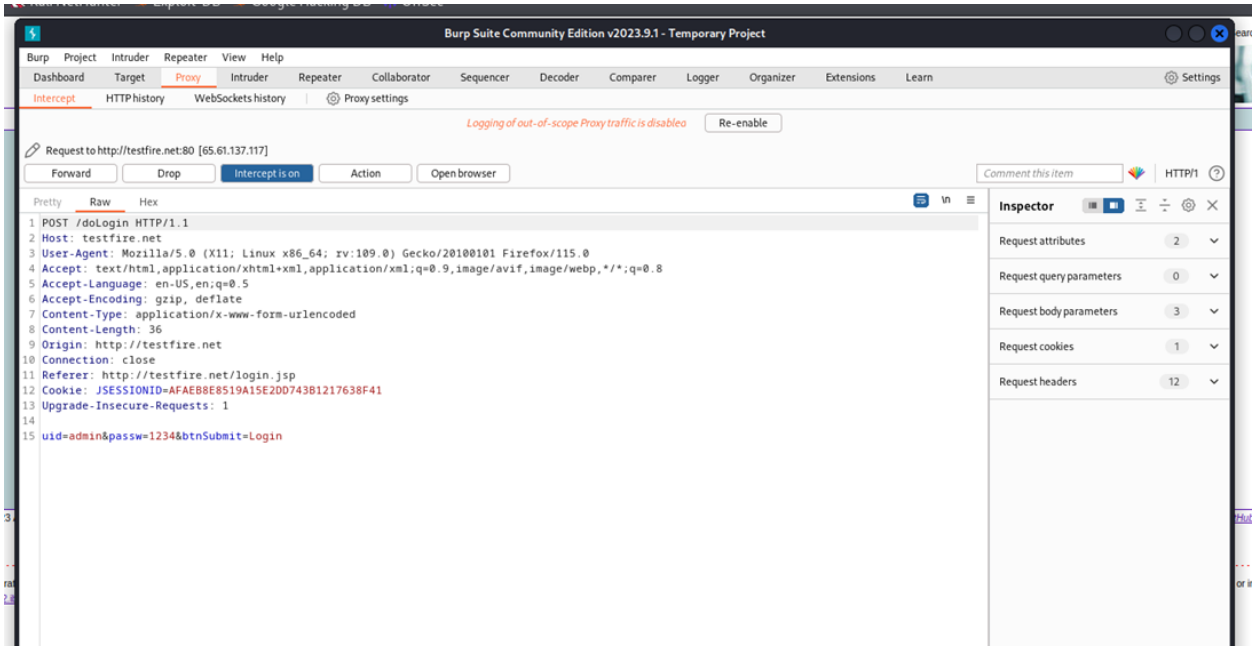


- · Choose burp as our default proxy on foxyproxy.

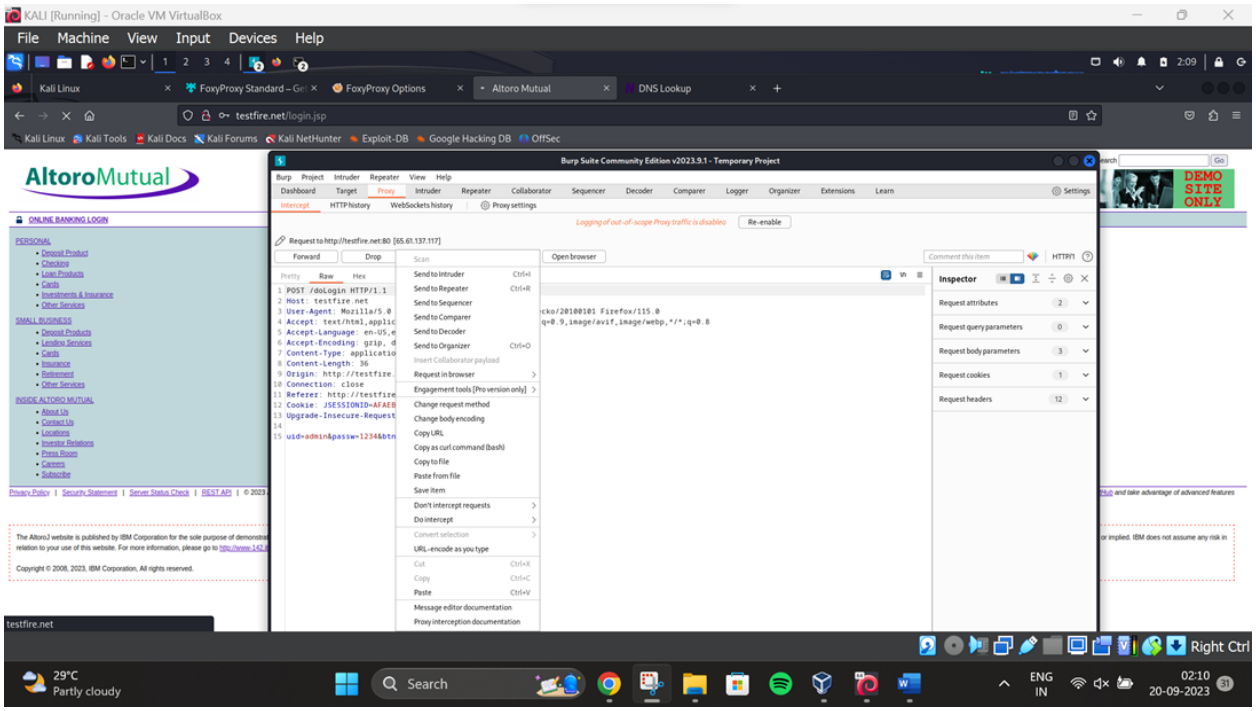


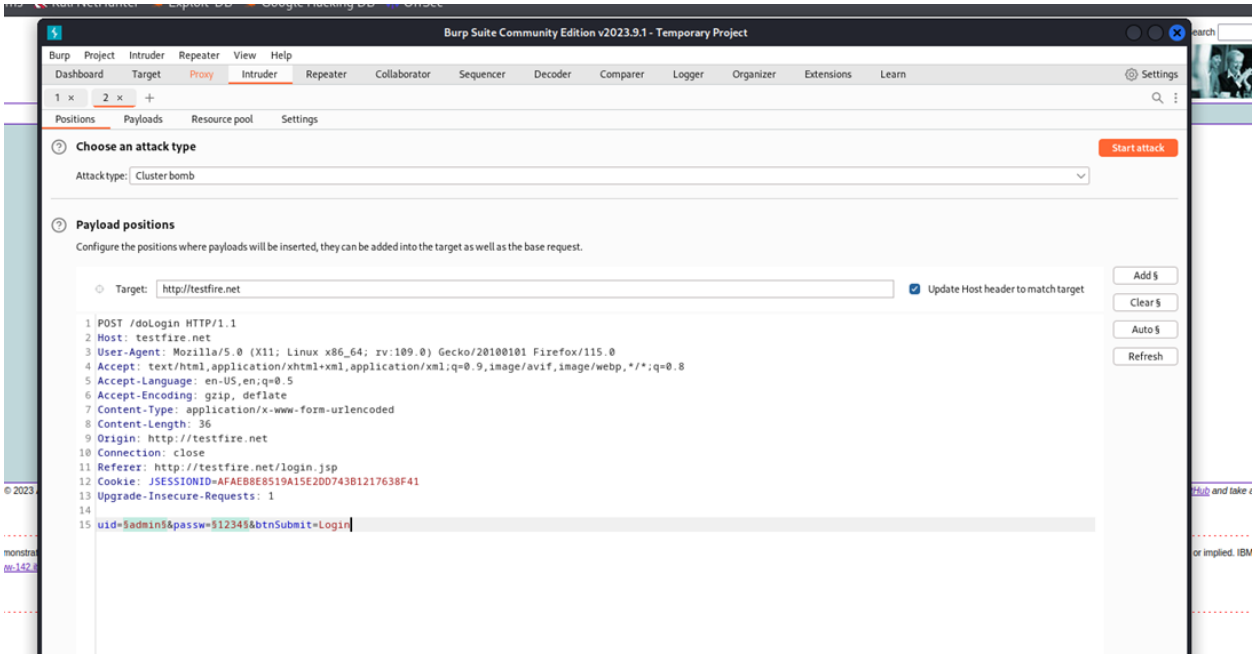
- Go to proxy and turn on the intercept and then click on the login page of the website and give in random username and password.



