ASSIGNMENT - 4

BurpSuite

Name : Bhumireddy Thanmaye

Reg No: 21BCI0013

Course: AI for Cyber Security With IBM Qradar (AI for Web Security)

Branch: Computer Science and Engineering with specialization in Information Security

Objective:

- What is a burp suite?
- Why burp suite?
- What are the features of burp suite?
- Test the vulnerabilities of testfire.net http://testfire.net

What is a Burp Suite ?

Burp Suite is a proxy program that enables us to track, examine, and alter requests made by our browsers before they are forwarded to a distant server.

Burp Suite is a prominent web application security solution. It gives us the ability to manually test for vulnerabilities, intercepts HTTP messages, and change a message's body and header.

It was created by a business with the alias Portswigger, whose creator Dafydd Stuttard also works there. BurpSuite is designed to be an all-in-one toolkit, and BApps are add-ons that may be installed to expand its functionality.

It is the most widely used tool among experts in online app security and bug bounty hunters. It is a better option than free substitutes like OWASP

ZAP because of how simple it is to use. The community edition of Burp Suite is accessible for free, whereas the professional edition and the enterprise edition need payment.

Why do we use Burp Suite ?

Burp Suite is a comprehensive framework that may be used to carry out several activities, including:

- Web crawling.
- Web application testing, both manually and automatically.
- Analysis of web applications.
- Vulnerability detection
- Burp Suite also has the advantage of being built into the Chrome browser.

What are the features of Burp Suite ?

Manual penetration testing features

Intercept everything your browser sees

Burp Suite's built-in browser works right out of the box - enabling you to modify every HTTP message that passes through it.

Quickly assess your target

Determine the size of your target application. Auto-enumeration of static and dynamic URLs, and URL parameters.

Speed up granular workflows

Modify and reissue individual HTTP and WebSocket messages, and analyze the response - within a single window.

Manage recon data

All target data is aggregated and stored in a target site map - with filtering and annotation functions.

Expose hidden attack surface

Find hidden target functionality with an advanced automatic discovery function for "invisible" content.

Break HTTPS effectively

Proxy even secure HTTPS traffic, using Burp Suite's built-in instrumented browser.

Work with HTTP/2

Burp Suite offers unrivaled support for HTTP/2-based testing - enabling you to work with HTTP/2 requests in ways that other tools cannot.

Work with WebSockets

WebSockets messages get their own specific history - allowing you to view and modify them.

Manually test for out-of-band vulnerabilities

Make use of a dedicated client to incorporate Burp Suite's out-of-band (OAST) capabilities during manual testing.

DOM Invader

Use Burp Suite's built-in browser to test for DOM XSS vulnerabilities more easily - with DOM Invader.

Assess token strength

Easily test the quality of randomness in data items intended to be unpredictable (e.g. tokens).

Advanced / custom automated attacks

Faster brute-forcing and fuzzing

Deploy custom sequences of HTTP requests containing multiple payload sets. Radically reduce time spent on many tasks.

Query automated attack results

Capture automated results in customized tables, then filter and annotate to find interesting entries / improve subsequent attacks.

Construct CSRF exploits

Easily generate CSRF proof-of-concept attacks. Select any suitable request to generate exploit HTML.

Facilitate deeper manual testing

See reflected / stored inputs even when a bug is not confirmed. Facilitates testing for issues like XSS.

Scan as you browse

The option to passively scan every request you make, or to perform active scans on specific URLs.

