VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

INFORMATION SECURITY MANAGEMENT LAB

EXPERIMENT-4

Functionalities of Burp Suite

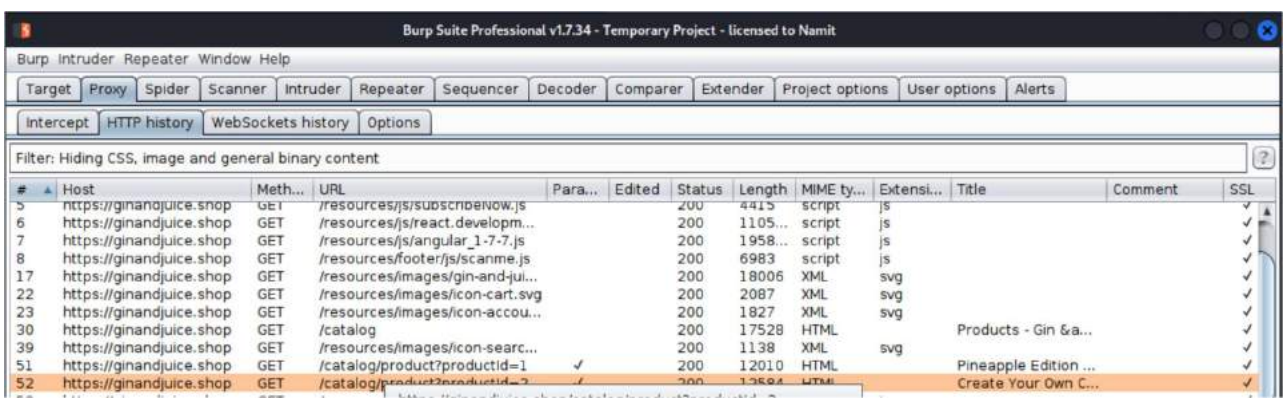
GROUP NO. :	**11**
TEAM MEMBER 1 :	**Namit Mehrotra**
REG. NO. :	**21BCE0763**
TEAM MEMBER 2 :	**Purva Sharma**
REG.NO :	**21BCE0169**
SUBJECT CODE :	**BCSE354E**
SUBJECT TITLE :	**Information Security Management**
LAB SLOT :	**L29+L30**
SEMESTER :	**Winter Semester 2023-2024**
GUIDED BY :	**NIHA K**

After documenting each feature of the tool and explaining its functionality in detail under experiment 3, provide a minimum of 6 to 10 scenarios (use case) where those functionality's will be used to solve the problems in Information Security Management along with detailed steps for each scenario.

1. Identifying Cross-Site Scripting (XSS) Vulnerabilities:

- **Scenario Question:** How would you utilize Burp Suite to identify and mitigate potential XSS vulnerabilities in the web application https://ginandjuice.shop/?
- **Scenario:** We suspect the web application https://ginandjuice.shop/ might be vulnerable to XSS attacks.
- **Steps**:
  1. **Proxy Setup**: Launch Burp Suite and configure your browser to use it as a proxy.
  2. **Interception**: Navigate to the target website and interact with different input fields.

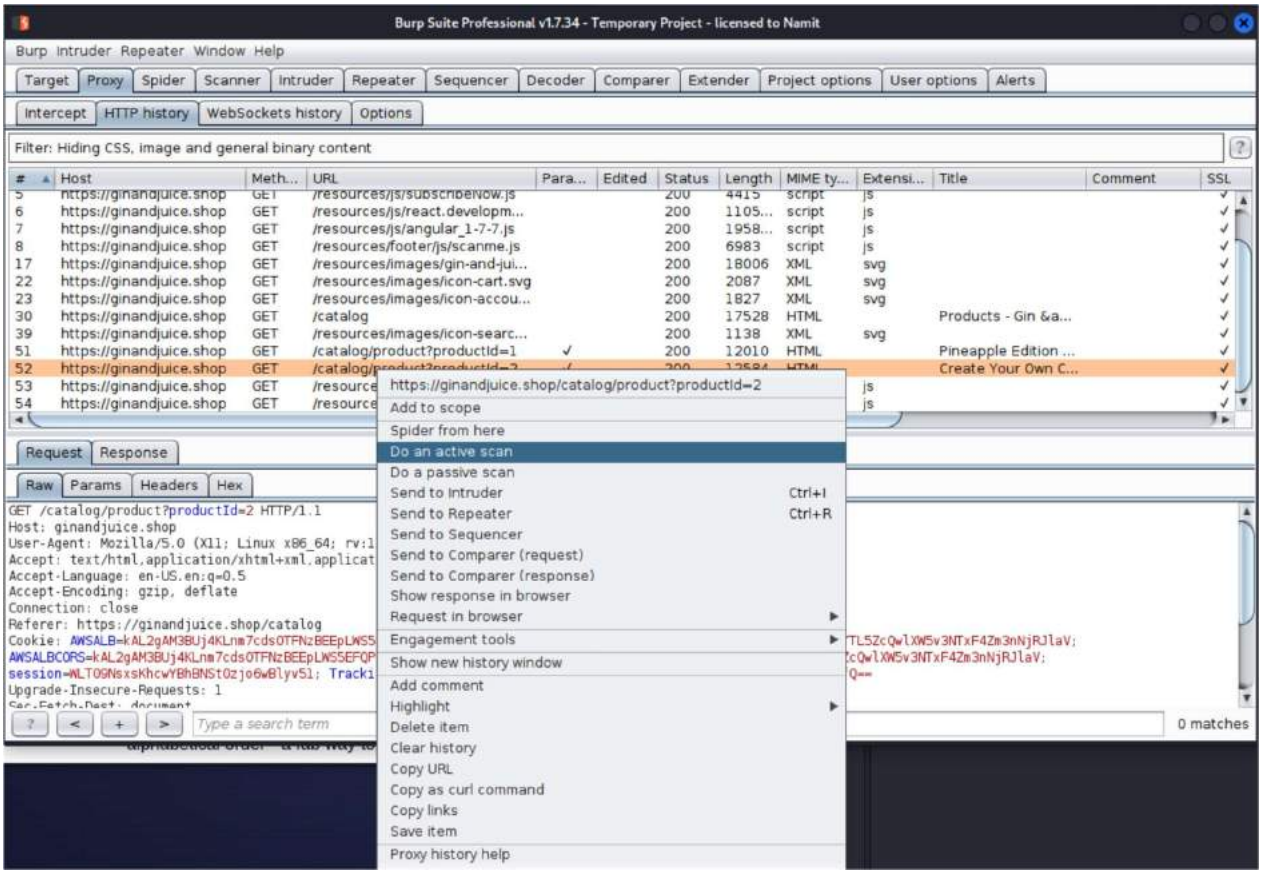
3. **Proxy Tab**: In Burp Suite, navigate to the Proxy tab and observe the requests/responses.

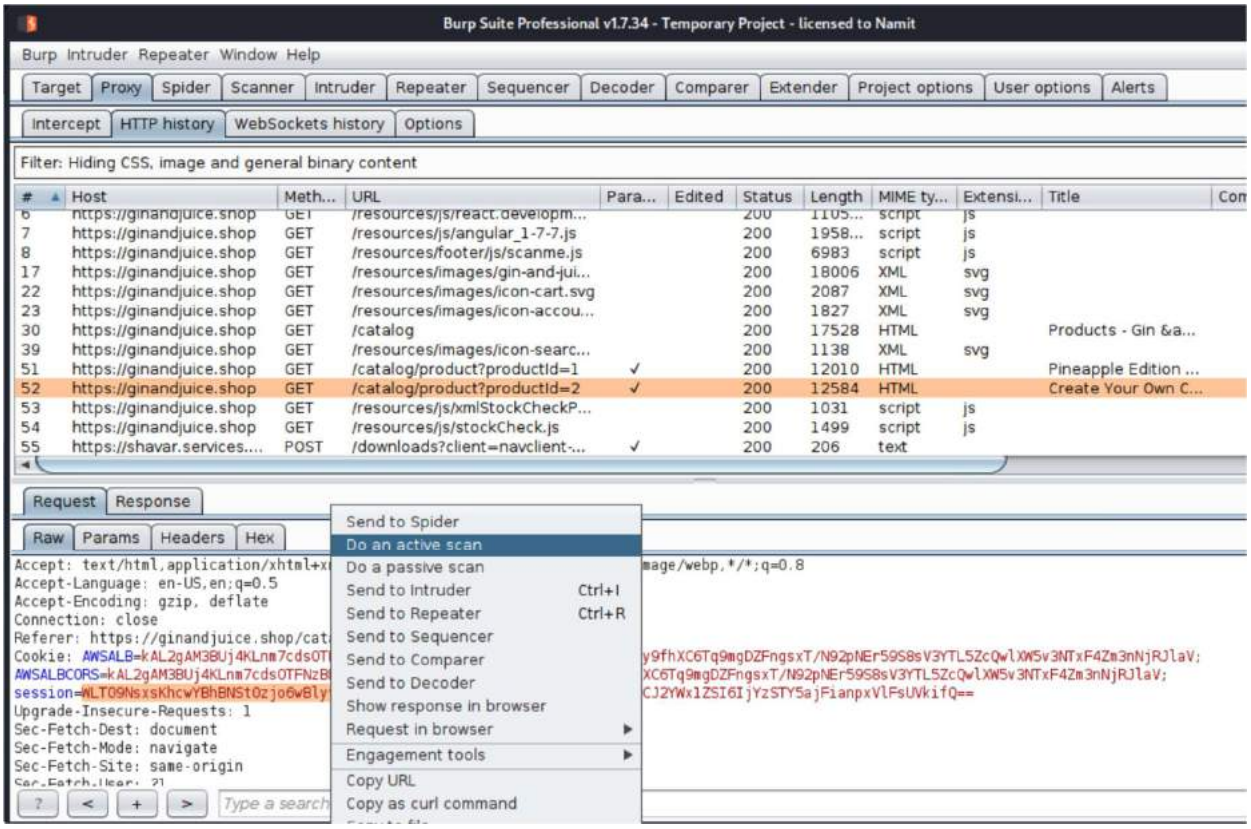


4. **Detection**: Look for suspicious input validation or encoding practices in the responses.

