

Experiment No. 4



Batch: A2**Roll No.:16010421119****Experiment No.:4****Aim:** Experimenting with BURP

(Any 3-5 VAPT-related tasks to be done on BURP).

Resources needed: Burp Suite, Web Application Target, Browser and Browser Configuration, Vulnerable Web Applications, Payloads and Wordlists, Proxy Configuration

Pre Lab/ Prior Concepts:

Students should have prior knowledge of Web Application Basics, HTTP Protocol and Requests/Responses, Proxy and Interception Concepts, HTTP Authentication Methods, Common Web Application Vulnerabilities, HTTP Session Management, HTML Forms and Input Validation, Security Headers.

Theory:

Burp Suite stands as a cornerstone in the arsenal of security professionals, offering a robust platform for Vulnerability Assessment and Penetration Testing (VAPT) on web applications. This section delves into key aspects of experimenting with Burp Suite, focusing on essential VAPT tasks that harness its capabilities.

1. **Automated Web Application Scanning:** Burp Suite's automated scanning feature empowers security analysts to identify vulnerabilities efficiently. By leveraging its crawler and scanner, security practitioners can initiate comprehensive scans, probing for common vulnerabilities such as SQL injection, Cross-Site Scripting (XSS), and more. The automated scanning process accelerates the identification of potential threats, providing a foundational understanding of the web application's security posture.
2. **Manual Web Application Testing:** The heart of Burp Suite lies in its manual testing capabilities. Analysts can intercept and manipulate HTTP requests and responses using the Proxy module, offering a hands-on approach to identify and exploit vulnerabilities. The "Repeater" tool enables detailed examination and modification of individual requests, allowing security professionals to craft targeted attacks, test input validation, and uncover subtle vulnerabilities that automated scanners might overlook.
3. **Session Handling and Authentication Bypass:** Burp Suite facilitates the examination of session management mechanisms and authentication controls within web applications. By intercepting and manipulating session tokens and cookies, security analysts can explore potential weaknesses. The suite's "Intruder" tool proves invaluable for testing authentication mechanisms, attempting brute force attacks, and validating the robustness of access controls. This task is instrumental in identifying vulnerabilities related to unauthorized access.

5. **Cross-Site Scripting (XSS) Detection:** Burp Suite excels in detecting and validating Cross-Site Scripting vulnerabilities. Analysts can use the suite to inject payloads into user input fields, testing the application's resistance to client-side attacks. The "Scanner" tool automates the identification of XSS vulnerabilities, while the "Repeater" tool allows for manual validation, understanding impact, and proposing mitigation strategies. This task enhances proficiency in identifying and addressing one of the most prevalent web application security risks.

Procedure:**Step 1: Download and Install Burp Suite**

1. Visit the official Burp Suite website (<https://portswigger.net/burp/communitydownload>) and download the platform's appropriate version (Community or Professional).
2. Install Burp Suite following the on-screen instructions.

Step 2: Configure Browser and Proxy Settings

1. Launch Burp Suite.
2. Configure web browser to use Burp Proxy:
3. Set the browser's proxy settings to use Burp Suite's proxy (default port: 127.0.0.1:8080).
4. Install the Burp Suite CA certificate for SSL/TLS interception in the browser.

Step 3: Explore the Target Web Application

1. Identify and select a target web application for testing by ensuring proper authorization.
2. Navigate to the target application through the configured browser, allowing Burp to intercept traffic.

Step 4: Automated Web Application Scanning

1. In Burp Suite, go to the "Target" tab, right-click on the target URL, and select "Add to Scope."
2. Switch to the "Scanner" tab and initiate an automated scan on the selected target.
3. Review the scan results in the "Scan" tab, identifying and categorizing vulnerabilities discovered by Burp's scanner.

Step 5: Manual Web Application Testing

1. In the "Proxy" tab, intercept and manipulate HTTP requests and responses to identify potential vulnerabilities.
2. Use the "Repeater" tool to manually send and modify HTTP requests, observing the application's responses.
3. Explore the "Intruder" tool to perform parameter-based attacks, such as brute force or fuzzing, on selected requests.
4. Document and validate any vulnerabilities discovered through manual testing.

