**Experiment 6**

**Shashwat Tripathi**

**D20A 64**

**Aim:** Use burp proxy to test web applications.

**Theory:**

Burp Suite is a leading web security tool used by security professionals, penetration testers, and developers to find vulnerabilities in web applications. Acting as an intermediary between your browser and the web server, Burp Suite captures, inspects, and modifies the data exchanged between them, helping identify security flaws such as SQL injection, Cross-Site Scripting (XSS), and more.

### **Tools offered by Burp Suite:**

1. **Intercepting Proxy**: The Intercepting Proxy is the core feature of Burp Suite. It sits between your web browser and the target server, capturing all HTTP/S traffic. This allows you to review, alter, and modify requests and responses before they’re processed. By doing so, you can simulate different types of attacks and understand how the application responds to them.
2. **Automated Scanner**: Available in the Professional Edition, the automated Scanner searches for common web vulnerabilities like SQL injection and XSS. The scanner can operate in two modes: passive scanning, where it observes the interaction without making any changes, and active scanning, where it probes for vulnerabilities by sending custom inputs. The tool provides detailed reports that outline the severity of each issue and suggest possible fixes.
3. **Intruder**: Intruder is designed to automate repetitive attacks. You can use it to brute-force login pages, test for parameter tampering, or fuzz input fields by injecting various payloads to see how the application handles them. It supports various attack modes, allowing you to customize the type of attack depending on the vulnerability you’re testing.
4. **Repeater**: The Repeater tool allows you to manually test individual web requests. You can capture a request, modify it in different ways (e.g., change the headers, query parameters, or cookies), and resend it multiple times. This is useful for testing how different inputs affect the web application, particularly for fine-tuning vulnerabilities or edge cases.
5. **Sequencer**: Sequencer helps evaluate the randomness of session tokens and other generated values used in web applications. By analyzing a large set of tokens, it determines if they are secure and sufficiently unpredictable. Weak token randomness can allow attackers to hijack sessions or bypass authentication mechanisms.
6. **Decoder**: Decoder is a simple utility for converting data between different formats like Base64, URL encoding, or hexadecimal. It allows testers to decode or encode web requests and responses, making it easier to understand or manipulate the data being transferred.
7. **Comparer**: Comparer helps you identify differences between two sets of data, such as HTTP requests or responses. For instance, after altering input parameters, you can compare responses to spot subtle differences, which could help detect vulnerabilities like parameter tampering.
8. **Spider**: The Spider tool automatically crawls a web application to map out its structure. By following links and gathering endpoints, the Spider ensures that all parts of the web application are accounted for, making sure you don’t miss any untested areas during the assessment.
9. **Extender**: Extender lets you add custom functionality to Burp Suite by integrating third-party extensions or building your own in Java, Python, or Ruby. This flexibility allows you to enhance Burp’s capabilities or link it with other security tools, providing a more tailored security testing experience.
10. **Dashboard**: The Dashboard is a centralized hub for tracking and managing automated tasks like scans. It gives you real-time insights into scan progress, task statuses, and vulnerability findings. You can pause, stop, or prioritize tasks and generate detailed reports for review and remediation planning.
11. **Burp's Browser**: Burp Suite comes with its own pre-configured browser, eliminating the need to manually adjust your external browser's settings to work with Burp. This makes testing quicker and easier, as the Burp browser automatically captures all traffic without additional setup.

### **Burp Repeater**

Burp Repeater is a tool that enables you to modify and send an interesting HTTP or WebSocket message over and over. By resending modified requests through Repeater, you can test how the application responds to various inputs, which is crucial for uncovering vulnerabilities like SQL injection or parameter tampering.

**Steps**:

* **Intercept a Request**: Capture a request using Burp’s Proxy tool.
* **Send to Repeater**: Right-click on the request and choose “Send to Repeater.”
* **Modify the Request**: Edit any part of the request (e.g., parameters, headers).
* **Send and Analyze**: Resend the modified request to the server and examine how the server responds.

You can use Repeater for all kinds of purposes, for example to:

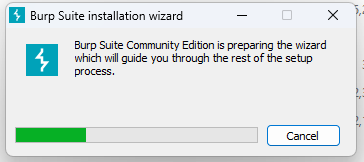
* Send a request with varying parameter values to test for input-based vulnerabilities.
* Send a series of HTTP requests in a specific sequence to test for vulnerabilities in multi-step processes, or vulnerabilities that rely on manipulating the connection state.
* Manually verify issues reported by Burp Scanner.
* Repeater enables you to work on multiple messages simultaneously, each in its own tab. Any modifications you make to a message are saved in the tab's history. You can easily manage large numbers of open tabs with the grouping function. For HTTP requests, you can also add notes to each tab.