5. **Scanner**: Use Burp's Scanner to automatically scan for XSS vulnerabilities.

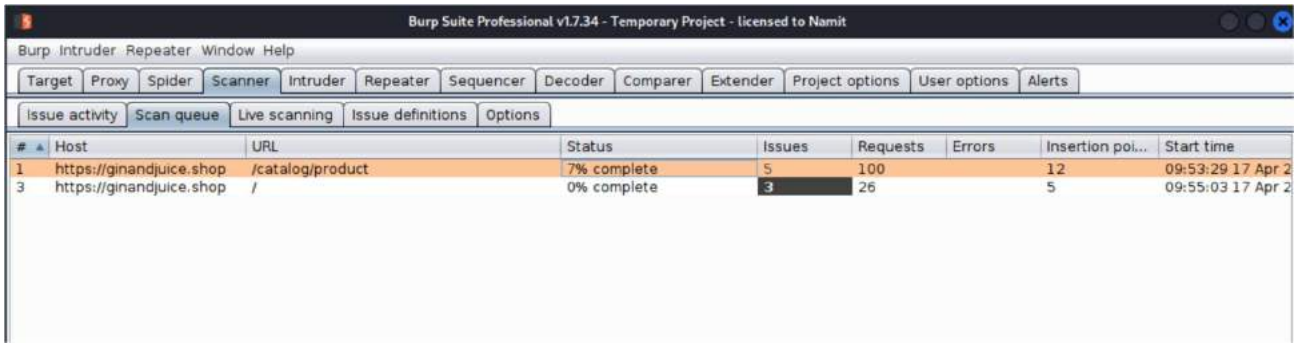
   - Configure the scanner to target input fields and parameters where XSS vulnerabilities are suspected.

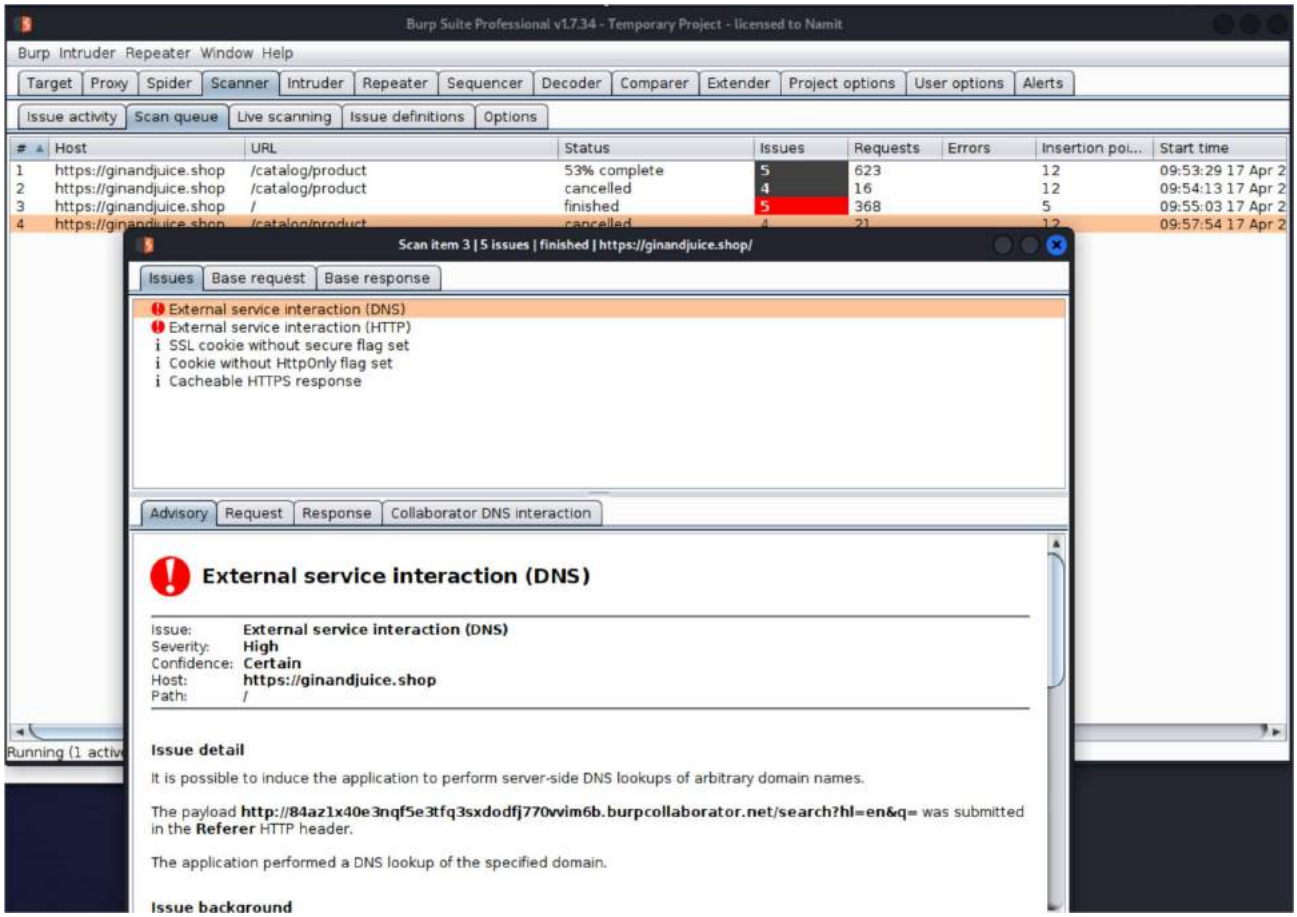




6. **Analysis & Reporting**: Analyze the findings reported by Burp's Scanner in the Scanner tab.

- Review any discovered XSS vulnerabilities and their severity.
- Generate a detailed report highlighting the identified vulnerabilities and recommended mitigation measures.
- Report any discovered vulnerabilities to the development team for remediation.





2. Authentication and Session Management Testing (Maintaining an authenticated session using Burp Suite)

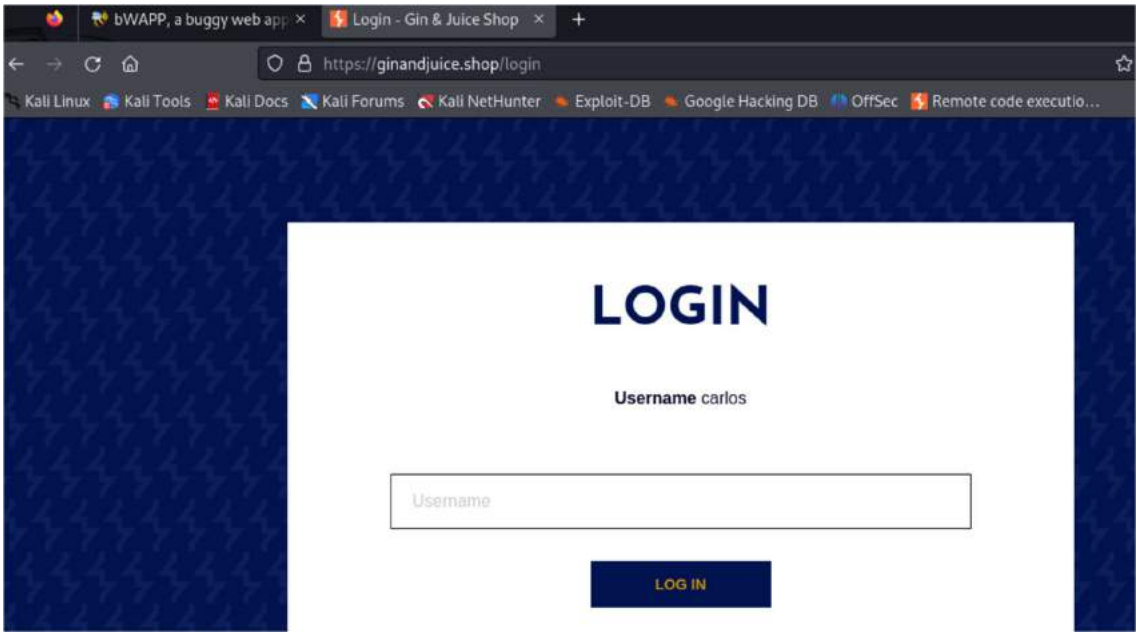
Scenario question: You need to evaluate the effectiveness of authentication and session management mechanisms on the website https://ginandjuice.shop/ to protect user accounts and sensitive data.

