**VULN Report**

**Web Application Security Tasks:**

**TASK 1** - Use Burp to capture HTTPs traffic

**TASK 2** - Use Burp to call a website's API without authentication and add a screenshot

**TASK 3** - Use Burp to pass XSS payloads to a website's API parameters and add screenshots of request and response with payloads

**TASK 4** - Find the CSRF attack and add the screenshot of PoC and the response.

* Bonus: You can perform additional security checks on the website and add screenshots with explanation.
* Use this

**Link:- https://demo.testfire.net/**

* for the above tasks

**Tasks Overview**

**What is BurpSuite and its Uses?**

* Burp Suite is a comprehensive platform developed by PortSwigger, primarily used for security testing and penetration testing of web applications. It serves as an essential tool for cybersecurity professionals, helping them identify vulnerabilities and assess the security posture of web applications.
* Key Features and Tools
* **Intercepting Proxy** :- Burp Suite’s core functionality is its intercepting proxy, which allows users to view and modify HTTP requests and responses in real-time. This feature enables penetration testers to analyze how web applications handle user inputs and to manipulate requests for testing purposes.
* **Web Crawling** :- The suite includes automated web crawling capabilities that help map out the structure of a web application. This process identifies various endpoints, which can then be analyzed for potential vulnerabilities. The spider tool is particularly useful for gathering more attack surfaces during the reconnaissance phase of testing.
* **Vulnerability Scanning** :- Burp Suite can automatically scan web applications for known vulnerabilities, providing detailed reports on security weaknesses. This feature is crucial for identifying issues before they can be exploited by malicious actors.
* **Intruder :-** The Intruder tool allows users to perform automated attacks such as brute force or dictionary attacks on input fields (e.g., login forms). It can test various payloads against parameters to identify weaknesses in authentication mechanisms.
* **Repeater :-** The Repeater tool enables users to manually modify and resend HTTP requests, making it easier to test specific inputs and observe server responses without the need to re-navigate through the application.
* **Sequencer :-** This tool assesses the randomness of tokens generated by the server (such as session tokens or CSRF tokens). It helps determine whether these tokens are sufficiently random to resist prediction attacks.
* Uses of Burp Suite :-
* **Penetration Testing** :- Burp Suite is widely used by security professionals to conduct penetration tests on web applications, identifying and exploiting vulnerabilities.
* **Security Audits** :- Organizations utilize Burp Suite to perform security audits, ensuring that their web applications comply with security standards.
* **Bug Bounty Programs** :- Many bug bounty hunters rely on Burp Suite due to its robust feature set, allowing them to discover and report vulnerabilities effectively.
* **Training and Education** :- Burp Suite serves as a learning tool in cybersecurity education, helping students understand web application security concepts through practical application.

**Task 1 -** Use Burp to capture HTTPs traffic

* Capturing HTTP traffic involves recording the data exchanged between a client (such as a web browser) and a server over the Hypertext Transfer Protocol (HTTP).
* This process is crucial for diagnosing network issues, analyzing performance, and ensuring security.
* Capturing HTTP traffic is essential for several reasons:
* Troubleshooting: Helps identify issues with web applications or services by providing detailed request and response information.
* Performance Analysis: Allows developers and network administrators to analyze response times and optimize performance.
* Security Monitoring: Facilitates the detection of anomalies or malicious activities within network communications.

**Task 2 -** Use Burp to call a website’s API without Authentication.

* It refers to the process of utilizing Burp Suite, a security testing tool, to interact with a web application's API (Application Programming Interface) without needing any form of user credentials or tokens that would typically be required for access.
* Burp Suite
* Burp Suite is a powerful platform used for web application security testing. It provides various tools for intercepting, modifying, and analyzing HTTP/S traffic between clients (like web browsers) and servers.
* API (Application Programming Interface)
* An API allows different software applications to communicate with each other. In the context of web applications, APIs often expose endpoints that can be accessed over HTTP/S to perform actions like retrieving data or submitting information.
* Without Authentication
* "Without authentication" implies that the API endpoints being accessed do not require any login credentials or security tokens. This could mean that the API is publicly accessible or has been misconfigured to allow access without proper security measures.

**Task 3 -** Use Burp to pass XSS payloads to a website's API parameters.

* It refers to the practice of using Burp Suite, a web application security testing tool, to inject Cross-Site Scripting (XSS) payloads into the parameters of an API request.
* XSS (Cross-Site Scripting)
* XSS is a type of security vulnerability that allows an attacker to inject malicious scripts into content that is then served to users. These scripts can execute in the context of the user's browser, potentially leading to data theft, session hijacking, or other malicious actions.
* API Parameters
* APIs often accept various parameters in requests, which can include data sent via query strings, headers, or body content. These parameters are crucial for the API's functionality and can be targets for XSS attacks if not properly validated or sanitized.
* Practical Implications
* Using Burp Suite to inject XSS payloads into API parameters involves several steps:
* Intercept API Requests: Configure Burp Suite as a proxy to capture requests made to the API. This allows you to see the parameters being sent.
* Modify Parameters: In Burp's interface, you can modify the captured requests by inserting XSS payloads into the API parameters. Common XSS payloads might include:

**<script>alert('XSS');</script>**

**"><img src=x onerror=alert('XSS')>**

* Send Modified Requests: After injecting the payloads, you send the modified requests to the API and observe how it processes them.
* Analyze Responses: Check the responses from the API to see if the injected scripts execute or if any errors occur. Successful execution indicates a potential XSS vulnerability.

**Task 4 -** Find the CSRF attack and add the screenshot of PoC and the response.

* It refers to the task of identifying a Cross-Site Request Forgery (CSRF) vulnerability in a web application, demonstrating it through a Proof of Concept (PoC), and capturing the response from the server after executing the attack.
* Cross-Site Request Forgery (CSRF)
* CSRF is a type of security vulnerability that allows an attacker to trick a user into executing unwanted actions on a web application in which they are authenticated.
* This exploit takes advantage of the trust that a web application has in the user's browser.
* For example, if a user is logged into their bank account, an attacker could send them a link that, when clicked, would transfer money from the user's account to the attacker's account without their consent.
* Finding a CSRF Attack

Finding a CSRF vulnerability typically involves:

* Identifying State-Changing Requests: Look for operations in the web application that change data or state (e.g., transferring funds, changing email addresses).
* Testing for Vulnerabilities: Attempt to craft requests that mimic legitimate actions but alter parameters to benefit the attacker.
* Proof of Concept (PoC)
* A PoC is a demonstration that shows how an exploit can be executed. In the context of CSRF:
* It often involves creating an HTML form or link that, when accessed by an authenticated user, sends a malicious request to the server.
* For example:

<**form action="https://example.com/transfer" method="POST">**

**<input type="hidden" name="account" value="attacker\_account">**

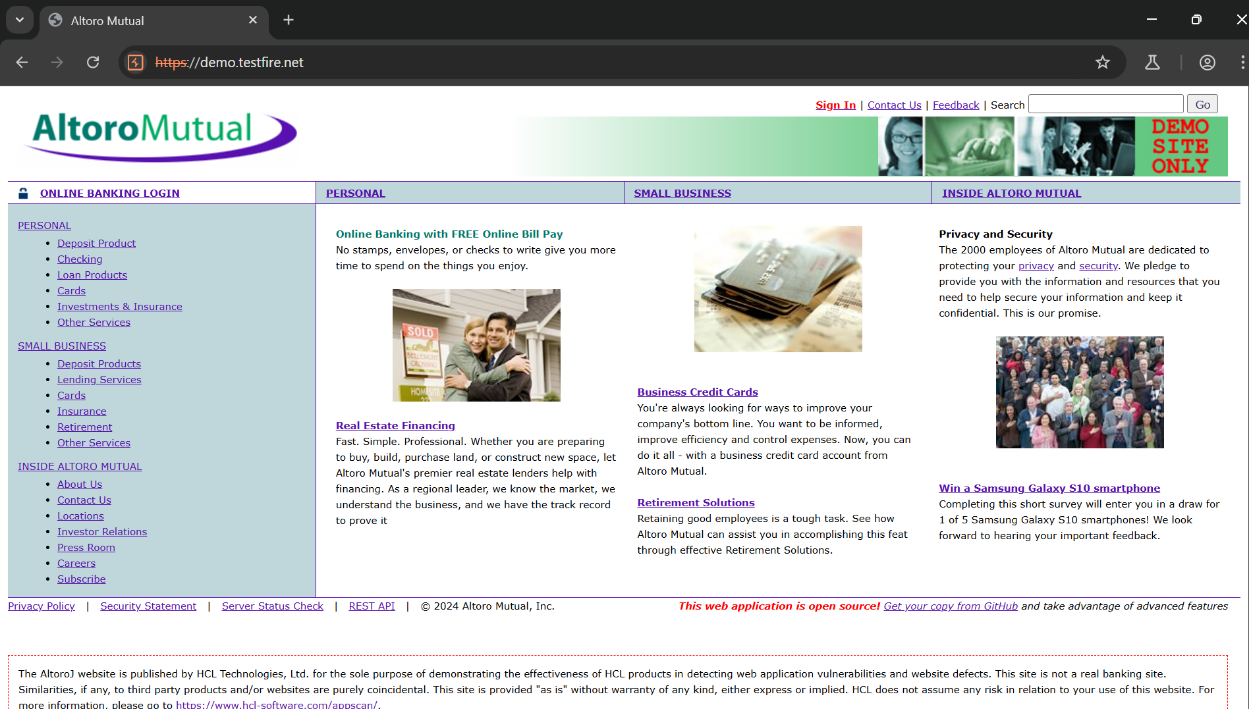
**<input type="hidden" name="amount" value="1000">**

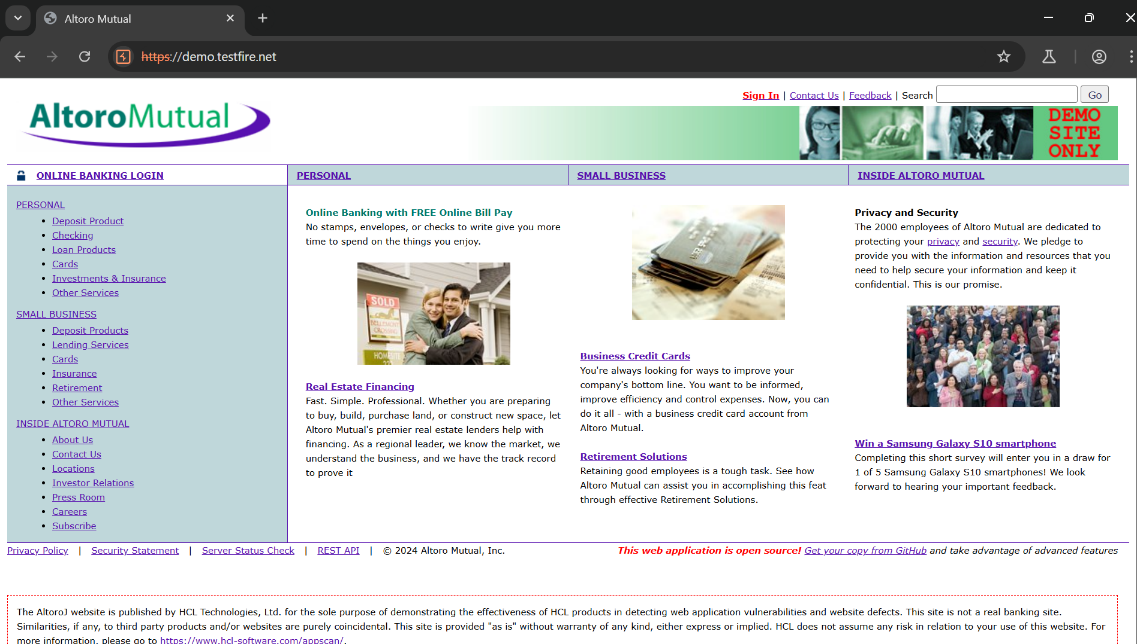
**<input type="submit" value="Transfer Funds">**

