ASSIGNMENT 4

Overview of Assignment

To write about Burp Suite, state its uses, and elaborate on its features. Try the burp suite on testfire.net to identify any vulnerabilities.

Burp Suite

Burp Suite is a widely used web application security testing tool developed by PortSwigger. It is designed for security professionals, ethical hackers, and penetration testers to identify and address security vulnerabilities in web applications. Burp Suite offers a comprehensive set of features and capabilities that make it a versatile tool for various security assessments and testing scenarios.

Burp Suite is primarily used for:

1) Web Application Scanning: It scans web applications to identify vulnerabilities such as SQL injection, cross-site scripting (XSS), and security misconfigurations. It allows security professionals to proactively find and address these issues.
2) Penetration Testing: Burp Suite can be used to perform penetration testing and ethical hacking activities. It helps testers identify and exploit vulnerabilities to assess an application's security posture.
3) Security Research: Security researchers use Burp Suite to analyze and understand web application behavior, especially when exploring new and emerging threats.
4) Web Application Development: Developers often use Burp Suite to test their own web applications during the development process to identify and fix security issues before deployment.

Some of it features are:

1) Proxy: Burp Suite acts as a proxy server, allowing you to intercept and inspect HTTP and HTTPS traffic between your browser and the web application you are testing. This feature is essential for understanding how an application works and for identifying security issues.
2) Scanner: The automated scanner feature in Burp Suite can identify a wide range of vulnerabilities in web applications, including SQL injection, XSS, and more. It provides detailed reports and remediation advice.
3) Repeater: Repeater allows you to manually manipulate and replay individual requests to a web application. This feature is useful for testing and verifying vulnerabilities.
4) Intruder: Intruder is a powerful tool for automated, customizable attacks against web applications. It helps in testing parameter manipulation, fuzzing, and brute-force attacks.
5) Sequencer: The Sequencer tool assesses the randomness and quality of session tokens and other data. It's useful for identifying predictable patterns in tokens that can lead to session fixation or similar vulnerabilities.
6) Decoder: Decoder assists in the decoding and encoding of data in various formats, including base64, URL, and more. This is handy for understanding and modifying data within requests and responses.
7) Extensions: Burp Suite supports extensions and add-ons, allowing users to customize and extend its functionality. The community has created a wide range of extensions to address specific needs.

Testing Burp Suite on testfire.com

1. Proxy Setup: Configure your browser to use Burp Suite as a proxy by setting the proxy settings in the browser to "127.0.0.1" on port 8080 (the default proxy listening port for Burp Suite).
2. Intercept Traffic: Start Burp Suite and ensure that the proxy interception is on. Visit the "testfire.net" website. Burp Suite will capture the requests and responses.
3. Passive Scanning: Let Burp Suite run in passive scanning mode to automatically identify common security issues.
4. Active Scanning: Run an active scan on the website to comprehensively identify vulnerabilities.
5. Analyze Results: Review the scan results and address any vulnerabilities or security issues identified by Burp Suite.

http://testfire.net/index.jsp [content parameter]

Summary

Severity:	**High**
Confidence:	**Firm**
Host:	**http://testfire.net**
Path:	**/index.jsp**

Issue detail

The value of the **content** request parameter is copied into the HTML document as plain text between tags. The payload **jftw8<a b=c>kg4p3** was submitted in the content parameter. This input was echoed unmodified in the application's response.

This behavior demonstrates that it is possible to inject new HTML tags and attributes into the returned document. An attempt was made to identify a full proof-of-concept attack for

injecting arbitrary JavaScript but this was not successful. You should manually examine the application's behavior and attempt to identify any unusual input validation or other obstacles that may be in place.

Request 1

```
GET /index.jsp?content=inside.htmjftw8%3ca%20b%3dc%3ekg4p3 HTTP/1.1
Host: testfire.net
Accept-Encoding: gzip, deflate
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
Accept-Language: en-US;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.5845.97 Safari/537.36
Connection: close
Cache-Control: max-age=0
Cookie: JSESSIONID=B8CFF6745F9E8C6109A3016DAEA32E49
Upgrade-Insecure-Requests: 1
Referer: http://testfire.net/
Sec-CH-UA: ".Not/A)Brand";v="99", "Google Chrome";v="116", "Chromium";v="116"
Sec-CH-UA-Platform: Windows
Sec-CH-UA-Mobile: ?0
Content-Length: 0
```