GROUP 2 VAPT REPORT

AltoroMutual

Target: Demo.testfire.net

Presented By:

Asharafraza Desai

NOVEMBER 23, 2025

Tables of Contents

1. Executive Summary

1.1 Purpose of Assessment:

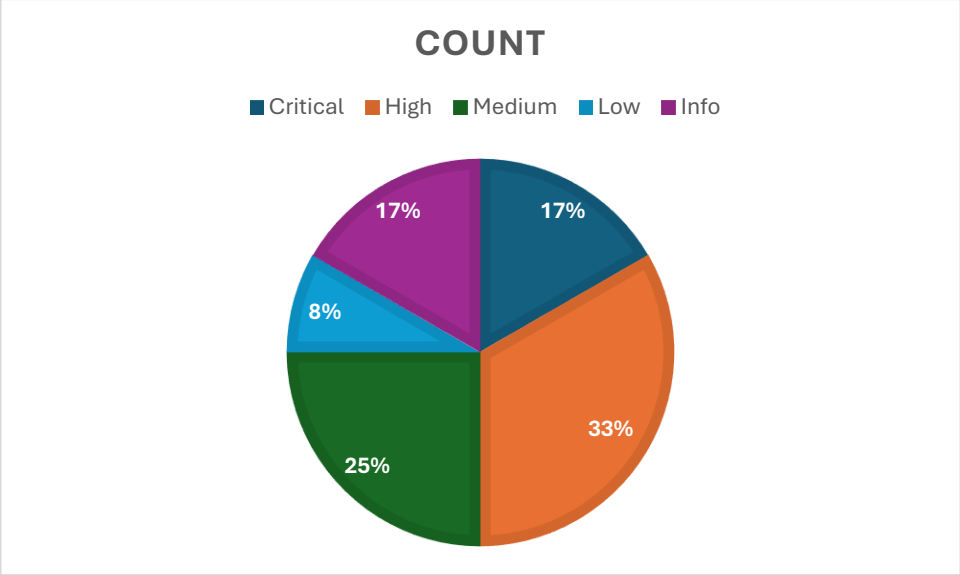
This security assessment was conducted to identify critical vulnerabilities in the Altoro Mutual online banking application (demo.testfire.net) that could lead to unauthorized access to customer accounts, financial data exposure, or system compromise. The engagement simulated real-world attack scenarios to evaluate the application's security controls and provide actionable recommendations for risk mitigation.

Objective of Assessment

- Identify security vulnerabilities in online banking platform
- Assess risks to customer data and financial assets
- Evaluate effectiveness of existing security controls
- Provide actionable remediation roadmap

1.2 Summary of Findings:

Number	Vulnerability	Severity
1	SQL Injection	Critical
2	Authentication Bypass	Critical
3	HTTP Request Smuggling	High
4	No Rate Limit	High
5	Static Session	High
6	Insecure Direct Object Reference (IDOR)	High
7	Reflected Cross-Site Scripting (XSS)	Medium
8	Clickjacking	Medium
9	Clear text Submission of Password	Medium
10	HTML Injection	Low
11	Default Credential	Info



2. Scope of Work:

- Target URL: https://demo.testfire.net
- Testing Type: Black Box Testing
- Testing Methodology: OWASP Web Application Penetration Testing Guidelines
- Testing Environment: Production Environment
- Tools Used:
  • Burp Suite
  • Nmap

3. Limitations and Assumptions:

- The tester did not engage in any testing outside the agreed-upon scope.
- Testing was conducted over 7 days to minimize disruption to the production environment.
- Testing activities included controlled exploitation of vulnerabilities that did not harm the production environment.
- Any payloads used to exploit vulnerabilities were documented in detail.
- Denial-of-Service (DoS) attacks were strictly prohibited during this engagement.

4. Findings & Vulnerabilities

4.1 SQL Injection

ID 1

Severity Critical

Affected URL http://testfire.net/login.jsp

Vuln-parameter uid & passw

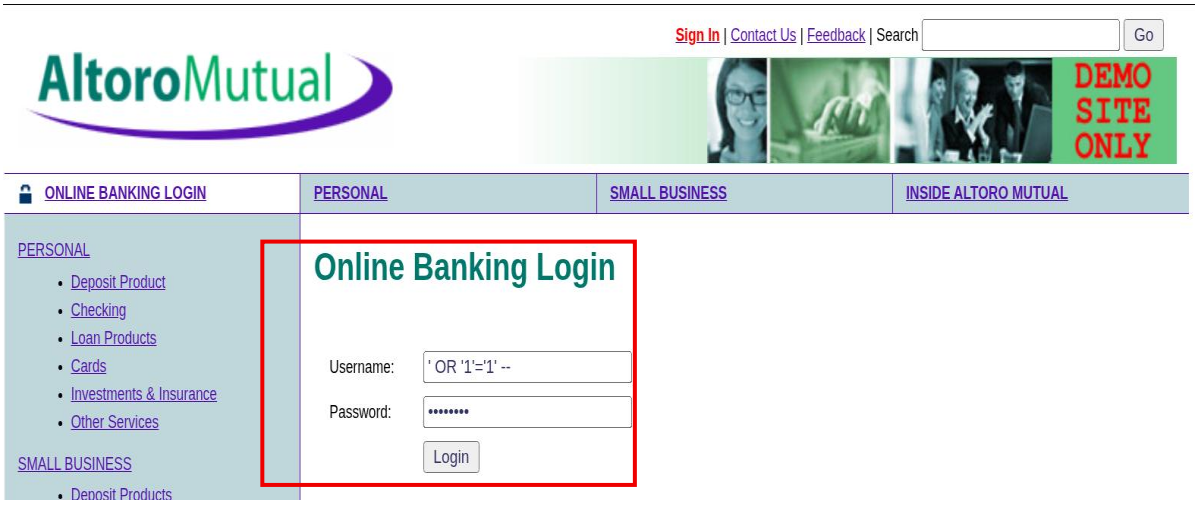
Description:

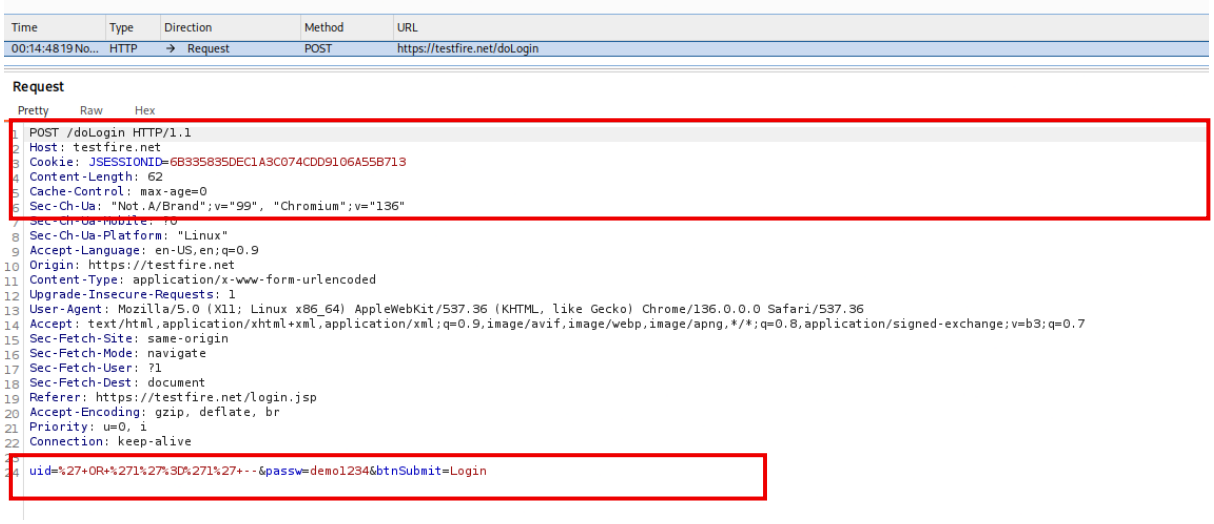
A critical SQL Injection vulnerability was identified in the login functionality, allowing attackers to bypass authentication and gain unauthorized access to user accounts. The application fails to properly sanitize user input in the username field, enabling the execution of malicious SQL queries.