Step 6: Session Handling and Authentication Bypass

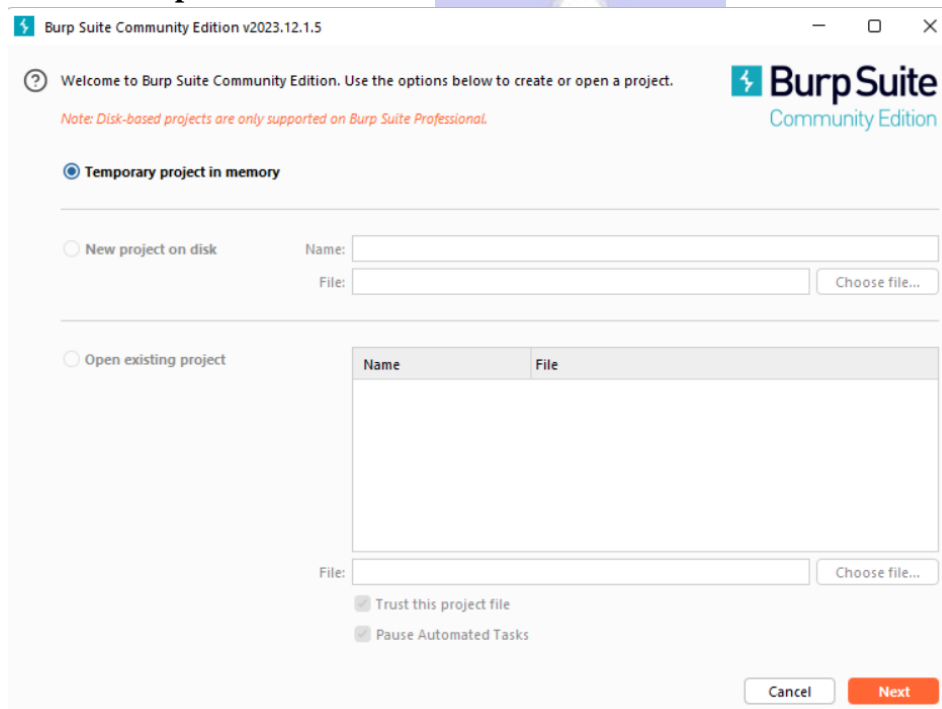
1. Capture and analyze login requests in the Burp Proxy history.
2. Use the "Intruder" tool to perform a password-spraying attack or brute force against the login page.
3. Explore session tokens and cookies using the "Session" and "Cookies" tabs in Burp.
4. Attempt to manipulate session data to bypass authentication or gain unauthorized access.
5. Document findings and propose recommendations for improving authentication controls.

Step 7: Cross-Site Scripting (XSS) Detection

- Use Burp Proxy to intercept and analyze requests and responses while interacting with the web application.
- Identify user input fields and inject simple payloads to test for potential XSS vulnerabilities.
- Utilize Burp's "Scanner" tool to automate the detection of XSS vulnerabilities.
- Confirm and validate XSS vulnerabilities by exploiting them and executing benign payloads.
- Provide recommendations for remediation and secure coding practices.

Step 8: Generate Reports and Document Findings

1. Generate detailed reports in Burp Suite summarizing the identified vulnerabilities, their severity, and recommended mitigations.
2. Document the entire testing process, including steps taken, tools used, and vulnerabilities discovered.
3. Collaborate with relevant stakeholders to address and remediate the identified vulnerabilities.

Output (Code with result Snapshot)**1) Launch Burp Suite.****2) Set the browser's proxy settings to use Burp Suite's proxy (default port: 127.0.0.1:8080).**

Settings

Tools > Proxy

Manage global settings

Proxy listeners

Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to use one of the listeners as its proxy server.

Running	Interface	Invisible	Redirect	Certificate	TLS Protocols	Support HTTP...
<input checked="" type="checkbox"/>	127.0.0.1:8080			Per-host	Default	<input checked="" type="checkbox"/>

Each installation of Burp generates its own CA certificate that Proxy listeners can use when negotiating TLS connections. You can import or export this certificate for use in other tools or another installation of Burp.

Import / export CA certificate Regenerate CA certificate

Request interception rules

Use these settings to control which requests are stalled for viewing and editing in the intercept tab.

☒ Intercept requests based on the following rules:

Enabled	Operator	Match type	Relationship	Condition
<input checked="" type="checkbox"/>	Or	File extension	Does not match	(^gif\$ ^jpg\$ ^png\$ ^css\$ ^js\$...
<input type="checkbox"/>	Or	Request	Contains parameters	
<input type="checkbox"/>	Or	HTTP method	Does not match	(get post)
<input type="checkbox"/>	And	URL	Is in target scope	

☐ Automatically fix missing or superfluous new lines at end of request

☒ Automatically update Content-Length header when the request is edited

- 3) Navigate to the target application(portswigger.net) through the burpsuite browser, allowing Burp to intercept traffic.