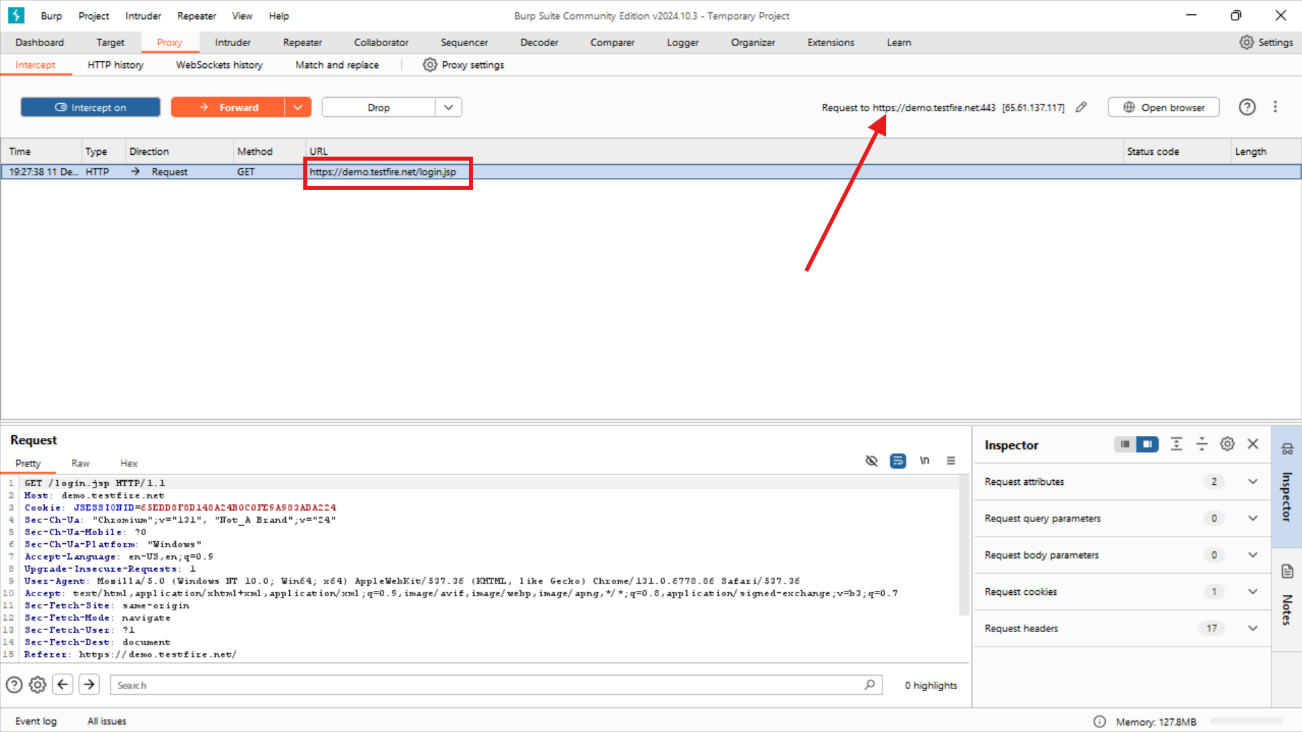
**</form>**

* Capturing the Response
* After executing the PoC:
* The response from the server is analyzed to confirm whether the unauthorized action was successfully performed (e.g., funds transferred).
* This response can provide insight into whether proper security measures are in place to prevent CSRF attacks.

**TASK - 1**

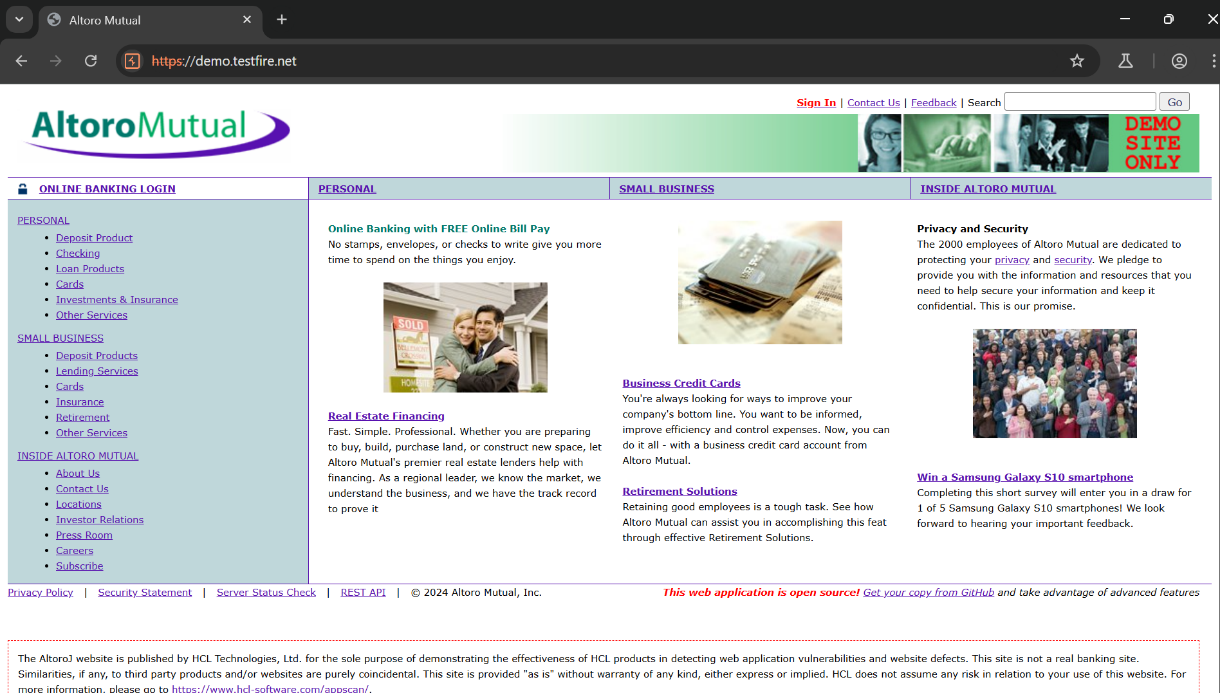
* Use Burp to capture HTTPs traffic
* Open the website
* Click on the sign in to caputre the HTTPs traffic

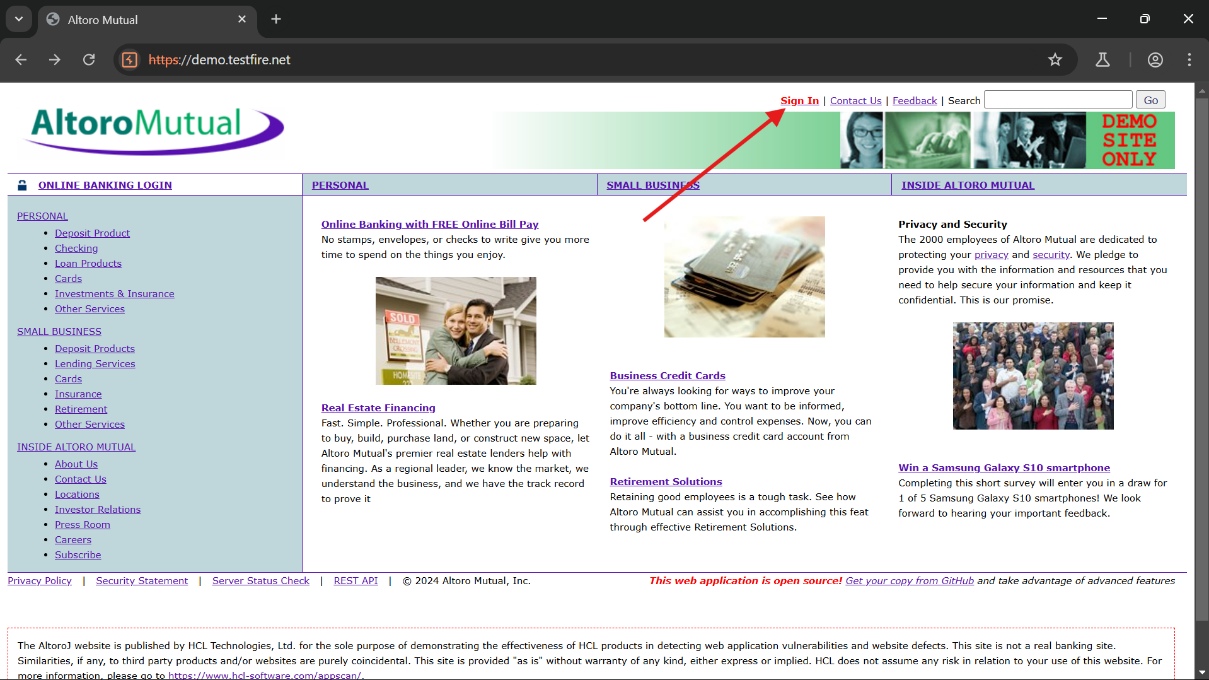


* Let’s capture the HTTPS traffic in Burpsuite
* We have captured the above HTTPS request from the website.
* Captured Request :- https://demo.testfire.net/login.jsp

**TASK – 2**

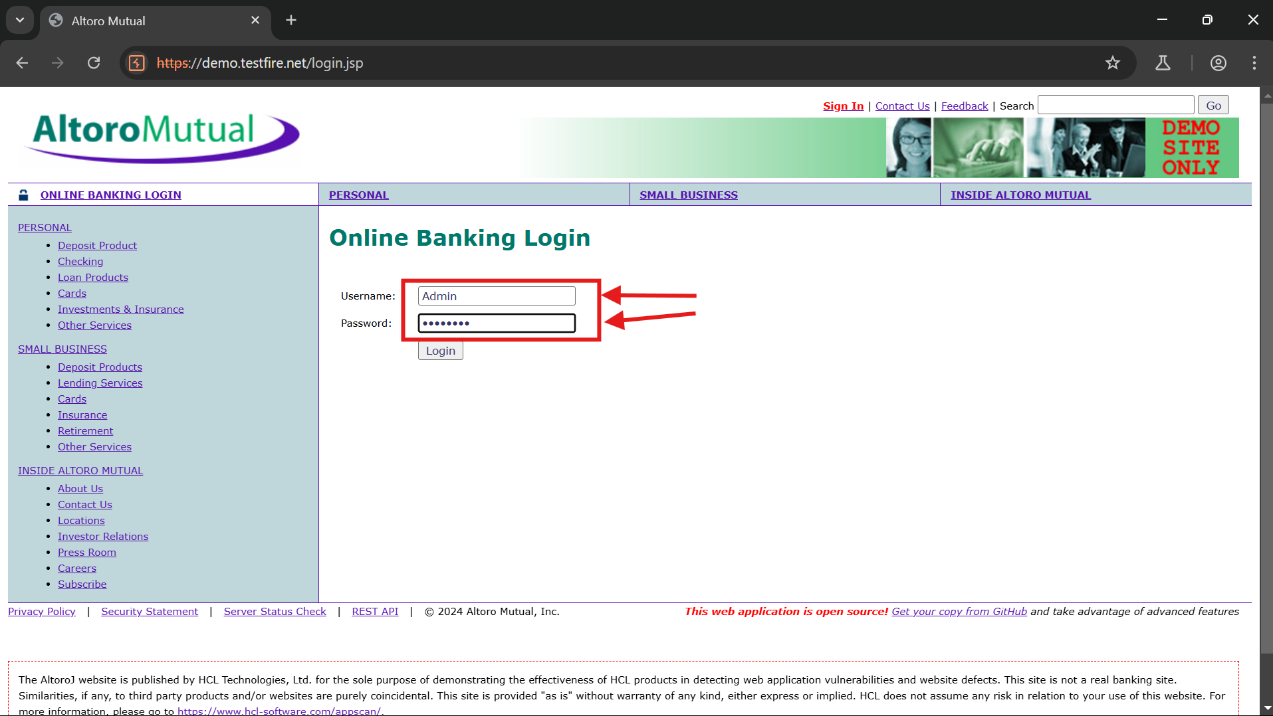
* **Use Burp to call a website's API without authentication and add a screenshot.**
* First visit the given Link