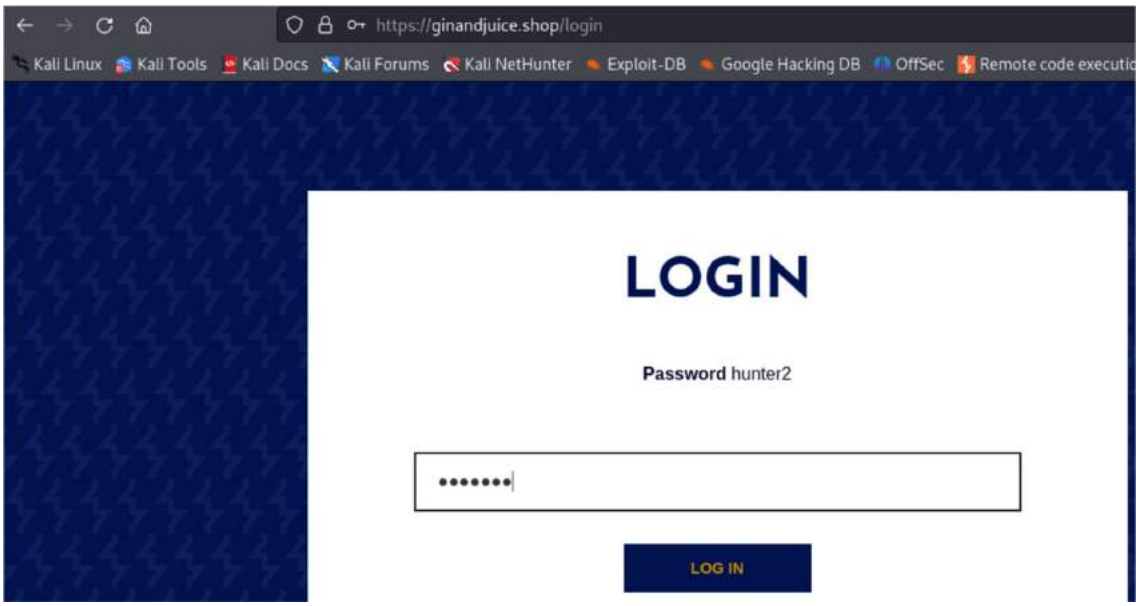
Scenario: We evaluate the effectiveness of authentication and session management mechanisms on the website https://ginandjuice.shop/ to protect user accounts and sensitive data.

Functionality: Burp Suite's "Repeater" and "Session Handling" tools enable manual testing of authentication and session management functionalities.

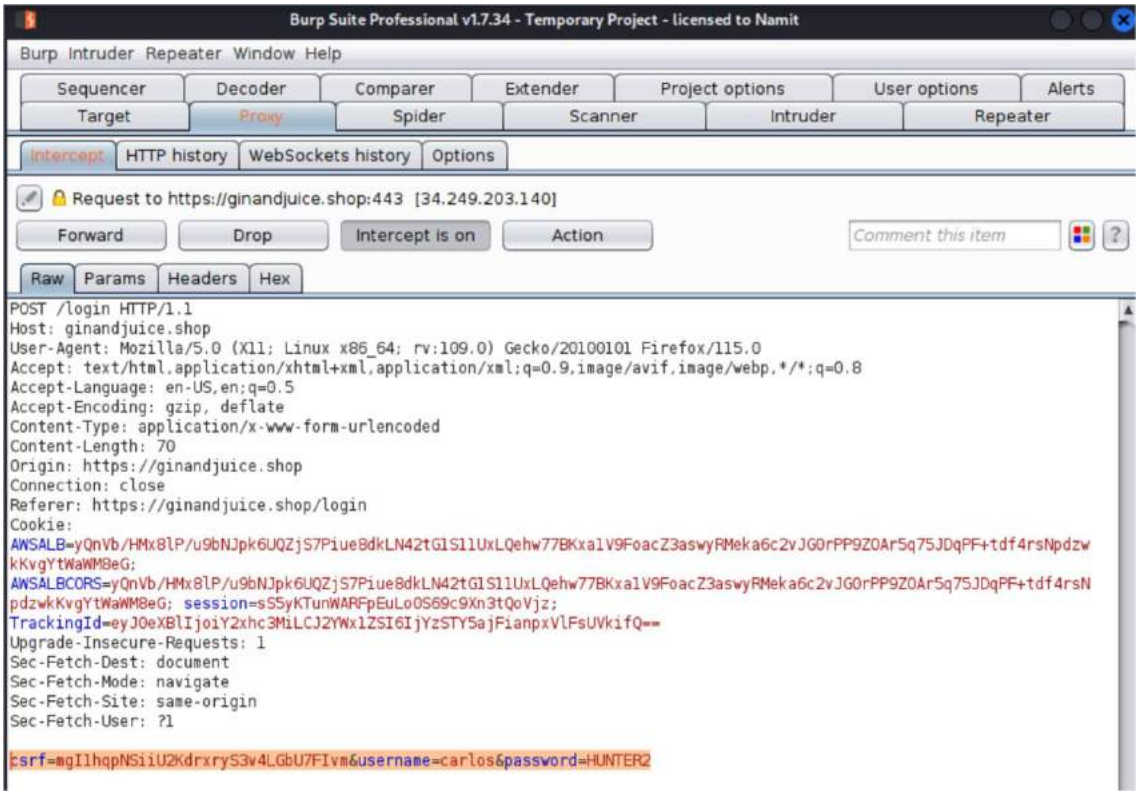
Steps:

1. **Capture Login Requests:**
   - Launch Burp Suite and configure it as a proxy.
   - Navigate to https://ginandjuice.shop/ and initiate the login process.