0a1b000f046547ea84fbc8b100780078.web-security-academy.net


WebSecurity Academy Excessive trust in client-side controls

LAB Not solved


Back to lab description >>

Home | My account | 0


WE LIKE TO SHOP




Lightweight "133t" Leather Jacket
★★★★★ \$1337.00
[View details](#)



Com-Tool
★★★★☆ \$20.25
[View details](#)



Six Pack Beer Belt
★☆☆☆☆ \$61.65
[View details](#)



Sprout More Brain Power
★★★★★ \$71.44
[View details](#)

Settings

Project > Scope

Manage global settings

Target scope

Use these settings to define exactly what hosts and URLs constitute the target for your current work. This configuration affects the behavior of tools throughout the suite.

☐ Use advanced scope control

Include in scope

Enabled	Prefix	Include subdomains
<input checked="" type="checkbox"/>	http://testfire.net/	<input type="checkbox"/>

Exclude from scope

Enabled	Prefix	Include subdomains
---------	--------	--------------------

Out-of-scope request handling

Use these settings to define how Burp handles any out-of-scope requests.

☐ Drop all out-of-scope requests

☒ Use suite scope [defined above]

☐ Use custom scope

4) Manual Web Application Testing:

- In the "Proxy" tab, intercept and manipulate HTTP requests and responses to identify potential vulnerabilities.

Web Security Academy

Excessive trust in client-side controls

LAB Not solved

Store credit: \$100.00

Lightweight "133t" Leather Jacket

\$1337.00

Home | My account | 1

Request to https://0a1b000046547ea84bcb00780078.web-security-academy.net:443 [34.246.129.62]

Forward Drop Intercept Action Open bro... Add notes

HTTP/2

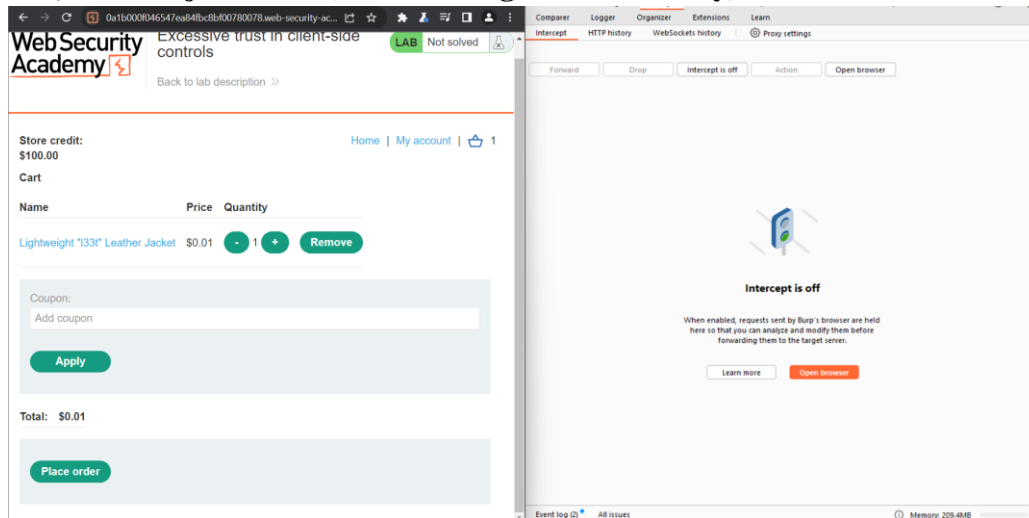
Raw

```

1 GET /academy/lab/index HTTP/2
2 Host: 0a1b000046547ea84bcb00780078.web-security-academy.net
3 Connection: Upgrade
4 Pragma: no-cache
5 Cache-Control: no-cache
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.160 Safari/537.36
7 Upgrade: websocket
8 Origin: https://0a1b000046547ea84bcb00780078.web-security-academy.net
9 Sec-WebSocket-Protocol: 13
10 Accept-Encoding: gzip, deflate, br
11 Accept-Language: en-US,en;q=0.9
12 Cookie: session=YTCFm1t12Kw3u088ag8f0xw3P3C48
13 Sec-WebSocket-Key: 3p6v0T860308uP8u7C3b8u=
14
15

```

- The price of 1 leather jacket is 1337\$ but after changing the parameters in burpsuite ,now it is just 0.01\$.(Business Logic Vulnerability)



- Use the "Repeater" tool to manually send and modify HTTP requests, observing the application's responses.

i) Using this request to manipulate and sending it to repeater.

