Vulnerability testing on practice website

Team: 2.3

Practice website chosen: testfire.net

1. Cross-Site Scripting (Stored)

CWE: CWE-79

OWASP Category: A03:2021 – Injection

Description: The product does not neutralize or incorrectly neutralizes user-controllable input before it is placed in output that is used as a web page that is served to other users.

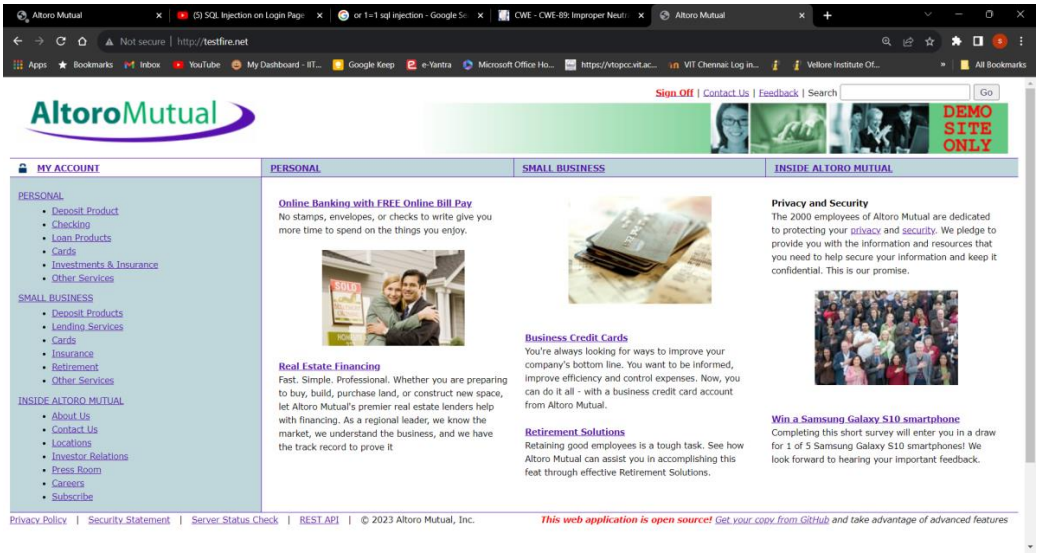
Business Impact: The most common attack performed with cross-site scripting involves the disclosure of information stored in user cookies. Typically, a malicious user will craft a client-side script, which -- when parsed by a web browser -- performs some activity (such as sending all site cookies to a given E-mail address). This script will be loaded and run by each user visiting the web site. Since the site requesting to run the script has access to the cookies in question, the malicious script does also.

Vulnerability Path : http://testfire.net

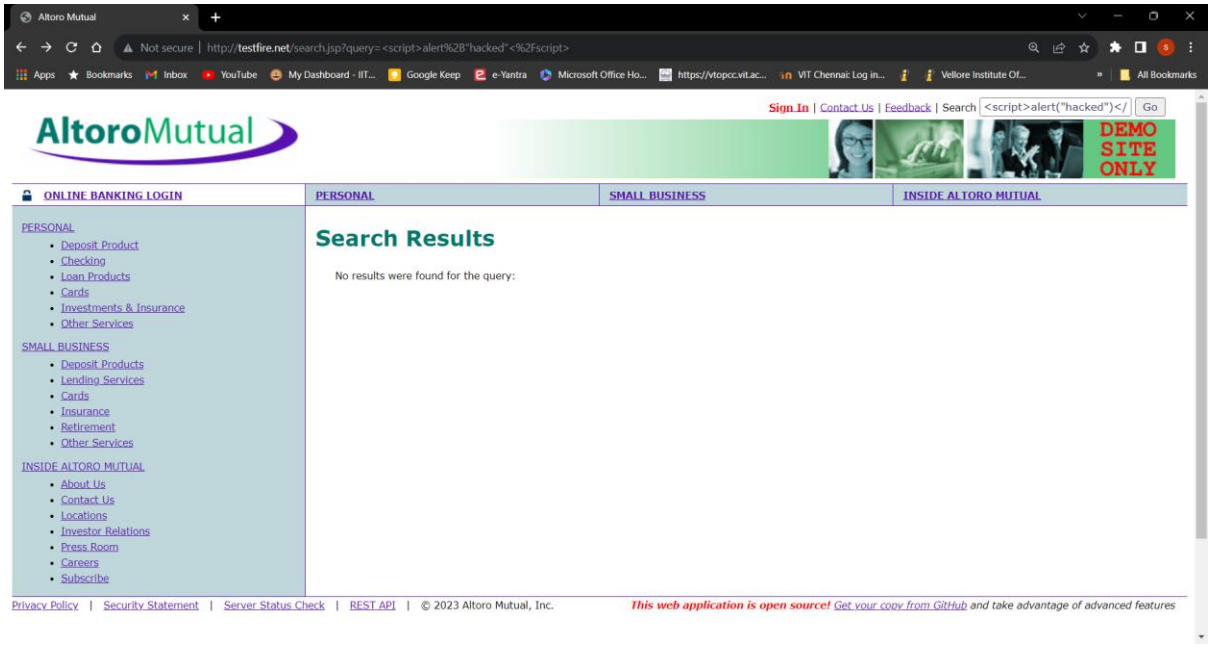
Vulnerability Parameter: http://testfire.net/search.jsp?query=

Steps to Reproduce :

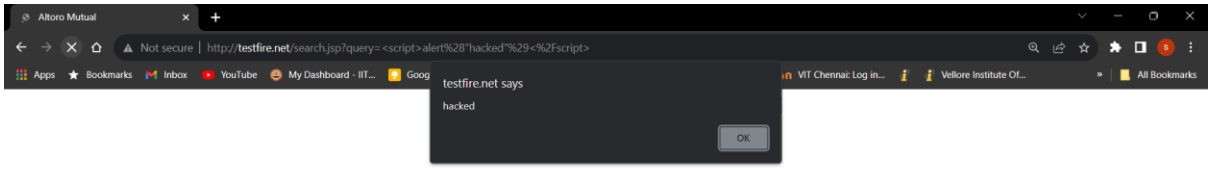
Step 1. Access the URL



Step 2: enter the script in the search box <script>alert"hacked"</script>



Step 3:-after entering the script content like" hacked" u will find the dialogue box as shown below.



Recommendation:

Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.

Examples of libraries and frameworks that make it easier to generate properly encoded output include Microsoft's Anti-XSS library, the OWASP ESAPI Encoding module, and Apache Wicket.

2. Html injection

CWE: CWE-80

OWASP Category: A03:2021 – Injection

Description: The product receives input from an upstream component, but it does not neutralize or incorrectly neutralizes special characters such as "<", ">", and "&" that could be interpreted as web-scripting elements when they are sent to a downstream component that processes web pages.

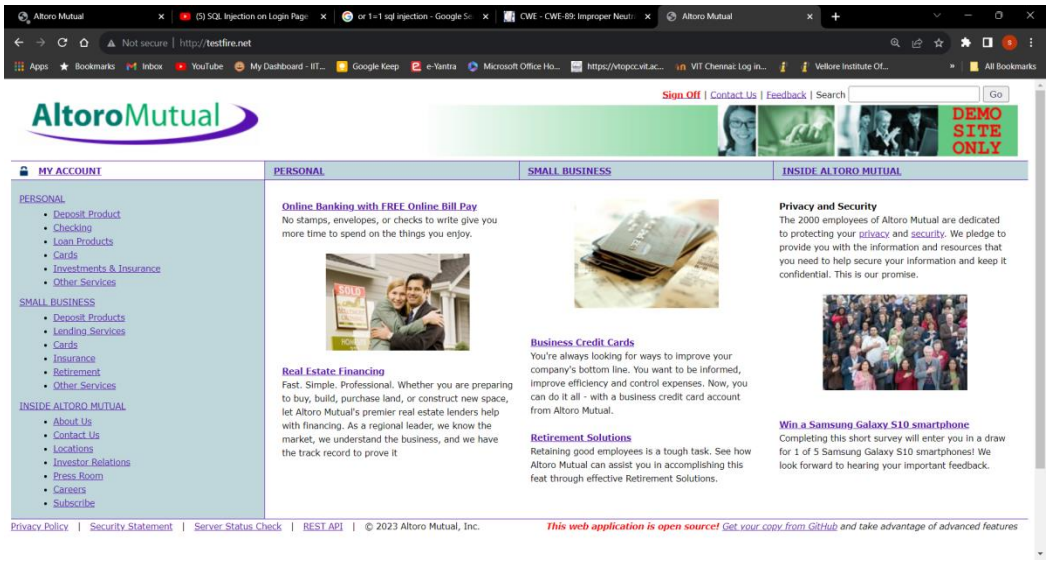
Business Impact: In some circumstances it may be possible to run arbitrary code on a victim's computer when cross-site scripting is combined with other flaws.