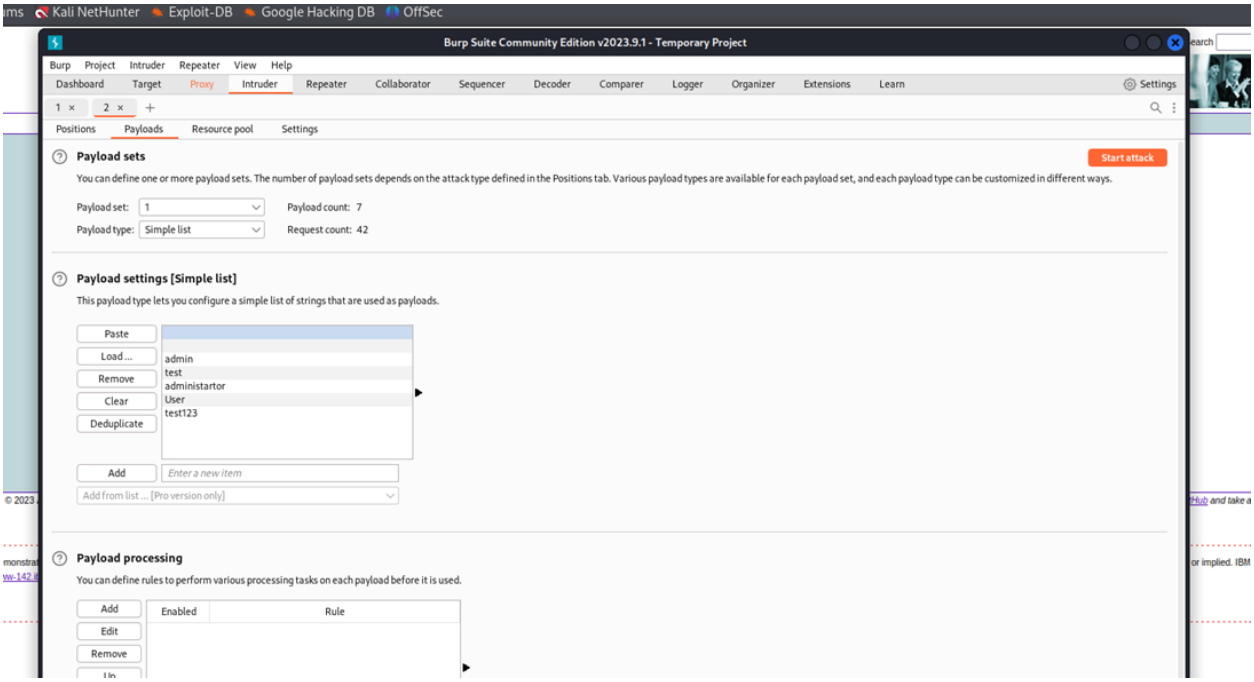


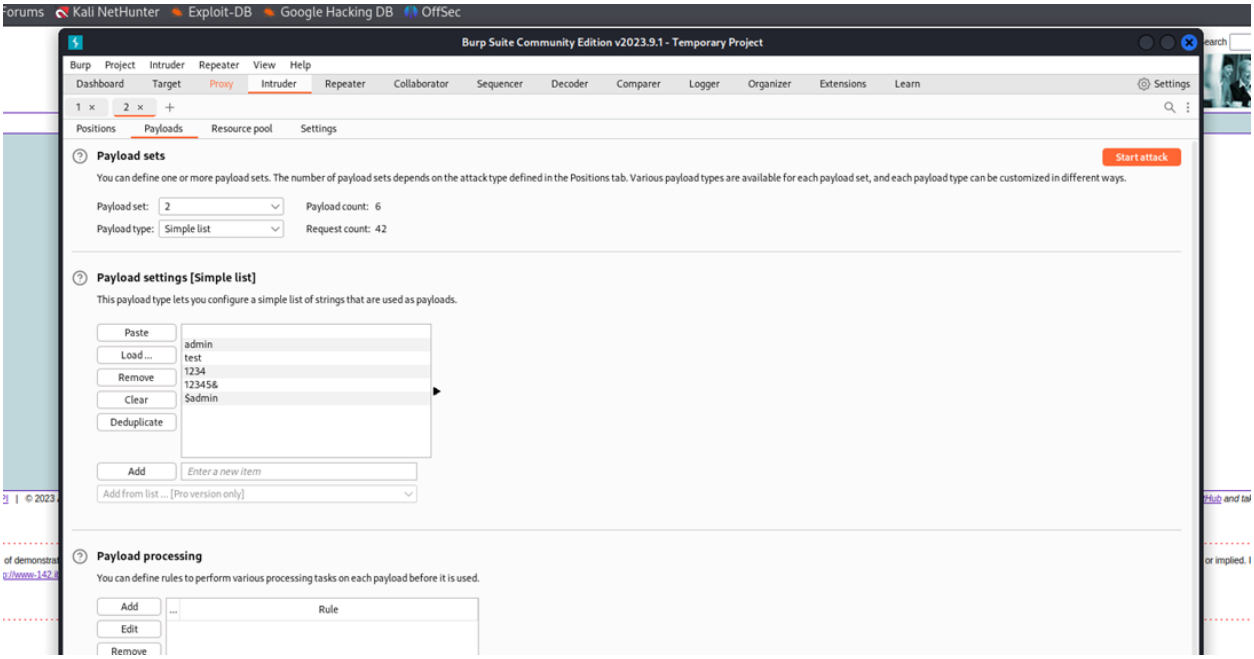
- Then we sent it to intruder and go to positions tab choose cluster bomb attack and select the given input of username as payload 1 and given input of password as payload 2 by clicking on add.



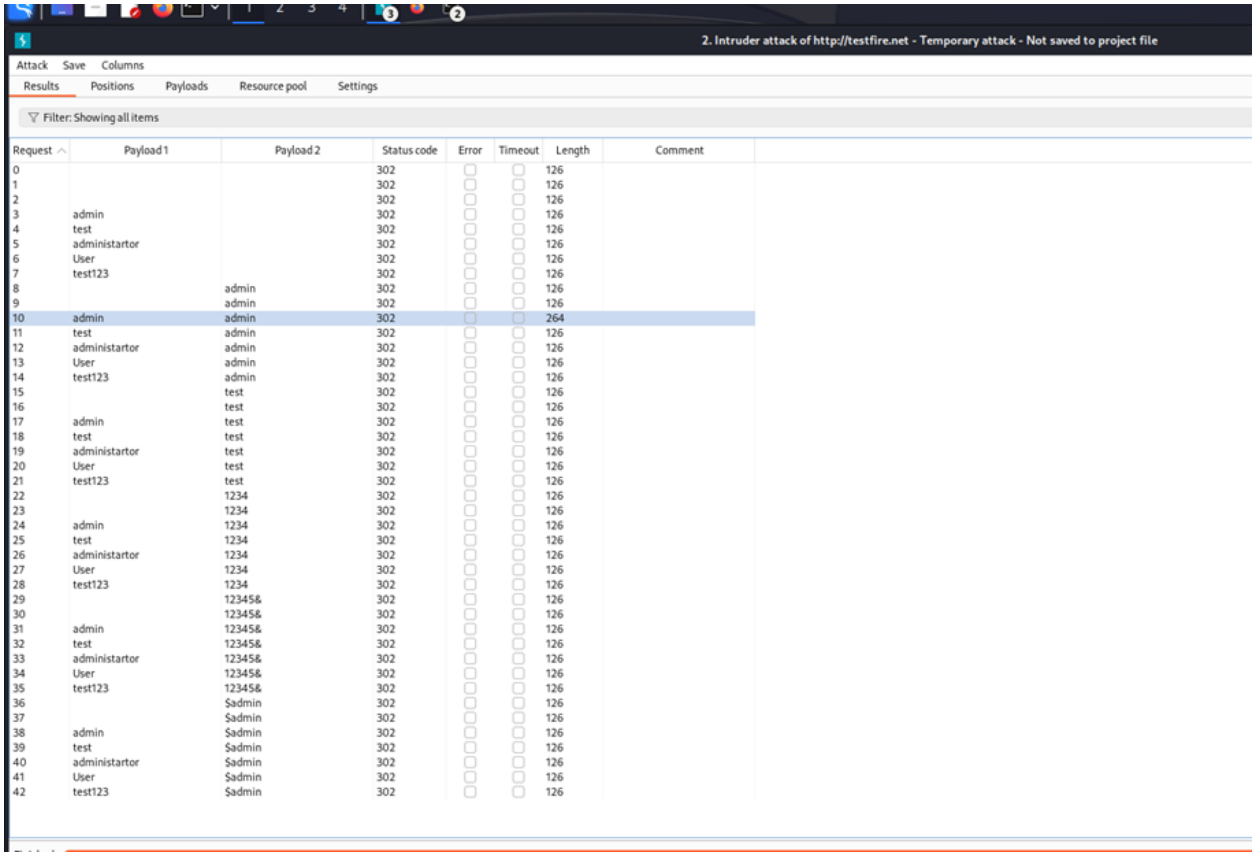


- Then we go to the payloads tab and select payload 1 that is our username in this case and choose simple text and below give some random expected usernames. We can also upload a file here but since I do not have one I did it this way. We do the same for payload 2 which is our passwords and then start the attack.