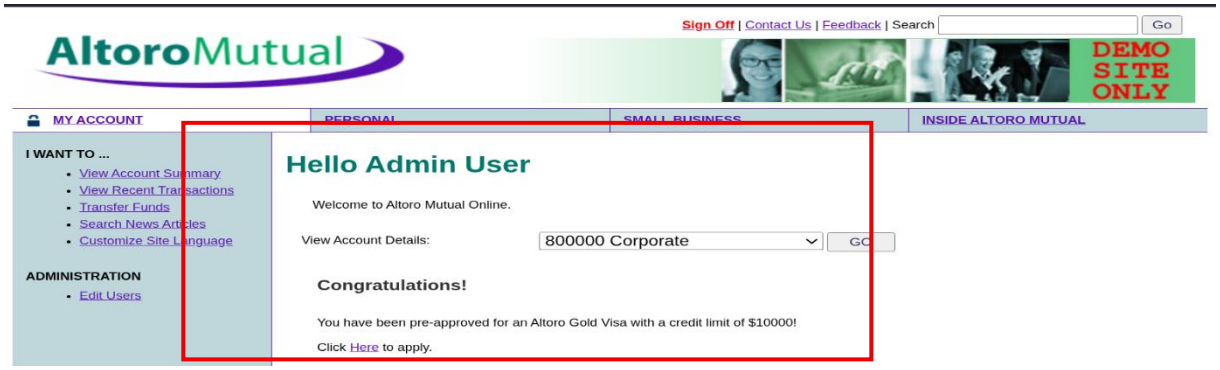
Proof of Concept (PoC):

Username: 'OR '1'='1' --

Password: [Anything]







Impact:

- Full database access

- Unauthorized user authentication bypass

- Data exfiltration

Recommendation:

- Use **prepared statements** and parameterized queries

- Implement input validation and sanitization

4.2 Authentication Bypass

ID **2**

Severity **Critical**

Affected URL http://testfire.net/login.jsp

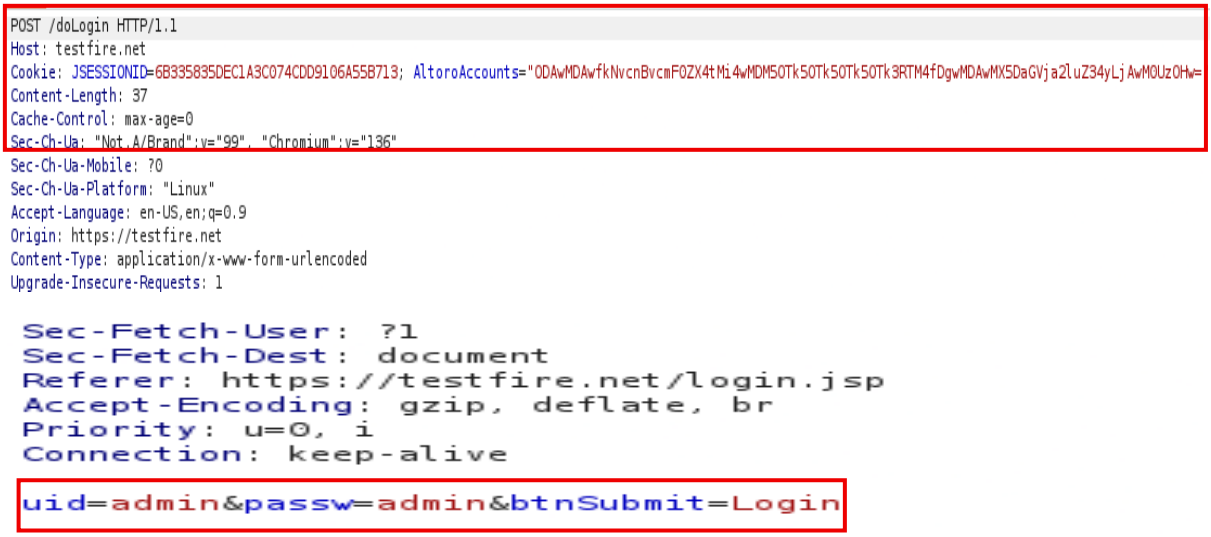
Description:

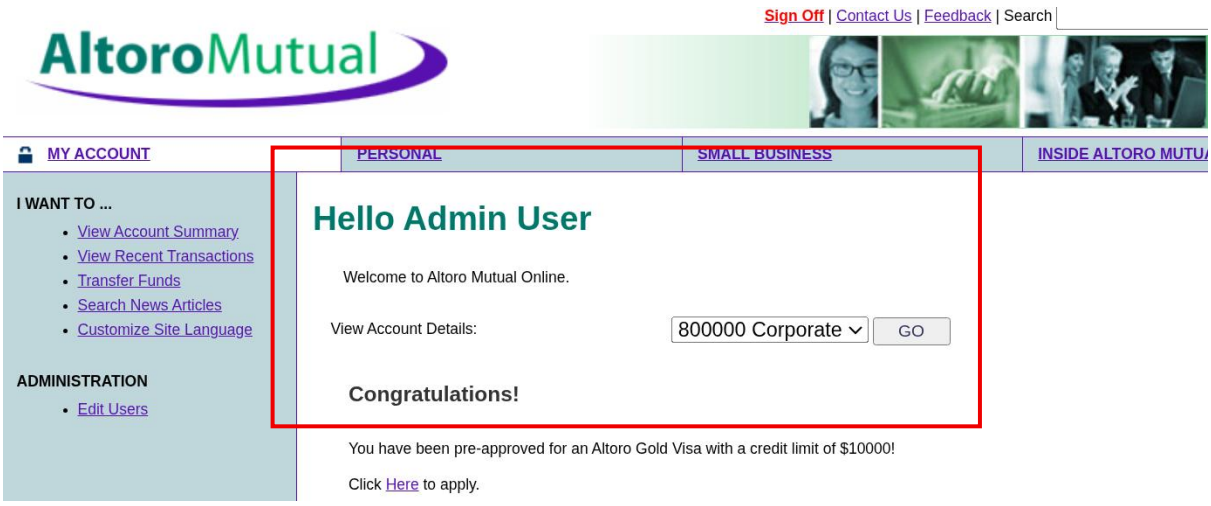
The login mechanism lacks proper authentication controls, allowing attackers to access the application using known default credentials. The system fails to enforce secure credential policies or detect common default password usage.

Proof-of-Concepts (PoC):

Username: Admin

Password: Admin





Impact:

- Unauthorized access to administrative functions and sensitive data

- Potential for financial fraud and data manipulation

Recommendation:

- Eliminate all default credentials and enforce strong password policies

- Implement account lockout mechanisms after failed login attempts

- Deploy multi-factor authentication for administrative accounts

4.3 HTTP Request Smuggling:

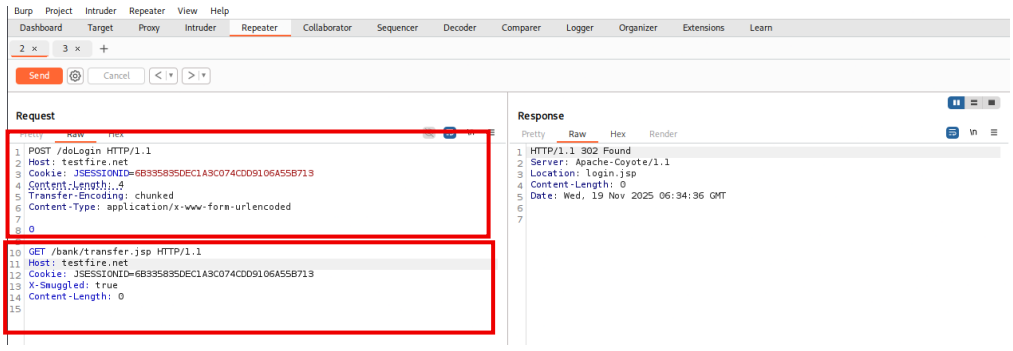
ID 3

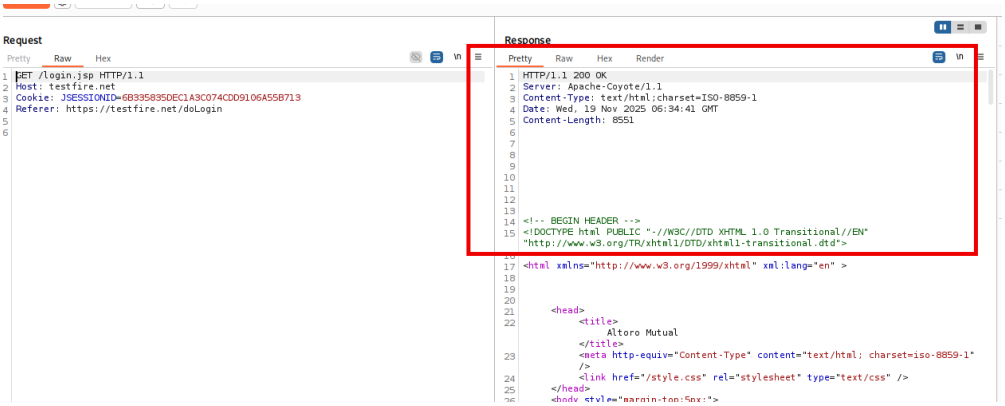
Severity High

Affected URL http://testfire.net/login.jsp

Description:

The application contains HTTP Request Smuggling vulnerabilities due to improper parsing of HTTP request headers. This allows attackers to manipulate request sequences and bypass security controls.