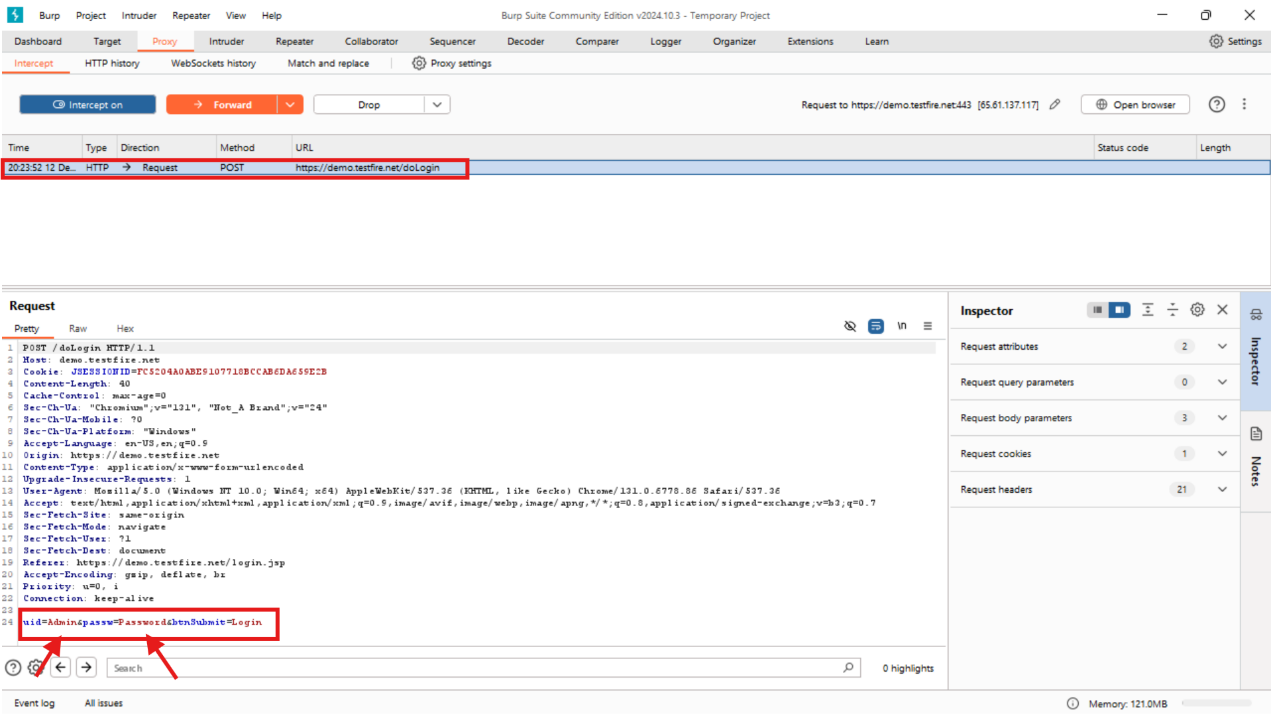


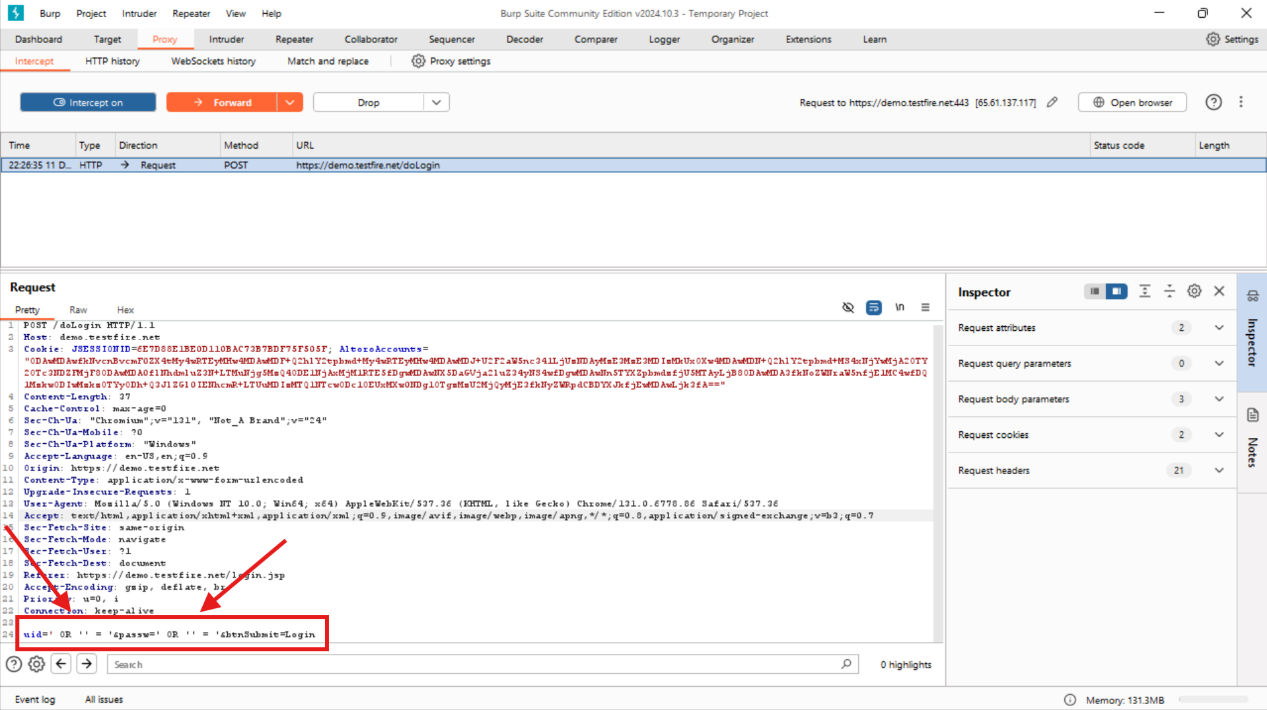
* Now click on sign in

And enter the credentials as

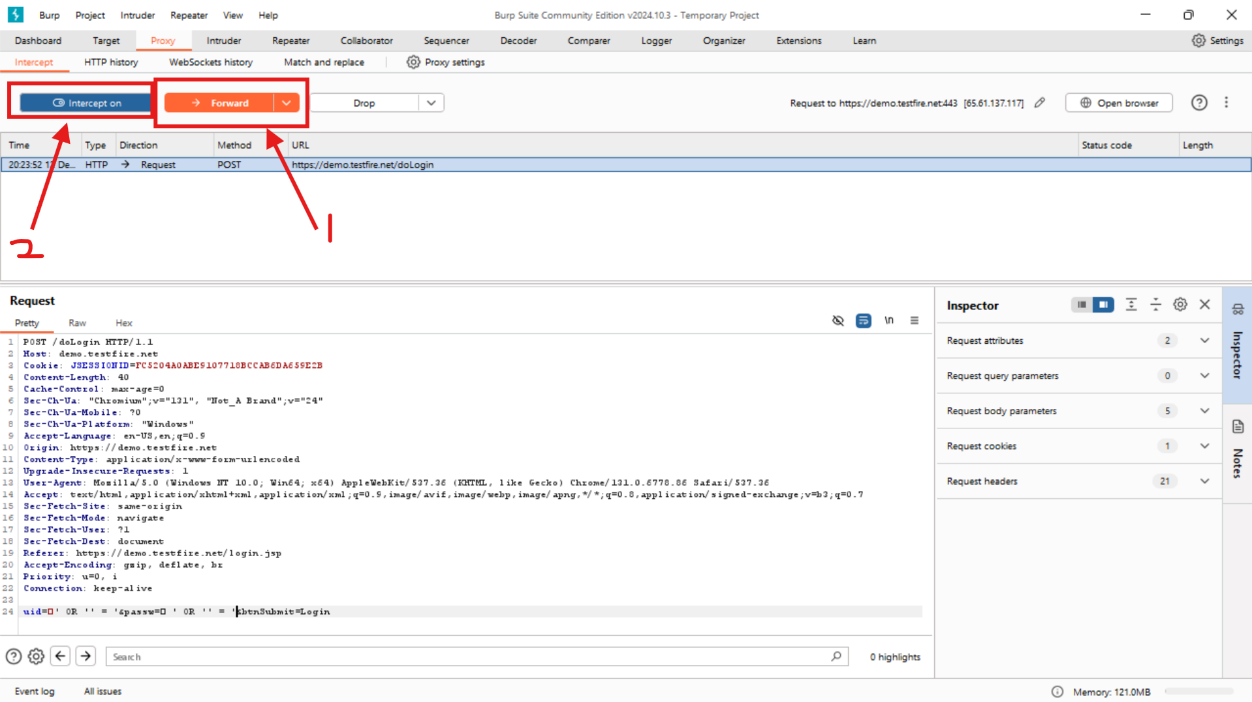
* Username :- Admin
* Password :- Password



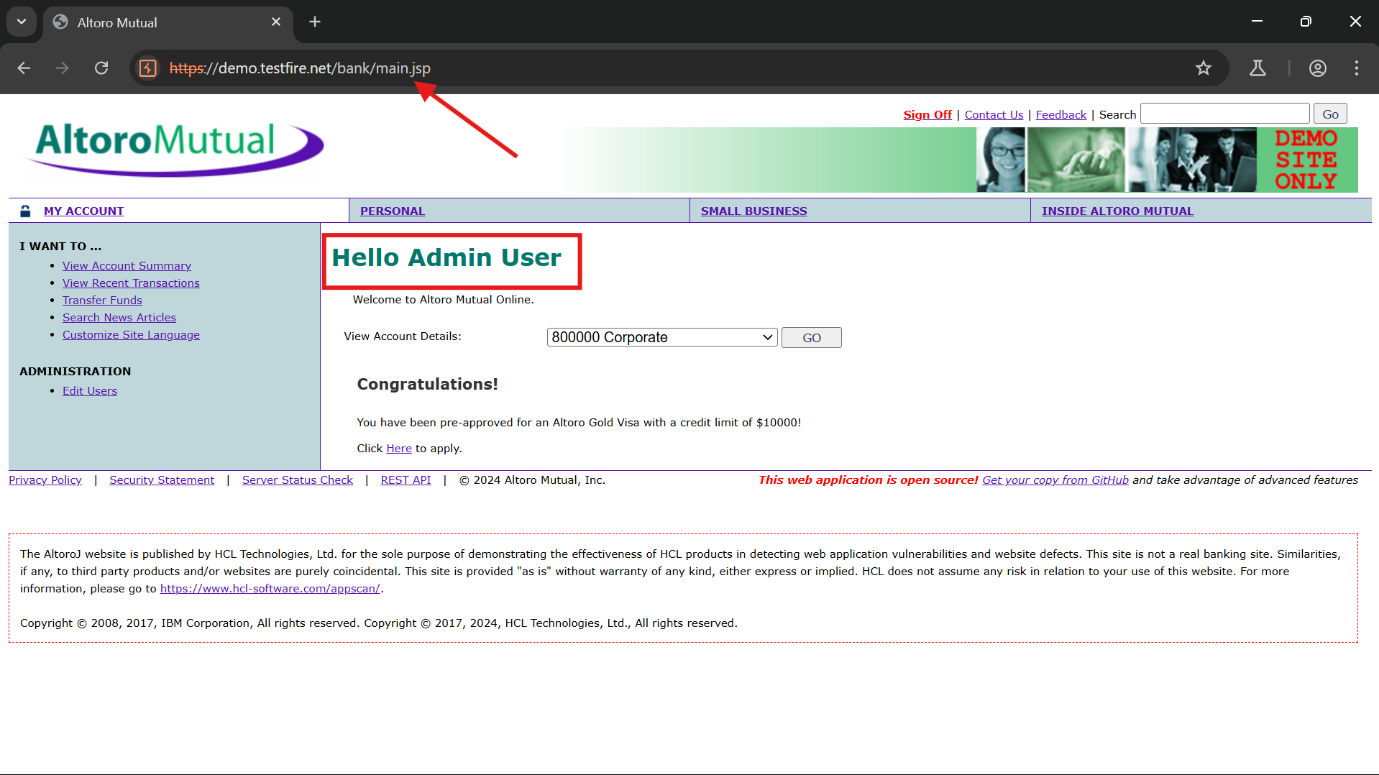
* Now switch on the intercept in burpsuite
* And now capture the request
* Now let’s modify the username and password with some SQL injection payloads.
* SQL injection payloads :-
* **' or "**
* **or #**
* **' OR '1**
* **' OR 1 -- -**
* **" OR "" = "**
* **" OR 1 = 1 -- -**
* **' OR '' = '**
* Atlast the following payload worked:-
* **' OR '' = '**