### **Practical Use Cases of Burp Suite**

1. **Input Validation Testing**: Input validation testing involves changing input values in web requests to see how well the application validates data. For example, by injecting SQL commands into a form field, you can test for SQL injection vulnerabilities. This helps identify how robust the application’s data handling processes are.
2. **Authentication Testing**: Burp Suite can be used to test the security of login mechanisms by altering session cookies, authentication tokens, or HTTP headers. This helps uncover vulnerabilities related to session management, such as bypassing authentication controls or identifying weak passwords.
3. **Tampering**: Request tampering involves altering web requests—changing parameters, cookies, or headers—to test how the application handles unexpected input. For example, manipulating a user ID in a request can help you discover whether an application improperly exposes sensitive data.
4. **Cross-Site Scripting (XSS) Testing**: Burp Suite can be used to manually inject malicious scripts into web requests to check if they are executed by a user’s browser. If a website reflects this malicious script without proper validation, it indicates an XSS vulnerability, which could allow attackers to steal user data or hijack sessions.

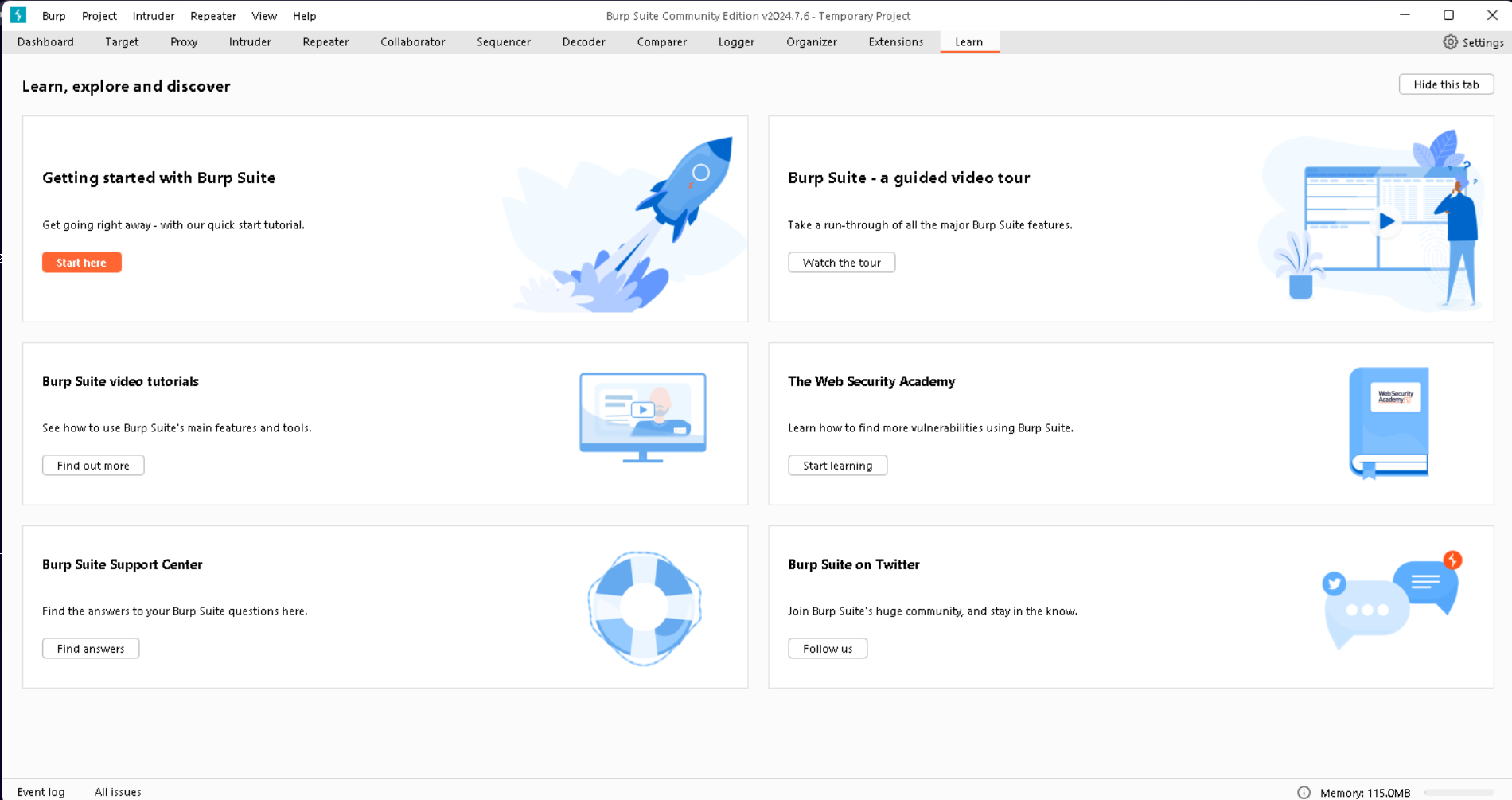
**Steps for Proxy:**

**Installing Burp Suite Community edition.**

****

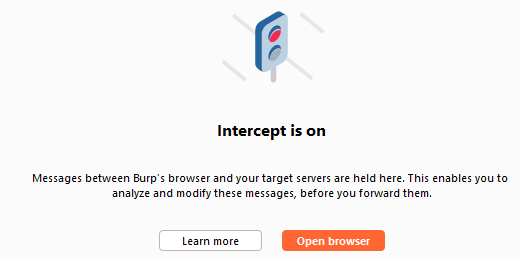
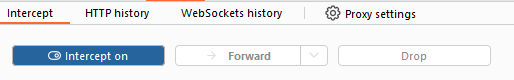


**Burp suite Homepage with its functionalities for security testing :**



**Experimenting with proxy functionality.**

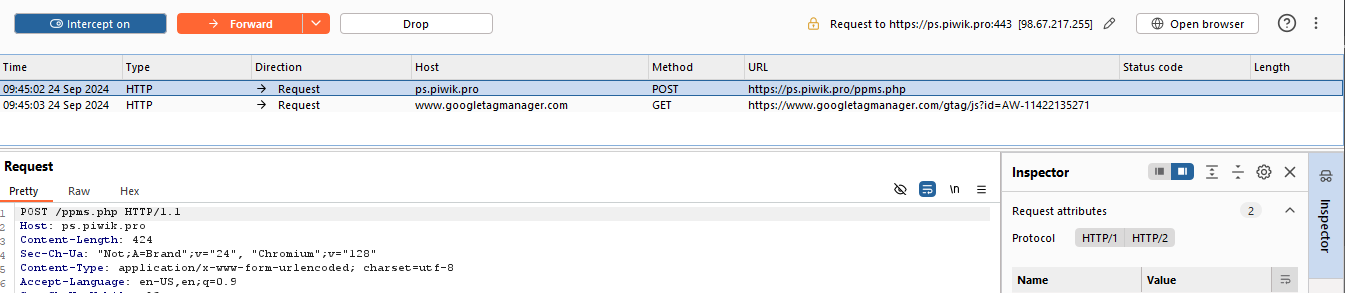
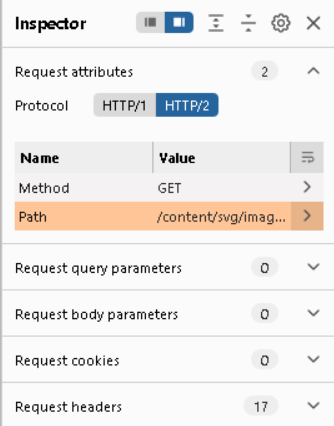
1. **Intercept which is by default off should be turned on.**
2. **Open the browser.**



1. **Search for anything on the browser.**

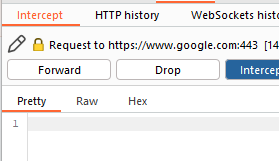
**Here for eg: https://portswigger.net**

* The request is intercepted by burp suite’s proxy browser.
* We can see the request on proxy’s intercept tab.
* The request is held here so that you can study it, and even modify it, before forwarding it to the target server.