Impact:

- Cache poisoning and security control bypass

- Unauthorized access to restricted endpoints

Recommendation:

- Implement strict HTTP protocol validation

- Update web server to latest secure version

4.4 No Rate Limit:

ID 4

Severity HIGH

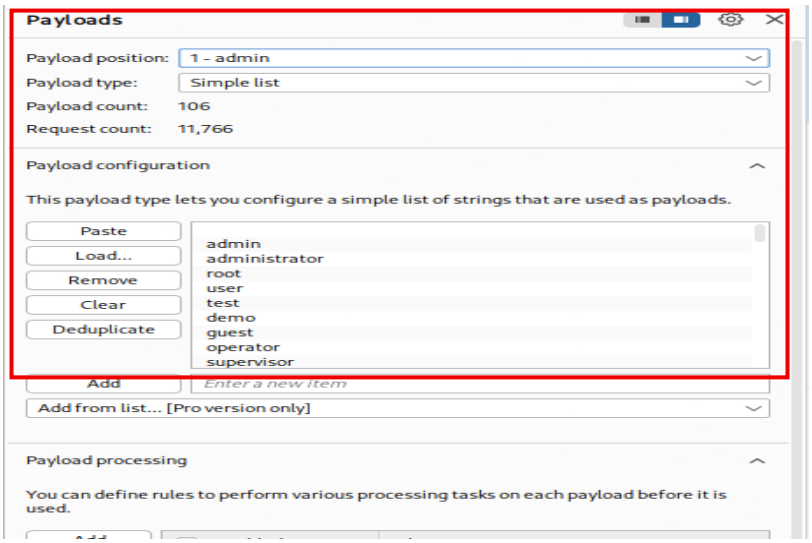
Affected URL http://testfire.net/login.jsp

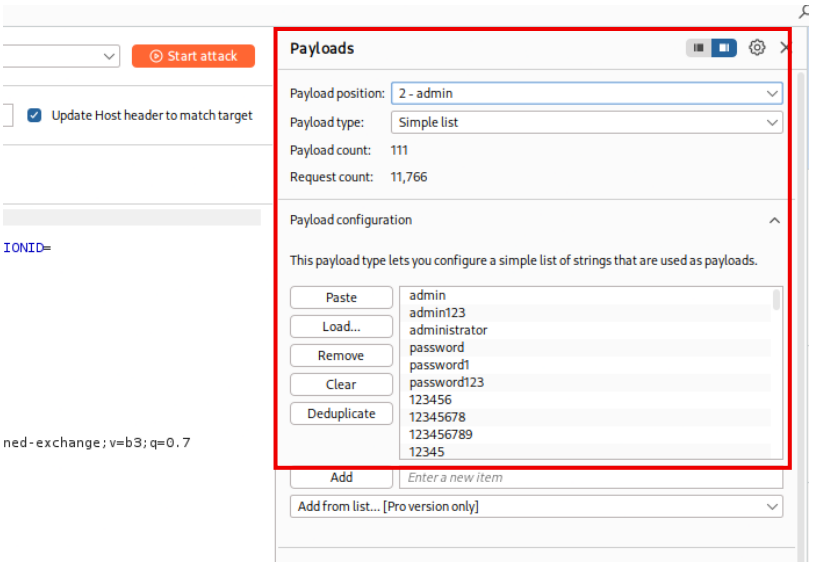
Description:

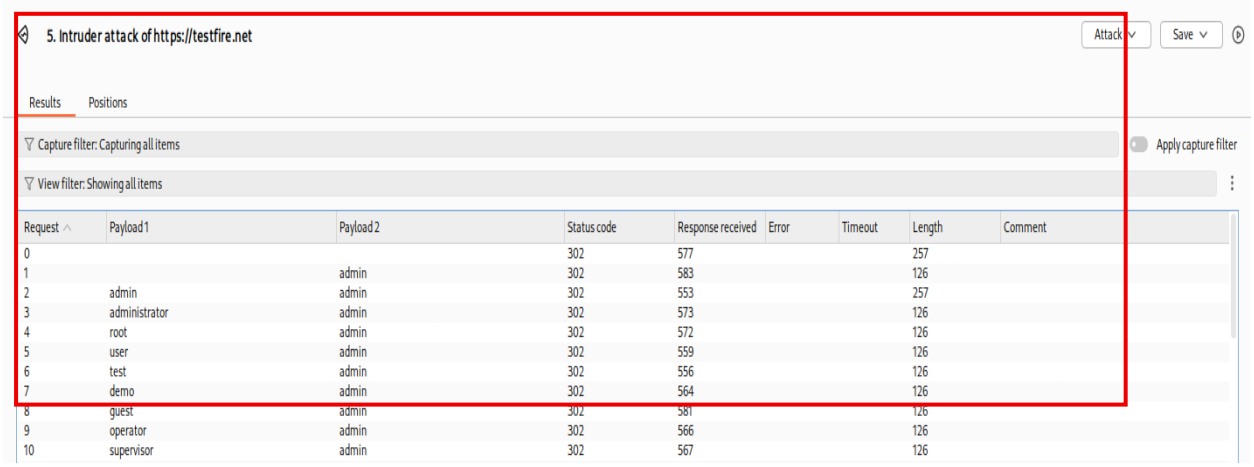
The login page allows unlimited login attempts without any rate limiting mechanisms, making it vulnerable to brute-force attacks.

Proof-of-Concepts (PoC):

The HTTP login request was intercepted and sent to Burp Suite Intruder, where dictionary-based brute-force attacks were successfully executed without any account lockout or throttling.







Impact:

- Account takeover through credential brute-forcing

- Increased risk of unauthorized access

Recommendation:

- Implement account lockout after 5-10 failed attempts

- Add rate limiting and CAPTCHA challenges

4.5 Static Session ID:

ID 5

Severity High

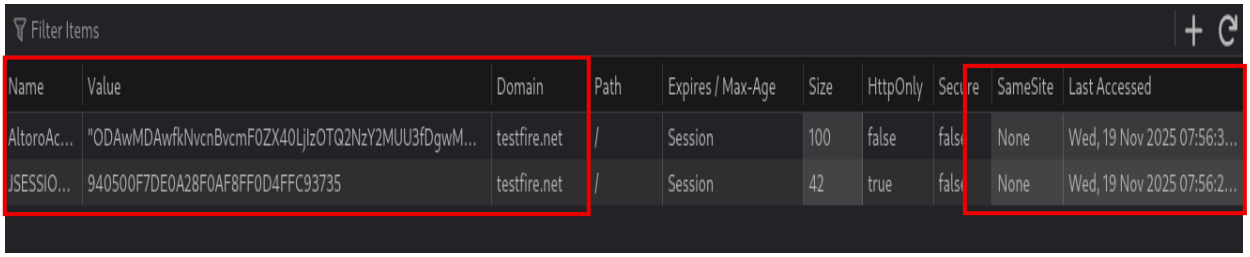
Affected URL http://testfire.net/login.jsp

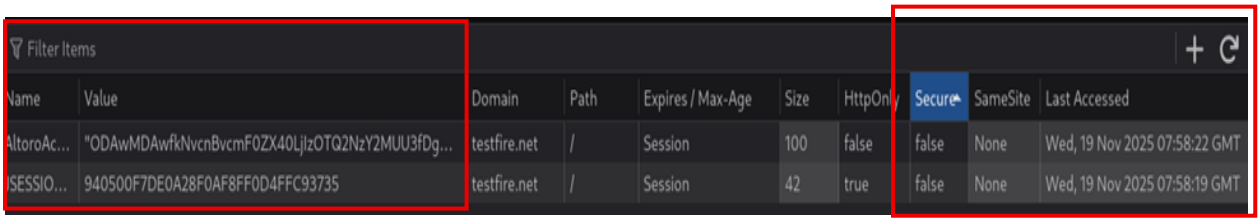
Description:

Session IDs remain static and do not change upon login or logout, making the application vulnerable to session fixation attacks.