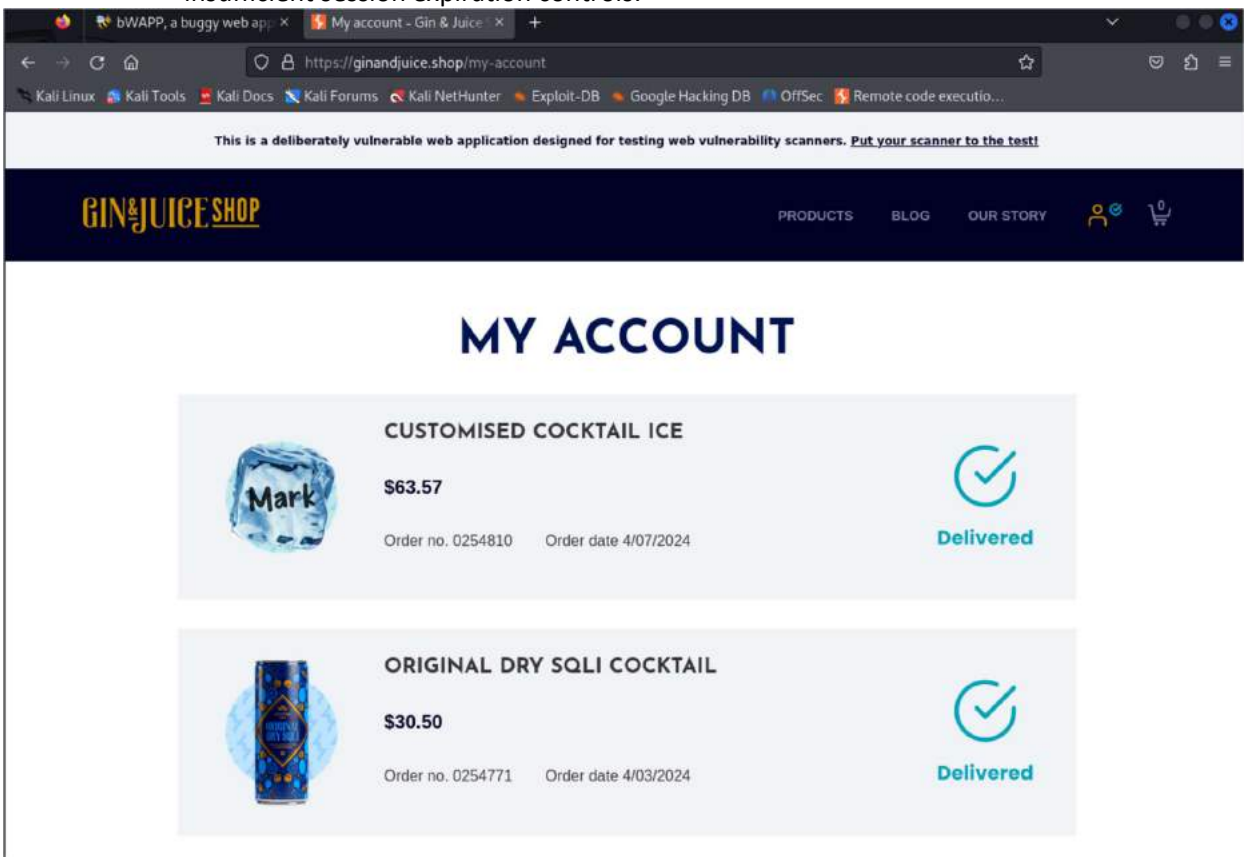




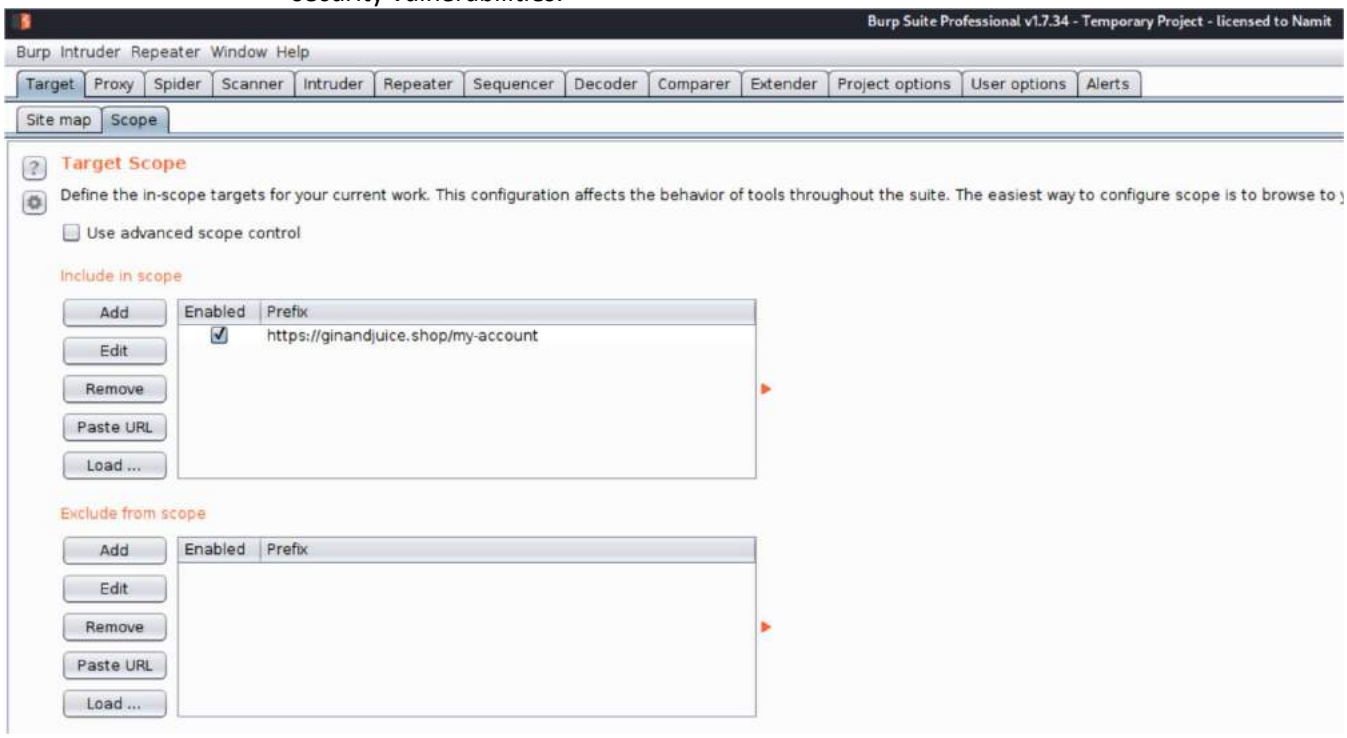
- Use Burp Suite's "Proxy" tool to capture the login requests and subsequent authenticated sessions.

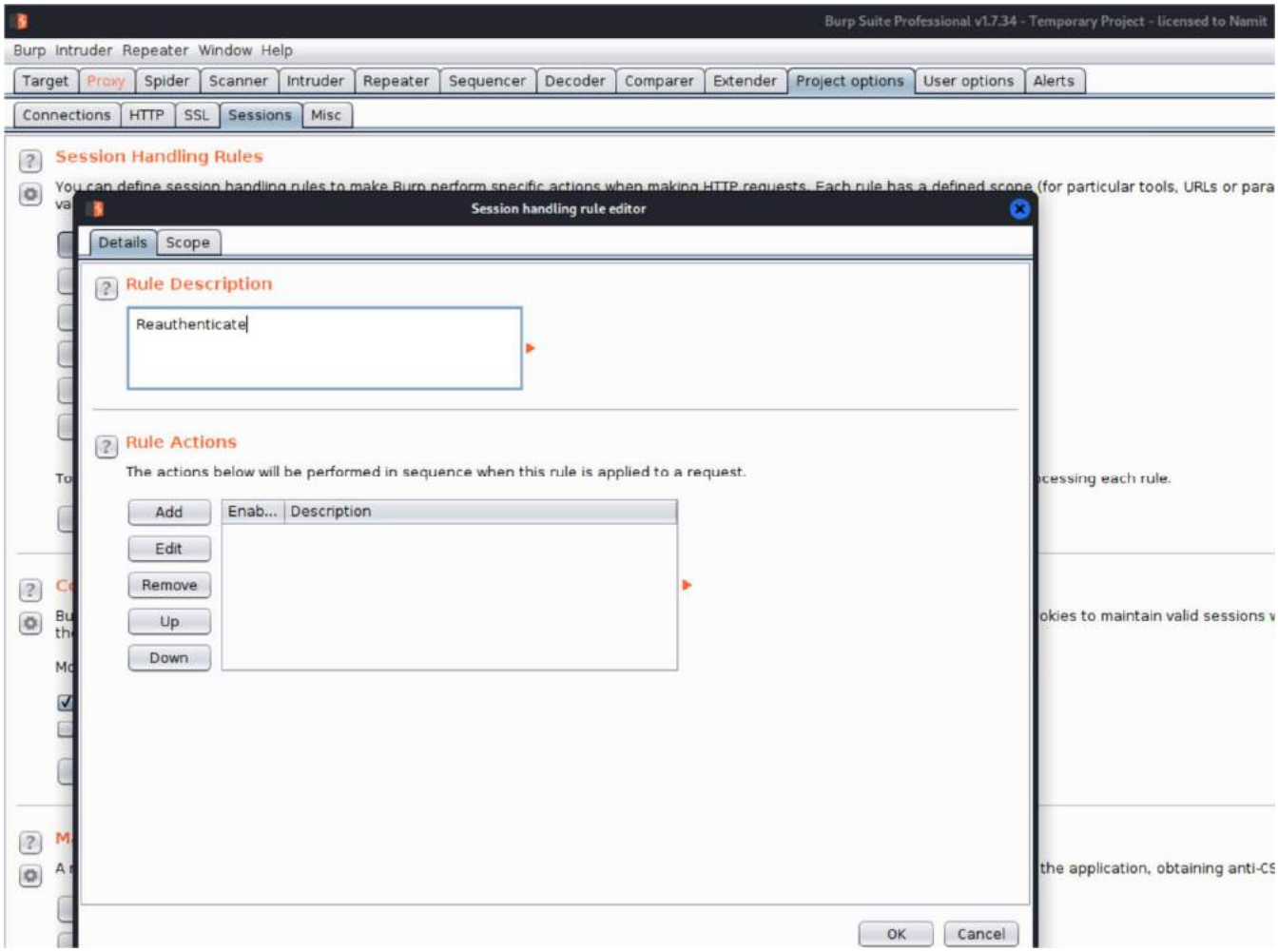


2. **Analyze Authentication Process:**
   - Examine the authentication process to identify potential weaknesses.
   - Look for indications of weak password policies, predictable session tokens, or insufficient session expiration controls.



3. **Monitor Session Cookies and Headers:**
    - Monitor session-related cookies and headers for consistency and adherence to security best practices.
    - Look for inconsistencies or deviations from expected behavior that may indicate security vulnerabilities.





Session handling rule editor

Details | Scope

Tools Scope

Select the tools that this rule will be applied to.

- [x] Target
- [x] Scanner
- [x] Repeater
- [x] Spider
- [x] Intruder
- [x] Sequencer
- [ ] Extender
- [ ] Proxy (use with caution)

URL Scope

Use the configuration below to control which URLs this rule applies to.

- ( ) Include all URLs
- (•) Use suite scope [defined in Target tab]
- ( ) Use custom scope

Parameter Scope

You can restrict the rule to requests containing specific parameters if required.

- [ ] Restrict to requests containing these parameters:

Edit

OK | Cancel

Session handling rule editor

Details | Scope

Rule Description

Reauthenticate

Rule Actions

The actions below will be performed in sequence when this rule is applied to a request.

Add | Enab... | Description

- Use cookies from the session handling cookie jar
- Set a specific cookie or parameter value
- Check session is valid
- Prompt for in-browser session recovery
- Run a macro
- Run a post-request macro
- Invoke a Burp extension

OK | Cancel

Session handling action editor - Reauthenticate

This action checks whether the session is valid, by either issuing the current request or running a predefined macro (sequence of requests), and inspecting the response. If the session is invalid, you can optionally perform a further action to obtain a new valid session.

Make request(s) to validate session:

- (•) Issue current request
- ( ) Run macro:

Add

Edit

[ ] Validate session only every 10 requests

Inspect response to determine session validity:

Location(s):
- [ ] HTTP headers
- [ ] Response body
- [x] URL of redirection target

Look for expression: login

Match type:
- (•) Literal string
- ( ) Regular expression

Case-sensitivity:
- ( ) Sensitive
- (•) Insensitive

Match indicates:
- (•) Invalid session
- ( ) Valid session

OK Cancel

Session handling action editor - Reauthenticate

Match indicates:
- (•) Insensitive
- (•) Invalid session
- ( ) Valid session

Define behavior dependent on session validity:

[x] If session is valid, don't process any further rules or actions for this request

[x] If session is invalid, perform the action below:

Run a macro

Select macro:

Add

Edit

Note that the request currently being processed by this session handling rule will still be issued, so the macro should not include this request unless it is necessary to issue it twice.

[x] Update current request with parameters matched from final macro response

- (•) Update all parameters except for:

Edit

- ( ) Update only the following parameters:

Edit

[ ] Tolerate URL mismatch when matching parameters (use for URL-agnostic CRSF tokens)

OK Cancel

Macro Recorder - Macro 1

Macro Recorder

Select the items from the proxy history that you wish to include in the macro, and click "OK". Note that to record a macro now using your browser you will need to ensure that proxy interception is turned off.