Vulnerability Path : http://testfire.net

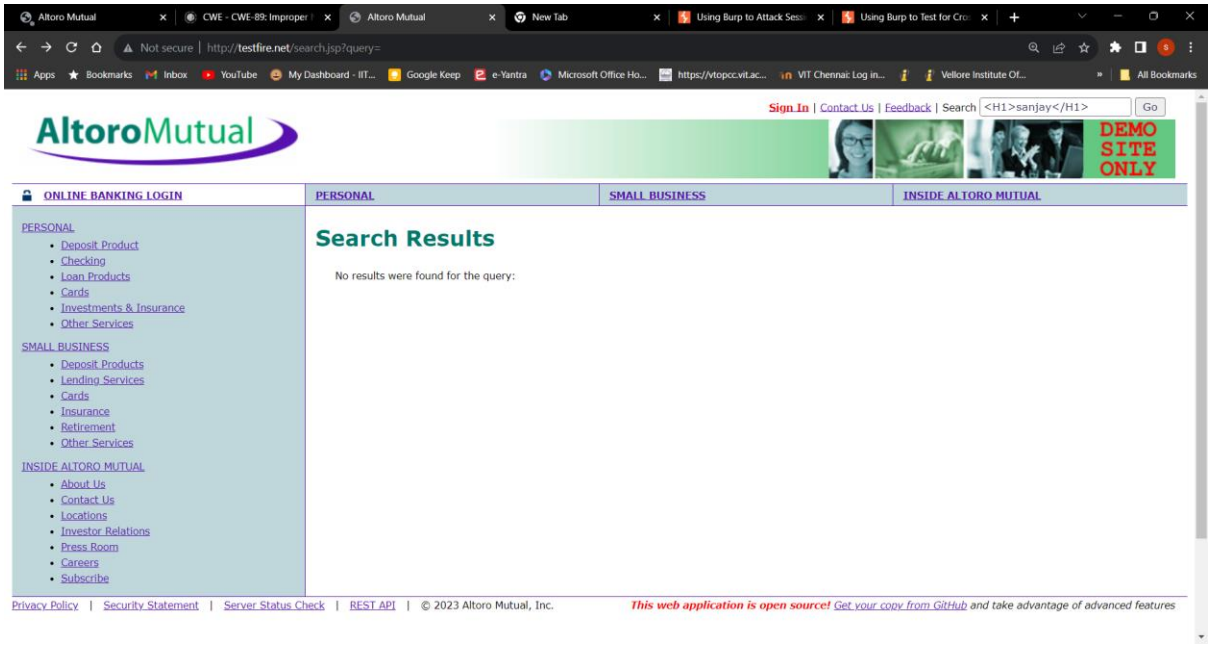
Vulnerability Parameter: http://testfire.net/search.jsp?query=

Steps to Reproduce :

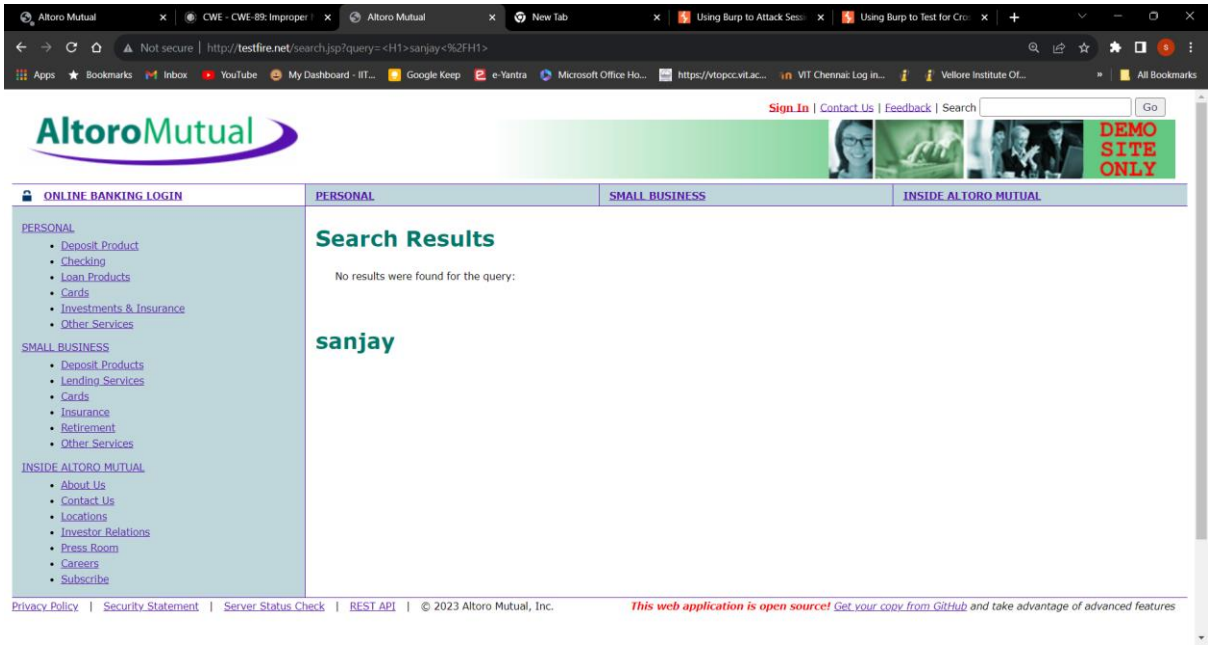
Step 1. Access the URL



Step 2: type the html script in the search box



Step 3: The script affects the webpage



3. No Session Management

CWE: CWE-384

OWASP Category: A07:2021 –Identification and Authentication Failures

Description: An attacker is able to force a known session identifier on a user so that, once the user authenticates, the attacker has access to the authenticated session.

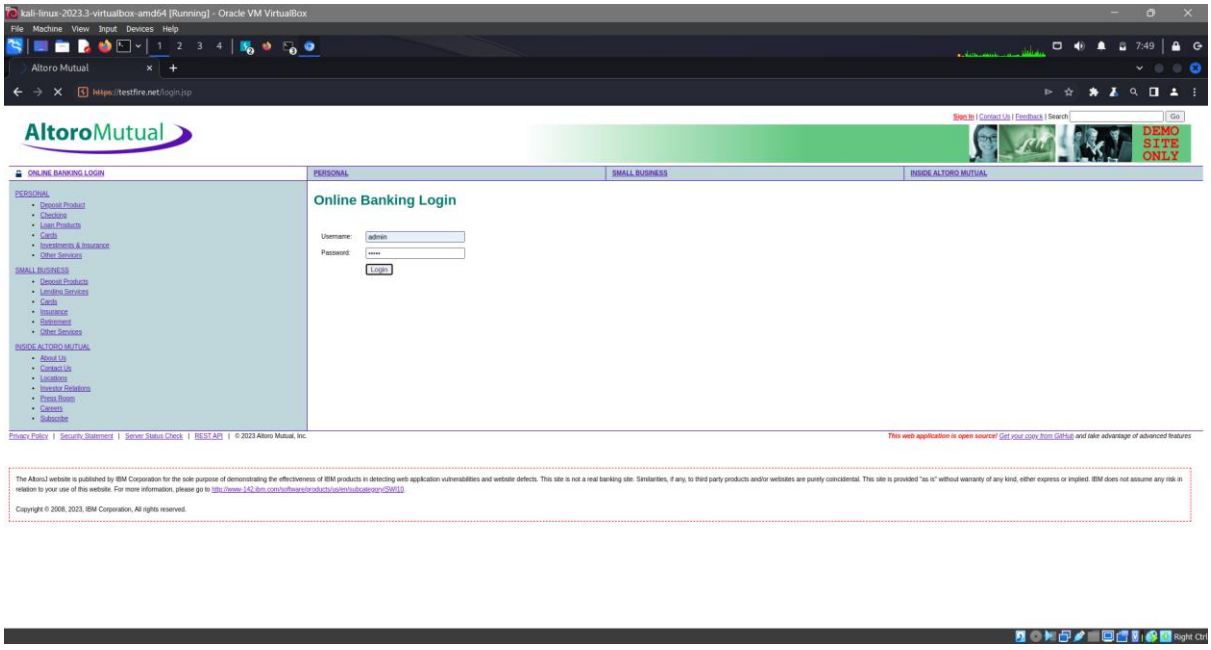
Business Impact: Without appropriate session management, you can run into several security problems, putting your users at risk. Common vulnerabilities caused by a lack of or poorly implemented session management include: Session hijacking