Proof-of-Concept (PoC):

Multiple login and logout cycles were performed with the "admin" user account, confirming that the same session ID value persists without regeneration or invalidation





Impact:

- Session hijacking and account takeover

- Persistent unauthorized access

Recommendation

- Regenerate session IDs after authentication

- Invalidate sessions upon logout

4.6 Reflected Cross-Site Scripting

ID **6**

Severity **Medium**

Affected URL:
http://testfire.net/search.jsp?query=%3Cscript%3Ealert%28%22XSS1%22%29%3C%2Fscript%3E

http://testfire.net/search.jsp?query=%3Cscript%3Ealert%28%22XSS2%22%29%3C%2Fscript%3E

Vuln-Parameter: query & name & query

Description:

The application fails to properly sanitize user input in search functionality, allowing Reflected Cross-Site Scripting (XSS) attacks. This vulnerability enables attackers to inject and execute malicious JavaScript in users' browsers.

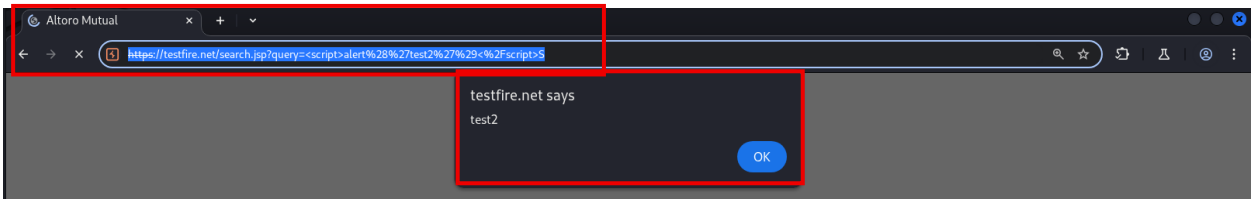
Proof-of-Concept (PoC):

During testing of "altoro.testfire.net", the search field was identified as vulnerable. When searching for "anything", the application returned to the search.jsp page with the parameter "query" containing the search value.

The JavaScript payload

“`<script>alert("XSS1")</script>”

was injected into the search parameter, which successfully executed and displayed an alert popup, confirming the XSS vulnerability.



Feedback

Our Frequently Asked Questions area will help you with many of your inquiries.
If you can't find your question, return to this page and use the e-mail form below.

IMPORTANT! This feedback facility is not secure. Please do not send any account information in a message sent from here.

To: **Online Banking**

Your Name: <script>alert("XSS")</script>

Your Email Address:

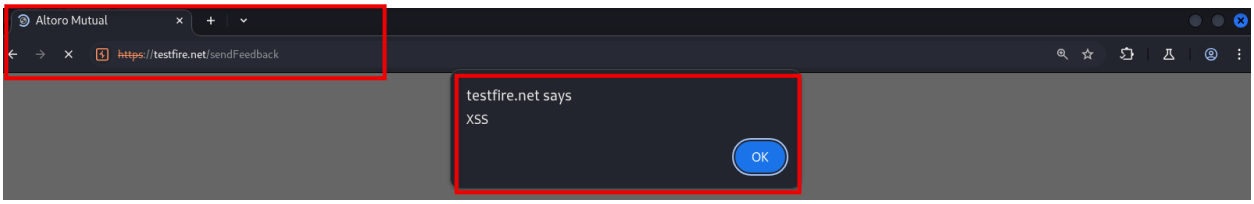
Subject:

Question/Comment:

Request

Pretty Raw Hex

```
1 POST /sendFeedback HTTP/1.1
2 Host: testfire.net
3 Cookie: JSESSIONID=6B9BCF6CE6D96AB3D50E877449C58EBF; AltoroAccounts=
"ODAwMDAwfkNvcnBvcmF0ZX4tNC40Njc3NzgwMjc2MzcxMzlFMTN8ODAwMDAxfkNoZWNraW5nfjQuNDY3NzgzMjc3NjE5OTQ0RTEzfA=="
4 Content-Length: 124
5 Cache-Control: max-age=0
6 Sec-Ch-Ua: "Not.A/Brand";v="99", "Chromium";v="136"
7 Sec-Ch-Ua-Mobile: ?0
8 Sec-Ch-Ua-Platform: "Linux"
```



Impact:

- Session hijacking and cookie theft

- Malicious redirects and defacement

Recommendation:

- Implement output encoding and input validation

- Use Content Security Policy (CSP) headers

4.7 Click Jacking

ID **7**

Severity **Medium**

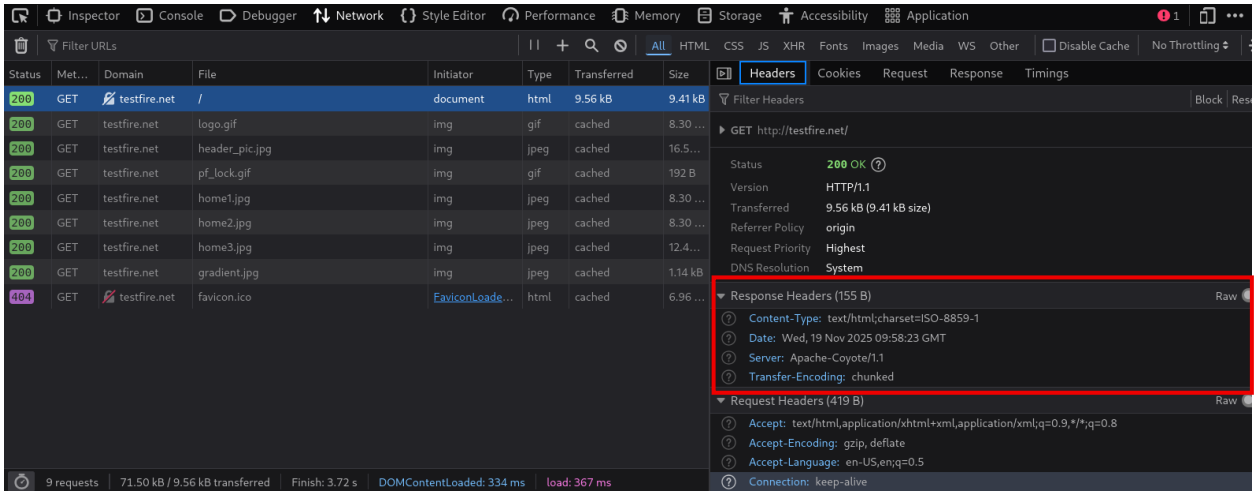
Affected URL http://demo.testfire.net/

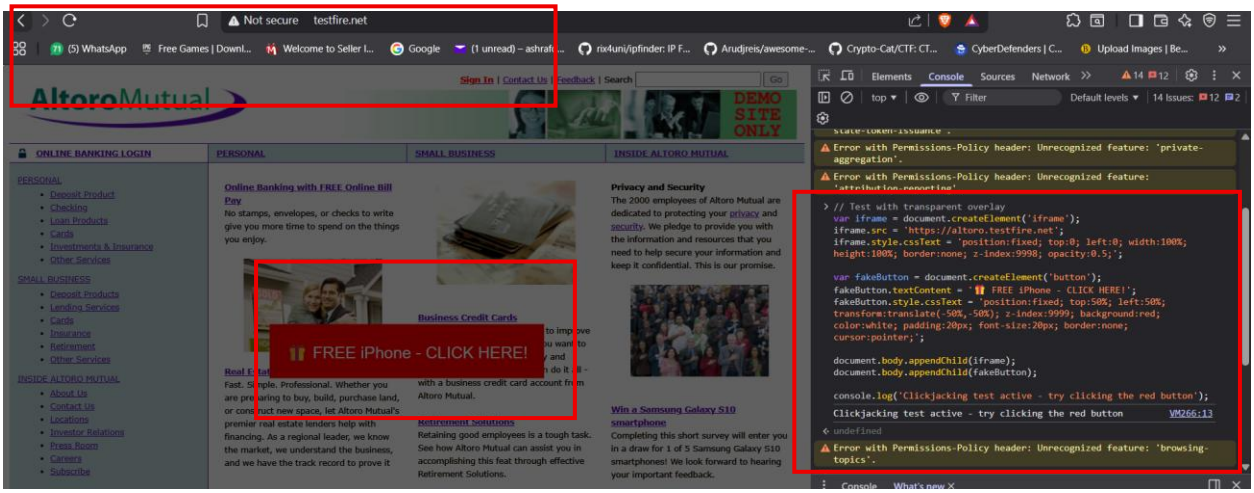
Description:

The application lacks proper frame-busting protections, making it vulnerable to clickjacking attacks that can trick users into performing unintended actions.