****

* After modifying , First Forward the request
* And now Switch off the intercept in burpsuite.



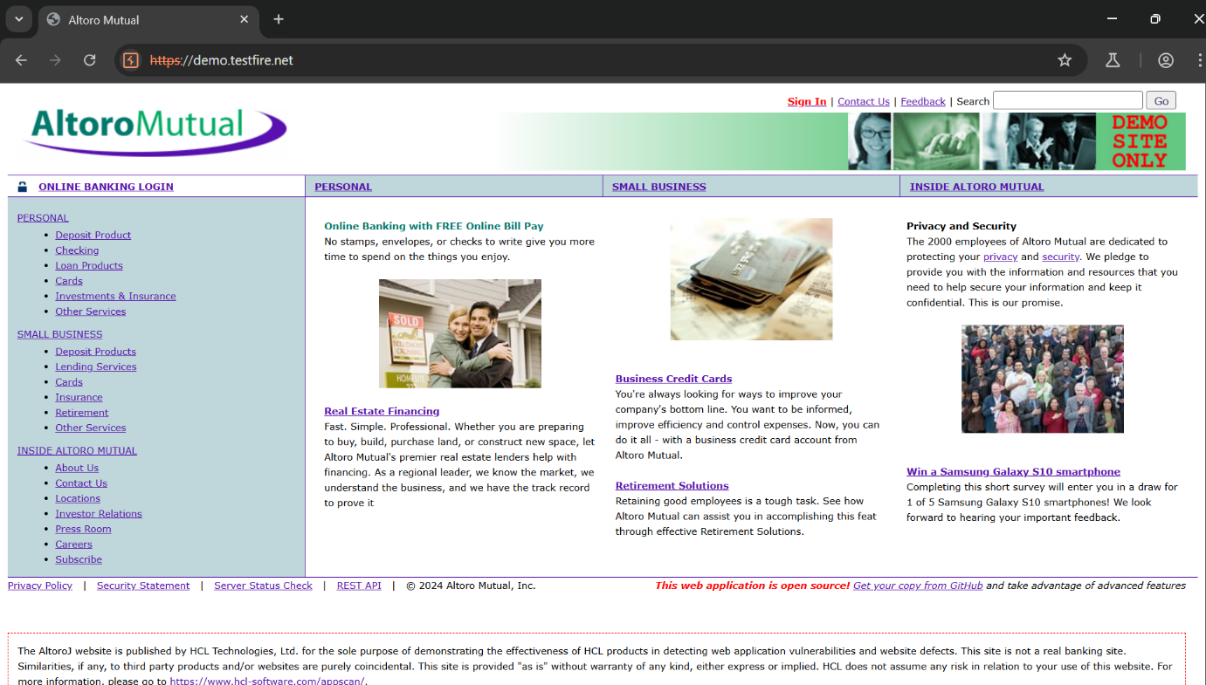
* Now visit the website



* Here we got the Admin Access without authentication.
* And we can also observe the change in URL directory.

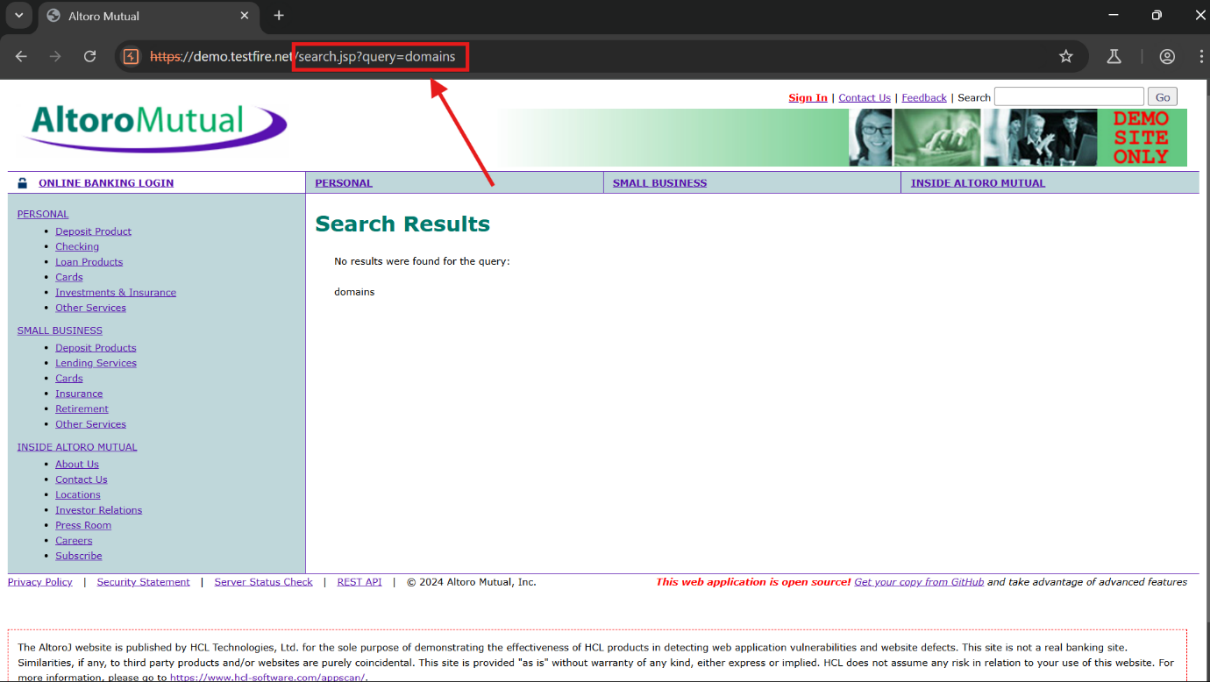
**TASK - 3**

* **Use Burp to pass XSS payloads to a website's API parameters and add screenshots of request and response with payloads**



Link:- <https://demo.testfire.net/>

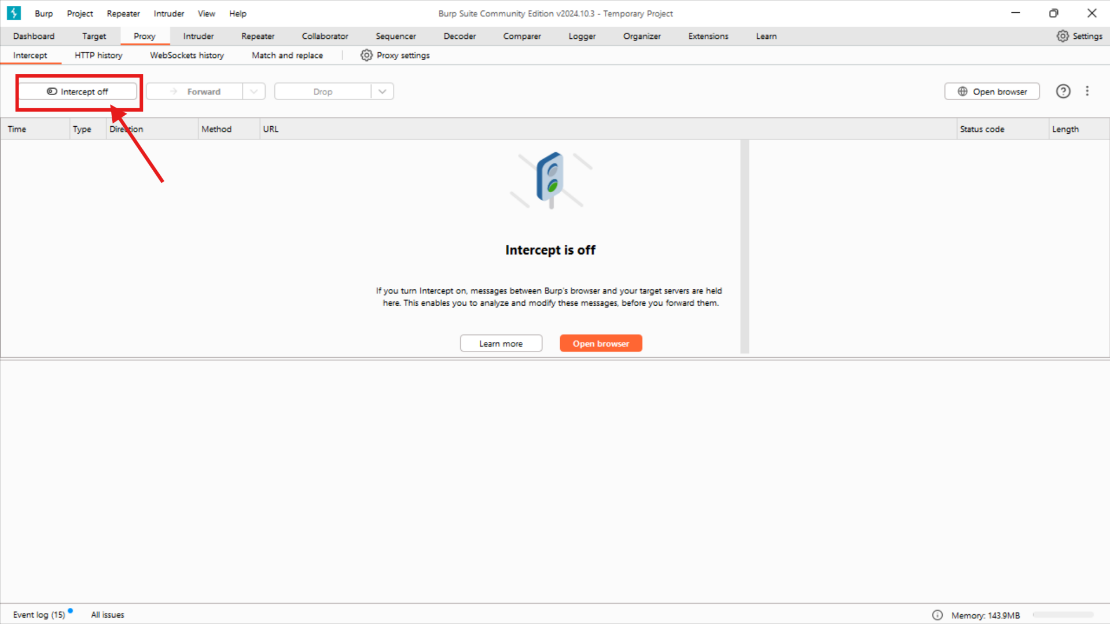
* Performing the reflected XSS attack on the above website.
* Randomly typing “domains” in the input filed