Vulnerability Path : http://testfire.net

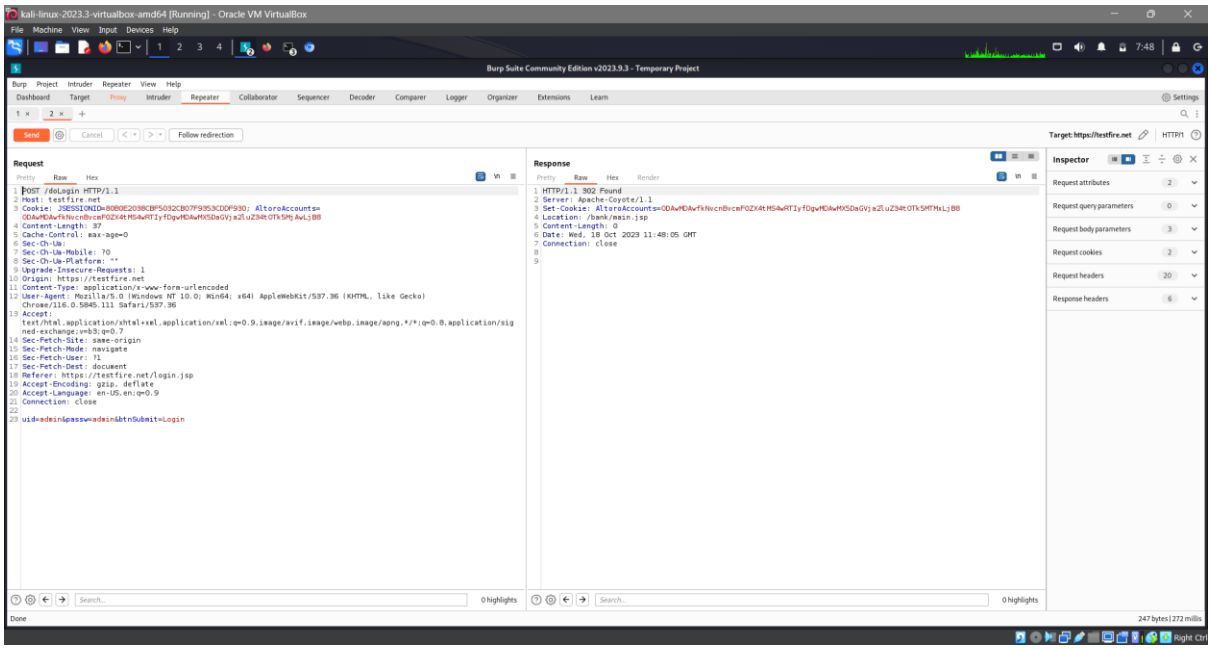
Vulnerability Parameter: http://testfire.net/login.jsp

Steps to Reproduce :

Step 1. Access the URL



Step 2.without the proper session management the burp can still access the request of session as shown.



Recommendation:

- Invalidate any existing session identifiers prior to authorizing a new user session.

4. Admin Login SQL Injection

CWE: CWE-284

OWASP Category: A03:2021-Injections

Description: Without sufficient removal or quoting of SQL syntax in user-controllable inputs, the generated SQL query can cause those inputs to be interpreted as SQL instead of ordinary user data. This can be used to alter query logic to bypass security checks, or to insert additional statements that modify the back-end database, possibly including execution of system commands.

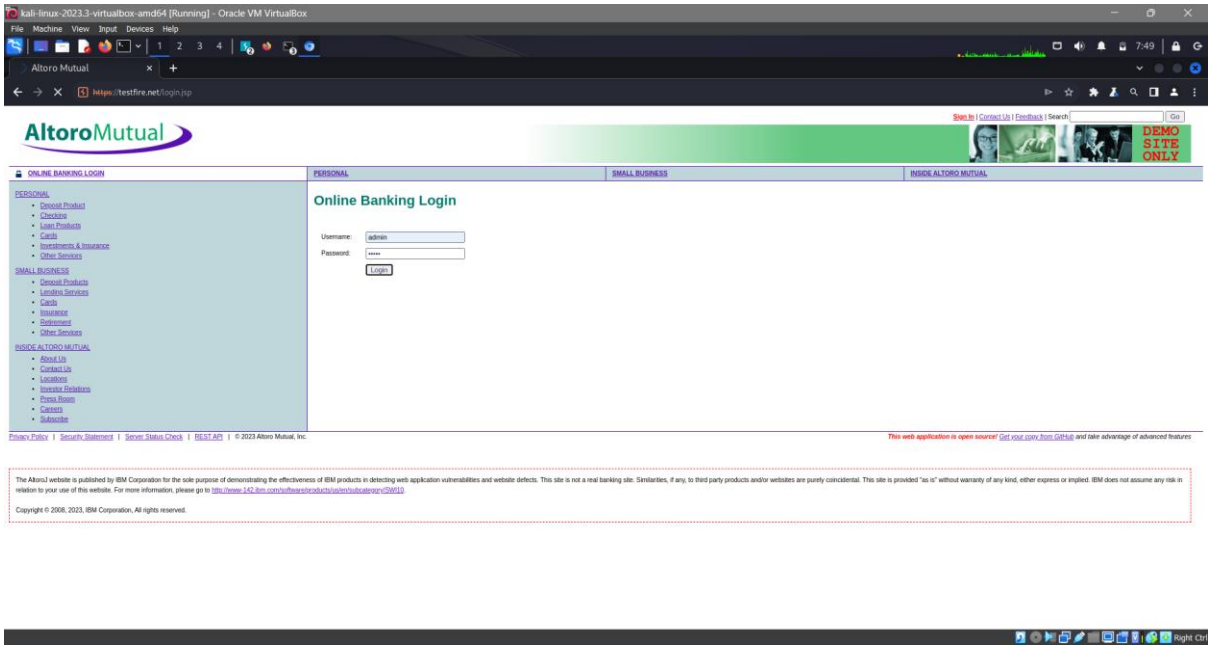
Business Impact: If authorization information is held in a SQL database, it may be possible to change this information through the successful exploitation of a SQL injection vulnerability