Proof-of-Concept (PoC):

I noticed that the "X-Frame-Options" is not found in the HTTP response, which indicated that the web application might be vulnerable to clickjacking. Therefore, I wrote a simple JavaScript code script to embed "altoro.testfire.net" in an external website..





Impact:

- Unauthorized fund transfers

- Account settings modification

- Forced user actions without consent

Recommendation:

- Implement X-Frame-Options: DENY header

- Deploy Content Security Policy frame-ancestors directive

- Use frame-busting JavaScript scripts

4.8 Cleartext Submission of Password:

ID **8**

Severity **Medium**

Affected URL http://testfire.net/login.jsp

Description:

Passwords are transmitted over HTTP instead of HTTPS.

Proof-of-Concept (PoC):

I tried to intercept the http request when I login in with

Username: jsmith

Password: demo1234

<u>Impact:</u>

- Man-in-the-Middle (MitM) attacks can intercept user credentials

- Unauthorized access risk due to exposed passwords in transit

<u>Recommendation:</u>

- Enforce HTTPS with TLS 1.2+

4.9 Password Field Autocomplete enabled

ID 9

Severity Low

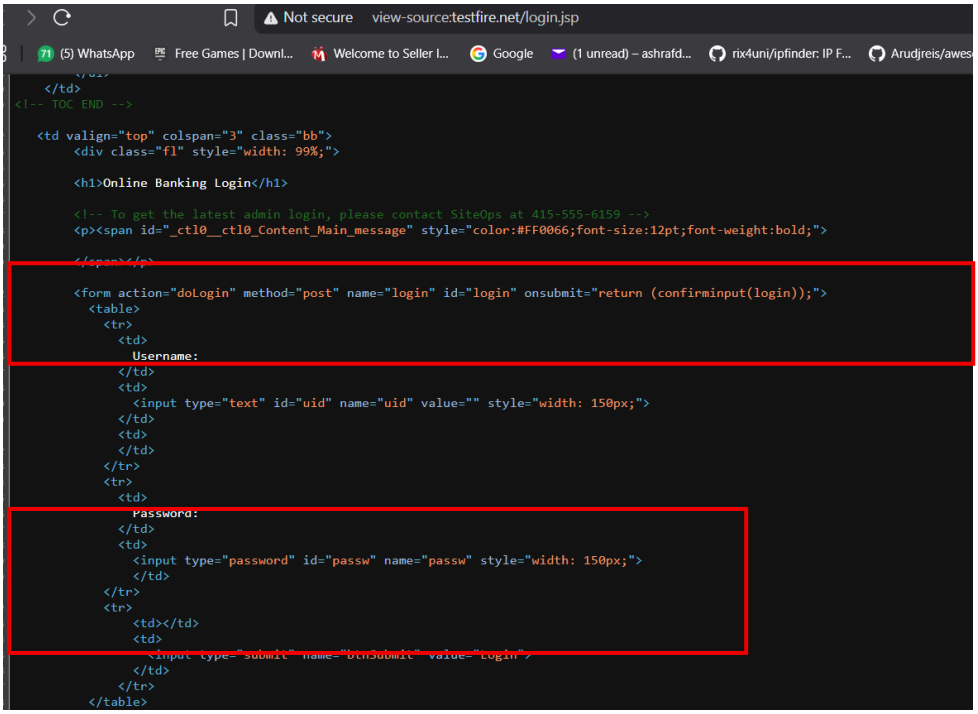
Affected URL http://testfire.net/login.jsp

Description:

The password field has autocompleted functionality enabled, allowing browsers to store credentials insecurely.

Proof-of-Concept (PoC):

The login form at /login.jsp contains password fields with autocomplete="on" attribute, enabling browser password saving on shared or public systems.



Impact:

- Credential exposure on shared computers

- Increased risk of unauthorized access

Recommendation:

- Disable autocomplete on password fields

- Implement secure session management

4.10 Default Credentials (admin/admin)

ID	10
Severity	Info
Affected URL	http://testfire.net/login.jsp

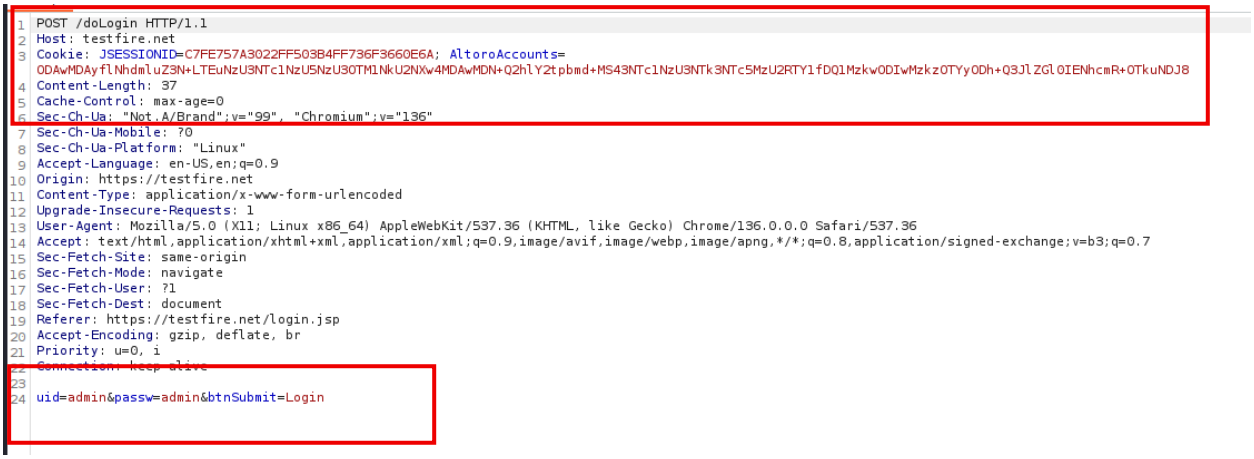
Description:

The application allows authentication using weak default credentials.

Proof-of-Concept (PoC):

Username: admin

Password: admin



```
POST /doLogin HTTP/1.1
Host: testfire.net
Cookie: JSESSIONID=C7FE757A3022FF503B4FF736F3660E6A; AltoroAccounts=
ODAwMDAyflNhdmluZ3N+LTEuNzU3NTc1NzU5NzU3OTM1NkU2NXw4MDAwMDN+Q2hlY2tpbmd+MS43NTc1NzU3NTk3NTc5MzU2RTY1fDQ1MzkwODIwMzkzOTYyODh+Q3JlZGl0IENhcmR+OTkuNDJ8
Content-Length: 37
Cache-Control: max-age=0
Sec-Ch-Ua: "Not.A/Brand";v="99", "Chromium";v="136"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Linux"
Accept-Language: en-US,en;q=0.9
Origin: https://testfire.net
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://testfire.net/login.jsp
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
Connection: keep-alive

uid=admin&passw=admin&btnSubmit=Login
```

Impact:

- Unauthorized access to admin panel and user data.

Recommendation:

- Enforce strong password policies.