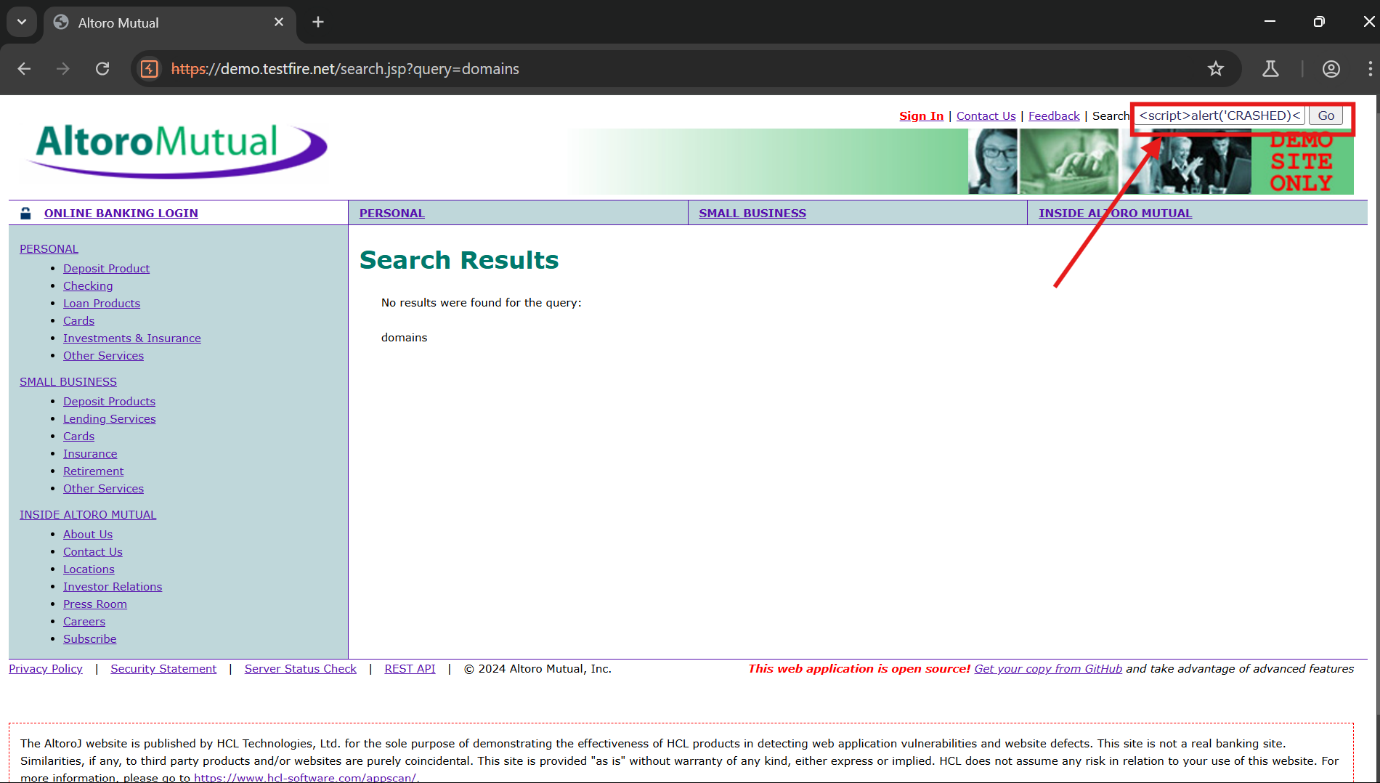
* After typing there is an output of no results,but
* There is a change in the URL
* It changed from

<https://demo.testfire.net/> ---------> <https://demo.testfire.net/search.jsp?query=domains>

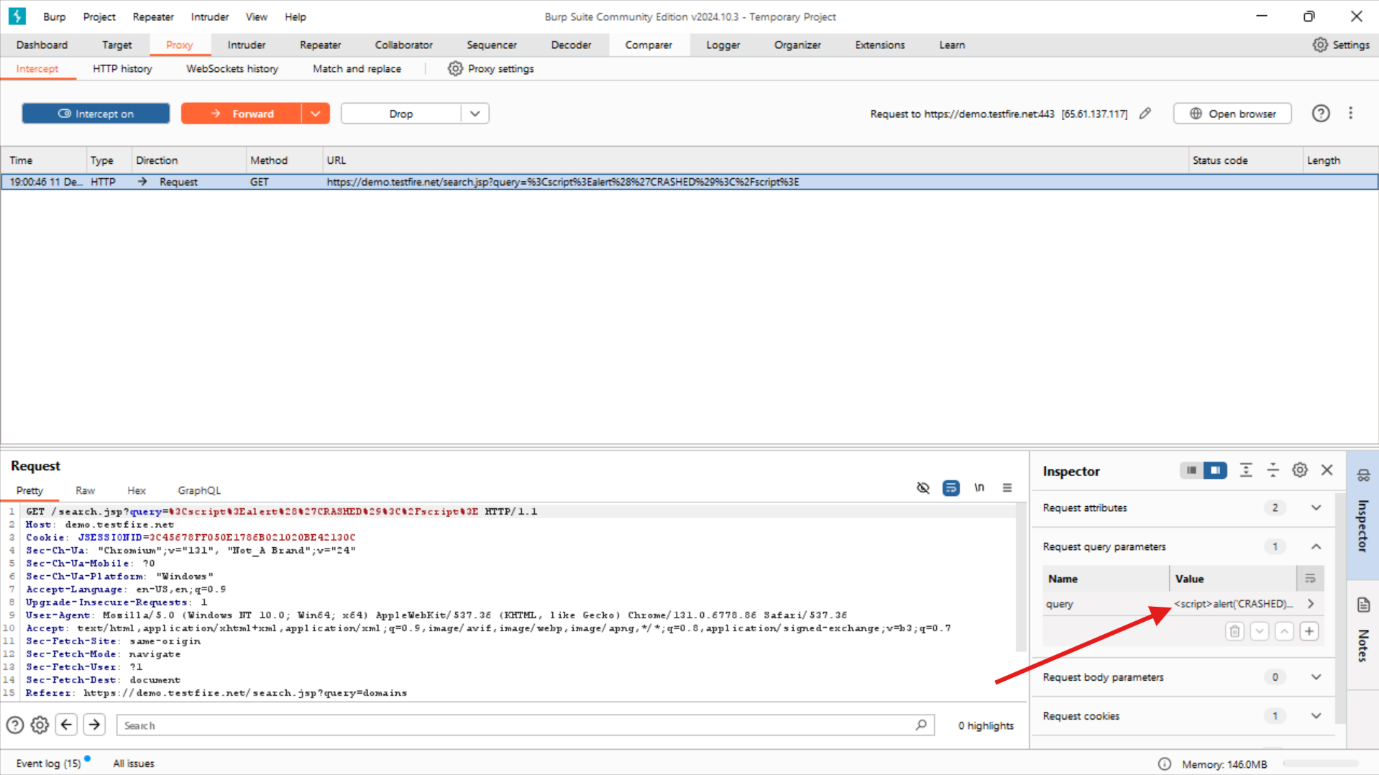
* So there is a chance of Reflected XSS attack
* Let’s perform a basic script for reflected XSS using BURP
* Open the Burpsuite and go to proxy and switch on the intercept



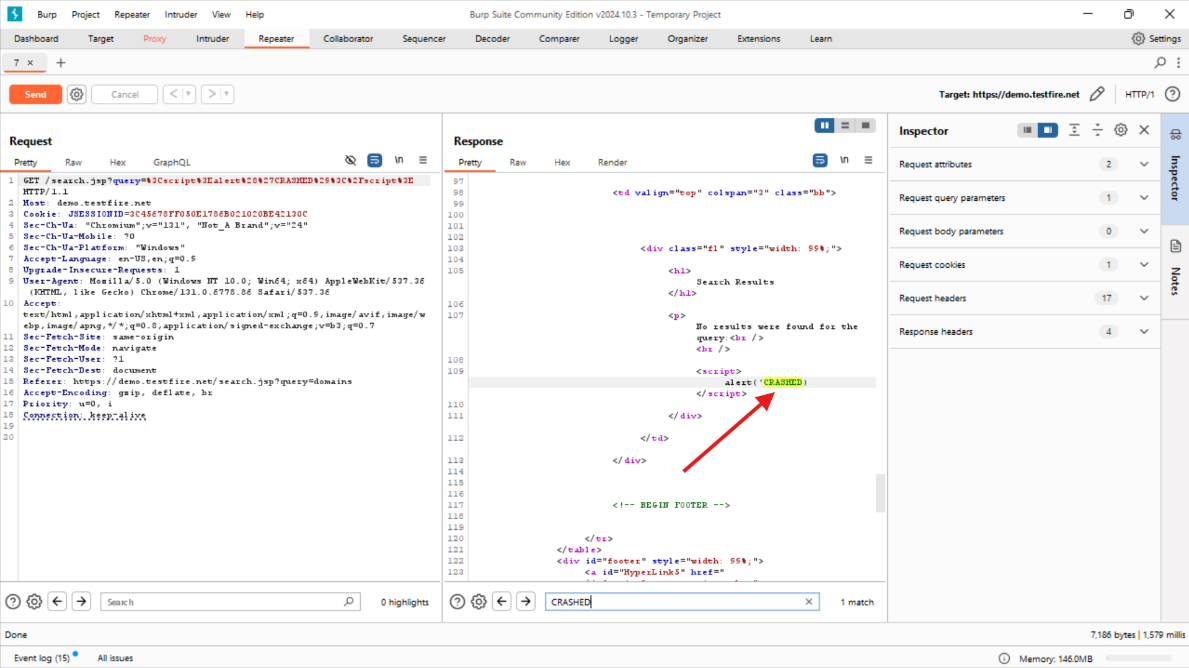
* Now type the above script in the input filed of the website

**<script>alert('CRASHED)</script>**

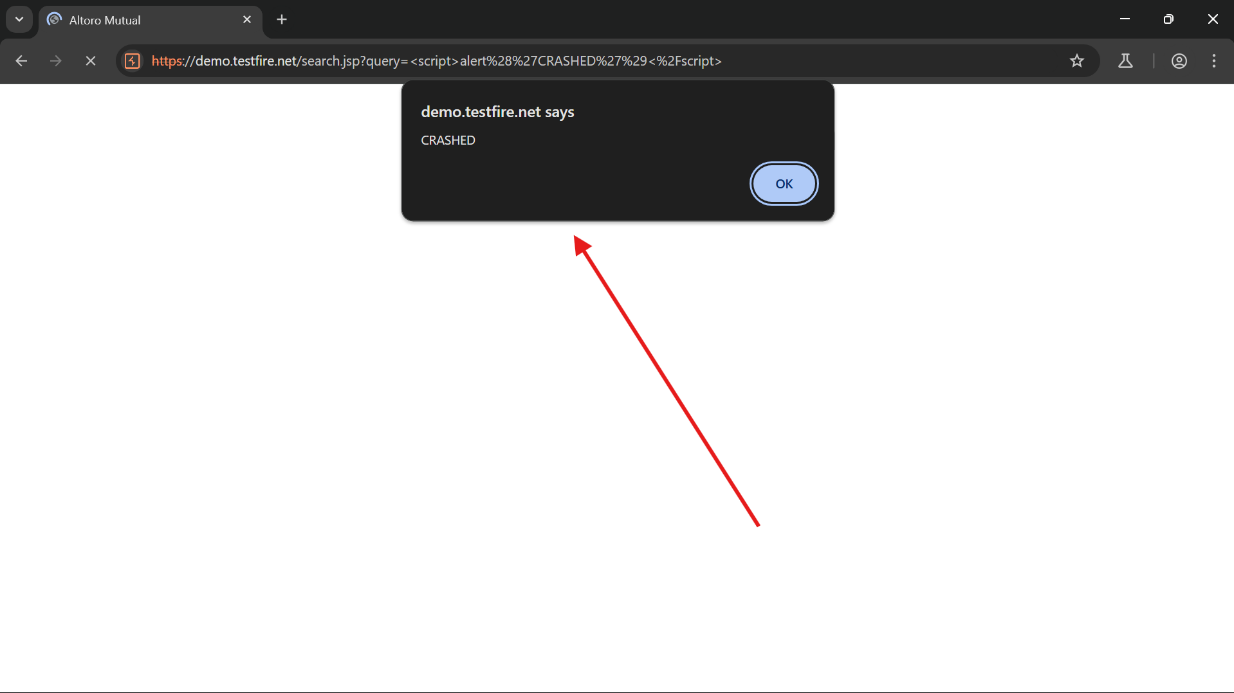
* Now press enter
* Observing the traffic in burpsuite

****

* We can see the script in the burpsuite and let’s send it to repeater
* And observe the response of the output for given script