Vulnerability Path: http://testfire.net

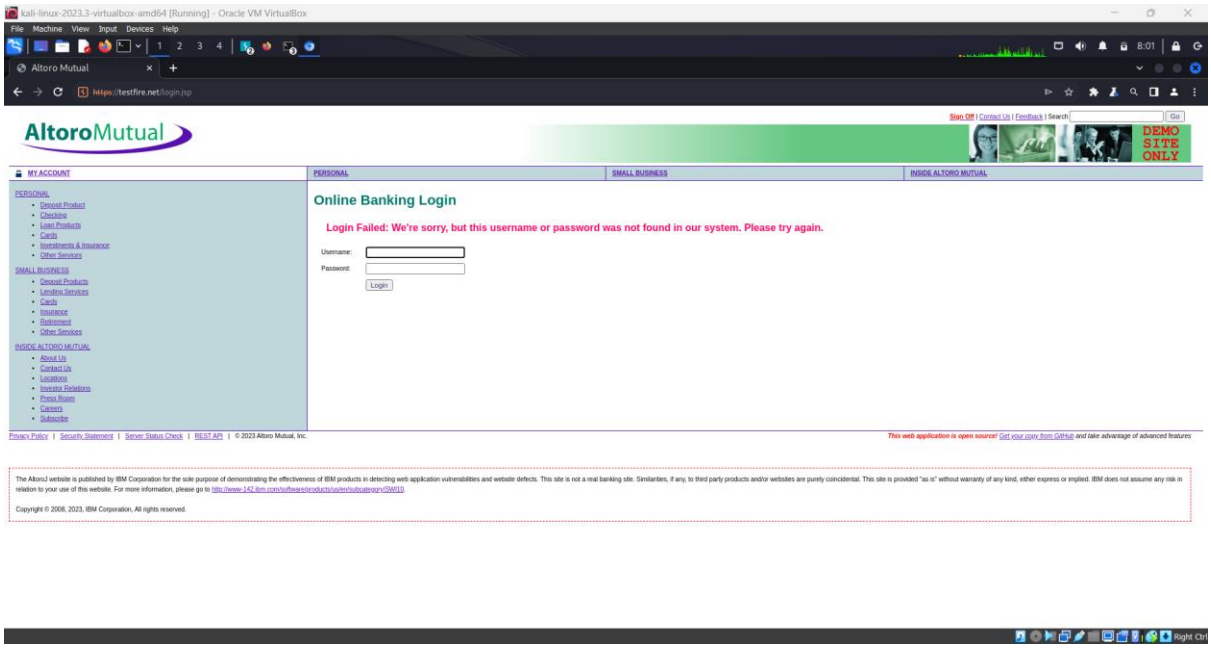
Vulnerability Parameter: http://testfire.net/login.jsp

Steps to Reproduce:

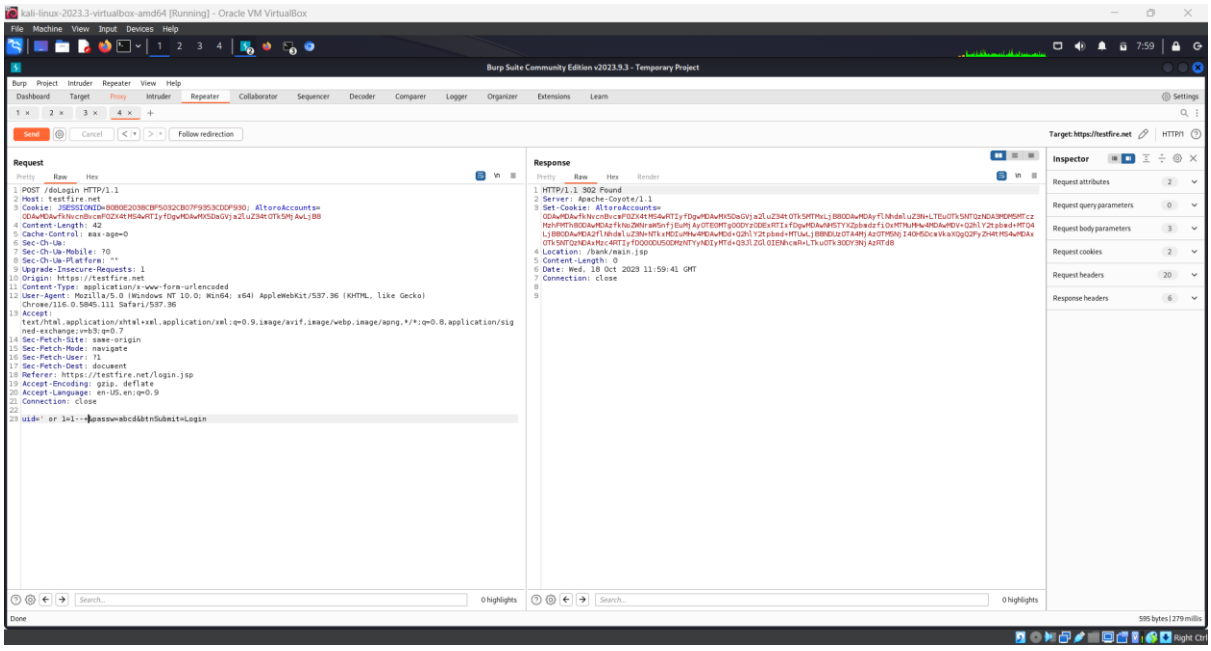
Step 1. Access the URL



Step 2:- Enter the login credentials in and try to validate as shown below.



Step 3:- in the backend we need to use the burp to configure the user with sql injection.



Recommendation:

Employ a layered approach to security that includes utilizing parameterized queries when accepting user input, ensuring that only expected data (white listing) is accepted by an application, and harden the database server to prevent data from being accessed inappropriately.

5. Default Credentials

CWE : CWE-1392

OWASP Category:A07:2021-Identification and Authentication Failures

Description:It is common practice for products to be designed to use default keys, passwords, or other mechanisms for authentication. The rationale is to simplify the manufacturing process or the system administrator's task of installation and deployment into an enterprise. However, if admins do not change the defaults, it is easier for attackers to bypass authentication quickly across multiple organizations.

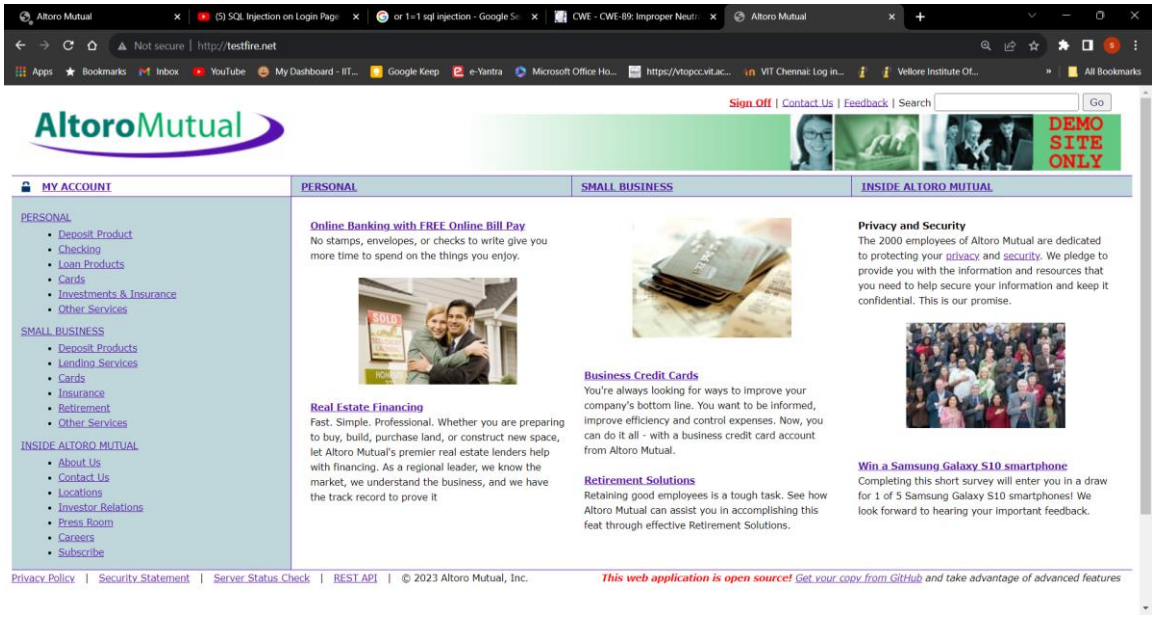
Business Impact:Attackers can easily obtain default passwords and identify internet-connected target systems. Passwords can be found in product documentation and compiled lists available on the internet.