- Disable default accounts upon deployment

4.11 Attribute Injection / HTML Tag Injection

ID 11

Severity Low

Affected URL http://testfire.net/

Description:

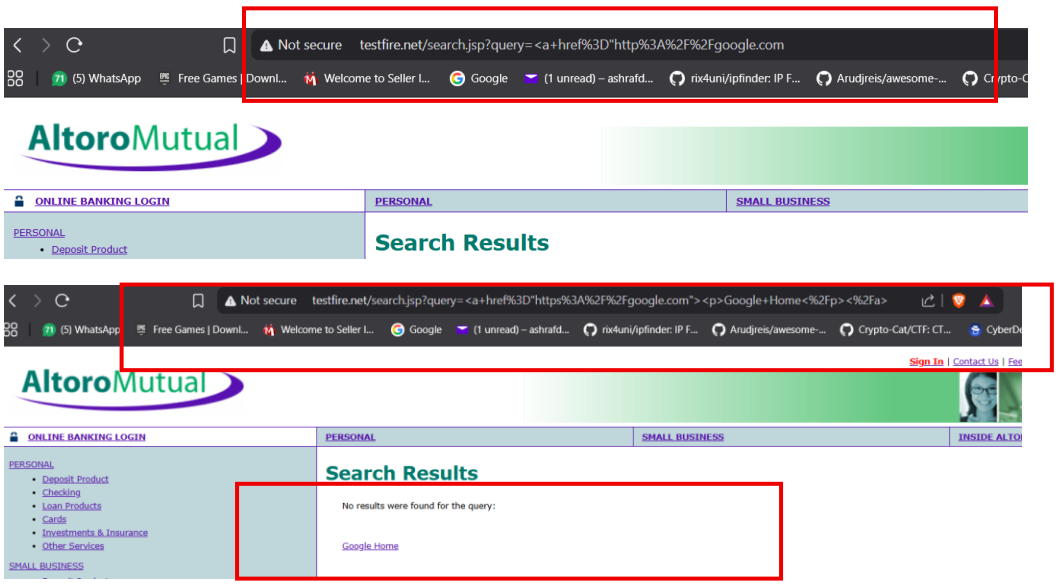
The application fails to properly sanitize user input in the search functionality, allowing HTML tag injection that can modify page content and redirect users.

Proof-of-Concept (PoC):

During testing in the search box, the payload

“ Google Home ”

was injected, which successfully rendered as a clickable link redirecting to an external website.



Impact:

- Potential phishing attacks

- User redirection to malicious websites

Recommendation:

- Implement proper output encoding

- Use Content Security Policy headers

4.12 Insecure Direct Object Reference (IDOR)

ID	12
Severity	Medium
Affected URL	http://testfire.net/bank/showAccount?listAccounts=800000 http://testfire.net/bank/showAccount?listAccounts=800002

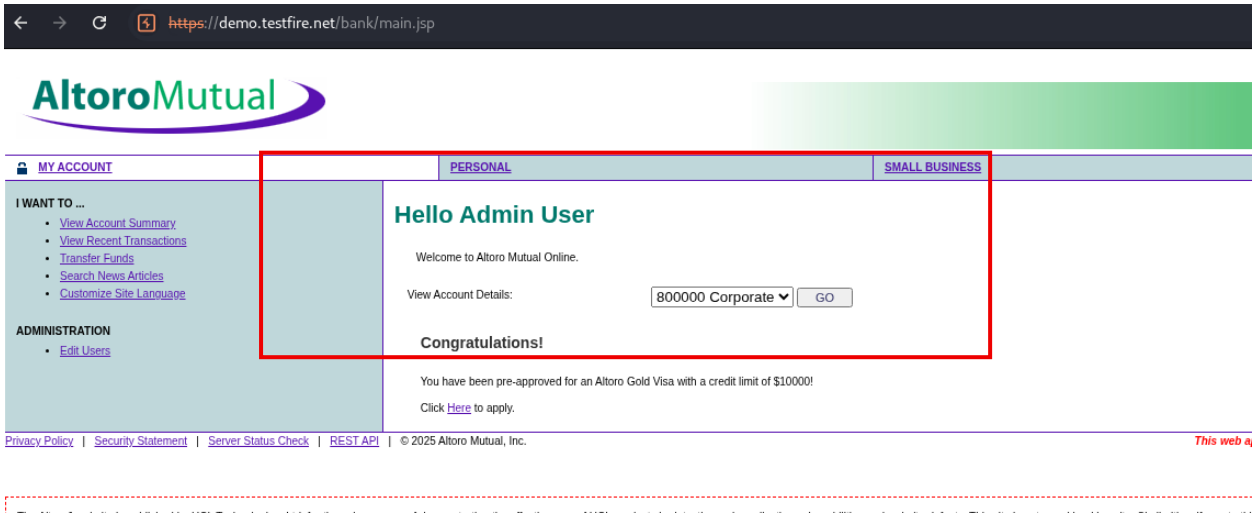
Description:

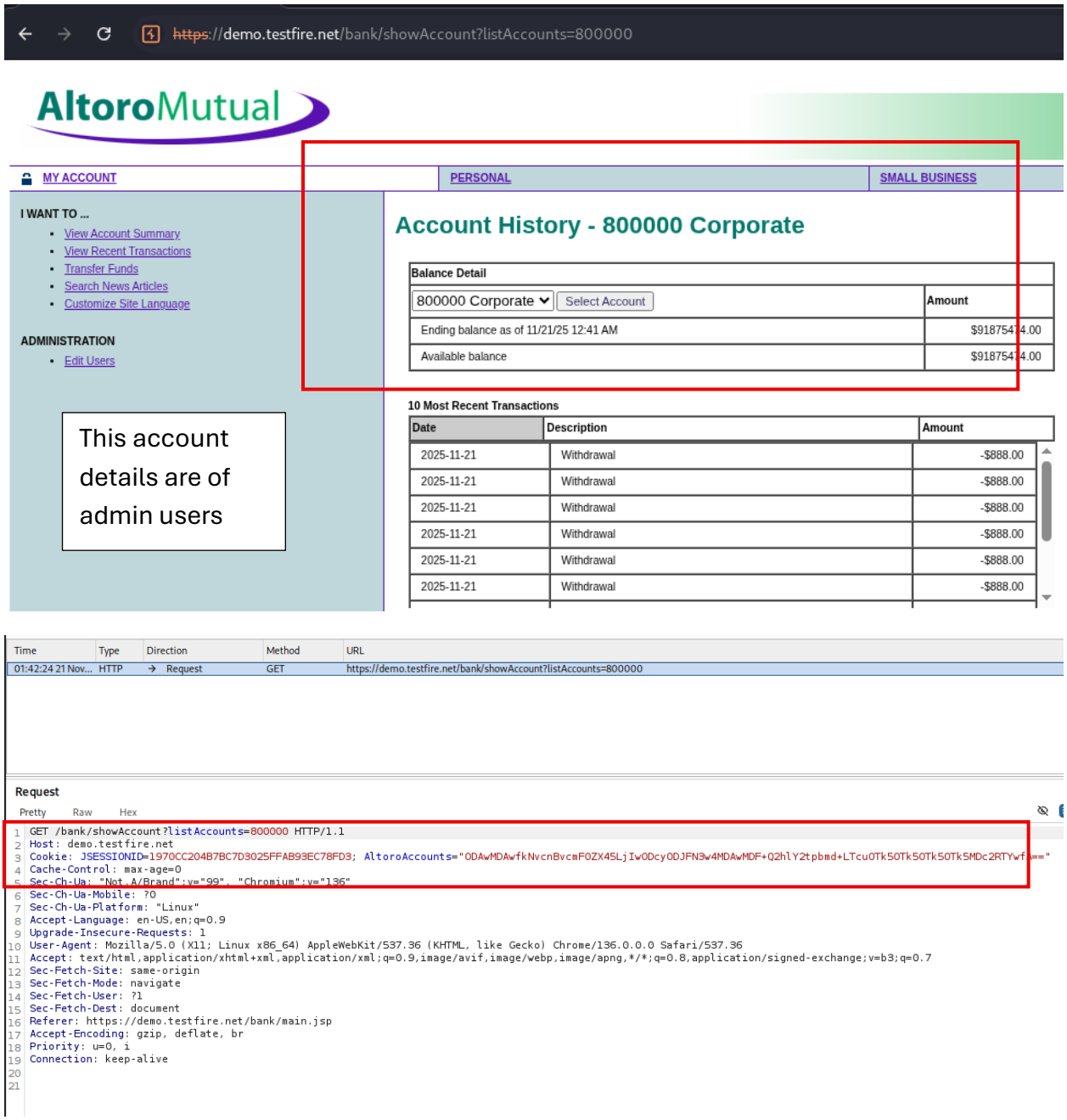
The application fails to implement proper access controls on account endpoints, allowing users to access unauthorized account data by manipulating account parameters.