Store credit:
\$100.00

Cart

Name	Price	Quantity
Lightweight "133t" Leather Jacket	\$1337.00	2
Lightbulb Moments	\$40.29	1

Coupon:

Add coupon

Apply

Total: \$2714.29

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cookies	Time	Listener port
501	https://0a1b00f046547ea...	GET	/academyLabHeader		✓	400	130	text				✓	34.246.129.62		16:12:08 18...	8080
502	https://0a1b00f046547ea...	POST	/cart		✓	302	100					✓	34.246.129.62		16:12:20 18...	8080
503	https://0a1b00f046547ea...	GET	/product?productId=1		✓	200	5283	HTML		Excessive trust in cli...		✓	34.246.129.62		16:12:23 18...	8080
504	https://0a1b00f046547ea...	GET	/academyLabHeader		✓	400	130	text				✓	34.246.129.62		16:12:24 18...	8080
505	https://0a1b00f046547ea...	GET	/cart		✓	200	8467	HTML		Excessive trust in cli...		✓	34.246.129.62		16:12:28 18...	8080
507	https://0a1b00f046547ea...	GET	/academyLabHeader		✓	400	130	text				✓	34.246.129.62		16:12:31 18...	8080
508	https://0a1b00f046547ea...	GET	/cart		✓	200	8467	HTML		Excessive trust in cli...		✓	34.246.129.62		16:13:16 18...	8080
509	https://play.google.com	POST	/log?format=json&hasfast=true...		✓	200	980	JSON				✓	172.217.166.46	NID=511=kyGp...	16:14:33 18...	8080
510	https://play.google.com	POST	/log?format=json&hasfast=true...		✓	200	980	JSON				✓	172.217.166.46	NID=511=K6eH...	16:14:33 18...	8080
511	https://0a1b00f046547ea...	POST	/cart		✓	302	85					✓	34.246.129.62		16:17:19 18...	8080
512	https://0a1b00f046547ea...	POST	/cart		✓	302	85					✓	34.246.129.62		16:17:22 18...	8080
513	https://0a1b00f046547ea...	GET	/cart		✓	200	8467	HTML		Excessive trust in cli...		✓	34.246.129.62		16:17:38 18...	8080
514	https://0a1b00f046547ea...	GET	/academyLabHeader		✓	400	130	text				✓	34.246.129.62		16:17:46 18...	8080

ii) Manipulating the product id from 1 to 3 and response is HTTP/2 302 found.

Send Cancel Follow redirection

Request

Raw Hex

```

1 POST /cart HTTP/2
2 Host: 0alb000f046547ea84fbc8bf00780078.web-security-academy.net
3 Cookie: session=yY2Prs1TlXJe0hXHM8aqME0z5sjP3CoN
4 Content-Length: 34
5 Cache-Control: max-age=0
6 Sec-Ch-Ua: "Chromium";v="121", "Not A(Brand";v="99"
7 Sec-Ch-Ua-Mobile: ?0
8 Sec-Ch-Ua-Platform: "Windows"
9 Upgrade-Insecure-Requests: 1
10 Origin: https://0alb000f046547ea84fbc8bf00780078.web-security-academy.net
11 Content-Type: application/x-www-form-urlencoded
12 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
    like Gecko) Chrome/121.0.6167.160 Safari/537.36
13 Accept:
    text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/a
    png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-User: ?1
17 Sec-Fetch-Dest: document
18 Referer: https://0alb000f046547ea84fbc8bf00780078.web-security-academy.net/cart
19 Accept-Encoding: gzip, deflate, br
20 Accept-Language: en-US,en;q=0.9
21 Priority: u=0, i
22
23 productId=1&quantity=-1&redir=CART

```

Response

Pretty Raw Hex Render

```

1 HTTP/2 302 Found
2 Location: /cart
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 0
5
6

```

Response

Pretty Raw Hex Render

```

1 HTTP/2 302 Found
2 Location: /cart
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 0
5
6

```

Inspector

Request attributes 2

Request query parameters 0


It's empty in here

Add


Request body parameters 3

Name	Value
productId	3
quantity	-1
redir	CART

- Explore the "Intruder" tool to perform parameter-based attacks, such as brute force or fuzzing, on selected requests.
 - ❖ This is the Login page for which we have to perform parameter based attack

Web Security Academy 

Username enumeration via different responses

LAB Not solved 

[Back to lab description >>](#)

[Home](#) | [My account](#)

Login

Invalid username

Username

Password

[Log in](#)

- ❖ This is raw URL information from HTTP history consisting of username and password.

```
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-User: ?1
17 Sec-Fetch-Dest: document
18 Referer: https://0a0d006503abbbd980ce5de800490053.web-security-academy.net/login
19 Accept-Encoding: gzip, deflate, br
20 Accept-Language: en-US,en;q=0.9
21 Priority: u=0, i
22
23 username=hello&password=world
```

- ❖ Setting the payload with various different usernames after sending the URL to intruders.