Vulnerability Path : http://testfire.net

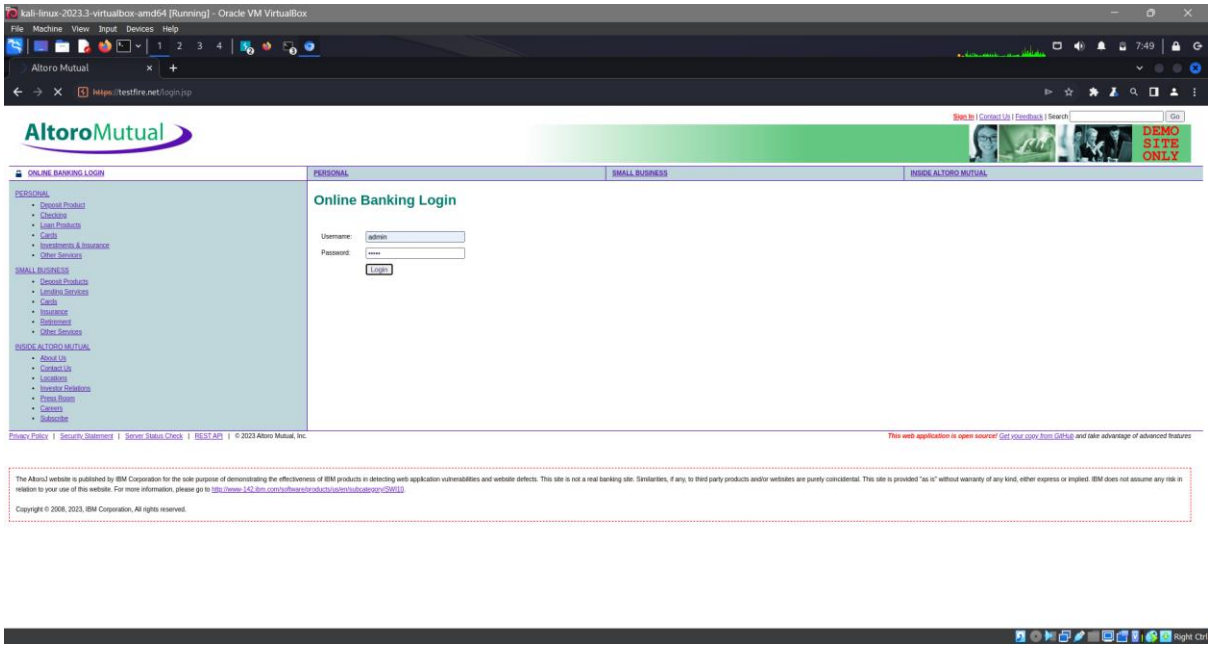
Vulnerability Parameter: http://testfire.net/login.jsp

Steps to Reproduce :

Step 1. Access the URL



Step 2: The website is based on apache tomcat server so we are able to login using a default user and password .



Recommendation:

- Prohibit use of default, hard-coded, or other values that do not vary for each installation of the product - especially for separate organizations.

6. Clickjacking (Improper Restriction of Rendered UI Layers or Frames)

CWE :CWE-1021

OWASP Category: A04-2021- insecure design

Description:it occurs whenever a web application is fetching a remote resource without validating the user-supplied URL. It allows an attacker to coerce the application to send a crafted request to an unexpected destination, even when protected by a firewall, VPN, or another type of network access control list (ACL).

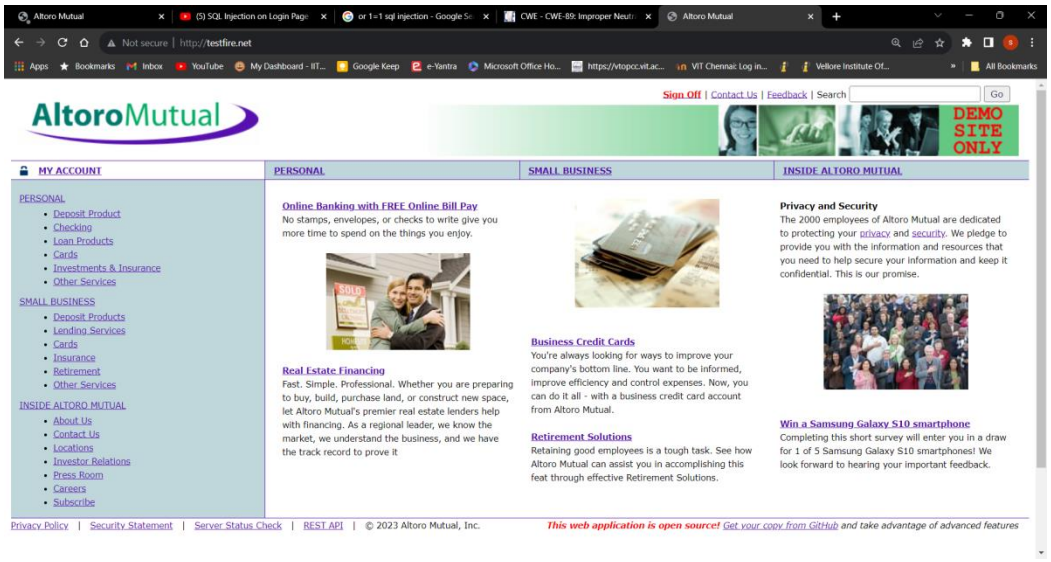
Business Impact: The common consequences of clickjacking include unauthorized actions, data theft, phishing, cookie theft, social engineering, reputation damage, legal and regulatory consequences, and user frustration. To prevent these, organizations should implement security headers, frame-busting techniques, user education, and regular security testing.

Vulnerability Path : http://testfire.net/index.jsp

Vulnerability Parameter: http://testfire.net/index.jsp

Steps to Reproduce :

Step 1:- Access the URl

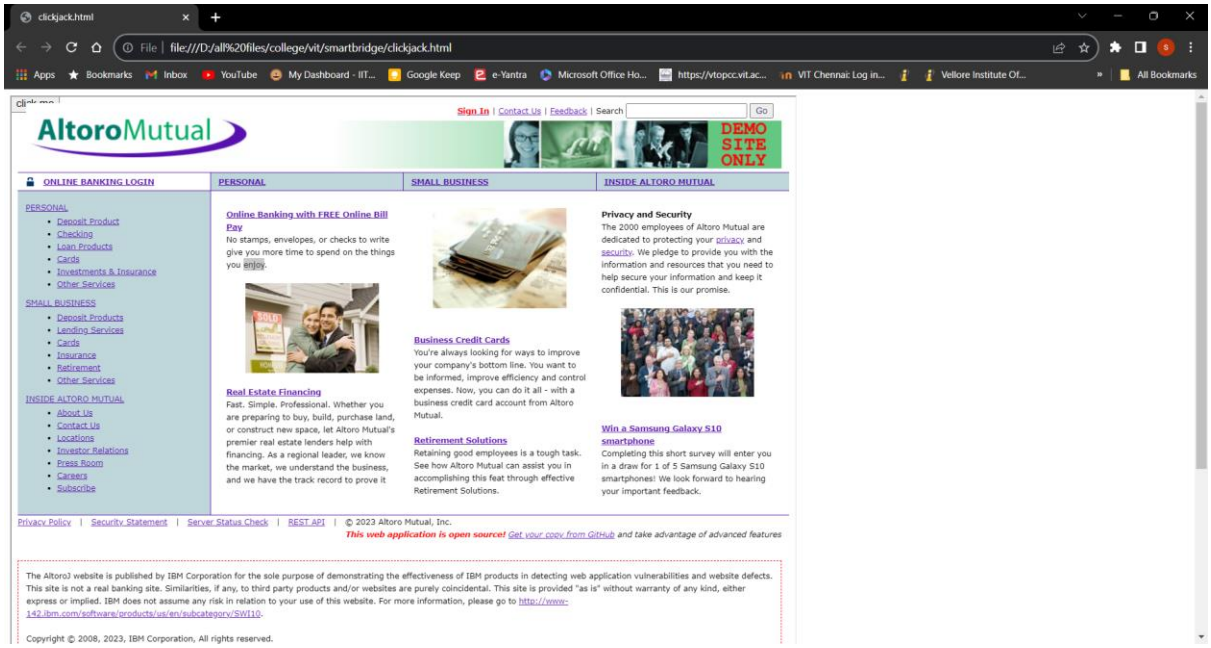


Step 2:- we write a html code as shown below when we run with the web address .

```
<head>
    <style>
        #target_website {
            position:relative;
            width:1000px;
            height:1000px;
            opacity:;
            z-index:2;
            }
        #decoy_website {
            position:absolute;
            width:300px;
            height:400px;
            z-index:1;
            }
    </style>
</head>

<body>
    <div id="decoy_website">
    <button>click me
    </button>
    </div>
    <iframe id="target_website" src="http://testfire.net/index.jsp">
    </iframe>
</body>
```

Step 3:- this will be the output of clickjacking the website with html code .