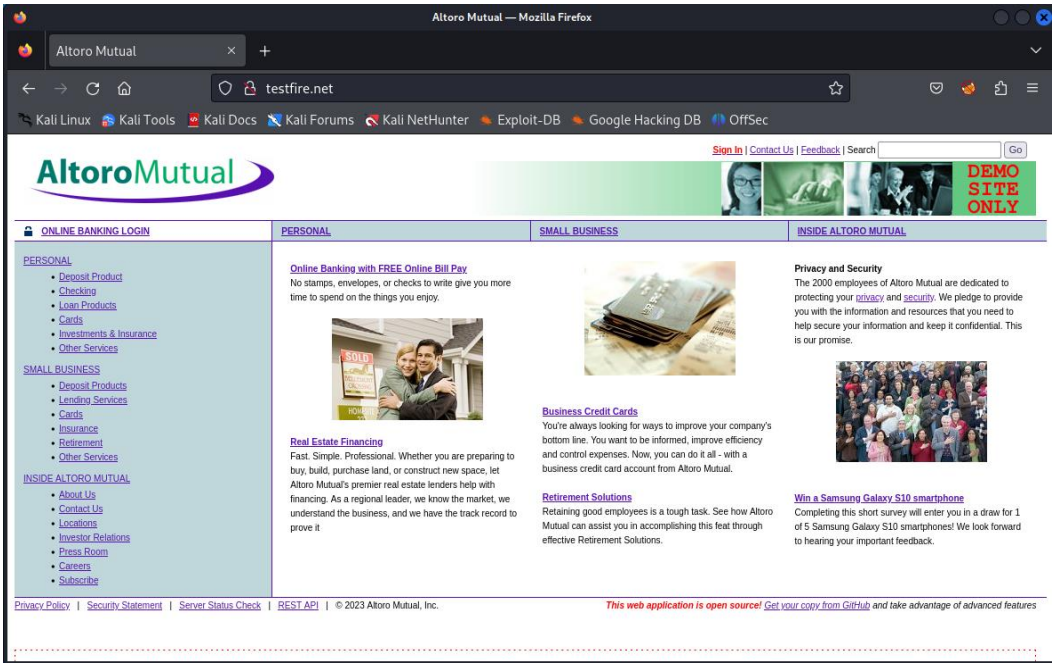
Automatically modify HTTP messages

Settings to automatically modify responses. Match and replace rules for both responses and requests.

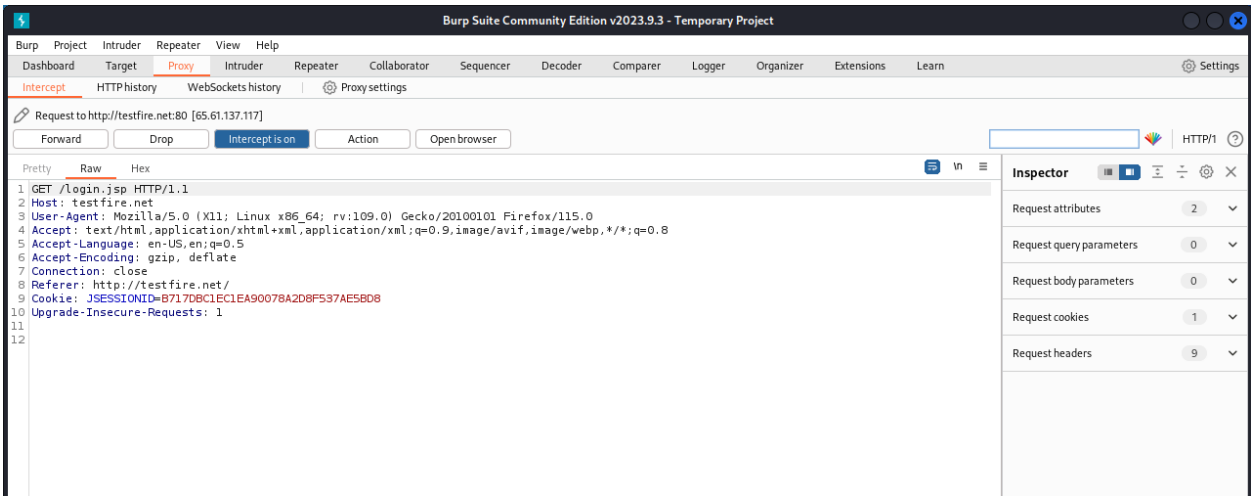
Test the Vulnerabilities of testfire.net using Burp suite ?