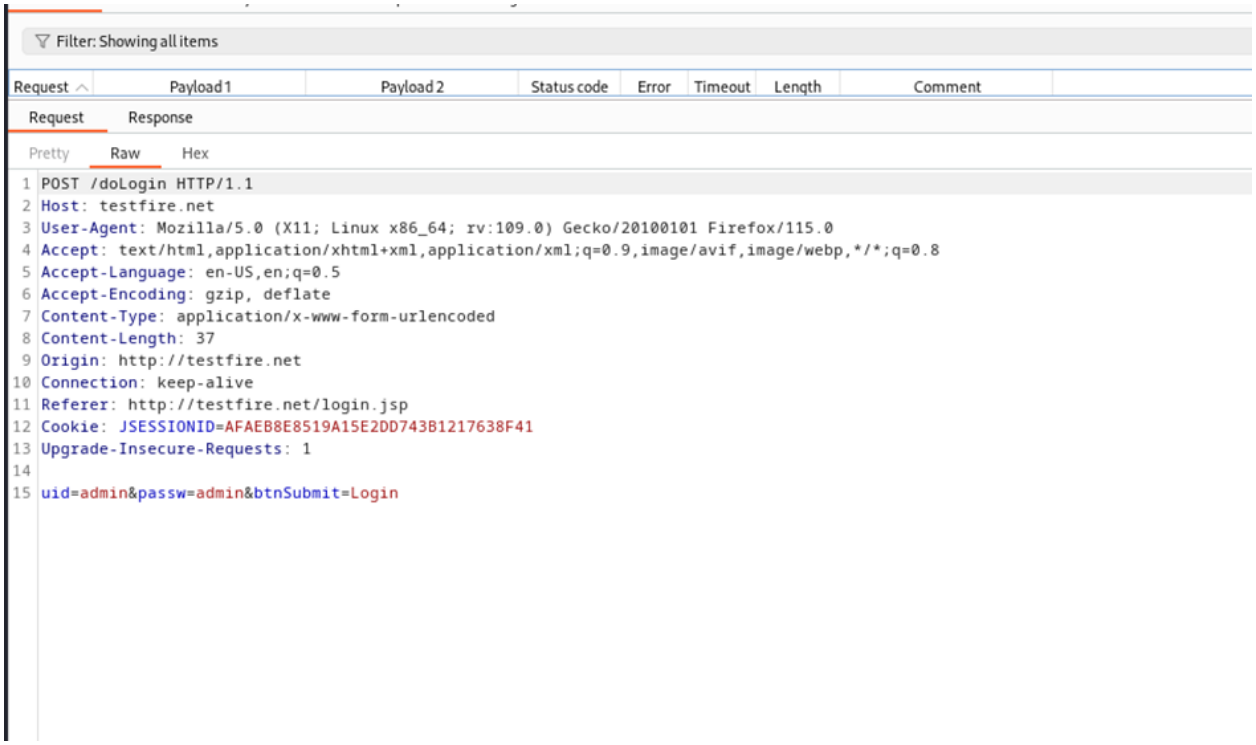




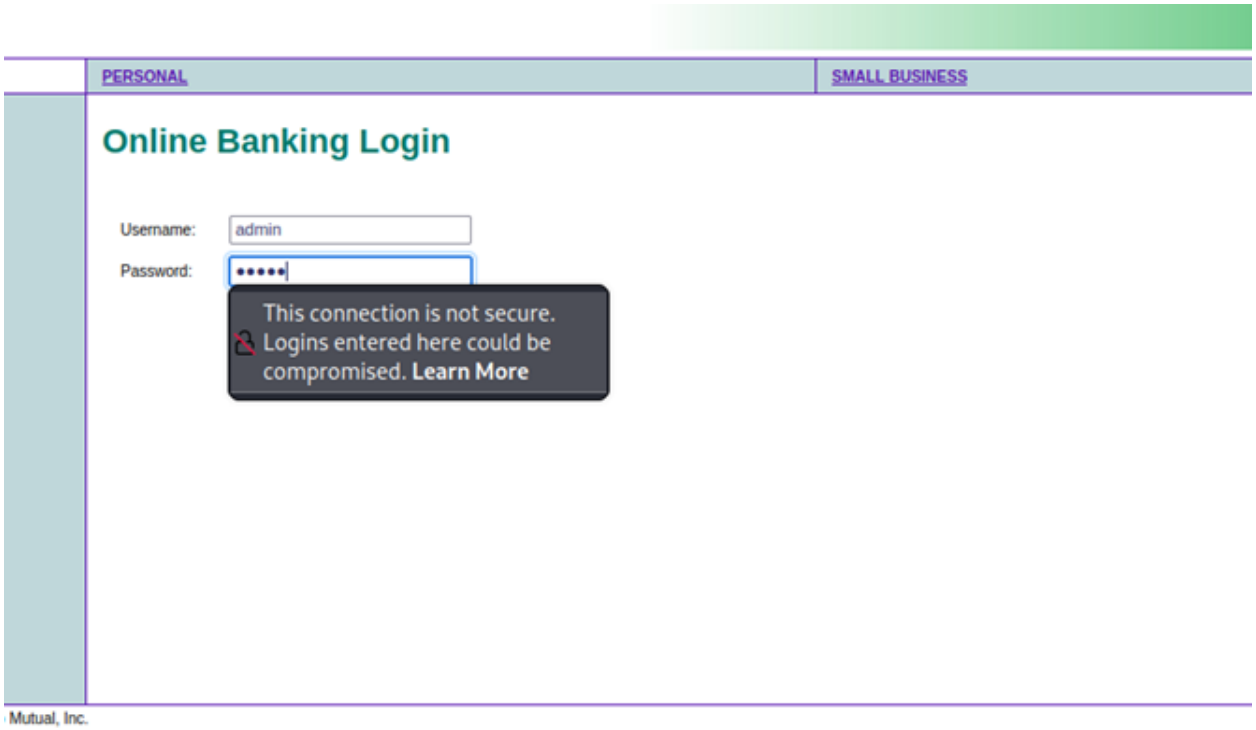
- After the attack is finished we can see that the highlighted admin admin has different length from the others. Thus it can be a probable solution. Upon checking Request and response we can assure that this is working.

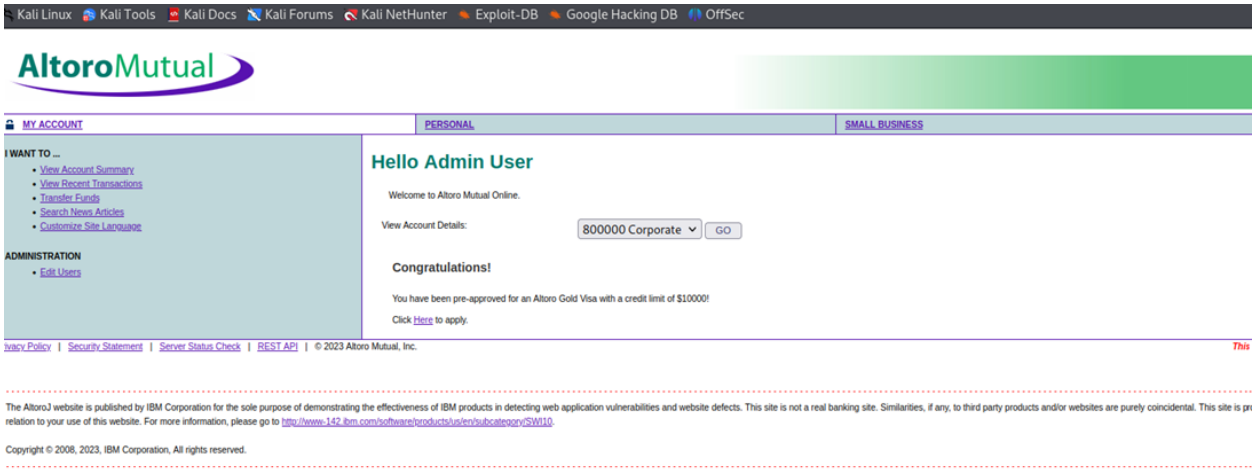


Request	Payload 1	Payload 2	Status code	Error	Timeout	Length	Comment
0			302			126	
1			302			126	
2			302			126	
3	admin		302			126	
4	test		302			126	
5	administartor		302			126	
6	User		302			126	
7	test123		302			126	
8		admin	302			126	
9		admin	302			126	
10	admin	admin	302			264	
11	test	admin	302			126	
12	administartor	admin	302			126	
13	User	admin	302			126	
14	test123	admin	302			126	
15		test	302			126	
16		test	302			126	
17	admin	test	302			126	
18	test	test	302			126	
19	administartor	test	302			126	
20	User	test	302			126	
21	test123	test	302			126	
22		1234	302			126	
23		1234	302			126	
24	admin	1234	302			126	
25	test	1234	302			126	
26	administartor	1234	302			126	
27	User	1234	302			126	
28	test123	1234	302			126	
29		12345&	302			126	
30		12345&	302			126	
31	admin	12345&	302			126	
32	test	12345&	302			126	
33	administartor	12345&	302			126	
34	User	12345&	302			126	
35	test123	12345&	302			126	
36		$admin	302			126	
37		$admin	302			126	
38	admin	$admin	302			126	
39	test	$admin	302			126	
40	administartor	$admin	302			126	
41	User	$admin	302			126	
42	test123	$admin	302			126	



- We then give the inputs in the login page and hence we are logged in.


