Assignment 4

- **Hardik Kankane (21BCE10413)**

Overview:

Write about Burp Suite and state its uses and also elaborate on the features of the burp suite. Try the burp suite on testfire.net to identify any vulnerabilities.

What is BURPSUITE?

Burp Suite, created by PortSwigger, is a widely utilized cybersecurity tool catering to the examination of web application security. This comprehensive suite finds extensive use among security experts, penetration testers, and web developers for pinpointing and rectifying vulnerabilities within web applications.

Key Components and Features of Burp Suite:

1. **Proxy Functionality**: Serving as a proxy server, Burp Suite stands intermediary between your browser and the target web application. This positioning enables the interception and examination of HTTP requests and responses, facilitating the streamlined identification of security concerns.
2. **Automated Scanner**: Burp Suite incorporates an automated scanning mechanism capable of traversing a web application autonomously. It adeptly recognizes prevalent security susceptibilities, such as SQL injection and cross-site scripting (XSS), expediting the detection of potential vulnerabilities.
3. **Repeater Tool**: Offering manual control, the Repeater feature empowers users to manipulate and resend individual HTTP requests. It proves invaluable for scrutinizing the effects of diverse input values on a web application and comprehending its ensuing responses.
4. **Intruder Functionality**: Burp Suite's Intruder tool specializes in automating bespoke assaults on web applications. It is customizable to execute an array of brute force or parameter-centered attacks, serving as a potent instrument in unearthing vulnerabilities.
5. **Sequencer Tool**: Tailored for assessing the quality of randomness in tokens and session identifiers generated by web applications, the Sequencer plays a pivotal role in pinpointing vulnerabilities linked to feeble or foreseeable session management.
6. **Decoder Capabilities**: Burp Suite incorporates a variety of encoding and decoding functions, enabling the analysis and manipulation of data across various formats like URL encoding, base64 encoding, and more.
7. **Extensibility via Plugins**: Burp Suite's extensibility feature facilitates the integration of customized plugins. This capability empowers security professionals to introduce new functionalities and automate specific tasks, and an array of community-contributed plugins is readily available.
8. **Target and Site Map Functionality**: These attributes are instrumental in structuring and visualizing information amassed during scanning and testing sessions. They

afford a methodical perspective of the target application and its associated vulnerabilities.

9. **Robust Reporting**: Following security assessments, Burp Suite is proficient in generating comprehensive reports. These reports encapsulate identified vulnerabilities, categorize their severity, and offer actionable recommendations for mitigation.

Diverse in its capabilities, Burp Suite proves invaluable for a multitude of tasks related to assessing the security of web applications and services. Here are some prevalent applications of Burp Suite:

1. **Web Application Scanning**: The automated scanner within Burp Suite adeptly navigates through a web application, pinpointing vulnerabilities and furnishing detailed reports. Its proficiency lies in uncovering critical issues like SQL injection, cross-site scripting (XSS), and an array of other security susceptibilities.
2. **Manual Testing**: Security experts leverage Burp Suite's proxy and interception functionalities to meticulously scrutinize and manipulate HTTP requests and responses. This hands-on approach facilitates the identification of vulnerabilities that automated tools may inadvertently overlook.
3. **Session Management Analysis**: Burp Suite's suite of tools tailored for session management analysis aids in assessing how a web application handles user sessions. This encompasses an evaluation of factors such as the predictability of session tokens and the effectiveness of session timeout mechanisms.
4. **Parameter Manipulation**: The Intruder tool housed within Burp Suite comes into play for automating parameter-based assaults, ranging from brute force tactics to fuzzing exercises. This serves to unveil vulnerabilities that might stem from inadequate input validation or filtering.
5. **Web Services Testing**: It finds application in scrutinizing the security of SOAP and RESTful web services, extending to the examination of XML and JSON payloads.
6. **Authentication Testing**: Burp Suite assumes the role of evaluating the security of login mechanisms, encompassing endeavors like username and password enumeration, as well as brute force attacks.
7. **Client-Side Testing**: Security professionals wield the capacity to dissect and test JavaScript code, cookies, and other client-side elements for security susceptibilities, such as DOM-based XSS.
8. **Custom Plugin Development**: The extensibility of Burp Suite is harnessed by security experts who craft tailored plugins. These serve to automate specific operations or seamlessly integrate with other tools, augmenting the suite's functionality.
9. **Penetration Testing**: Burp Suite frequently features in penetration testing endeavors, standing as a stalwart instrument for detecting and demonstratively illustrating security frailties within web applications and APIs.

It's important to note that Burp Suite should be used only on web applications and services for which you have explicit authorization or legal permission to test. Unauthorized or irresponsible use of such tools can have legal and ethical implications. Example of implementation of Burpsuite on testfire.net

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
```

Response 1

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/html;charset=ISO-8859-1
Content-Length: 6930
Date: Mon, 25 Sep 2023 07:09:44 GMT
Connection: close


<!-- BEGIN HEADER -->
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.
...[SNIP]...
<p>Failed due to The requested resource (/static/inside.htmjftw8<a b=c>kg4p3) is not available</p>
...[SNIP]...
```