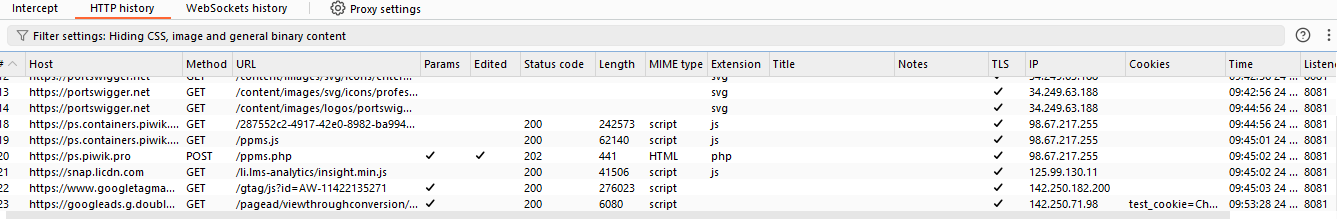


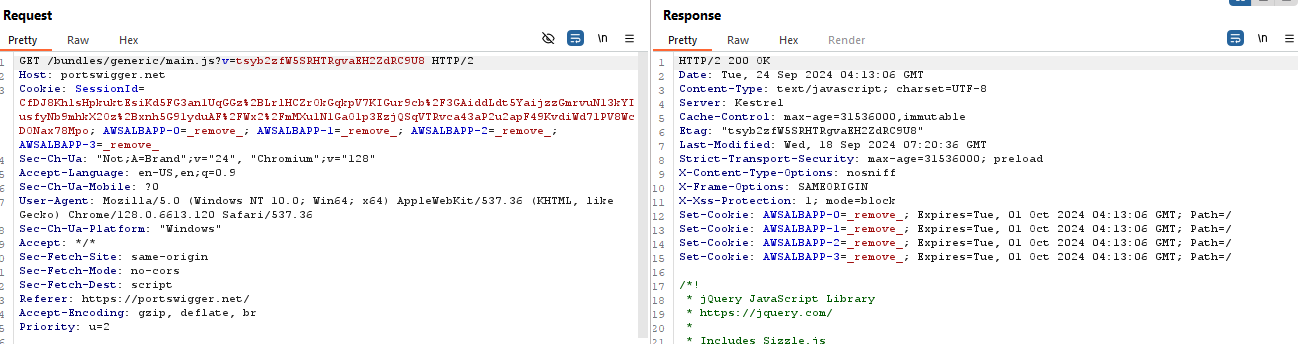
1. **Click the Forward button** to send the intercepted request. Click Forward again to send any subsequent requests that are intercepted, until the page loads in Burp's browser. The Forward button sends all the selected requests.

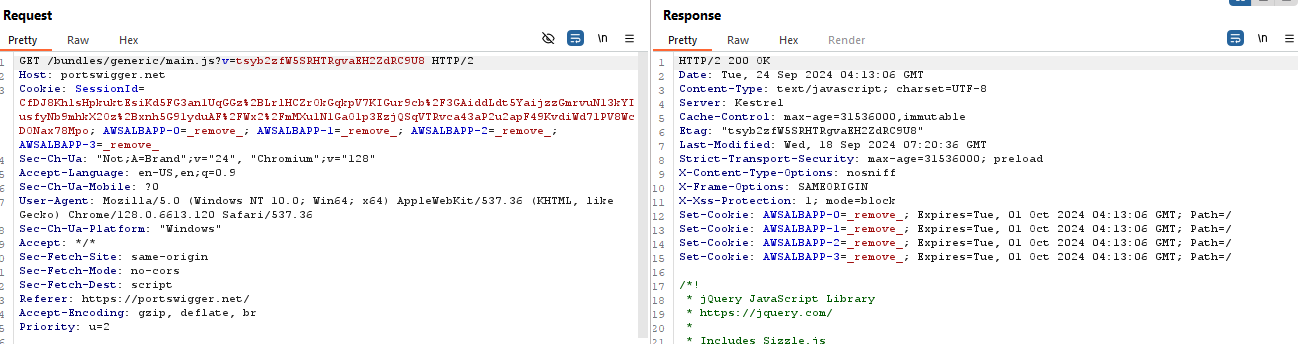
****

1. **We can turn off the interception and check the HTTP history now.**

Clicking on any of the entries in history shows the raw HTTP request, along with the corresponding response from the server.







This lets you explore the website as normal and study the interactions between Burp's browser and the server afterward, which is more convenient in many cases.

**Conclusion:**

The necessity of using Burp Suite became apparent. Various tools provided by Burp Suite were researched and learned. The proxy tool in Burp Suite was utilized, and the repeater tool was examined. The implementations were conducted by installing the Burp Suite Community Edition.