Dashboard Target **Proxy** Intruder Repeater Collaborator

1 x 2 x +

Positions **Payloads** Resource pool Settings

Payload sets

You can define one or more payload sets. The number of payload sets depends on the

Payload set: 1 Payload count: 101

Payload type: Simple list Request count: 0

Payload settings [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Paste Load ... Remove Clear Deduplicate Add Add from list ... [Pro version only]

carlos
root
admin
test
guest
info
adm
mysql
user

Enter a new item

❖ Intruder Attack of the raw URL.

4. Intruder attack of https://0a0d006503abbbd980ce5de800490053.web-security-academy.net Attack

Results Positions Payloads Resource pool Settings

Filter: Showing all items

Request	Payload	Status code	Error	Timeout	Length	Comment
0			✓			
1	carlos		✓			
2	root		✓			
3	admin		✓			
4	test		✓			
5	guest		✓			
6	info		✓			
7	adm		✓			
8	mysql		✓			
9	user		✓			
10	administrator		✓			
11	oracle		✓			

Request

Pretty Raw Hex

```

1 POST /login HTTP/2
2 Host: 0a0d006503abbbd980ce5de800490053.web-security-academy.net
3 Cookie: session=Be8bNfXT4sLruEfrd4frwiS9Ur1R01AG
4 Content-Length: 29
5 Cache-Control: max-age=0
6 Sec-Ch-Ua: "Chromium";v="121", "Not A(Brand";v="99"
7 Sec-Ch-Ua-Mobile: 70
8 Sec-Ch-Ua-Platform: "Windows"
9 Upgrade-Insecure-Requests: 1
10 Origin: https://0a0d006503abbbd980ce5de800490053.web-security-academy.net
11 Content-Type: application/x-www-form-urlencoded
12 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.160 Safari/537.36
13 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
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-User: 71
17 Sec-Fetch-Dest: document
18 Referer: https://0a0d006503abbbd980ce5de800490053.web-security-academy.net/login
19 Accept-Encoding: gzip, deflate, br
20 Accept-Language: en-US,en;q=0.9
21 Priority: u=0, i
22 Connection: keep-alive
23
24 username=hello&password=world

```

4) Cross-Site Scripting (XSS) Detection.

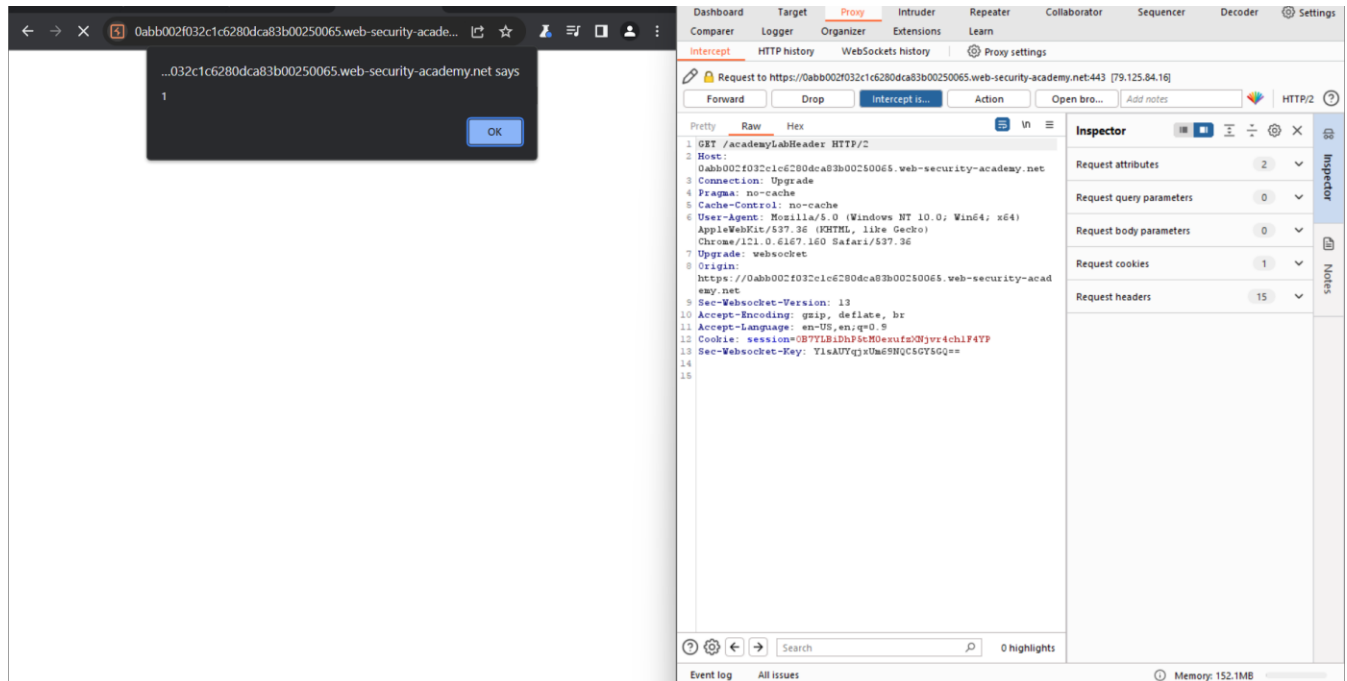
- Use Burp Proxy to intercept and analyze requests and responses while interacting with the web application.(vulnerability in search functionality)