Intercept is on

Logging of out-of-scope Proxy traffic is disabled Re-enable

Filter: Hiding CSS, image and general binary content

#	Host	Meth...	URL	Para...	Edited	Status	Length	MIME ty...	Exten
29	https://ginandjuice.shop	GET	/resources/images/icon-accou...			200	1827	XML	svg
34	https://ginandjuice.shop	GET	/my-account			302	604		
36	https://push.services.m...	GET	/			101	240		
37	https://ginandjuice.shop	GET	/login			200	8077	HTML	
39	https://ginandjuice.shop	POST	/login	✓		200	8427	HTML	
42	https://ginandjuice.shop	POST	/login	✓		302	696		
43	https://content-signatur...	GET	/chains/remote-settings.cont...			304	189		chain
44	https://ginandjuice.shop	GET	/my-account			200	13239	HTML	
45	https://firefox.settings.s...	GET	/v1/buckets/security-state/coll...	✓		200	3608	JSON	
47	https://ginandjuice.shop	GET	/resources/images/delivered.s...			200	6385	XML	svg
48	https://ginandjuice.shop	GET	/resources/images/delivered-...			200	6572	XML	svg
52	https://content-signatur...	GET	/chains/onecrl.content-signat...			200	5859	script	chain
53	https://ginandjuice.shop	GET	/resources/images/check-circl...			200	1550	XML	svg

Request | Response

Raw | Params | Headers | Hex

```
POST /login HTTP/1.1
Host: ginandjuice.shop
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 70
Origin: https://ginandjuice.shop
Connection: close
Referer: https://ginandjuice.shop/login
Cookie:
AWSALB=f1KVBHUP3ajaghcWD8XvYGfbHET0fzkfcVXqL4/Y5oYwUoI6xMuKAZ5HpJtmqRgvBvcPOS4WwdGINLmwOt4wQR8k6vggSP19HdNji7rtzcK0
Cy5Ot+tccthTlV30;
```

? < + > Type a search term 0 matches

OK Cancel

Macro Editor

Macro Editor

Use the configuration below to define the items that are included in the macro, and the order they will be issued. You can configure how parameters and cookies are handled for each item. You can also test the macro to confirm it is working correctly.

Macro description: Macro 1

Macro items:

#	Host	Method	URL	Status	Cookies received	Derived parameters
1	https://ginandjuice.shop	GET	/login	200	AWSALB, AWSALBCORS	
2	https://ginandjuice.shop	POST	/login	200	AWSALB, AWSALBCORS	csrf
3	https://ginandjuice.shop	POST	/login	302	AWSALB, AWSALBCORS, ...	csrf

Configure item
Move up
Move down
Remove item

Request | Response

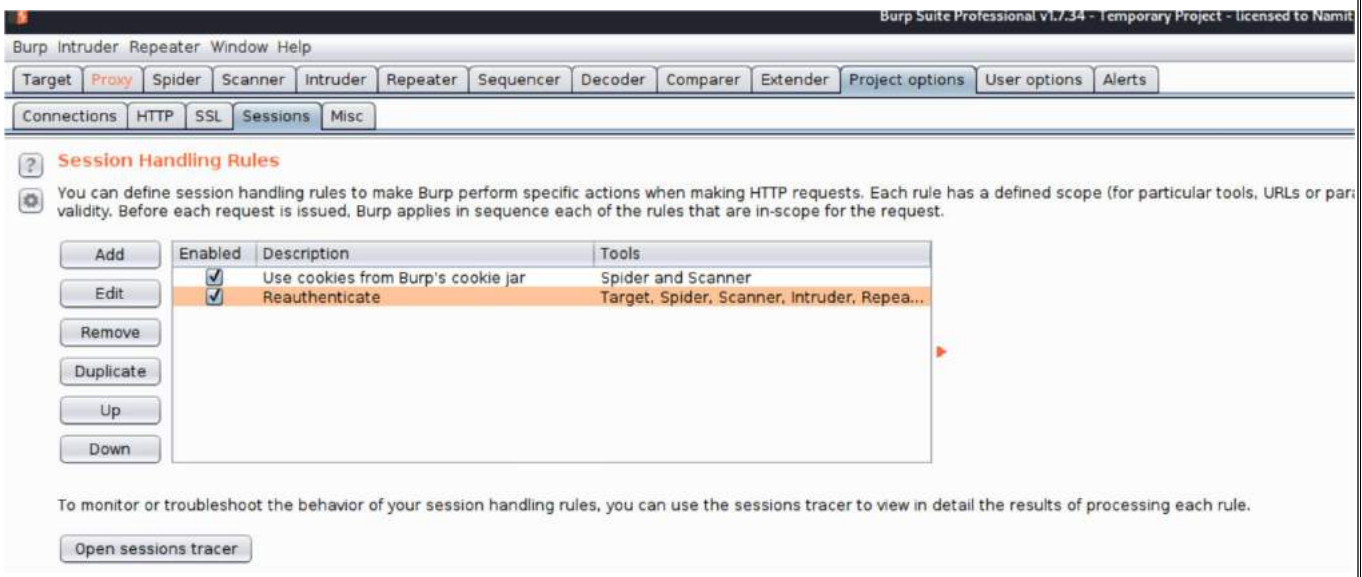
Raw | Params | Headers | Hex

```
GET /login HTTP/1.1
Host: ginandjuice.shop
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://ginandjuice.shop/
Connection: close
Cookie:
AWSALB=gVpXglIWxer1fFq+D8C//z214bRtj5VHsAQuBmNvgbGzAc4tzsGl7fSBdQZdv3QmijDCFYywUY+Dmq6TTDI1P22lOrZc99eXbwPUrLwsVKY7wLQSaiJ/kbYfw1ev;
AWSALBCORS=gVpXglIWxer1fFq+D8C//z214bRtj5VHsAQuBmNvgbGzAc4tzsGl7fSBdQZdv3QmijDCFYywUY+Dmq6TTDI1P22lOrZc99eXbwPUrLwsVKY7wLQSaiJ/kbYfw1e
```

? < + > Type a search term 0 matches

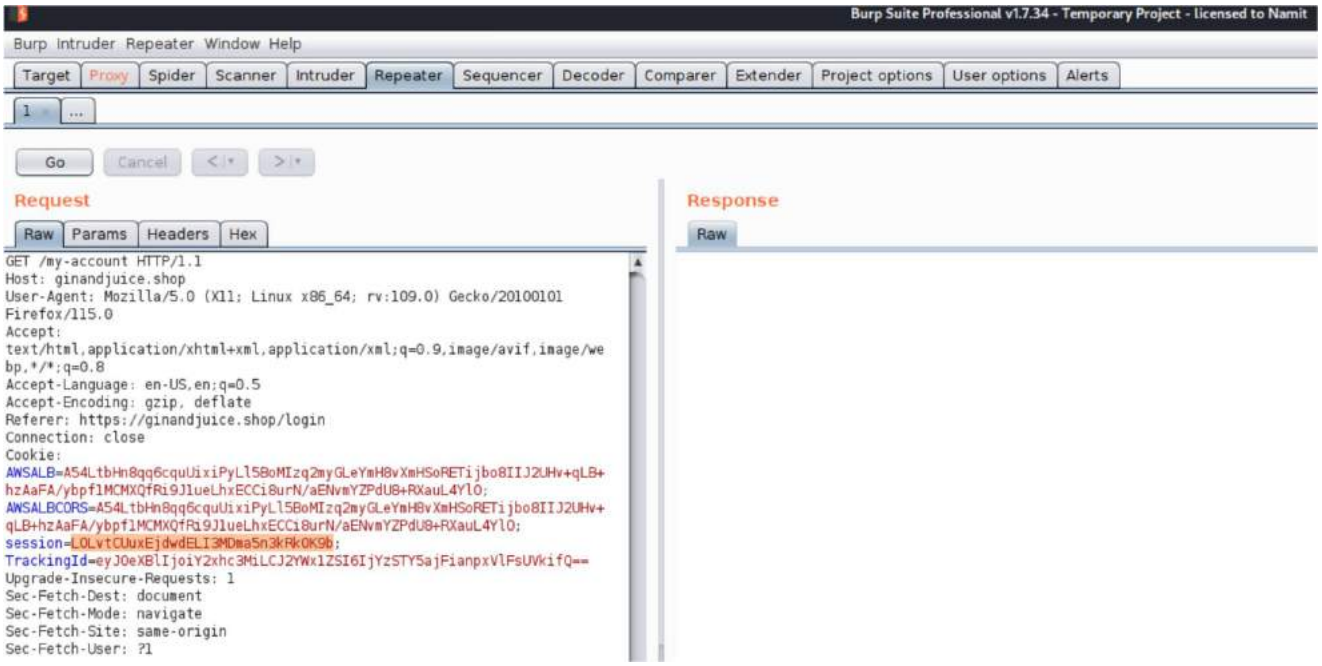
Re-record macro
Re-analyze macro
Test macro

OK Cancel

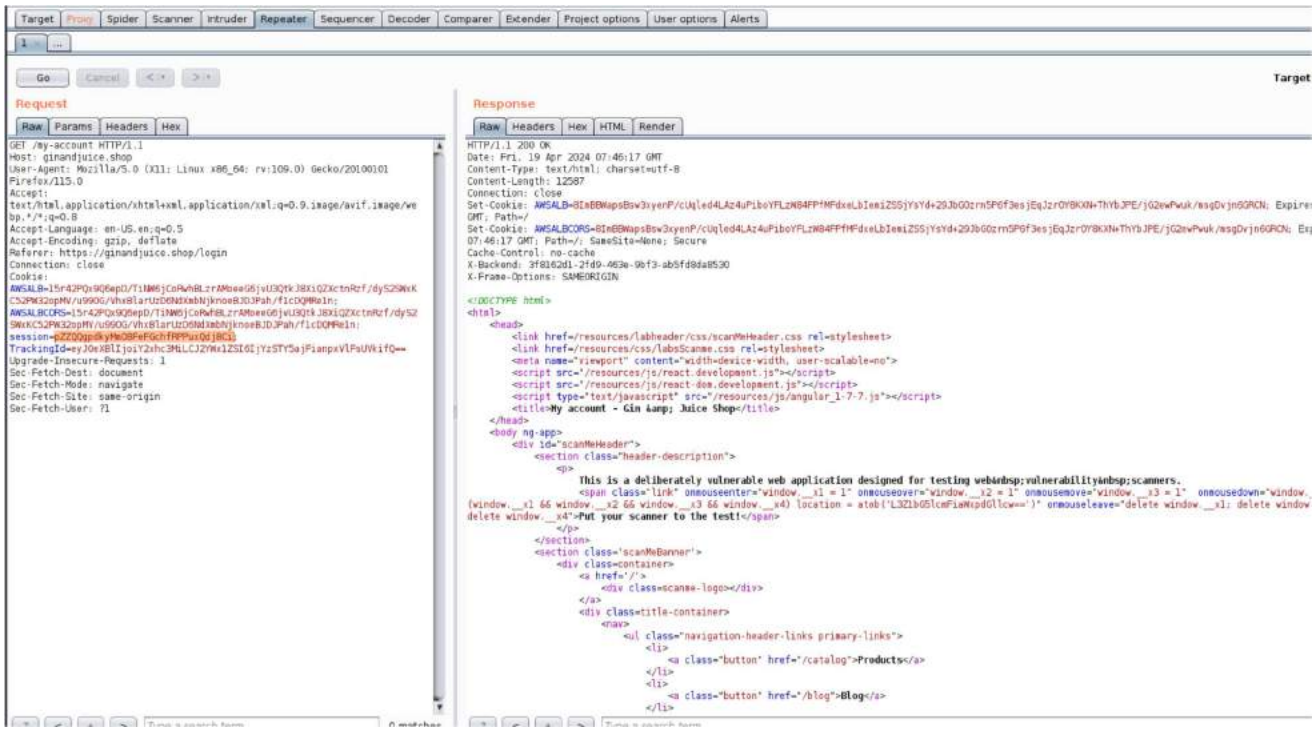


4. **Manipulate Parameters:**

- Utilize the "Repeater" tool in Burp Suite to manipulate session tokens or authentication parameters.