PERSONAL
SMALL BUSINESS
Online Banking Login
Username:
admin
Password:
This connection is not secure.
Logins entered here could be
compromised. Learn More
Mutual, Inc.


Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
AltoroMutual
MY ACCOUNT
PERSONAL
SMALL BUSINESS
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.
ivacy Policy | Security Statement | Server Status Check | REST API | © 2023 Altoro Mutual, Inc.
This
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 pr
relation to your use of this website. For more information, please go to http://www-142.ibm.com/software/products/us/en/subcategory/SW110.
Copyright © 2008, 2023, IBM Corporation, All rights reserved.

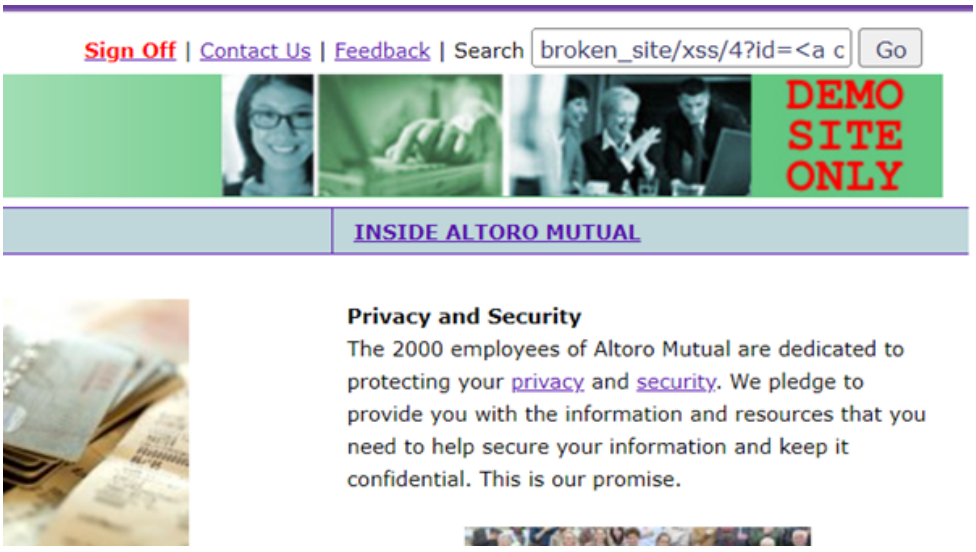
3. Cross Site Scripting (XSS)

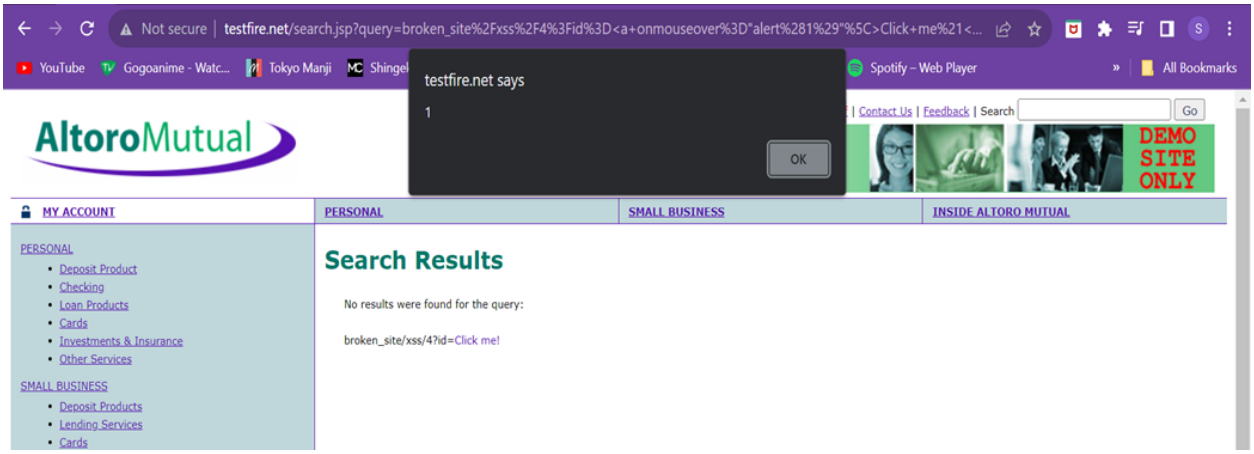
CWE-87: Improper Neutralization of Alternate XSS Syntax

Description: The product does not neutralize or incorrectly neutralizes user-controlled input for alternate script syntax.
Business Impact: Cross-Site Scripting (XSS) attacks in business can lead to data theft, user trust erosion, reputation damage, financial losses, legal consequences, operational disruptions, increased security costs, loss of competitive advantage, and customer trust issues.

Testing:

Injecting a basic a tag and seeing how the site reacts to that. Payload is **broken_site/xss/4?id=Click me!**. What we do here is basically create a link that says click me, and when we hover the mouse over it, the alert box should execute.



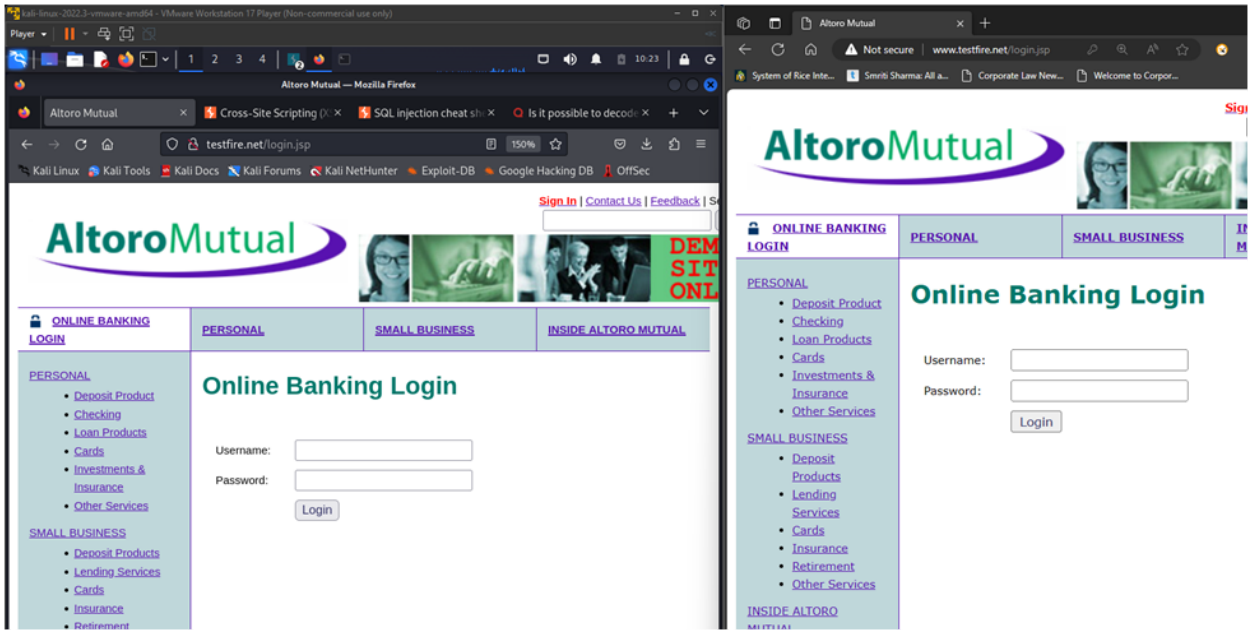


4. Improper Session Management

CWE ID for Session Fixation: CWE-384 (Session Fixation)

Description: Attacker sets or fixes a user's session ID, leading to unauthorized access. This can occur when session management is improperly implemented or lacks adequate security controls.

Business Impact: The business impact of CWE-384 (Session Fixation) can be significant, potentially leading to unauthorized access to user accounts. This can result in data breaches, loss of sensitive information, reputation damage, and financial losses due to legal and remediation costs.