Recommendation:

- Segment remote resource access functionality in separate networks to reduce the impact of SSRF
- Enforce “deny by default” firewall policies or network access control rules to block all but essential intranet traffic.
- Enforce the URL schema, port, and destination with a positive allow list
- Do not send raw responses to clients
- Disable HTTP redirection

7. Anti-CSRF Token missing

CWE : CWE-352

OWASP Category:A05:2021 – Security Misconfiguration

Description:The web application does not, or cannot, sufficiently verify whether a well-formed, valid, consistent request was intentionally provided by the user who submitted the request.

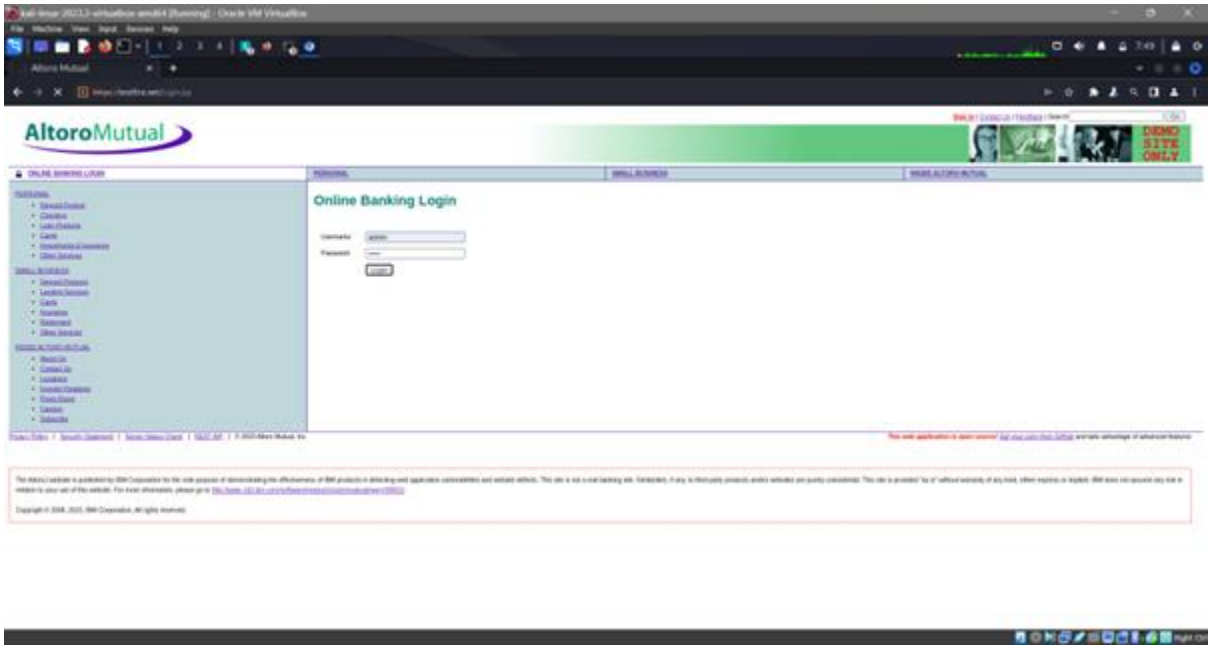
Business Impact:It can result in damaged client relationships, unauthorized fund transfers, changed passwords and data theft—including stolen session cookies.

Vulnerability Path : http://testfire.net/login.jsp

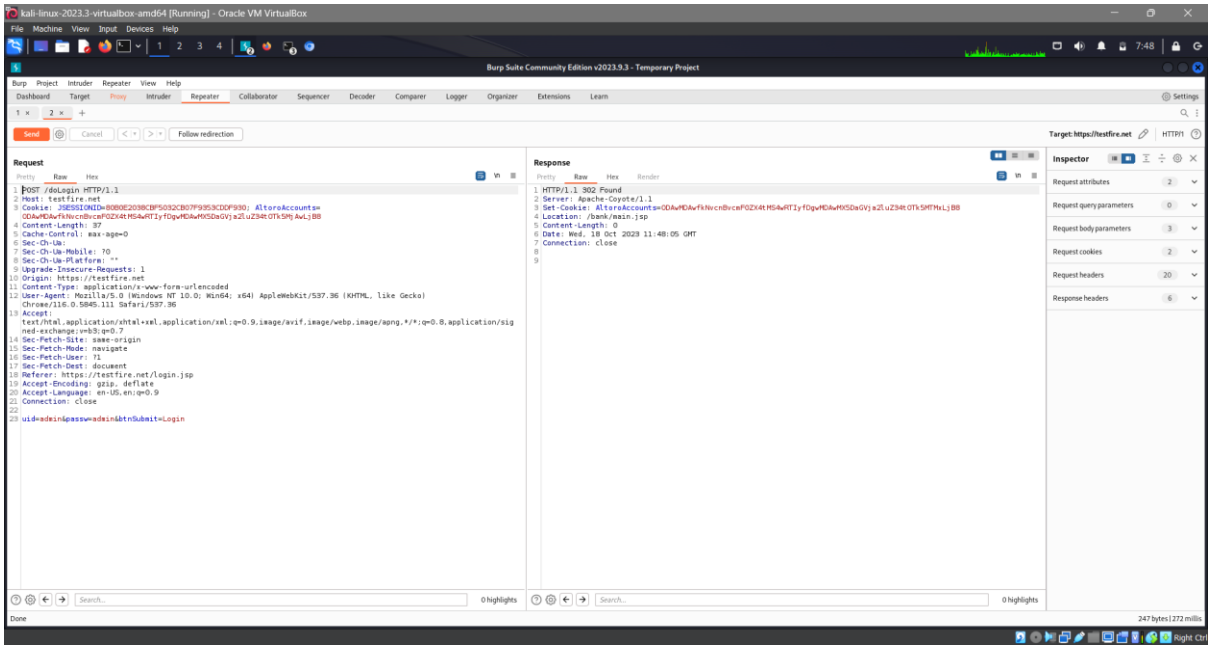
Vulnerability Parameter: http://testfire.net/login.jsp

Steps to Reproduce :

Step 1. Access the URL



Step 2: Now intercept with the burp proxy and send request in the repeater



Recommendation:

- The most effective way to protect against CSRF vulnerabilities is to include within relevant requests an additional token that is not transmitted in a cookie.

- validate that Host and Refer headers in relevant requests are both present and contain the same domain name

8. Web Server Transmits Cleartext Credentials

CWE :CWE- 319

OWASP Category: A02-2021- Cryptographic Failures

Description: The product transmits sensitive or security-critical data in cleartext in a communication channel that can be sniffed by unauthorized actors.