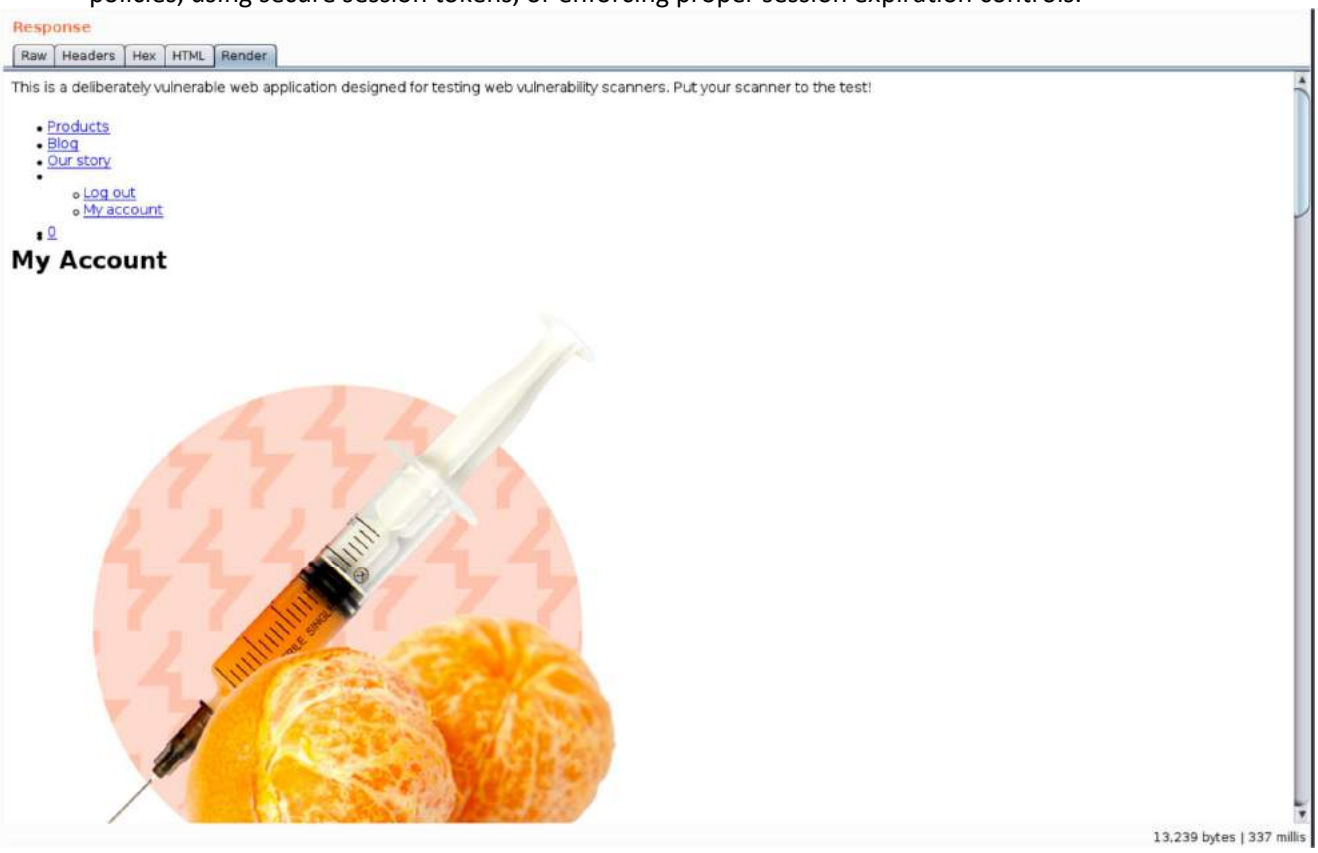


- Test for vulnerabilities such as session fixation or session hijacking by modifying session-related values.



5. **Generate Report:**

- Document any identified vulnerabilities or weaknesses in authentication and session management mechanisms.
- Provide detailed recommendations for improvement, such as implementing stronger password policies, using secure session tokens, or enforcing proper session expiration controls.



By following these steps, We can effectively assess the security of authentication and session management mechanisms on the website https://ginandjuice.shop/, helping to identify and mitigate potential risks to user accounts and sensitive data.

3. Testing for SQL Injection:

Scenario Question: Can you demonstrate how Burp Suite can be used to assess a web application's susceptibility to SQL injection attacks and verify the findings?

Scenario: We want to check if the website https://ginandjuice.shop/ is vulnerable to SQL injection attacks.

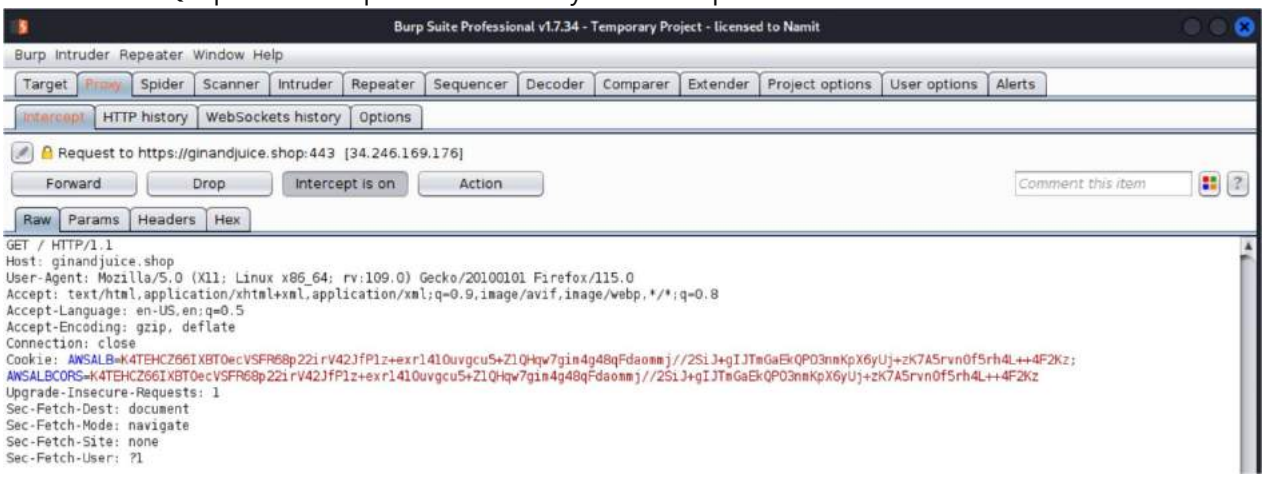
Steps:

1. **Proxy Setup:**

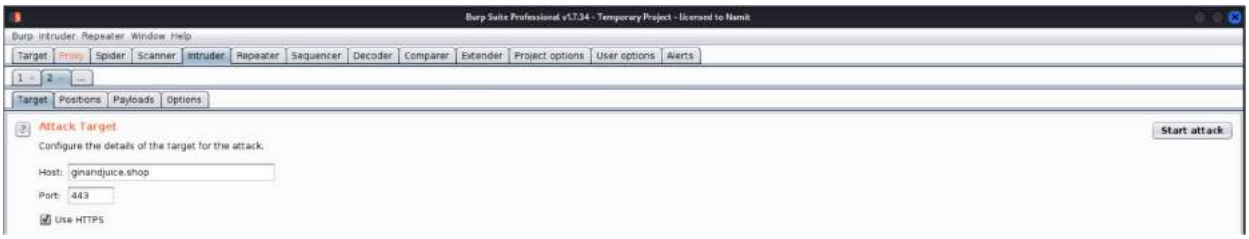
- Configure Burp Suite as a proxy and ensure interception is enabled.
- Navigate to https://ginandjuice.shop/ using your browser.