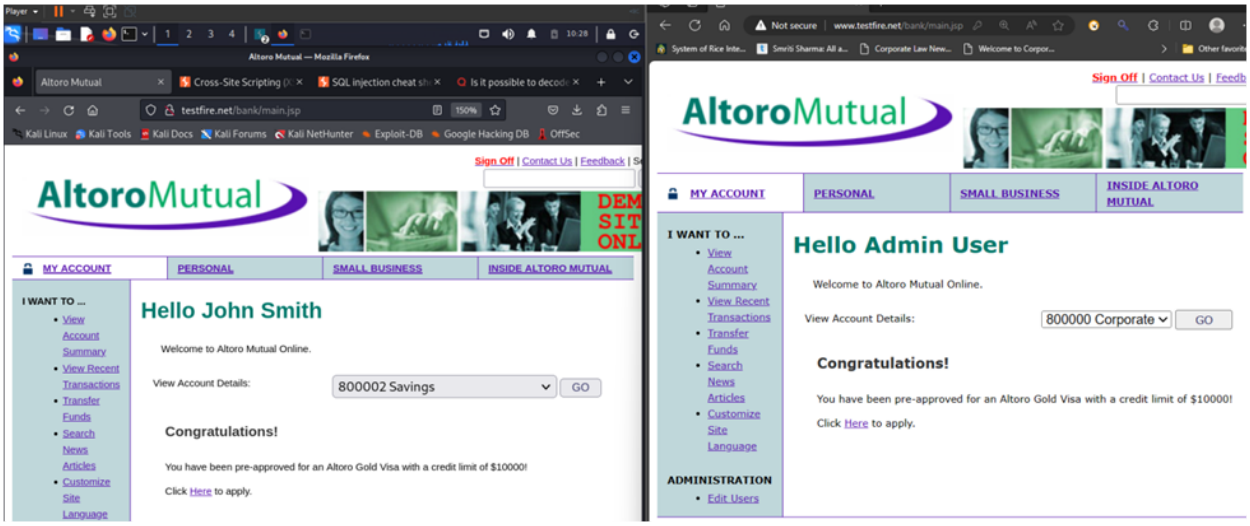


We have accessed the testfire.net website on two different browsers(one on virtual machine and another on main device).

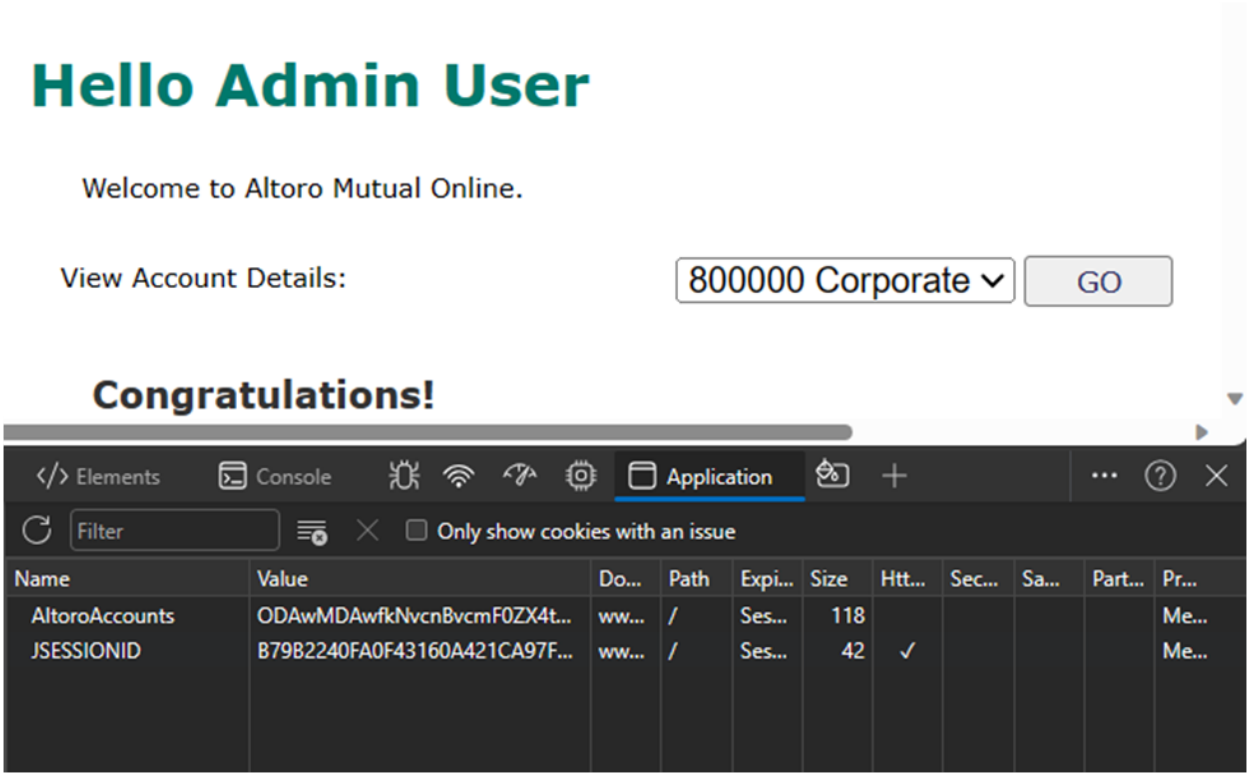
We will login as admin (u_name = admin && pass = admin) in the main machine and as John Smith (u_name = jsmith && pass = Demo1234) in the virtual machine.

Post login, we will fetch the JSESSION_ID of admin logged in account and copy that, intercept a request in the John Smith's account and replace John Smith's Session Id with admin's.

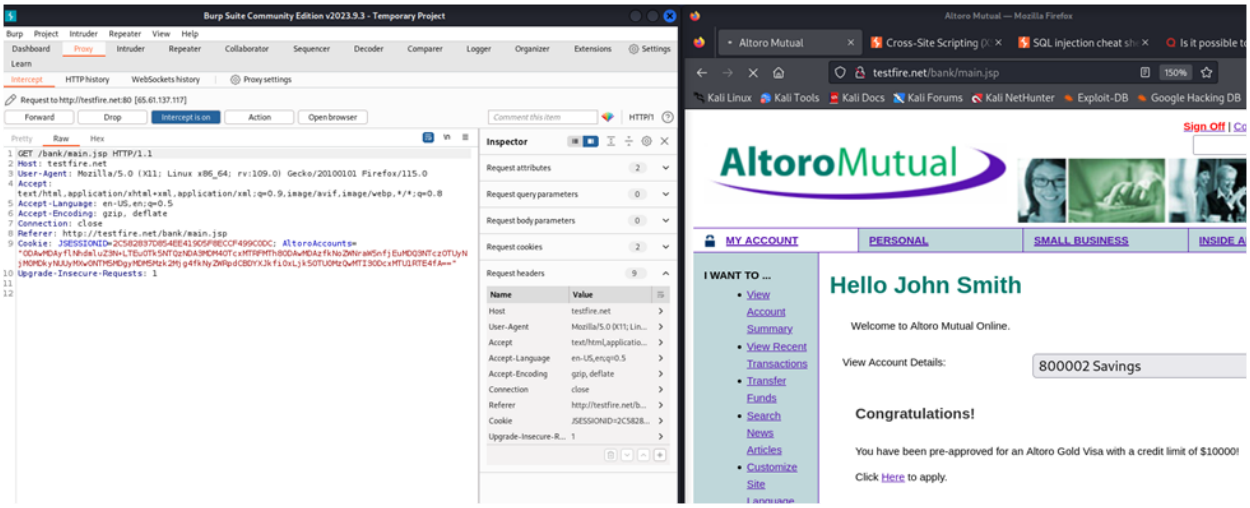
If we are able to access the admins account after this then we could possibly say that theres a security flaw of insecure Session management.



- Logged in to respective accounts.