Test Website: http://testfire.net

Software Used : **Burp Suite**

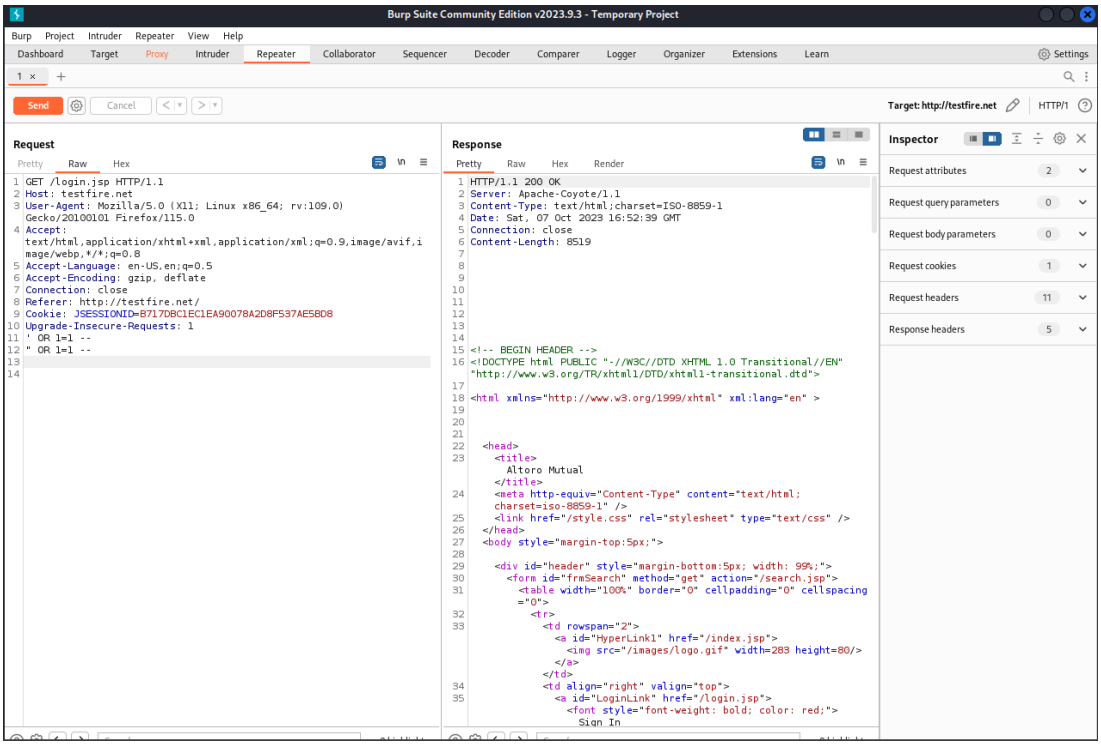


Open test fire website in a browser use foxyproxy so it can transmit to burp.Go to burpsuite an don the intercept to get the traffic interception to burpsuite