❖ This is a blog website where we have to inject `<script>alert(1)</script>` in search functionality and it will show pop-up box on top.

The screenshot shows a web browser displaying the 'Web Security Academy' lab interface. The lab title is 'Reflected XSS into HTML context with nothing encoded'. The search bar contains the payload `<script>alert(1)</script>`. To the right, the Burp Suite interface is visible, showing the intercepted HTTP request in the 'HTTP history' panel and the 'Inspector' panel displaying the request details, including headers and body.

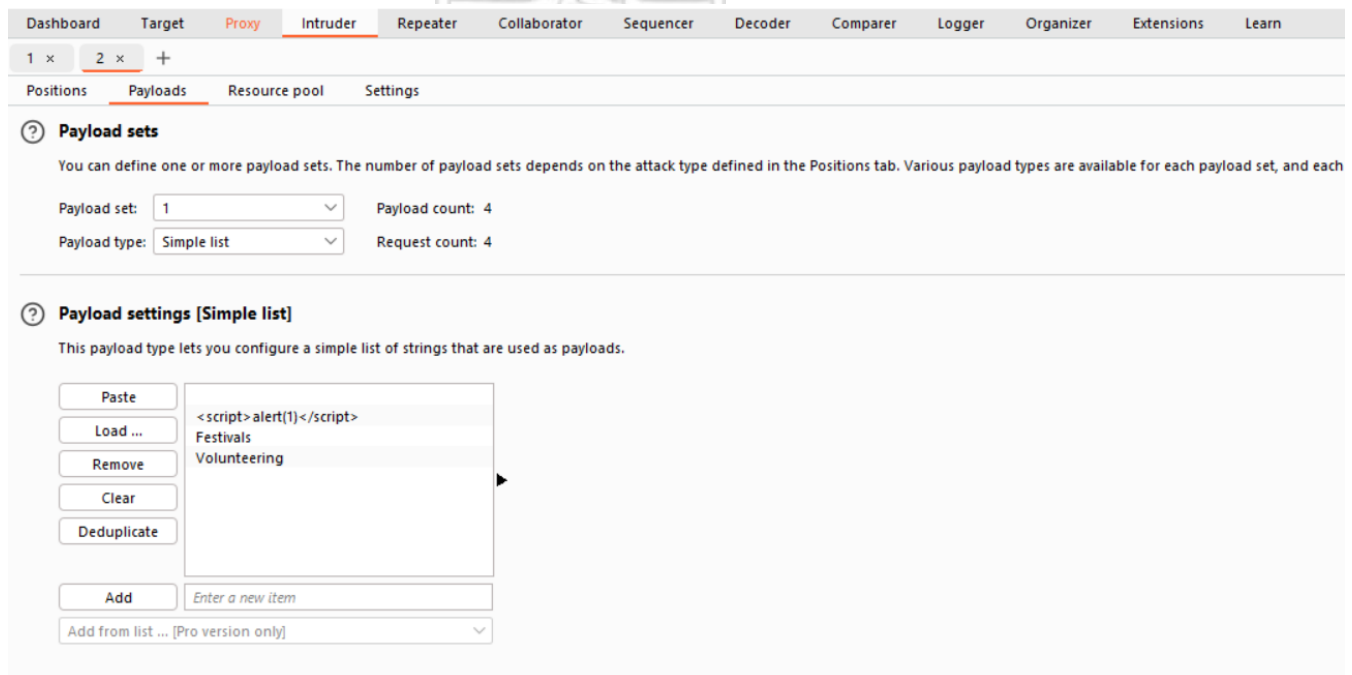
❖ Sending the raw URL to intruder.