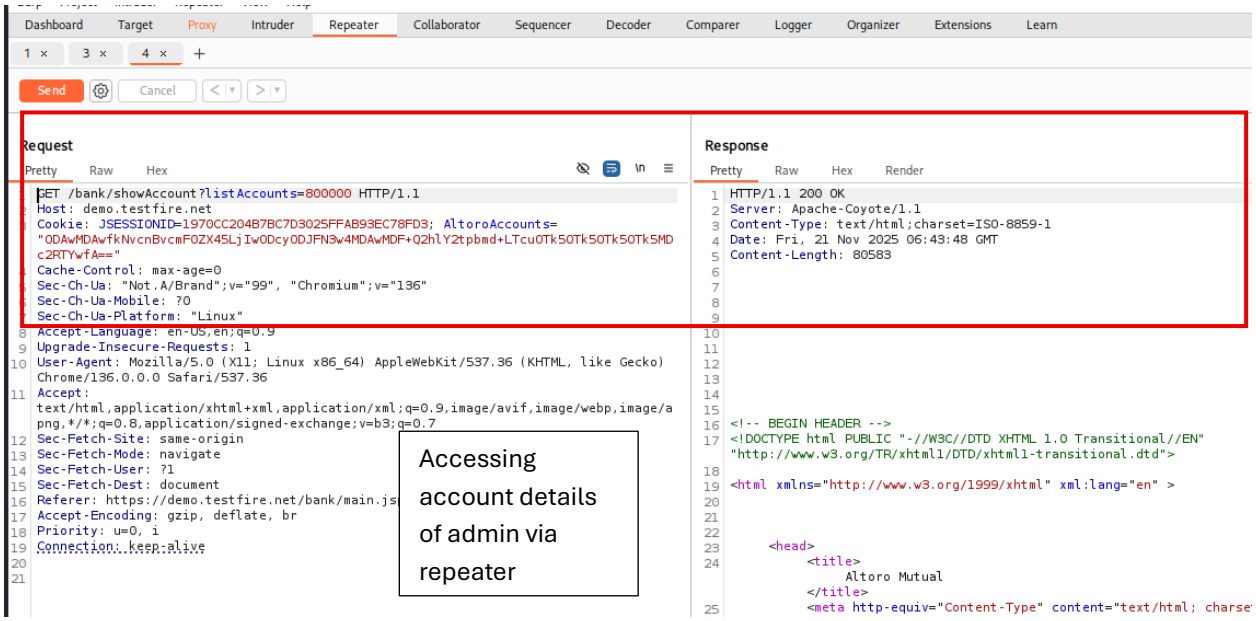
Proof-of-Concept (PoC):

While logged in as user 'jsmith', modifying the account number parameter in the GET request to a corporate account number successfully retrieved unauthorized corporate account details.

Admin User

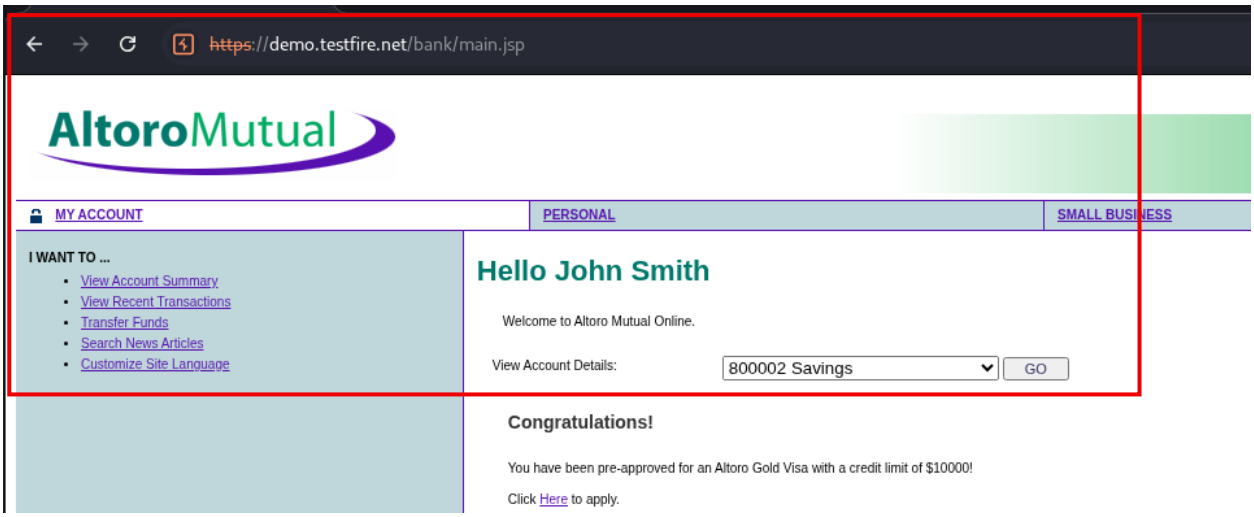


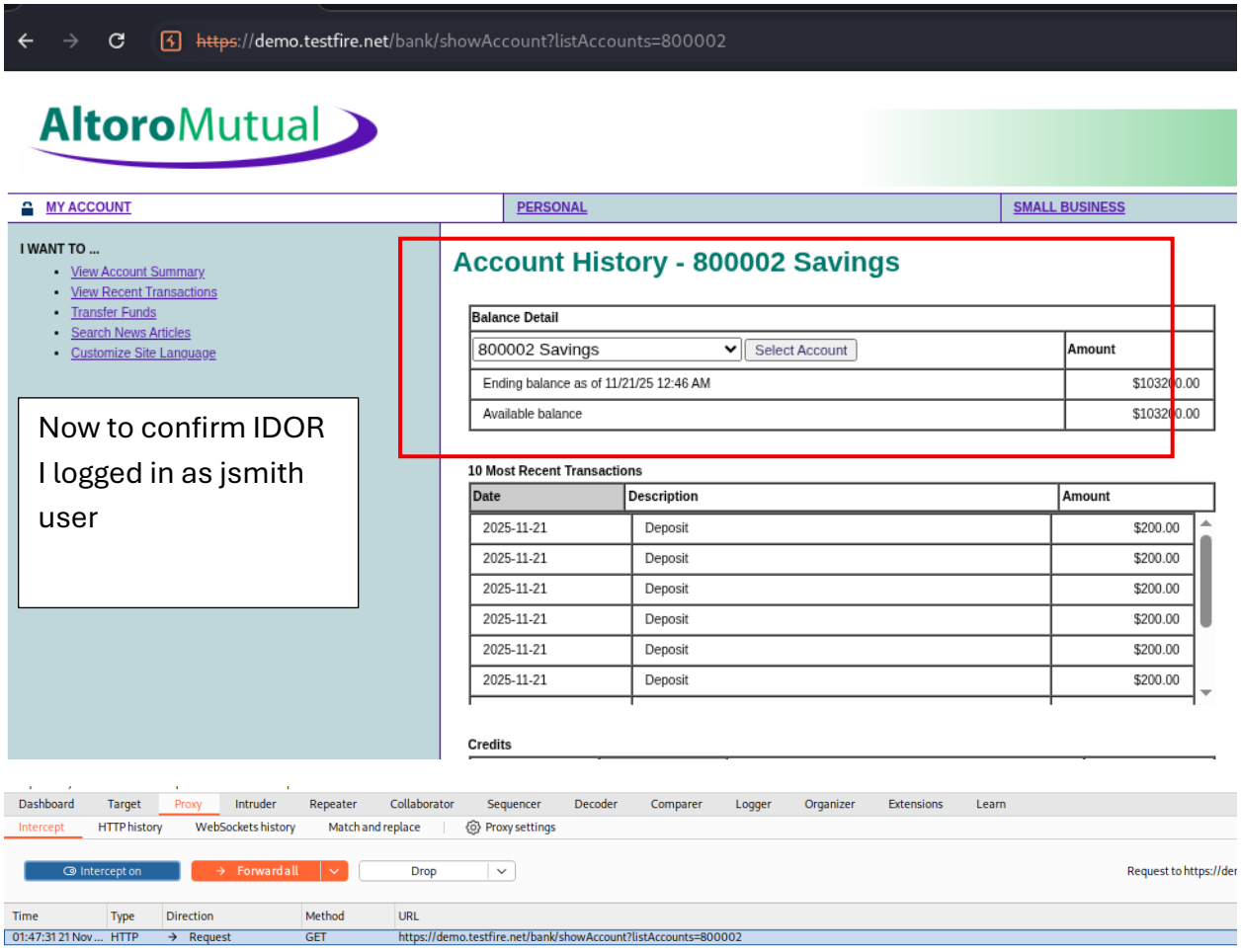

https://demo.testfire.net/bank/showAccount?listAccounts=800000
AltoroMutual
MY ACCOUNT
PERSONAL
SMALL BUSINESS
I WANT TO ...
View Account Summary
View Recent Transactions
Transfer Funds
Search News Articles
Customize Site Language
ADMINISTRATION
Edit Users
This account details are of admin users
Account History - 800000 Corporate
Balance Detail
800000 Corporate
Select Account
Amount
Ending balance as of 11/21/25 12:41 AM
Available balance
10 Most Recent Transactions
Date
Description
Amount
2025-11-21 Withdrawal -$888.00
Time
Type
Direction
Method
URL
01:42:24 21 Nov... HTTP Request GET https://demo.testfire.net/bank/showAccount?listAccounts=800000
Request
Pretty
Raw
Hex
GET /bank/showAccount?listAccounts=800000 HTTP/1.1
Host: demo.testfire.net
Cookie: JSESSIONID=1970CC204B7BC7D3025FFAB93EC78FD3; AltoroAccounts="ODAwMDAwfkNvcnBvcmF0ZX45LjIwODcyODJFN3w4MDAwMDF+Q2hlY2tpbmd+LTcu0Tk50Tk50Tk50Tk5MDc2RTYwfA=="
Cache-Control: max-age=0
Sec-Ch-Ua: "Not_A/Brand";v="99", "Chromium";v="136"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Linux"
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://demo.testfire.net/bank/main.jsp
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
Connection: keep-alive