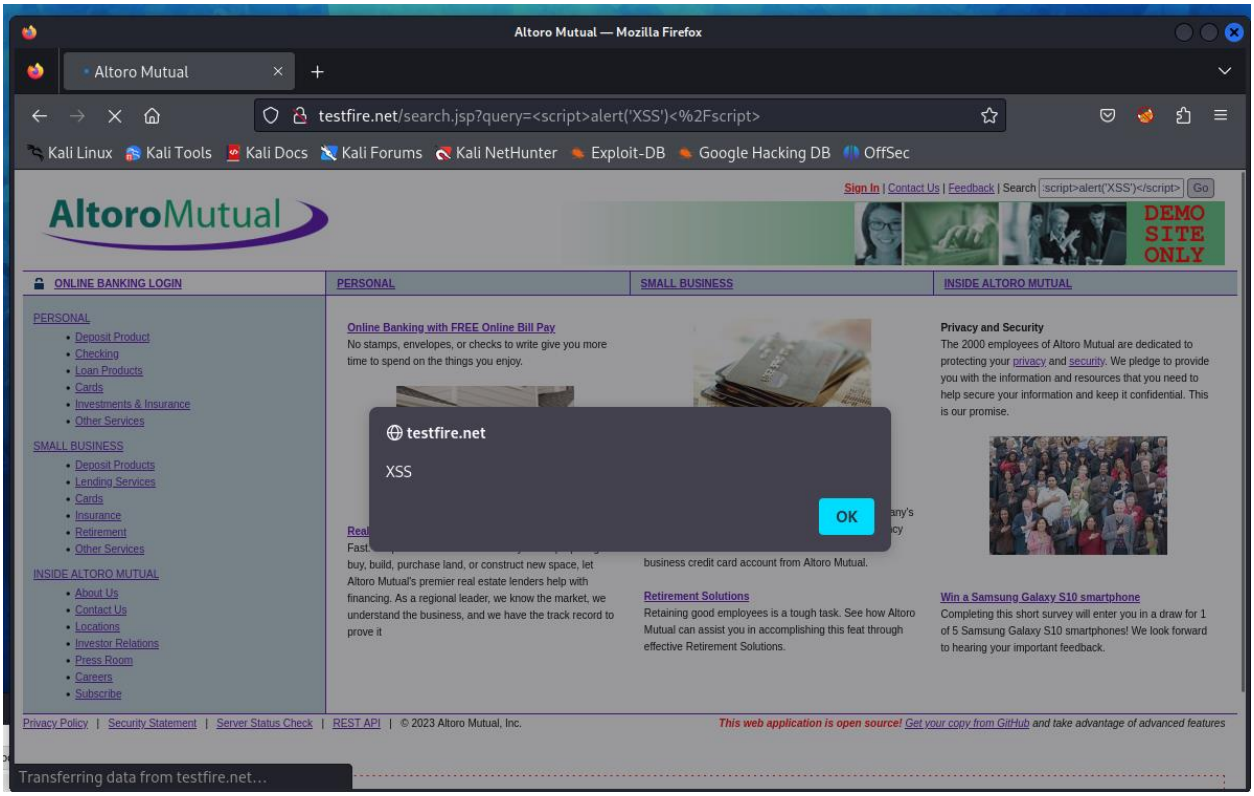


Click on different sections in test fire website and in proxy intercept right click, send to repeater/intruder for attacks after finding any vulnerabilities

Trying sql injection payloads to observe the website changes does that have this security issue /not



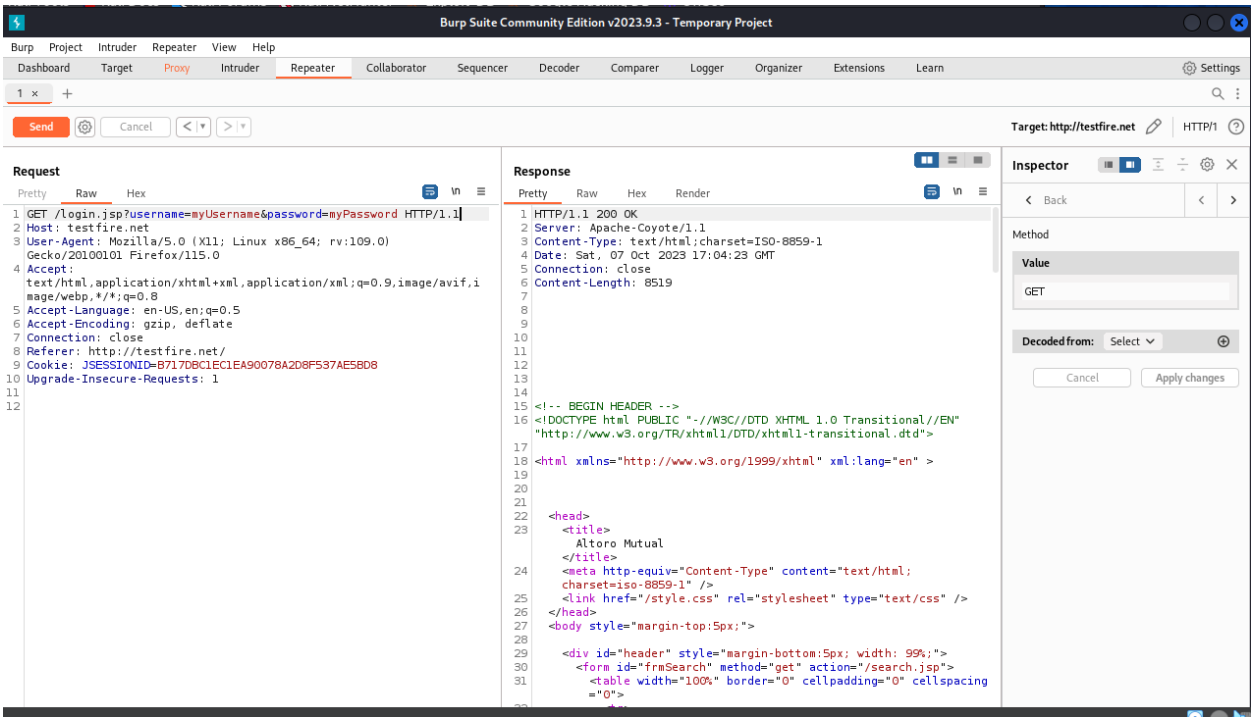
Adding XSS malicious script to website to know its has this security issue /not