The screenshot shows the Burp Suite interface. The top panel displays a list of requests, including the one with the payload `<script>alert(1)</script>`. The bottom panel shows a detailed view of the request and response. The request is a GET request to the search endpoint, and the response is an HTML page showing the search results for the injected payload. The response body contains the text '0 search results for '<script>alert(1)</script>''.



- Identify user input fields and inject simple payloads to test for potential XSS vulnerabilities.

❖ Setting the payload with different results in search functionality.



- ❖ Initilize the attack.
- ❖ In 2nd request you can see in response that the search result is '0' but still shows pop-up exploiting the vulnerability.

Filter: Showing all items

Requ...	Payload	Status code	Error	Timeout	Length	Comment
0		200	<input type="checkbox"/>	<input type="checkbox"/>	6385	
1		200	<input type="checkbox"/>	<input type="checkbox"/>	9136	
2	<script>alert(1)</script>	200	<input type="checkbox"/>	<input type="checkbox"/>	6385	
3	Festivals	200	<input type="checkbox"/>	<input type="checkbox"/>	6925	
4	Volunteering	200	<input type="checkbox"/>	<input type="checkbox"/>	6912	

Request Response

Pretty Raw Hex Render

```

77 </a>
78 <p>
79 |
80 </p>
81 </section>
82 </header>
83 <header class="notification-header">
84 </header>
85 <section class=blog-header>
86 <h1>
87 0 search results for '<script>
  alert(1)
</script>'
88 </h1>
89 <hr>
90 </section>
91 <section class=search>
92 <form action=/ method=GET>
93 <input type=text placeholder='Search the blog...' name=search>

```

search

❖ In 3rd request you can see in response that the search result is '1' for Festivals in blog website.

Results Positions Payloads Resource pool Settings

Filter: Showing all items

Requ...	Payload	Status code	Error	Timeout	Length	Comment
0		200	<input type="checkbox"/>	<input type="checkbox"/>	6385	
1		200	<input type="checkbox"/>	<input type="checkbox"/>	9136	
2	<script>alert(1)</script>	200	<input type="checkbox"/>	<input type="checkbox"/>	6385	
3	Festivals	200	<input type="checkbox"/>	<input type="checkbox"/>	6925	
4	Volunteering	200	<input type="checkbox"/>	<input type="checkbox"/>	6912	

Request Response

Pretty Raw Hex Render

```

77 </a>
78 <p>
79 |
80 </p>
81 </section>
82 </header>
83 <header class="notification-header">
84 </header>
85 <section class=blog-header>
86 <h1>
87 1 search results for 'Festivals'
88 </h1>
89 <hr>
90 </section>
91 <section class=search>
92 <form action=/ method=GET>
93 <input type=text placeholder='Search the blog...' name=search>
94 <button type=submit class=button>
  Search
95 </button>

```

search

Post Lab Questions: -

- 1. How does Burp Suite fit into the broader VAPT workflow? Discuss its role in conjunction with other security tools and methodologies. Consider how the information obtained from Burp Suite can inform decision-making in the context of enhancing web application security.**

Ans: Burp Suite's Role in the VAPT Workflow

Burp Suite is a critical tool within the Vulnerability Assessment and Penetration Testing (VAPT) workflow, particularly for web application security assessments. Here's how it fits in:

- 1. Vulnerability Scanning:**

- Identify vulnerabilities: Burp Scanner automates vulnerability detection in web applications, identifying common issues like SQL injection, XSS, and broken authentication.
- Complement vulnerability scanners: While powerful, standalone scanners sometimes miss vulnerabilities. Burp Suite allows manual testing and deeper analysis to complement them.
- Targeted testing: Information from Burp's sitemap and interception tools can guide focused vulnerability scans on specific areas of the application.

- 2. Manual Penetration Testing:**

- Interception and manipulation: Burp Proxy intercepts and modifies HTTP traffic, enabling testers to manipulate requests and responses, test for logic flaws, and bypass security controls.
- Fuzzing and brute-forcing: Burp Intruder automates fuzzing and brute-forcing attacks on various application parameters, aiding in discovering hidden vulnerabilities.
- Sequence testing: Burp Sequencer helps analyze the randomness of session tokens and other security-sensitive values, uncovering potential weaknesses.