- Fetched the Admin Session ID value.(B79B2240FA0F43160A421CA97F1343C6)



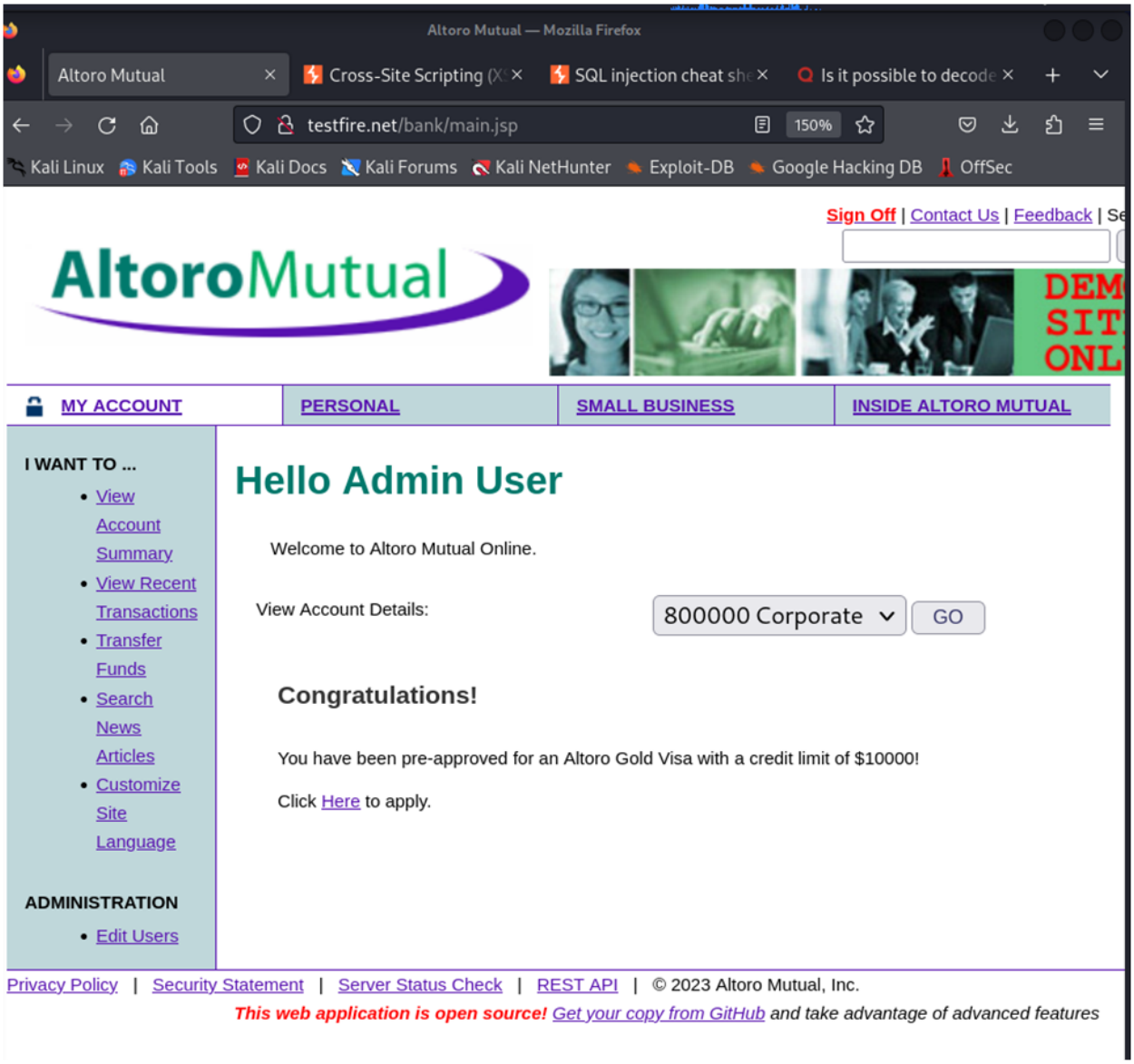
- Intercepted a request from John Smith's account.

```
GET /bank/main.jsp HTTP/1.1
Host: 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
Connection: close
Referer: http://testfire.net/bank/main.jsp
Cookie: JSESSIONID=2C582837D854EE419D5F8ECCF499C0DC; AltoroAccounts=
"ODAwMDAyflNhdmluZ3N+LTEuOTk5NTQzNDA3MDM4OTcxMTRFMTh8ODAwMDAzfkNoZWNraW5nfjEuMDQ3NTczOTUyN
jM0MDkyNUUyMXw0NTM5MDgyMDM5Mzk2Mjg4fkNyZWRpdCBDYXJkfi0xLjk5OTU0MzQwMTI3ODcxMTU1RTE4fA=="
Upgrade-Insecure-Requests: 1
```

- The intercepted request has JSESSIONID value as 2C582837D854EE419D5F8ECCF499C0DC

Pretty Raw Hex

```
GET /bank/main.jsp HTTP/1.1
Host: 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
Connection: close
Referer: http://testfire.net/bank/main.jsp
Cookie: JSESSIONID=B79B2240FA0F43160A421CA97F1343C6; AltoroAccounts=
"ODAwMDAyflNhdmluZ3N+LTEuOTk5NTQzNDA3MDM4OTcxMTRFMTh8ODAwMDAzfkNoZWNraW5nfjEuMDQ3NTczOTUyN
jM0MDkyNUUyMXw0NTM5MDgyMDM5Mzk2Mjg4fkNyZWRpdCBDYXJkfi0xLjk5OTU0MzQwMTI3ODcxMTU1RTE4fA=="
Upgrade-Insecure-Requests: 1
```



- Replaced the JSESSIONID with that of admin's

Logged in as admin

5.Broken Access Control:

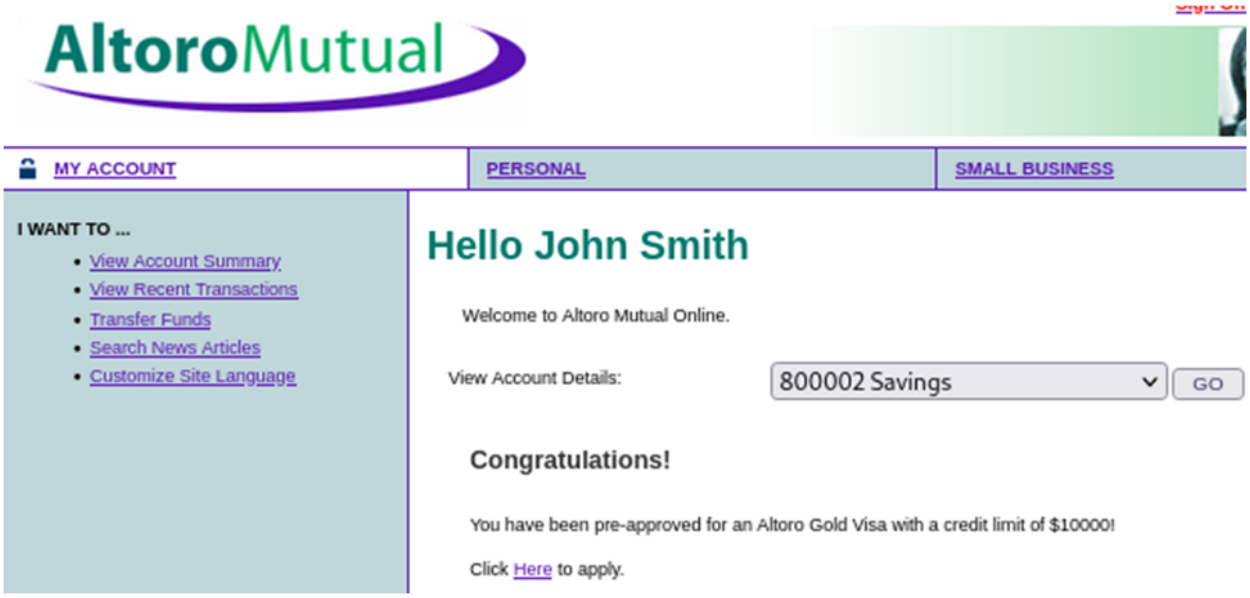
We will first login as john smith and then try to access bank account linked to any other user to view their transaction by modifying the url parameter (IDOR attack).

CWE-284: Improper Access Control.

Business Impact:

Data Breaches, Loss of Confidential Information, Financial Loss, Reputation Damage, Regulatory Compliance Issues, Operational Disruption, Lawsuits and Legal Liabilities, Resource Wastage, Loss of Competitive Advantage, Customer Trust Erosion