Accessing account details of admin via repeater

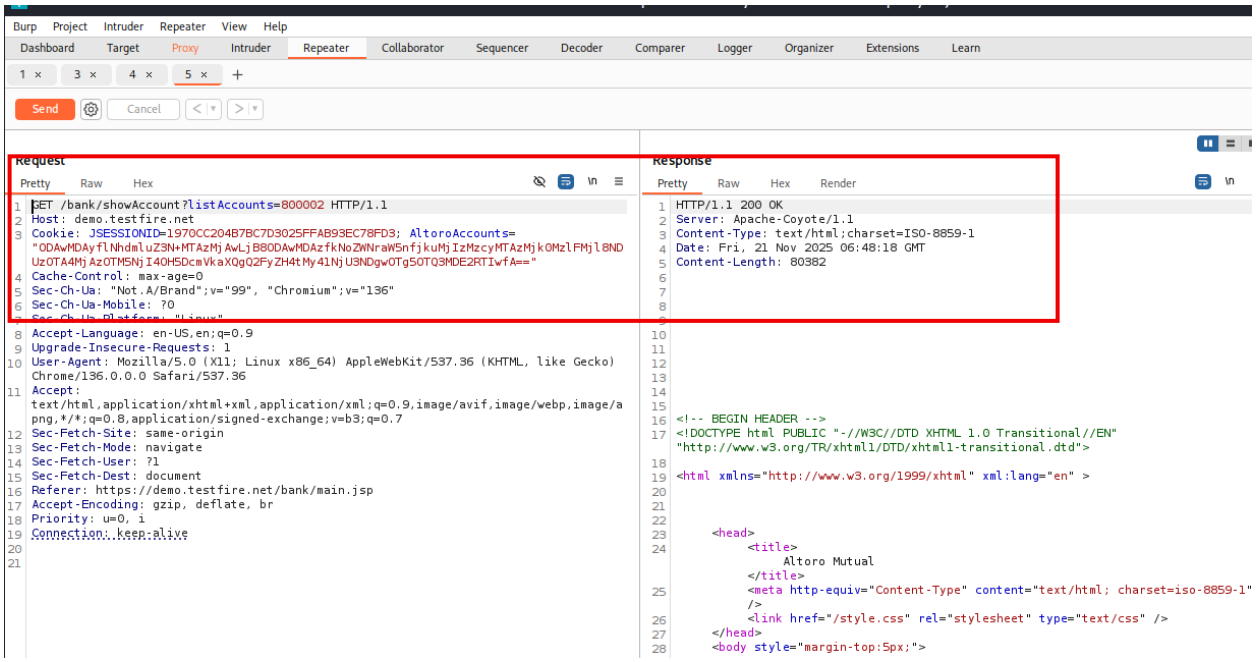
Now Jsmith as User

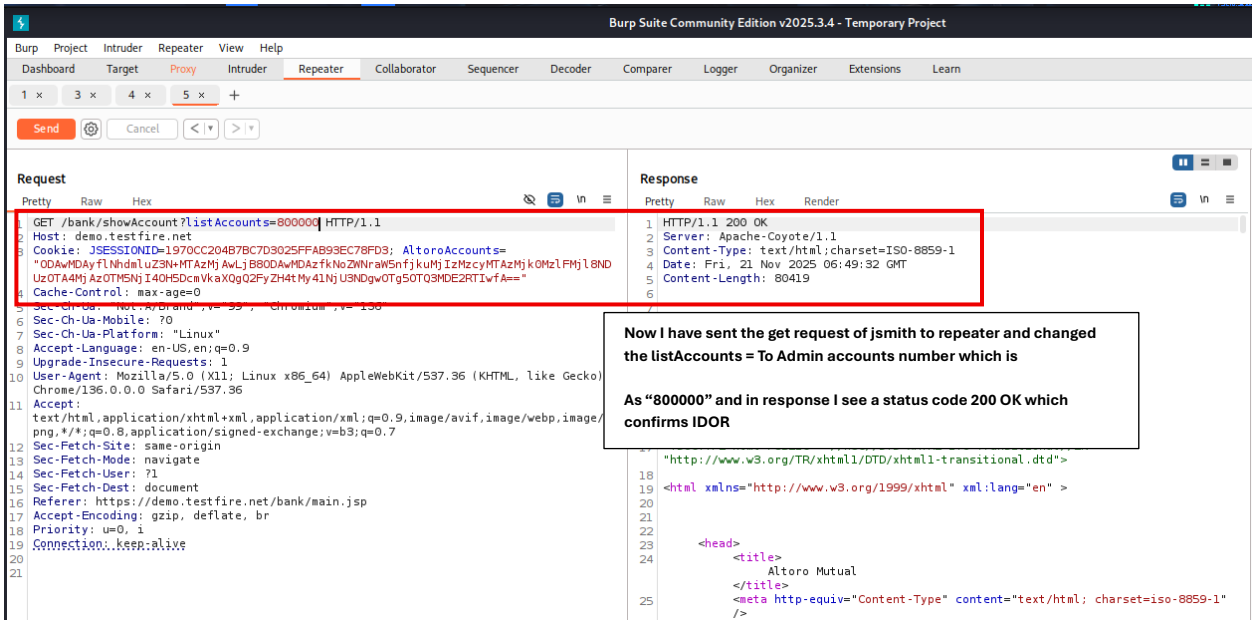



https://demo.testfire.net/bank/showAccount?listAccounts=800002
AltoroMutual
MY ACCOUNT
PERSONAL
SMALL BUSINESS
I WANT TO ...
View Account Summary
View Recent Transactions
Transfer Funds
Search News Articles
Customize Site Language
Now to confirm IDOR I logged in as jsmith user
Account History - 800002 Savings
Balance Detail
800002 Savings
Select Account
Amount
Ending balance as of 11/21/25 12:46 AM
$103200.00
Available balance
$103200.00
10 Most Recent Transactions
Date
Description
Amount
2025-11-21
Deposit
$200.00
Credits

Dashboard
Target
Proxy
Intruder
Repeater
Collaborator
Sequencer
Decoder
Comparer
Logger
Organizer
Extensions
Learn
Intercept
HTTP history
WebSockets history
Match and replace
Proxy settings
Intercept on
Forward all
Drop
Request to https://der
Time
Type
Direction
Method
URL
01:47:31 21 Nov ...
HTTP
Request
GET
https://demo.testfire.net/bank/showAccount?listAccounts=800002


Request
Pretty
Raw
Hex
GET /bank/showAccount?listAccounts=800002 HTTP/1.1
Host: demo.testfire.net
Cookie: JSESSIONID=1970CC204B7BC7D3025FFAB93EC78FD3; AltoroAccounts=
"ODAwMDAyflNhdmluZ3N+MTAzMjAwLjB8ODAwMDAzfkNoZWNraW5nfjkuMjIzMzcyMTAzMjk0MzlFMjl8NDUzOTA4MjAzOTM5NjI4OH5DcmVkaXQgQ2FyZH4tMy41NjU3NDgwOTg5OTQ3MDE2RTIwfA=="
Cache-Control: max-age=0
Sec-Ch-Ua: "Not.A/Brand";v="99", "Chromium";v="136"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Linux"
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://demo.testfire.net/bank/main.jsp
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
Connection: keep-alive





Impact:

- Unauthorized access to sensitive financial data

- Data confidentiality breach across customer accounts

Recommendation:

- Implement proper authorization checks on all object references

- Use indirect reference maps instead of direct object references