2. **Interception:**

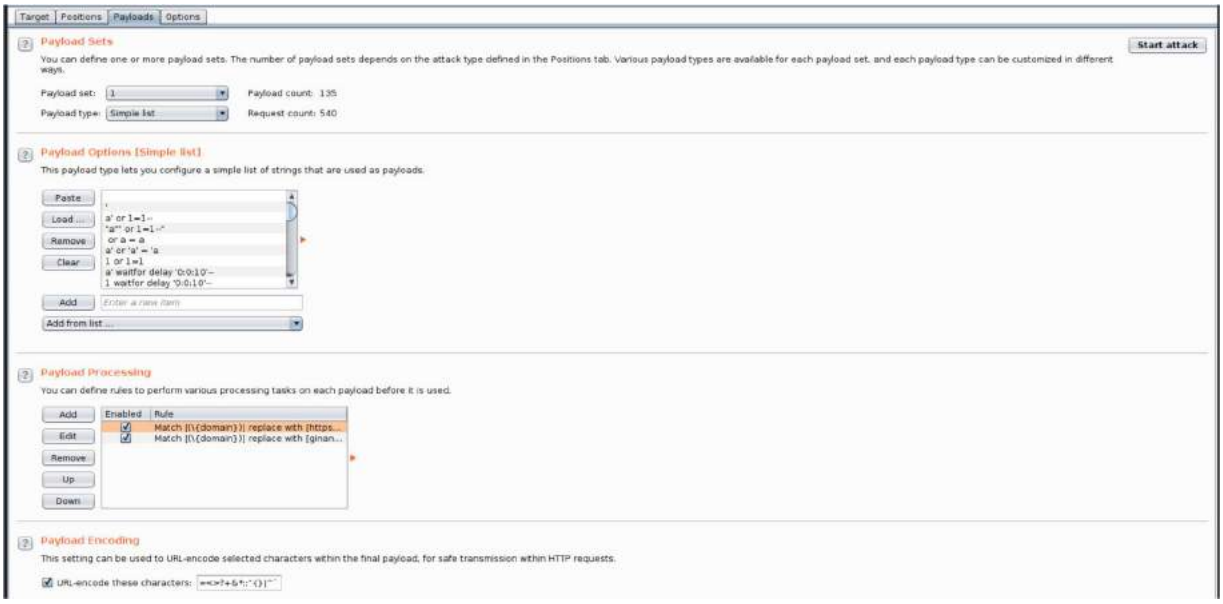
- Interact with input fields on the ginandjuice.shop website, such as search or login forms.
- Observe SQL queries in requests in the Proxy tab of Burp Suite.



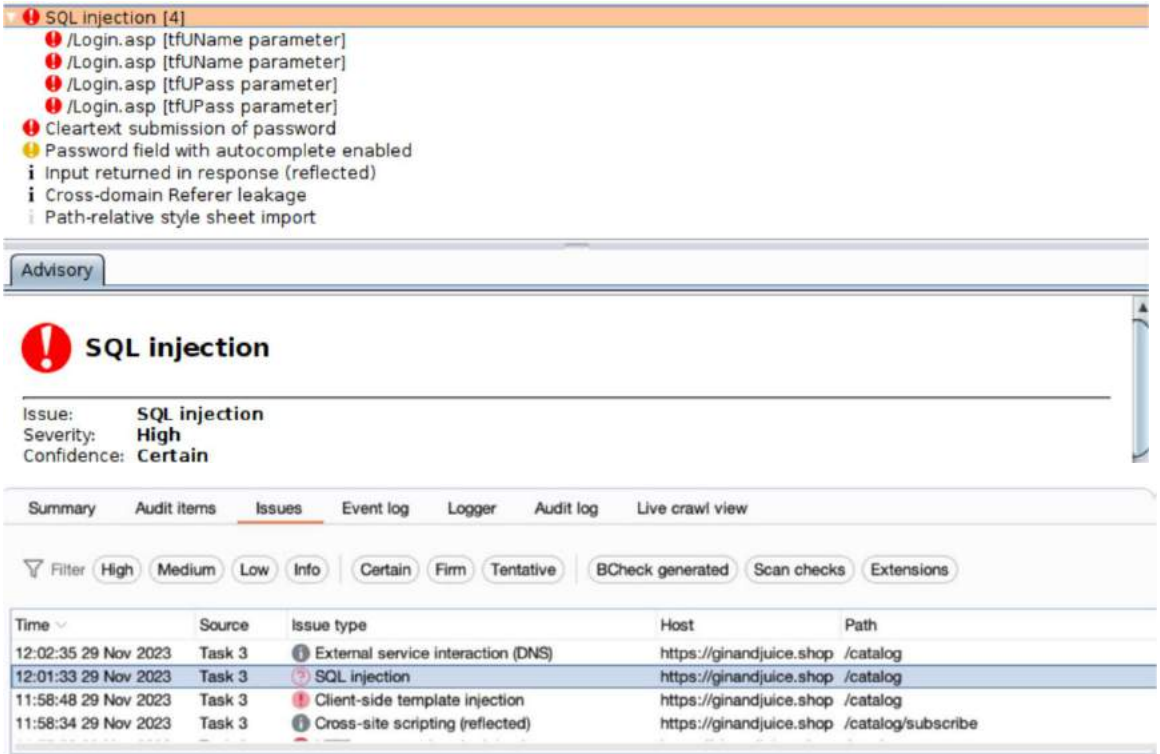
3. **Intruder Tool**: Use Burp's Intruder tool to modify parameters and observe SQL responses.



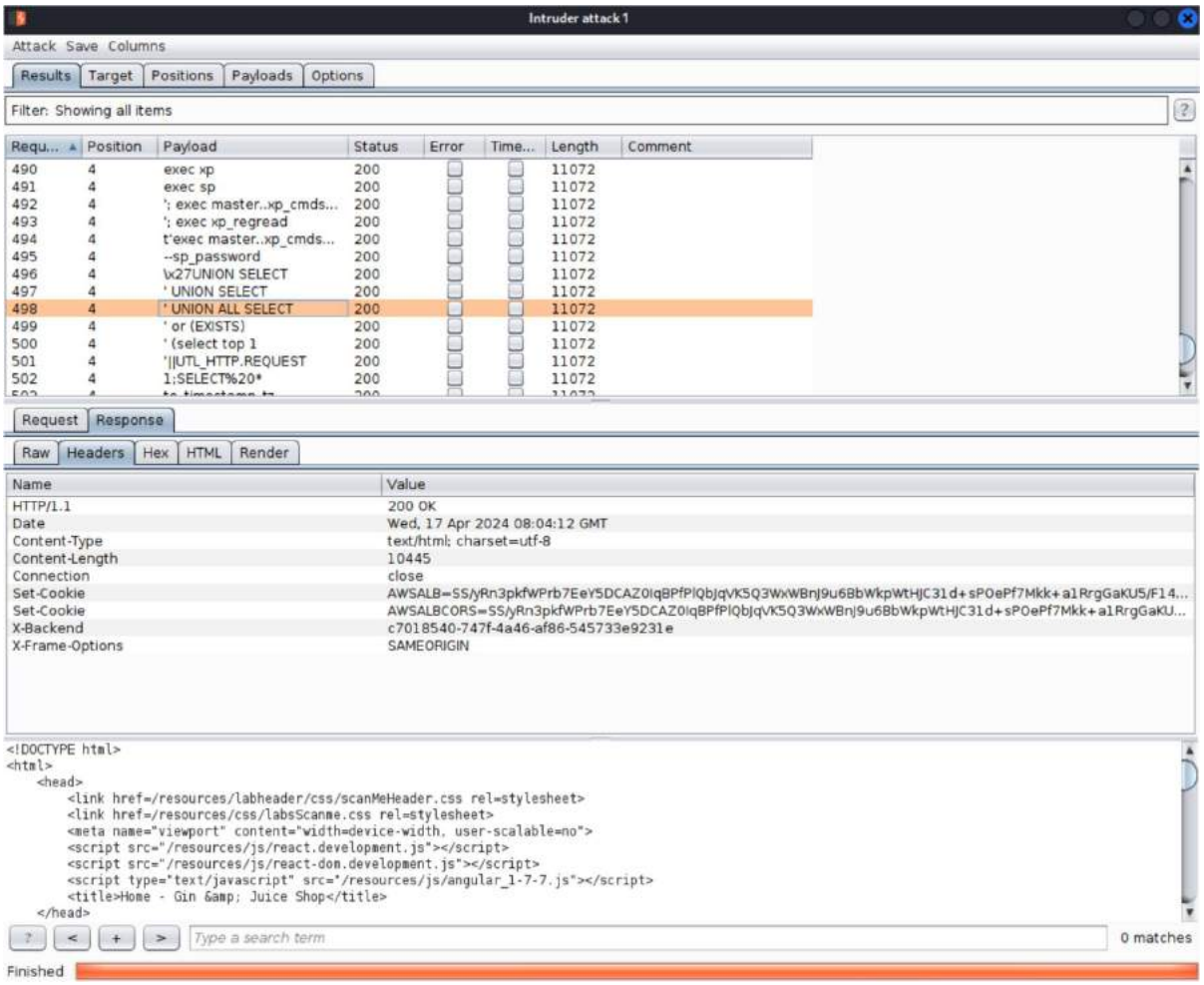
4. **Payload Crafting**: Craft SQL injection payloads and inject them through various input fields.