- Logged in as John Smith

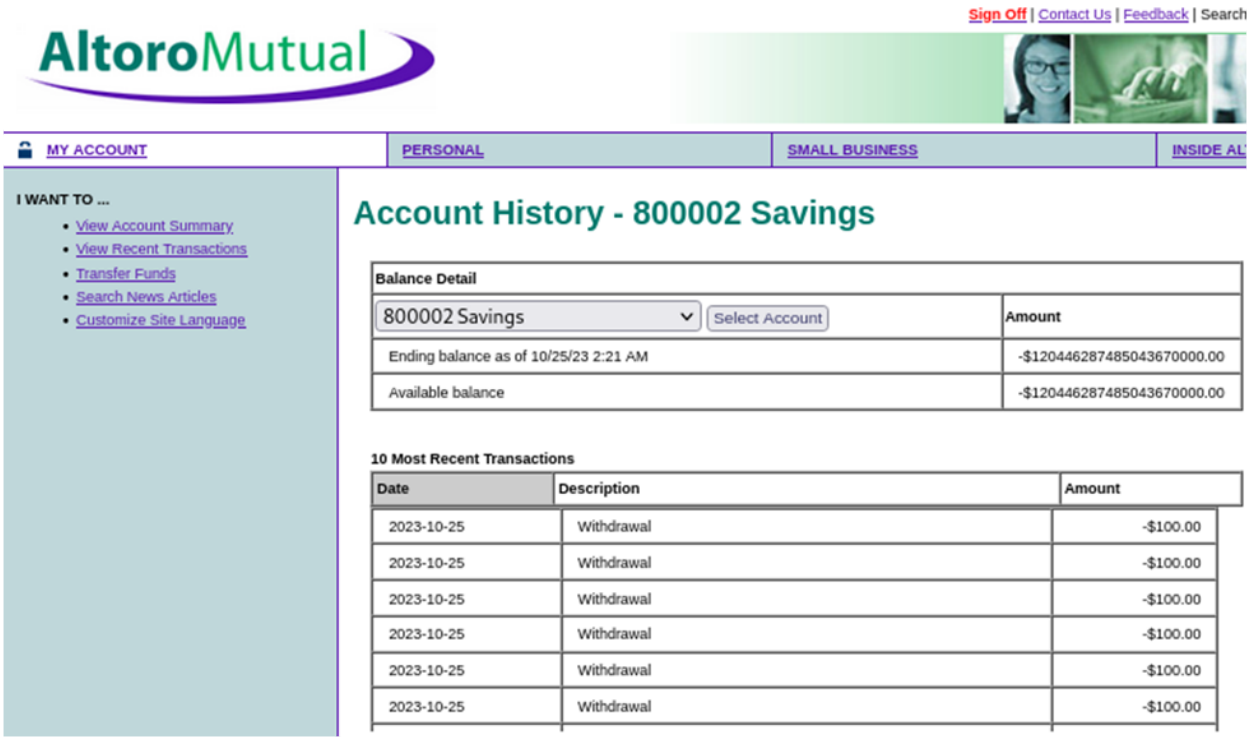


Here if we go to the view account summary section, we will notice an option to search the account history of all the accounts we posses. John Smith has access to the following accounts:

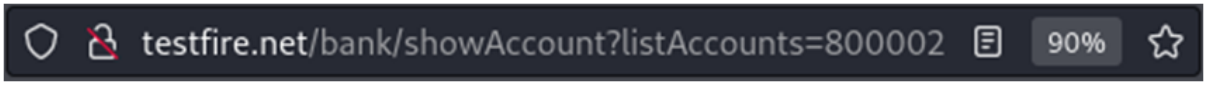
- 800002 – Savings
- 800003 – Checking
- 4539082039396288 – Credit card

If we access his Savings account, we notice account history where the 10 most recent transactions are of $100 each.

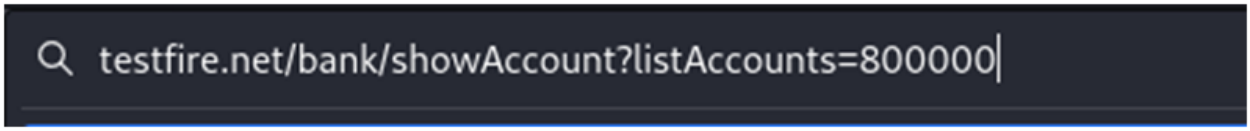
Having a look at the URL we can see the request parameter to be as listAccount=800002.



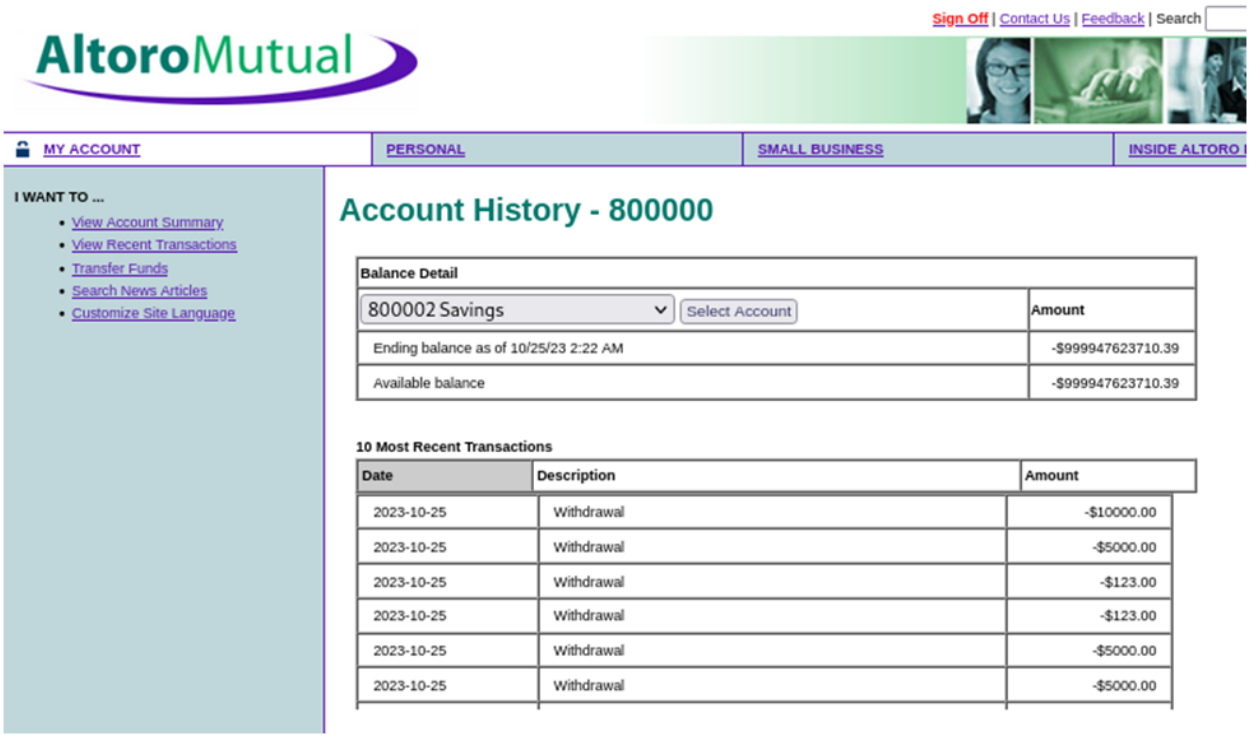
John Smith's Savings account's URL -



From here if we modify the URL to the one shown in the image below with account 800000 which technically doesn't belong to john smith (800000 Belongs to the admin user), we shouldn't be able to access it.



If we send this url request to the server we get a 200 OK response and apparently we are able to view the transaction history of account number 800000 as all the recent transaction values have changed.



Hence we have successfully exploited the IDOR (InDirect Object Reference) vulnerability.

6. HTML injection attack + ClickJacking

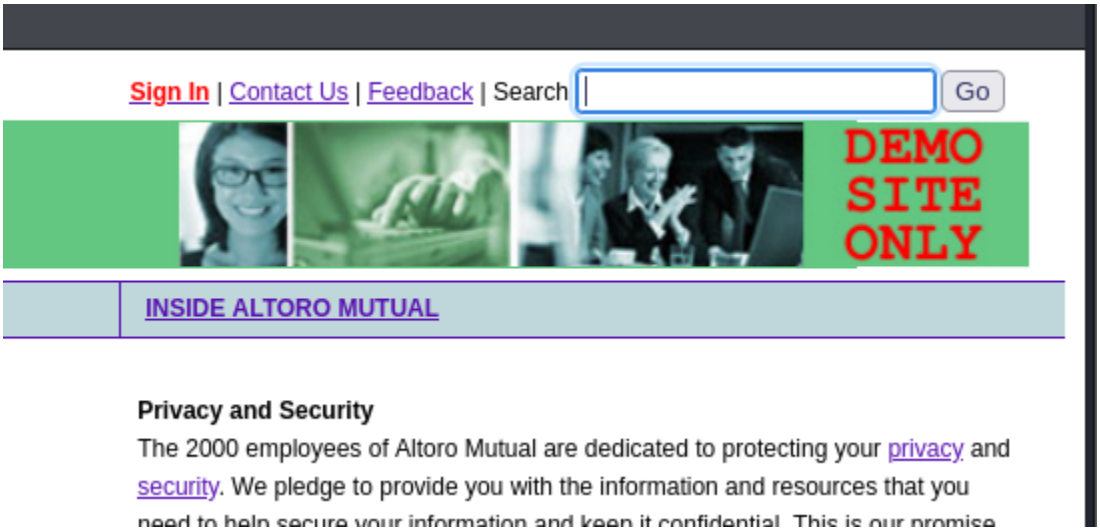
CWE=601: URL Redirection to untrusted site('Open Redirect')

Business Impact:

The impact of CWE-601 (Open Redirect) includes loss of trust, data breaches, financial loss, legal consequences, brand damage, and operational disruption.

Here a combination of man in the middle attack and html injection can be used to inject an html payload that can return a link to a malicious copy of the login page of the legitimate website seeking the credentials from the user.