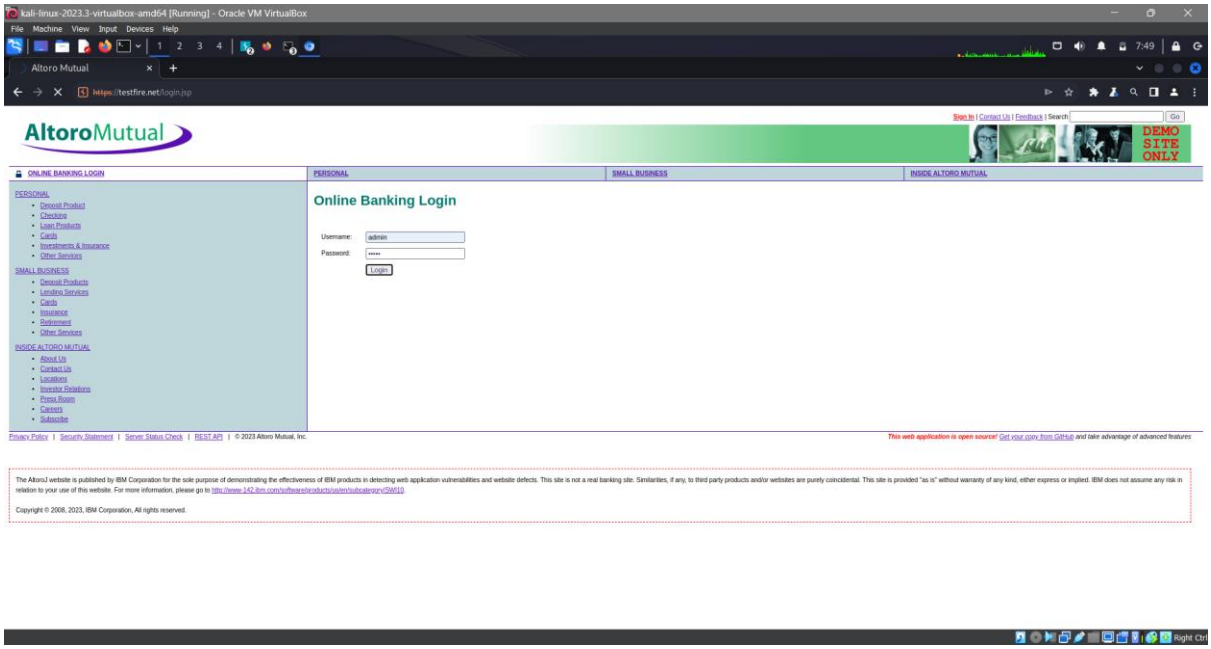
Business Impact: The technical impact of CWE-319 is that anyone can read the information by gaining access to the channel being used for communication. This can expose sensitive or confidential information to unauthorized parties, such as Passwords, Credit card numbers, Personal data.

Vulnerability Path : http://testfire.net/index.jsp

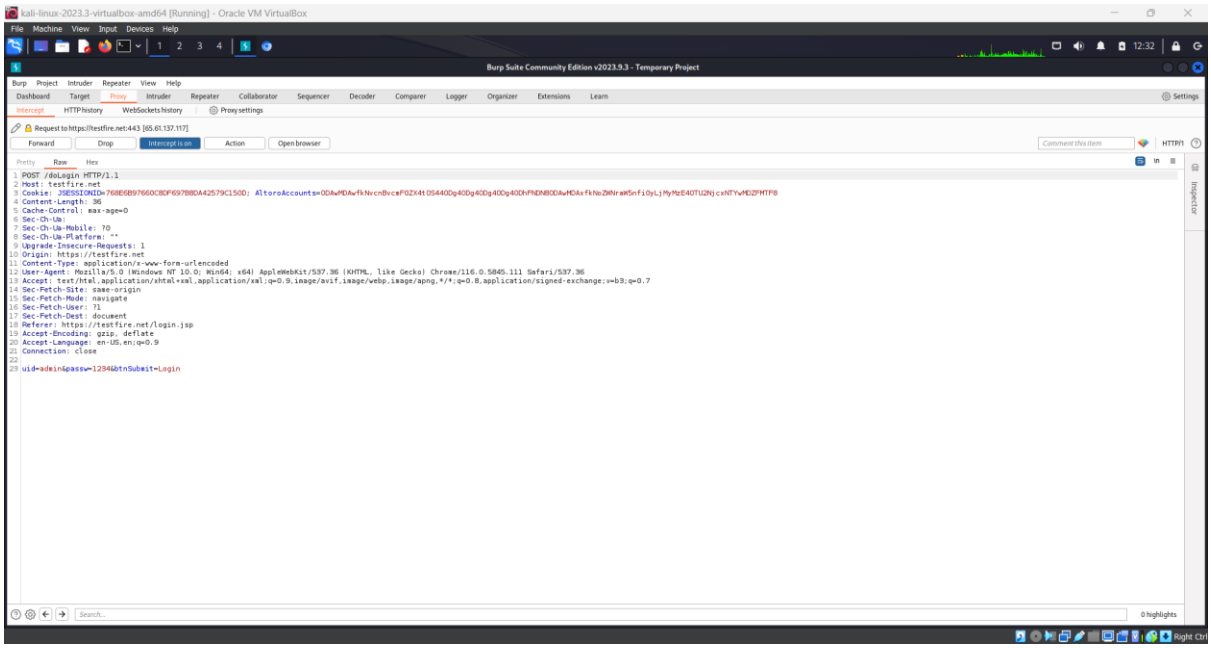
Vulnerability Parameter: **http://testfire.net/login.jsp**

Steps to Reproduce :

Step 1: Access url :



Step 2: Intercept the login with burp and you can view the credentials



Recommendation:

Before transmitting, encrypt the data using reliable, confidentiality-protecting cryptographic protocols.

9. Secure Http flag

CWE : CWE-1004

OWASP Category: A05:2021 – Security Misconfiguration

Description:The software uses a cookie to store sensitive information, but the cookie is not marked with the HttpOnly flag.

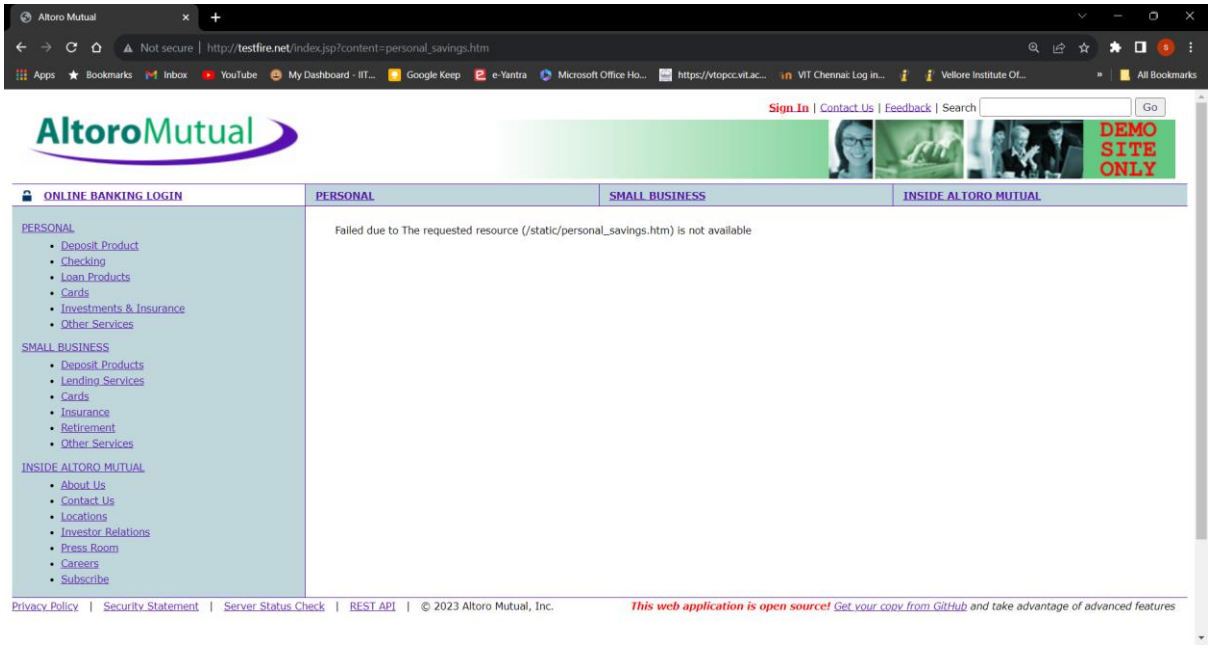
Business Impact: This measure makes certain client-side attacks, such as cross-site scripting, slightly harder to exploit by preventing them from trivially capturing the cookie's value via an injected script.