- 3. Integration with other tools:**

- Web vulnerability scanners: Burp integrates with scanners like Nessus and Acunetix for consolidated reporting and streamlined workflows.
- Security frameworks: Tools like Metasploit and Kali Linux can leverage Burp for advanced exploitation and post-exploitation activities.
- Web development tools: Integration with developer tools like browser extensions allows for seamless debugging and vulnerability verification.

- 4. Decision-making for enhanced security:**

- **Prioritization:** Burp helps prioritize vulnerabilities based on severity, exploitability, and potential impact, guiding remediation efforts.
- **Exploitation and impact testing:** By manually exploiting vulnerabilities through Burp, testers can assess their practical impact and guide mitigation strategies.
- **Proof-of-concept creation:** Burp enables testers to create proof-of-concept attacks, demonstrating the vulnerability's existence and urgency to stakeholders.

Limitations of Burp Suite:

- **Not a silver bullet:** Burp requires skilled users and can be complex for beginners.
- **Focus on web applications:** It's not designed for other attack vectors like network or system vulnerabilities.
- **False positives:** Scanner results need manual verification to avoid wasting time on non-existent vulnerabilities.

2. **What methods did you employ in the XSS detection task to identify and validate potential vulnerabilities? How did the Scanner tool and manual validation with the Repeater tool contribute to the overall XSS testing process? Discuss the impact and potential risks associated with XSS vulnerabilities.**

Ans:

XSS Detection Methods:

1. Static Analysis:

- **Source Code Review:** Analyzing the application's source code for instances where user input is not properly sanitized before being outputted.

2. Dynamic Analysis:

- **Fuzzing and Payload Injection:** Sending various payloads through input fields to identify potential points of injection where user-controlled data may be reflected or stored without proper sanitization.
- **Browser-based Testing:** Using tools like Burp Suite's Scanner, which automatically sends payloads and analyzes responses for indications of XSS vulnerabilities.

Burp Suite Scanner and Repeater in XSS Testing:

1. Scanner Tool:

- **Automated Scanning:** The Scanner tool in Burp Suite automates the process of sending malicious payloads to various input points in the application.

- **Identification of Potential XSS:** It identifies potential XSS vulnerabilities by analyzing how the application responds to injected payloads.

2. Repeater Tool:

- **Manual Validation:** The Repeater tool allows security professionals to manually validate and explore potential XSS vulnerabilities identified by the Scanner.
- **Fine-tuning Payloads:** Security analysts can use Repeater to fine-tune payloads, observe how the application reacts, and confirm the existence of XSS vulnerabilities.
- **Contextual Understanding:** Repeater provides a more interactive and contextual understanding of how user-input is processed and how potential vulnerabilities can be exploited.

Impact and Risks Associated with XSS Vulnerabilities:

- 1. Data Theft:** Attackers can steal sensitive user data, such as login credentials or personal information, by injecting malicious scripts that capture and send data to a remote server.
- 2. Session Hijacking:** XSS can be used to hijack user sessions, allowing attackers to impersonate legitimate users and perform unauthorized actions.
- 3. Malicious Redirection:** Attackers can redirect users to malicious websites, potentially leading to phishing attacks or the delivery of malware.
- 4. Defacement:** XSS can be used to deface websites by injecting malicious scripts that modify the content displayed to users.
- 5. Cookie Theft:** If session cookies are not properly secured, attackers can use XSS to steal these cookies and gain unauthorized access to user accounts.
- 6. Client-Side Attacks:** XSS attacks occur on the client side, impacting users directly and making them susceptible to various client-side attacks.

In summary, Burp Suite's Scanner and Repeater tools contribute to the overall XSS testing process by automating the identification of potential vulnerabilities and providing a platform for manual validation and fine-tuning. XSS vulnerabilities pose significant risks, including data theft, session hijacking, and client-side attacks, highlighting the importance of thorough testing and mitigation measures. It's crucial for organizations to address XSS vulnerabilities promptly to ensure the security of their web applications and user data.

Outcomes: CO2: Comprehend purpose of Anonymity and Foot printing.

Conclusion: (Conclusion to be based on the objectives and outcomes achieved)

In this experiment learnt about various vapt tasks using burpsuite.

Signature of faculty in charge with date

References:

1. <https://portswigger.net/burp/documentation>
2. <https://tryhackme.com/room/burpsuitebasics>
3. <https://www.softwaretestinghelp.com/how-to-use-burp-suite/>