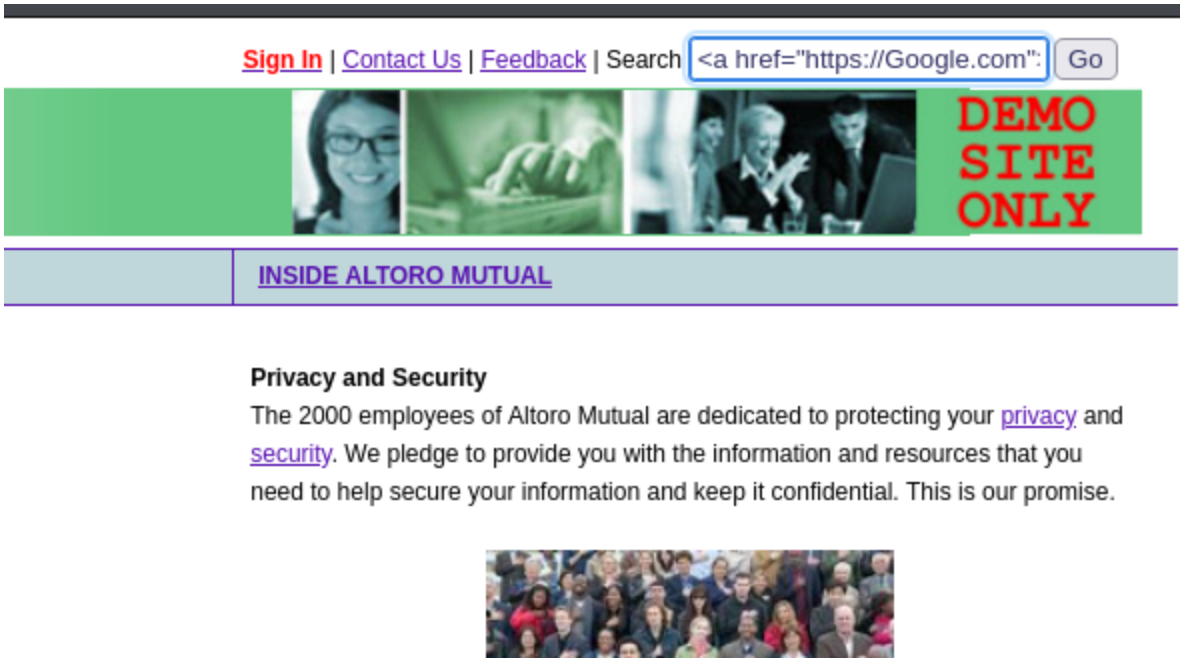
The search bar is vulnerable to html injection



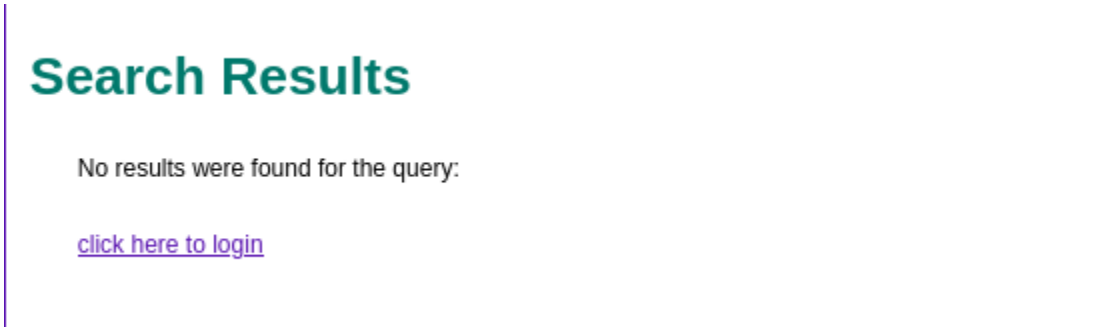
HTML payload:

click here to login

This payload's href link can be modified in certain way that it redirects to the malicious login page.



Executing the payload gives the following result.



When the click here to login hyperlink is clicked, we are redirected to the website that is linked to it.

7. Improper Input Validation

The website allows user to transfer amount greater than the user has in their account.

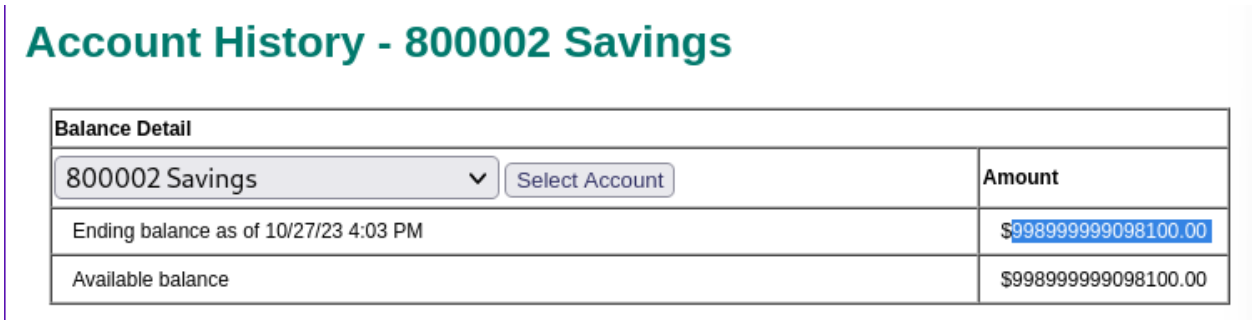
CWE (Common Weakness Enumeration): CWE-132

Business Impact:

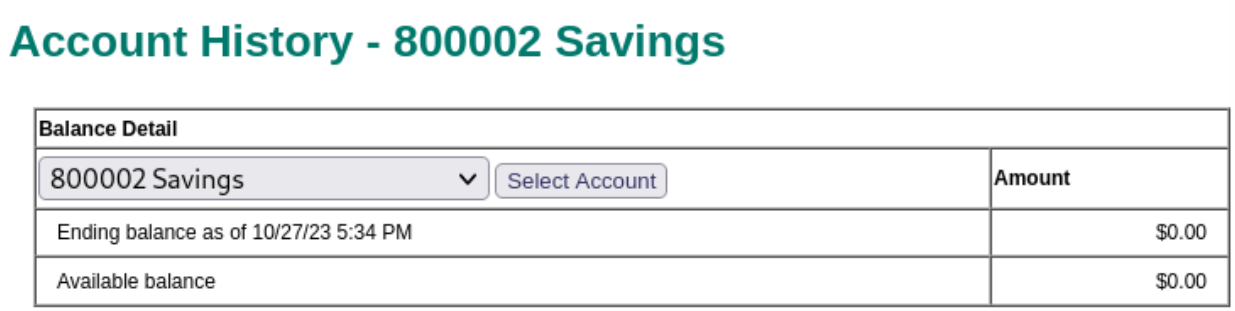
- Financial losses: Unauthorized transfers could result in substantial financial losses for both the business and affected users.
- Reputation damage: Such a security flaw can erode user trust and damage the reputation of the company.

- Legal and regulatory consequences: Violating financial regulations can result in fines and legal actions against the organization.
- Customer churn: Users may leave the platform due to concerns about their financial security.

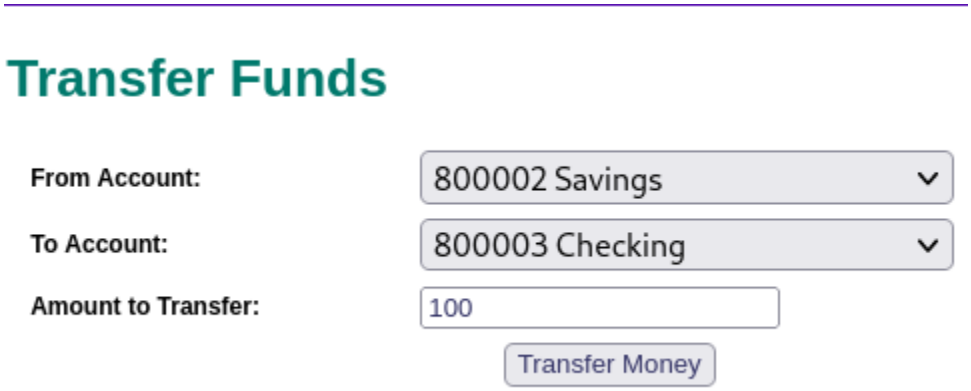
Wearing off the account 800002.



Amount to be debited to make the account balance empty is 998999999098100



Now trying to transfer amount even after the balance has worn out.



Transfer Funds

From Account: 800002 Savings

To Account: 800002 Savings

Amount to Transfer:

Transfer Money

100.0 was successfully transferred from Account 800002 into Account 800003 at 10/27/23 4:05 PM.

The transfer is successful as shown and the amount is credited in the respective account.