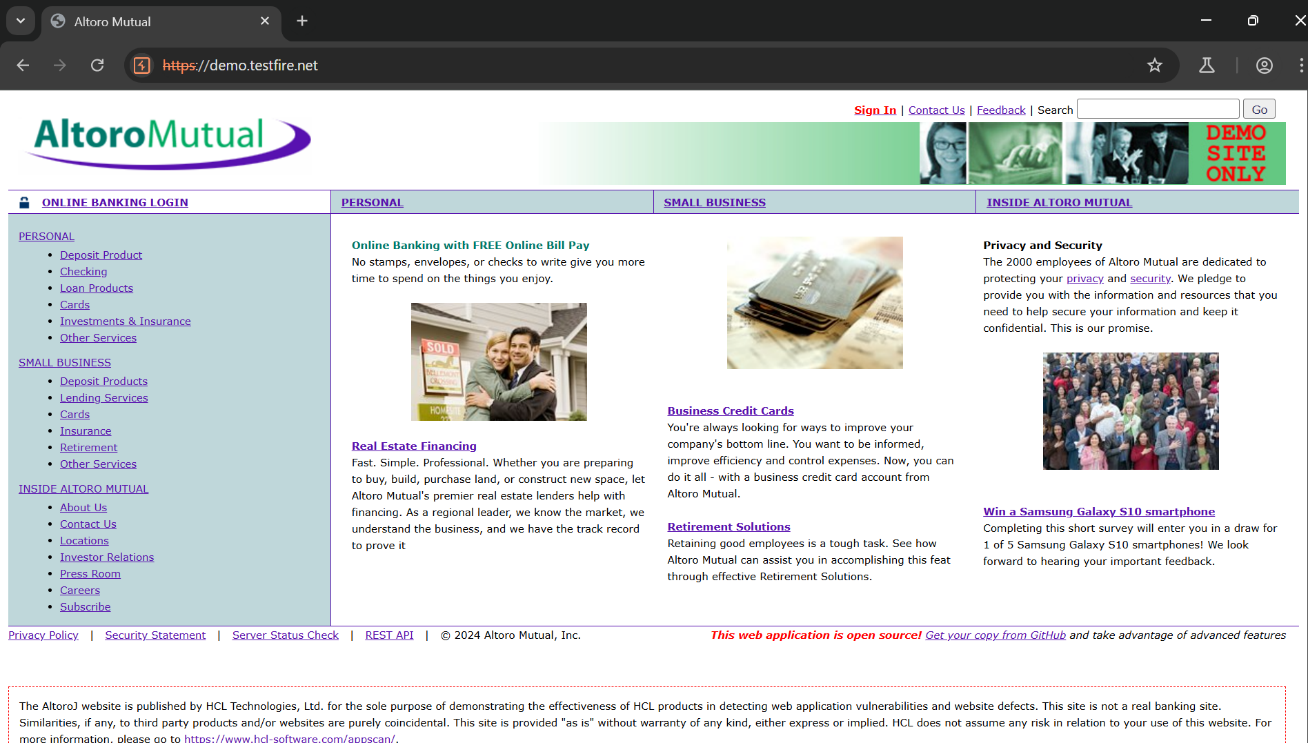


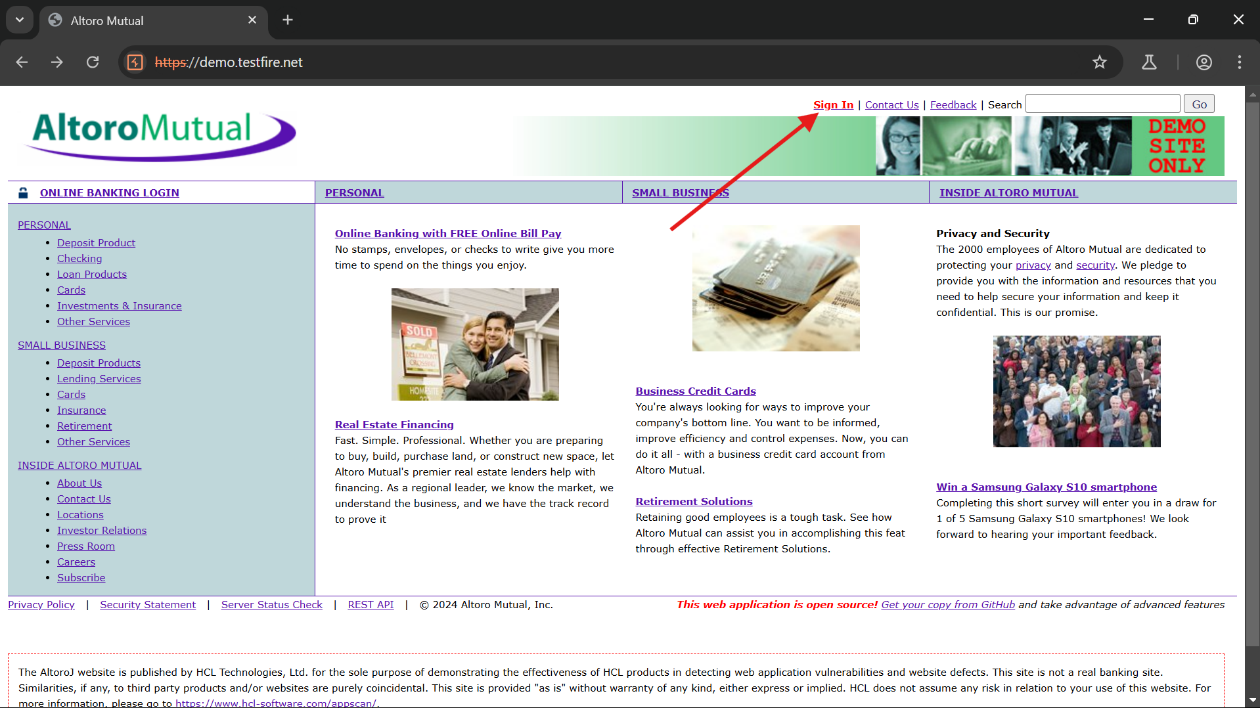
* Here we can see that our script is relfecting in the ouput
* Now lets send the response in the current browser

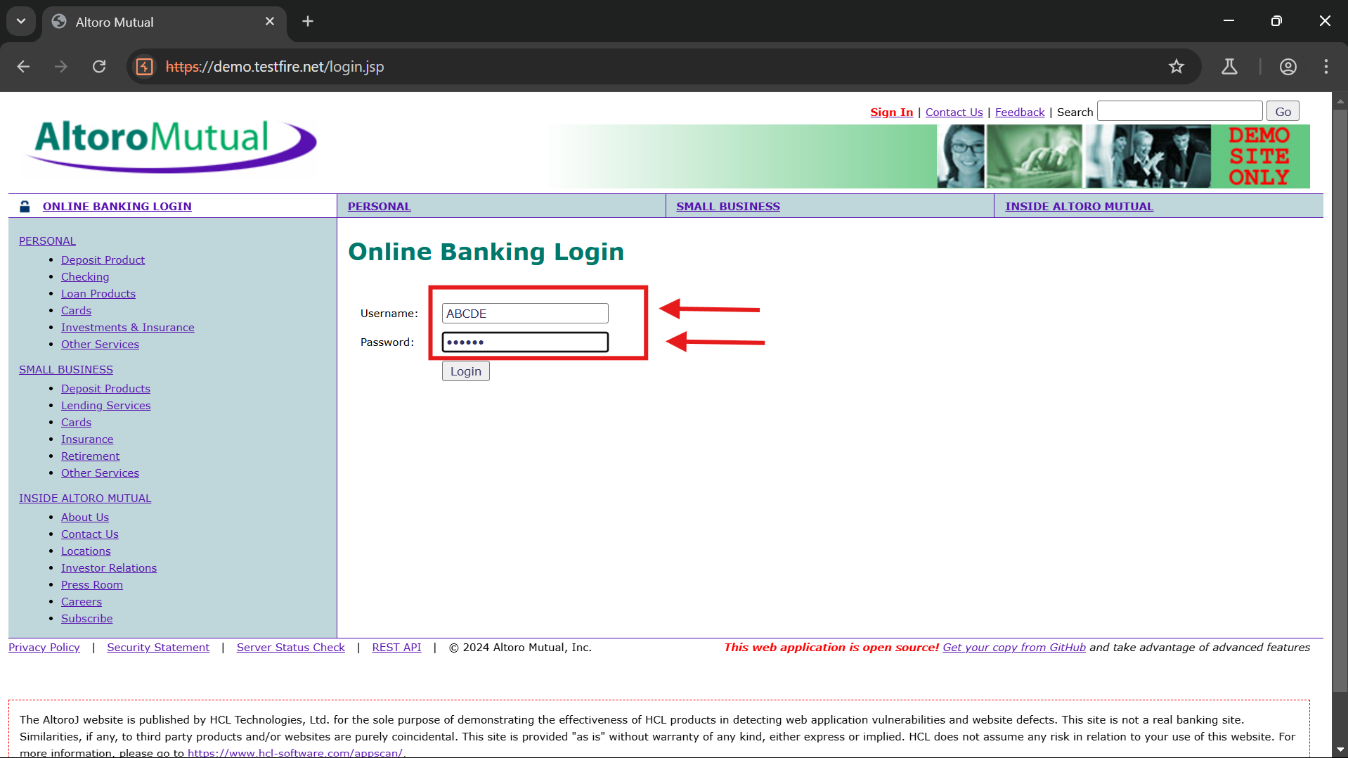
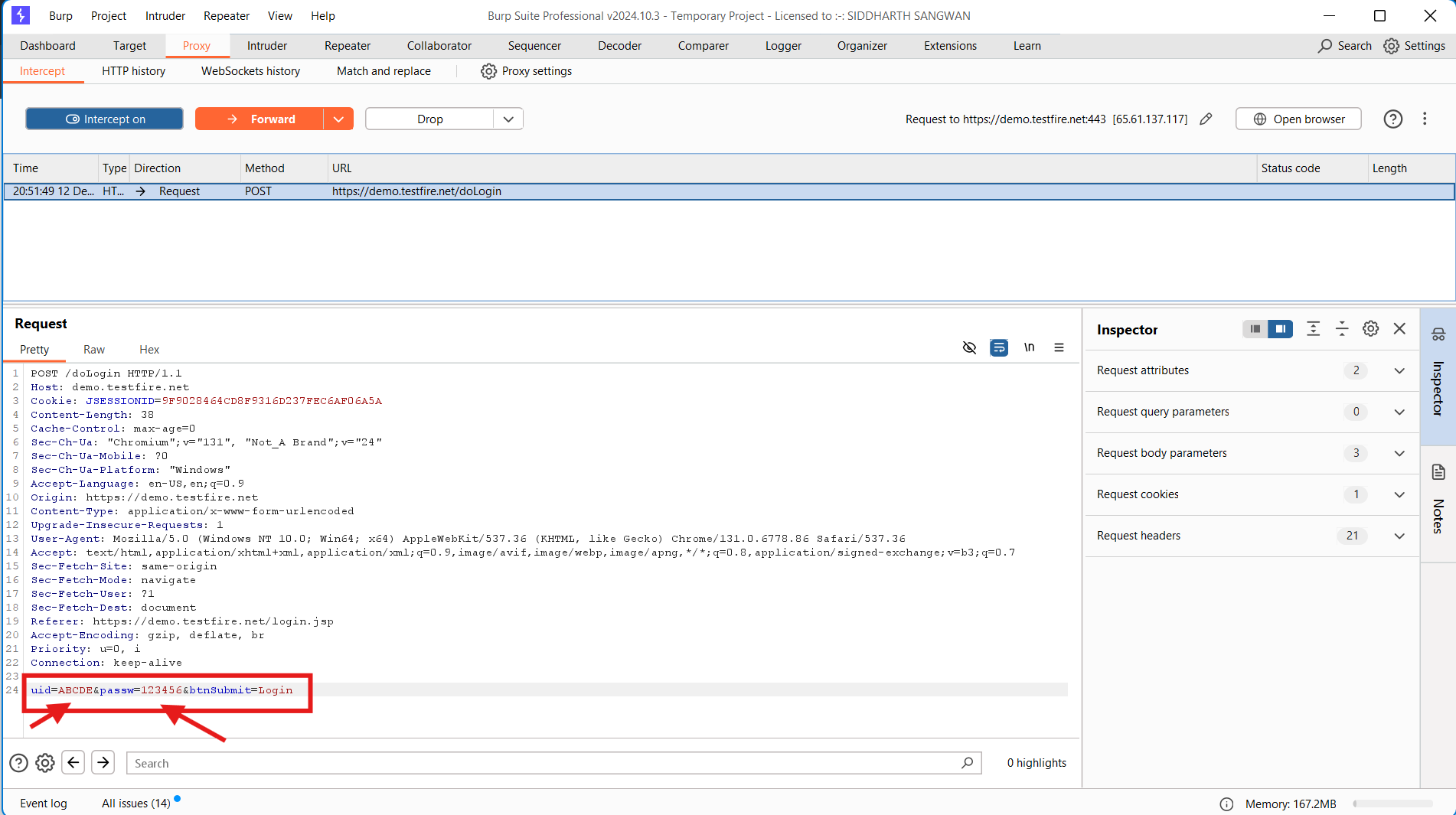
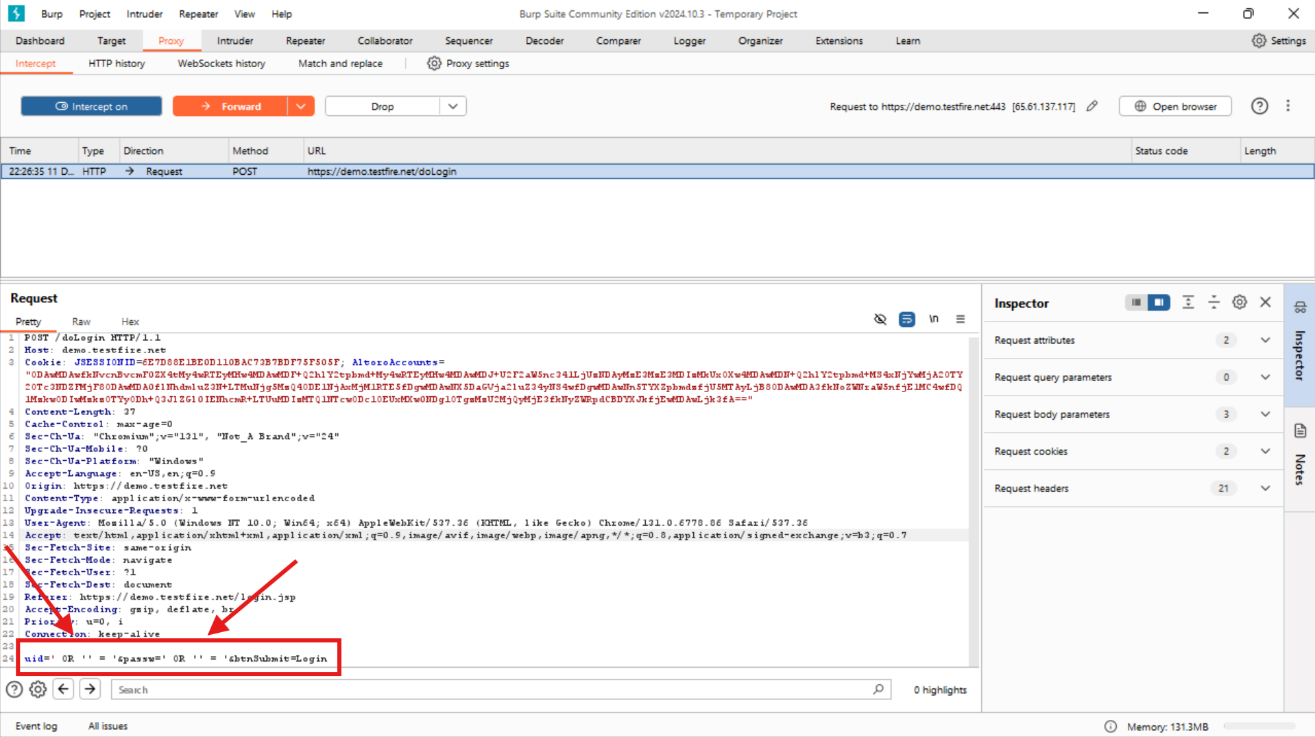