Vulnerability Path : http://testfire.net/index.jsp

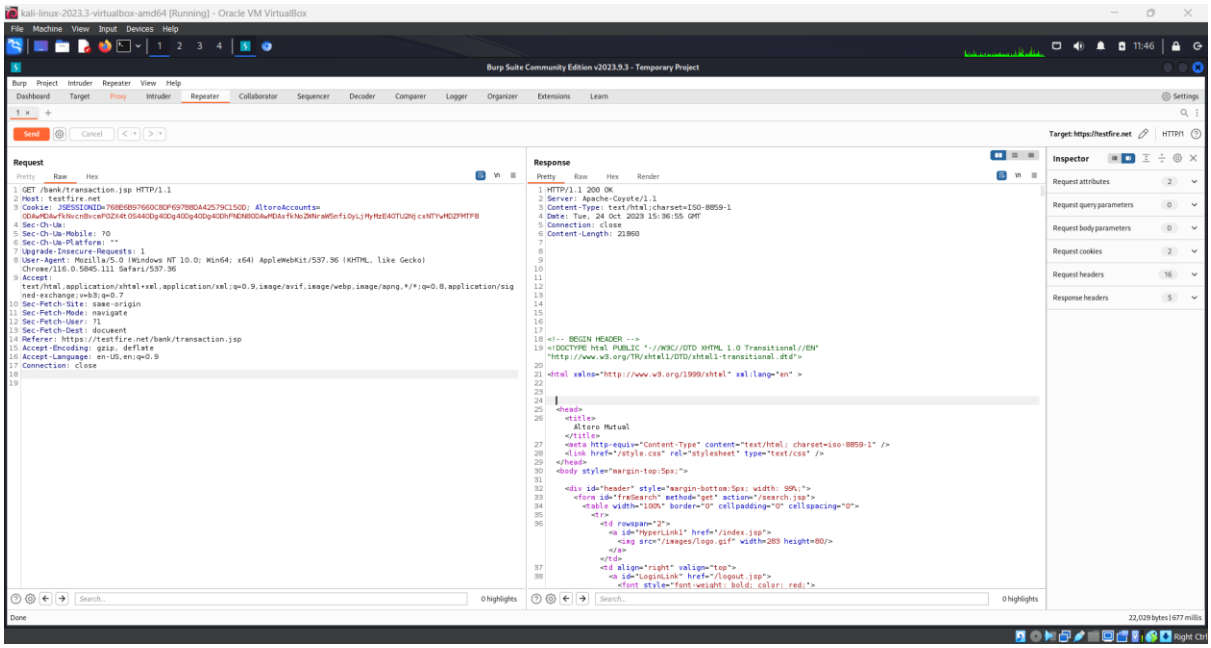
Vulnerability Parameter: http://testfire.net/index.jsp

Steps to Reproduce :

Step 1. Access the URL



Step 2: Now intercept with burp proxy and run the request in the repeater.



Recommendation:

- You specifically require legitimate client-side scripts within your application to read or set a cookie's value, you should set the HttpOnly flag by including this attribute within the relevant Set-cookie directive

10. Password field autocomplete

CWE :CWE-522

OWASP Category: A04:2021 – Insecure Design

Description: The product transmits or stores authentication credentials, but it uses an insecure method that is susceptible to unauthorized interception and/or retrieval.

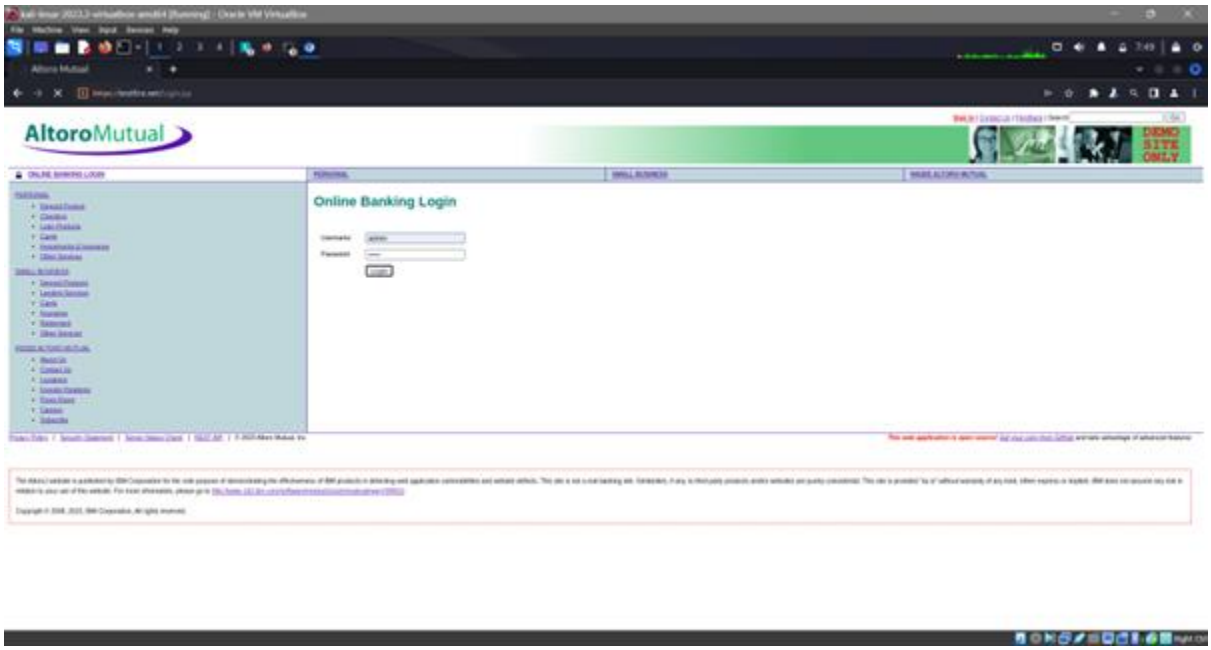
Business Impact: An attacker could gain access to user accounts and access sensitive data used by the user accounts.

Vulnerability Path : http://testfire.net/login.jsp

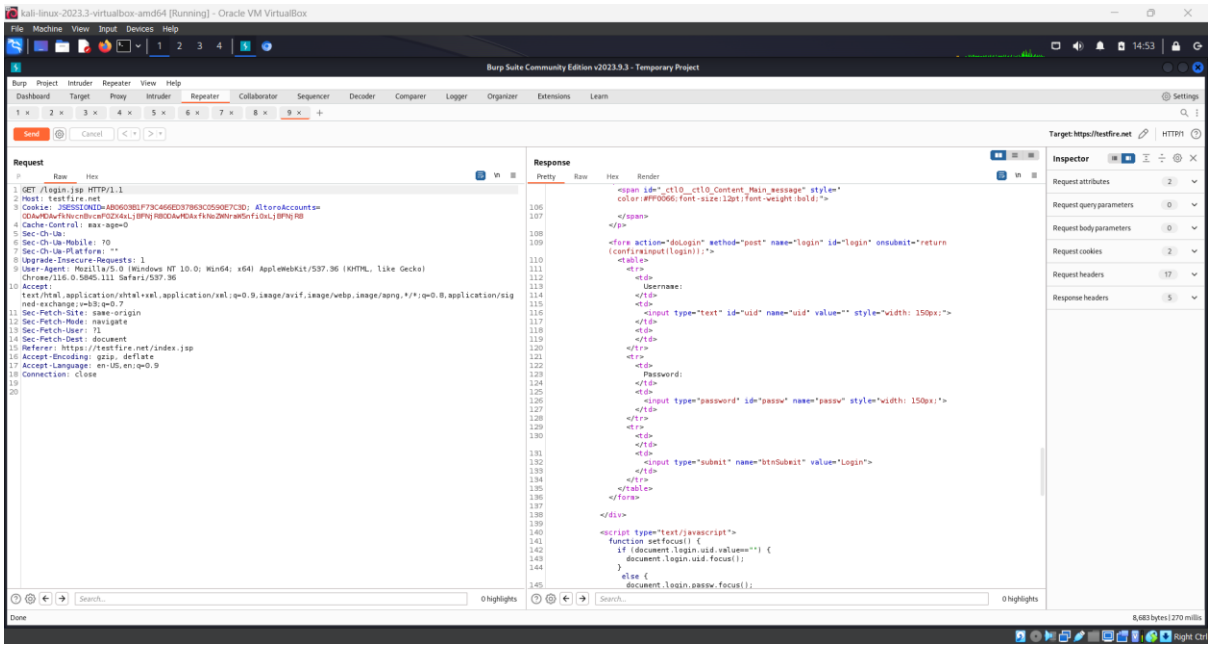
Vulnerability Parameter: http://testfire.net/login.jsp

Steps to Reproduce :

Step 1. Access the URL



Step 2: intercept with burp and check the html script.



Recommendation:

Use an appropriate security mechanism to protect the credentials. Make appropriate use of cryptography to protect the credentials.