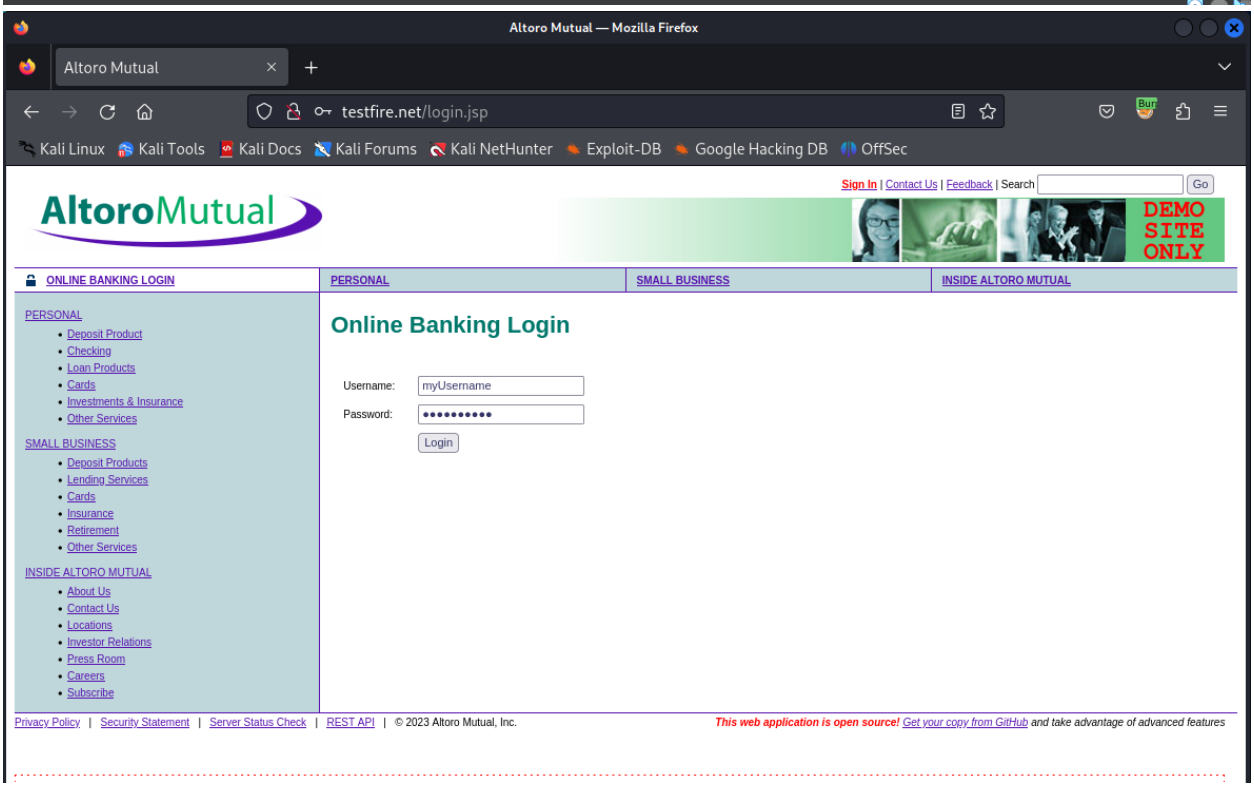


Its transferring the data

Here for doing the authentication testing by using username and password parameters when i gave my own to repeater its accepting 200 ok

GET /login.jsp?username=myUsername&password=myPassword HTTP/1.1





1.Username Enumeration: An attacker could use this login form to guess valid usernames by repeatedly changing the `username` parameter and

analyzing the responses for differences in error messages or behavior. This could lead to a username enumeration vulnerability.

2.Insecure Communication: If the application is not using HTTPS (SSL/TLS) to secure the login process, it could be vulnerable to eavesdropping and man-in-the-middle attacks.

3. Weak Passwords: If the application doesn't enforce strong password policies, users may choose weak or easily guessable passwords, making it vulnerable to brute-force attacks.

4. Lack of CAPTCHA: Without CAPTCHA or other anti-automation measures, the application may be vulnerable to automated brute-force attacks.

5. Session Management: While not directly visible in this request, session management issues can be present. If the application doesn't securely manage sessions after successful login, it could lead to session fixation or session hijacking vulnerabilities.

6. Cookie Security: The `Cookie` header contains a session ID. If this session ID is not properly secured or if the session ID is exposed inappropriately, it could be vulnerable to session-related attacks.

7. Referrer Leakage: The `Referer` header reveals the referring URL. If sensitive information is inadvertently exposed in the URL or if the referrer header is mishandled, it could lead to information leakage.

8. Upgrade-Insecure-Requests Header : If the application doesn't properly enforce HTTPS, the `Upgrade-Insecure-Requests` header might not be effective in securing the request.

We can also observe this type of security vulnerability if the web application does not flow the security things properly