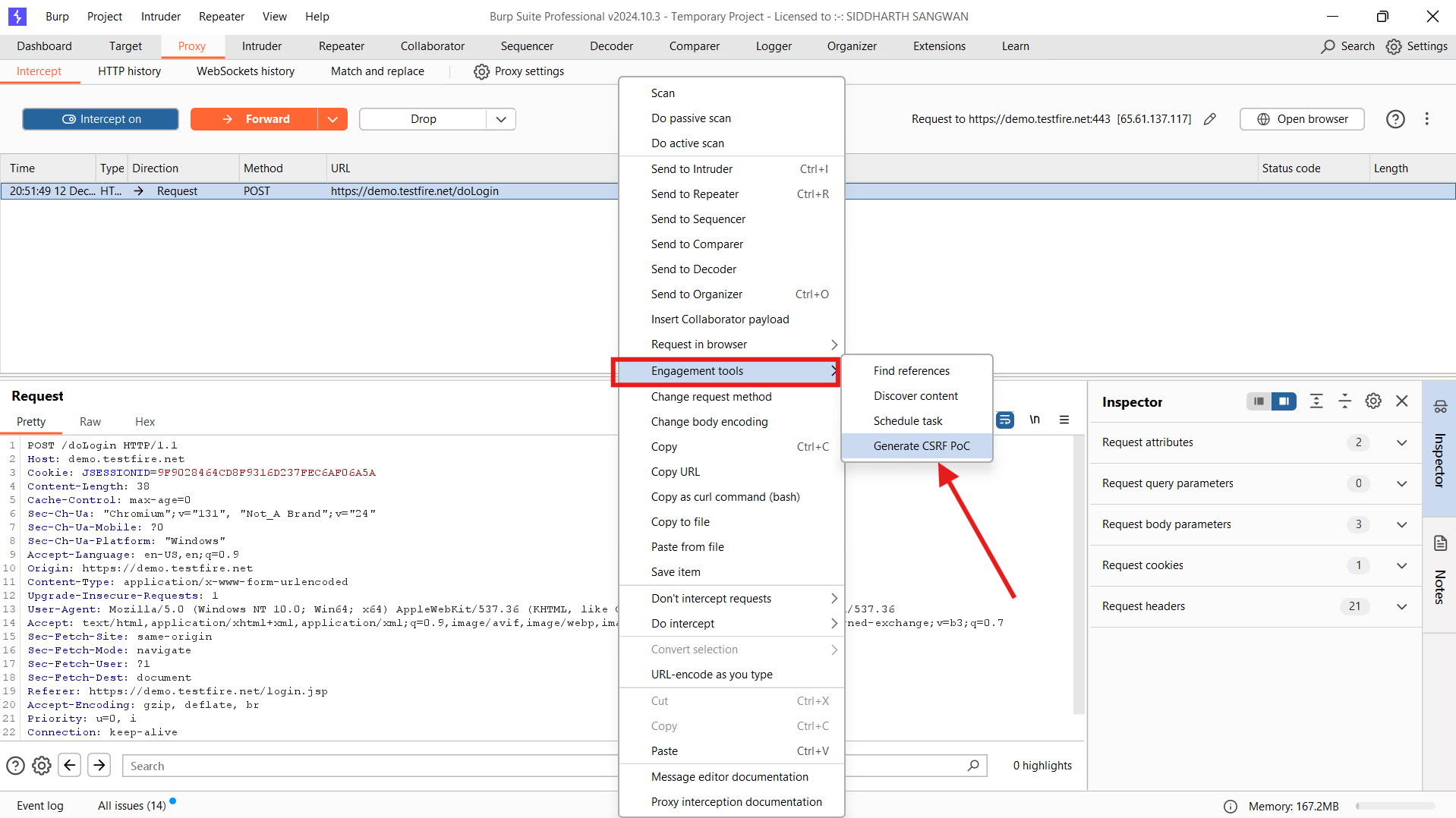
* Here we can see that **CRASHED** is reflected as per the given script.
* Hence the site is vulnerable of Reflected XSS.

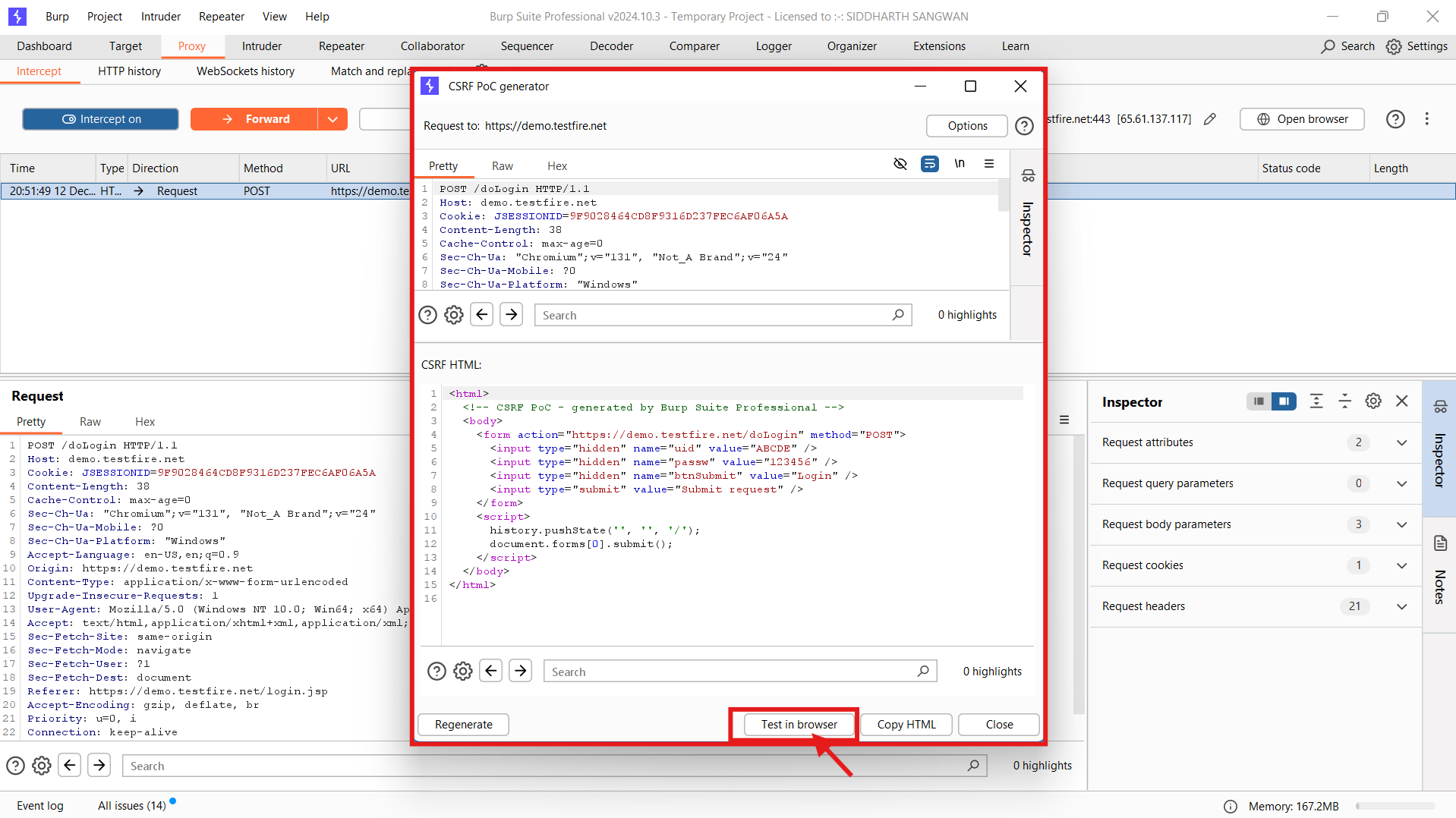
**TASK – 4**

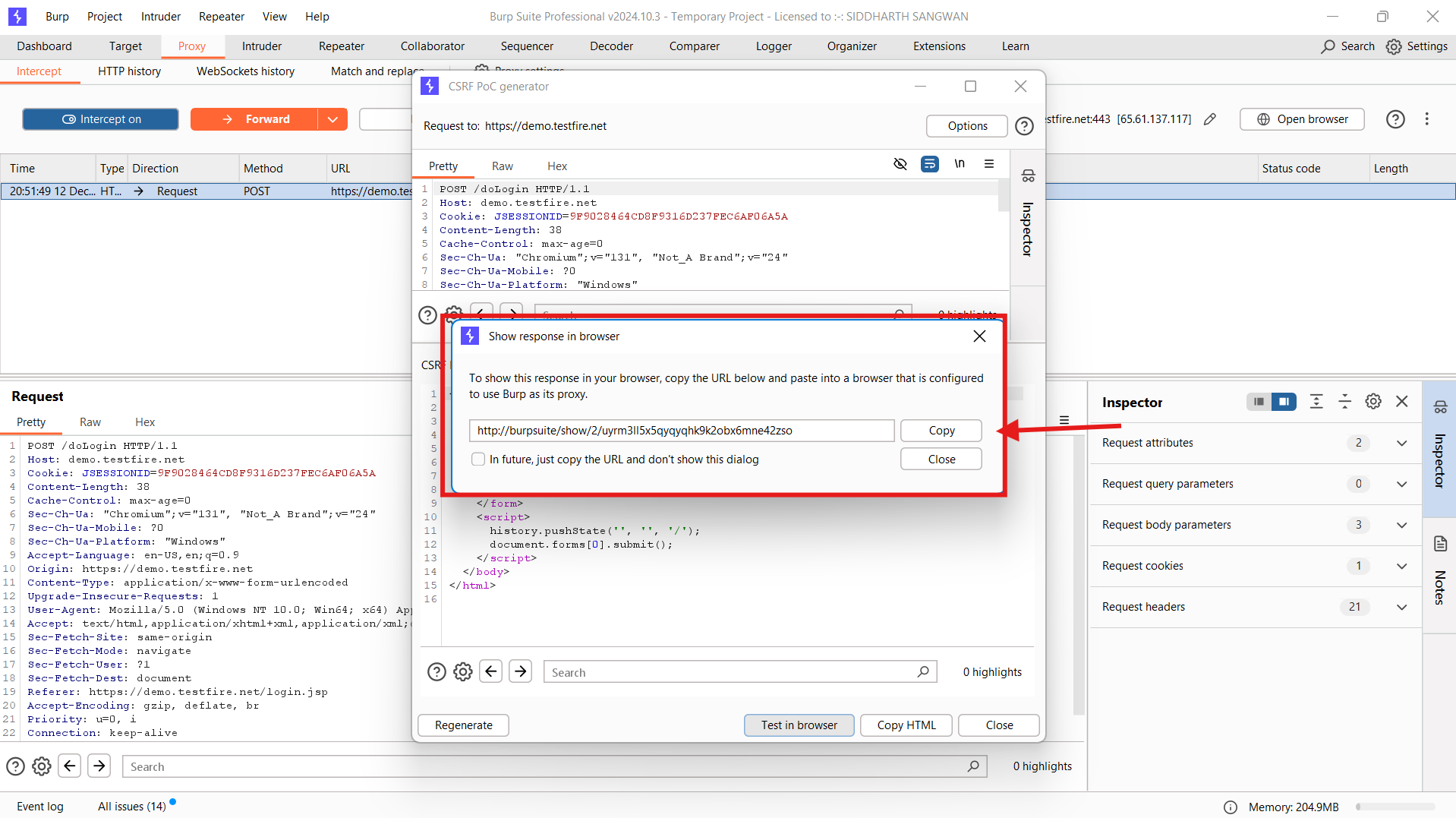
* **Find the CSRF attack and add the screenshot of PoC and the response.**
* ****First let’s visit the website.
* Now click sign in.

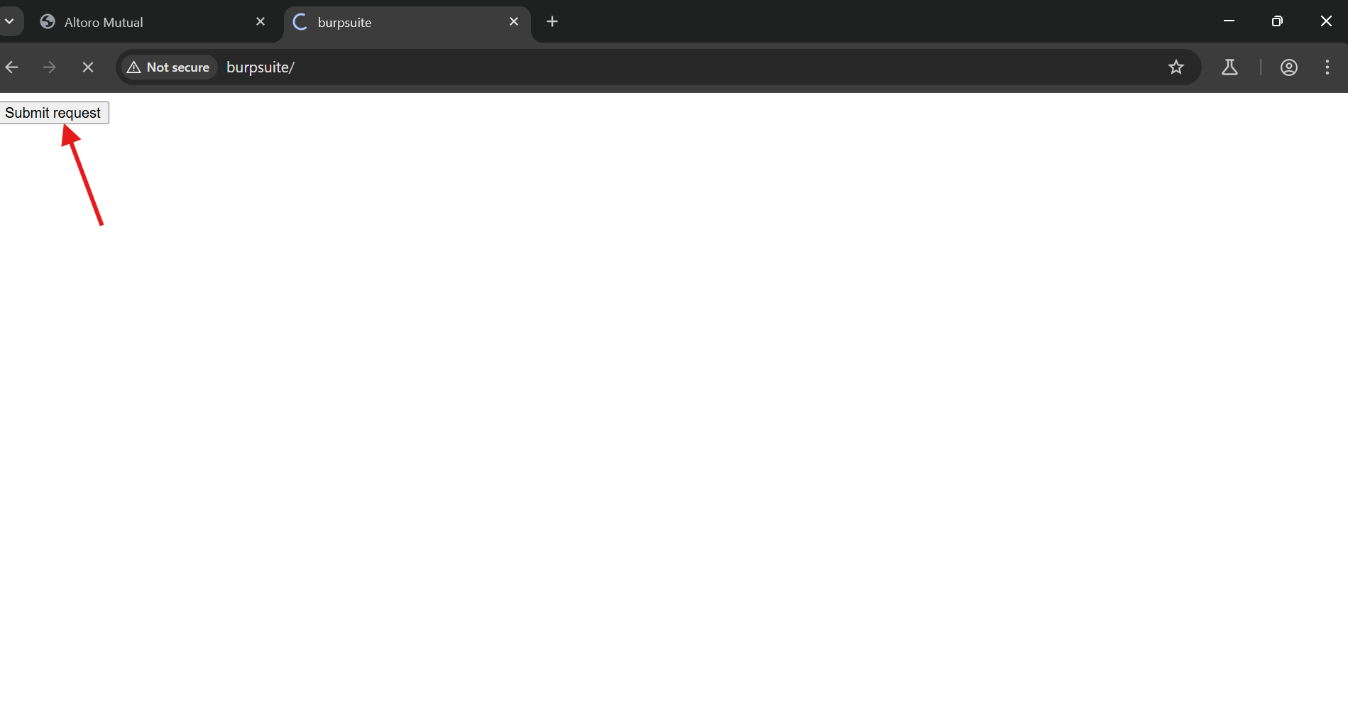
****

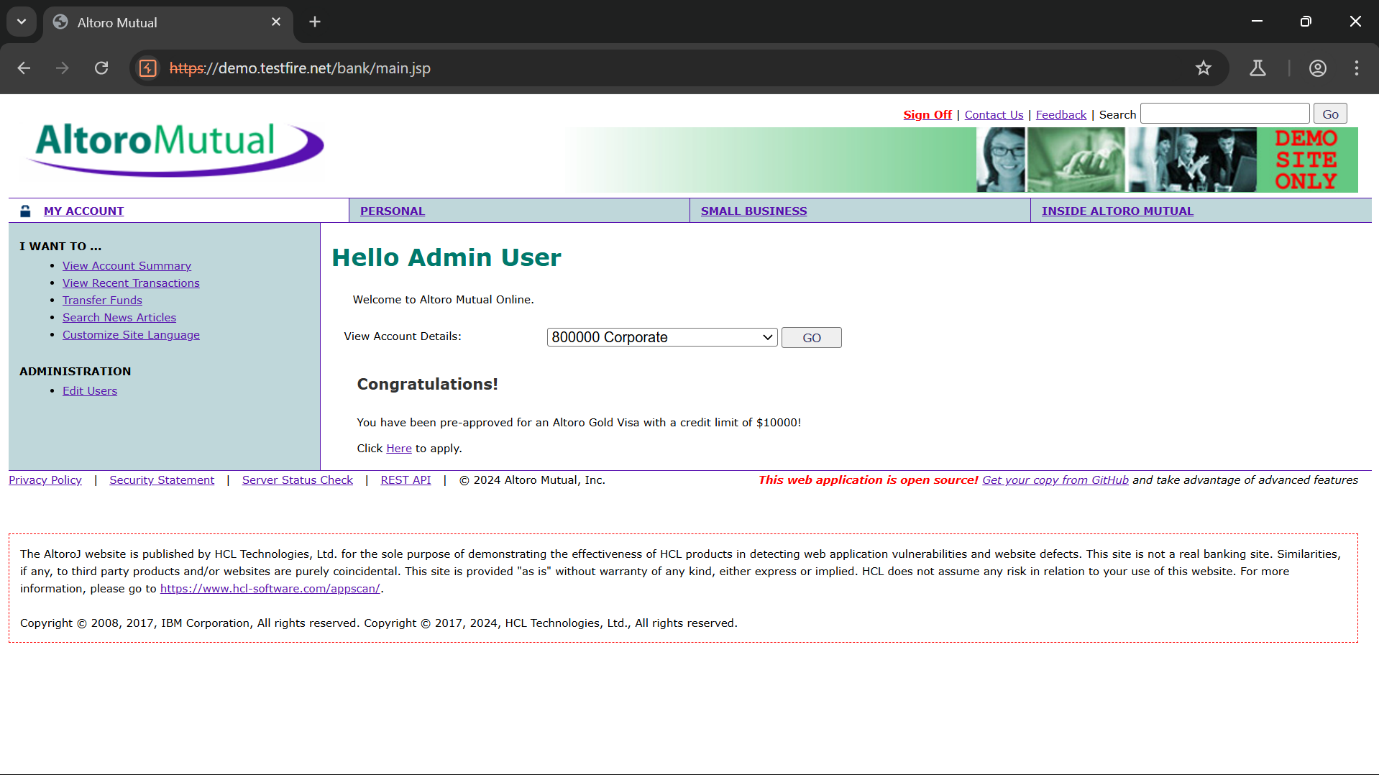
* Now let’s enter the Random credentials as follows :-
* Username :- ABCDE
* Password :- 123456
* Now Switch on the Intercept in Burpsuite.
* And capture the above Request.
* As we know before that there is a sql injection Vulnerability in the above site.
* So Let’s take the advantage of it and modify it with the following payload :-
* **' OR '' = '**
* Now let’s generate a POC.
* Now Right click on the Request
* And Select Engagement Tools.
* In that click Generate CSRF POC.



* Now we will be getting the response of the generated CSRF POC.
* After that we have to test that in the browser by clicking the Test in browser.
* Now click on copy and paste it in the browser.



* Now paste it in the Browser and Click submit.
* After submitting the request , we will receive the admin access.



* Hence CSRF attack has been done by generating POC.

**NOTE:-** The above attack can only be done in Burpsuite Professional.