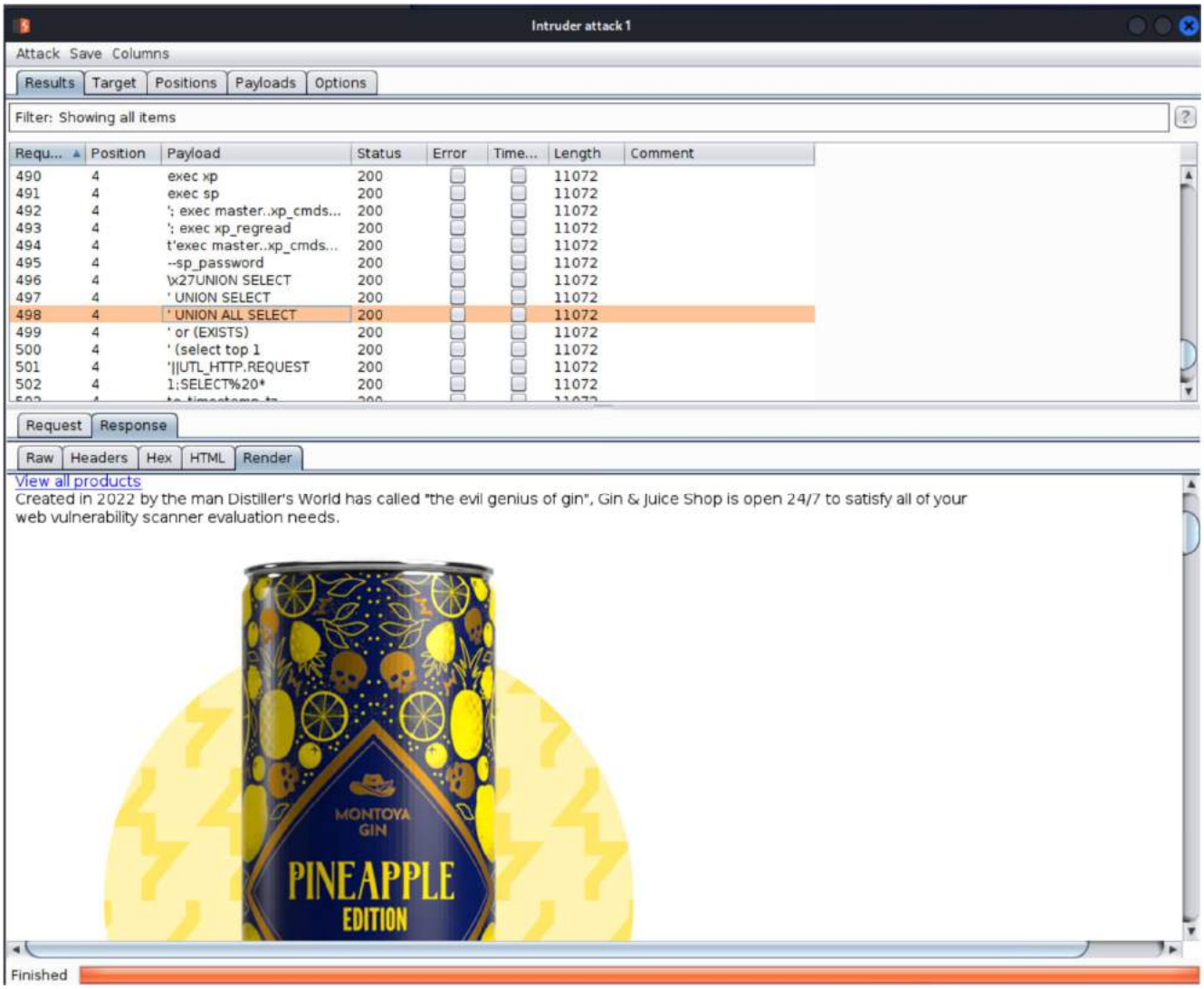
5. **Response Analysis**: Analyze server responses for any SQL errors or unusual behavior.



6. **Scanner**: Utilize Burp's Scanner to automate SQL injection testing and verify the findings.



- Utilize Burp Suite's Render tool to render the response in different formats (e.g., HTML, JSON).
- Examine the rendered response for any unexpected behavior or content that may indicate SQL injection vulnerabilities.



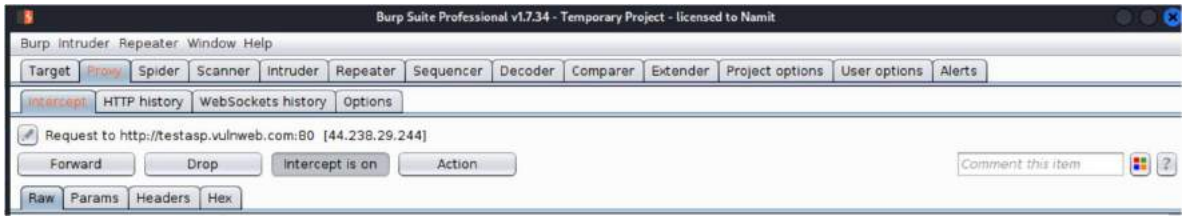
4. Session Hijacking:

Scenario Question: In what ways can Burp Suite be employed to assess the security of a web application against session hijacking attacks, and how would you document the results?

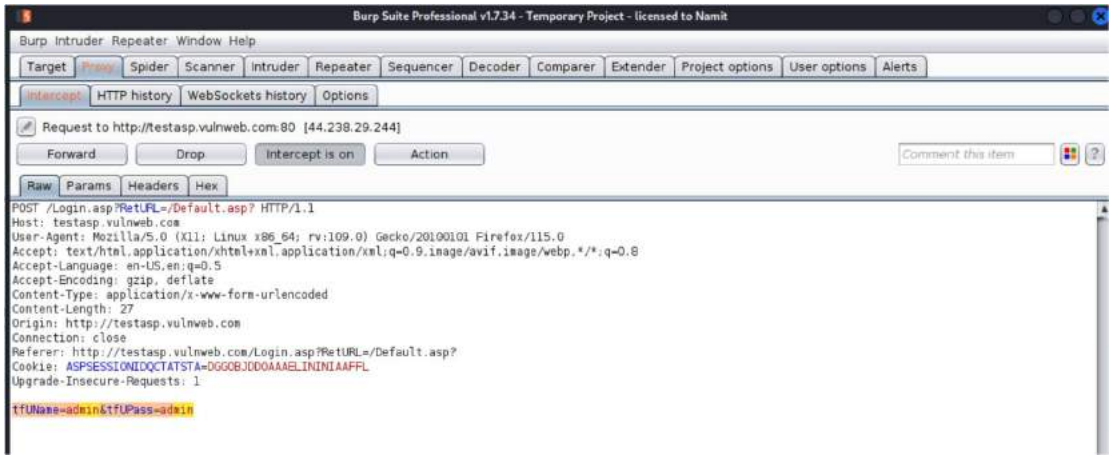
Scenario: We want to test the security of https://testasp.vulnweb.com/Login.asp?RetURL=/Default.asp? against session hijacking attacks.

- **Steps**:

1. **Interception**: Intercept a user's session token using Burp's Proxy.

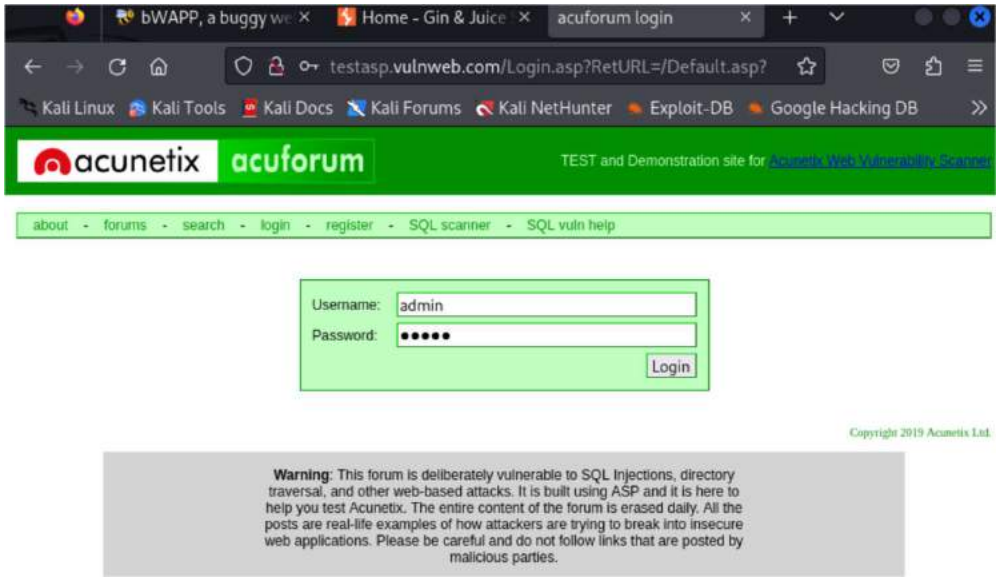


2. **Session Handling Rules**: Copy the session token and use Burp's Session Handling Rules to set it as your own session.



3. **Browsing**: Browse the application to see if you gain unauthorized access to sensitive data or perform unauthorized actions.

4. **Monitoring**: Monitor for any session expiration or invalidation mechanisms.



5. **Documentation**: Document any successful or unsuccessful attempts and their implications.

5. **Sensitive Data Exposure**:

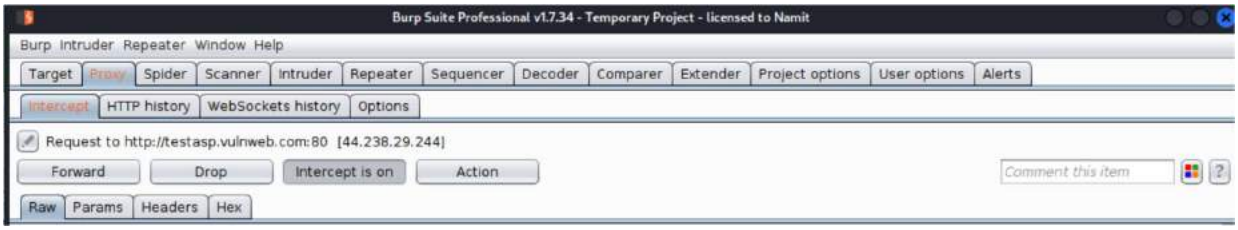
Scenario Question: Can you demonstrate how Burp Suite can be used to identify and mitigate sensitive data exposure vulnerabilities in the web application http://testasp.vulnweb.com/Login.asp?RetURL=/Default.asp?

Scenario: You suspect that the web application testasp.vulnweb.com/Login.asp?RetURL=/Default.asp is exposing sensitive information

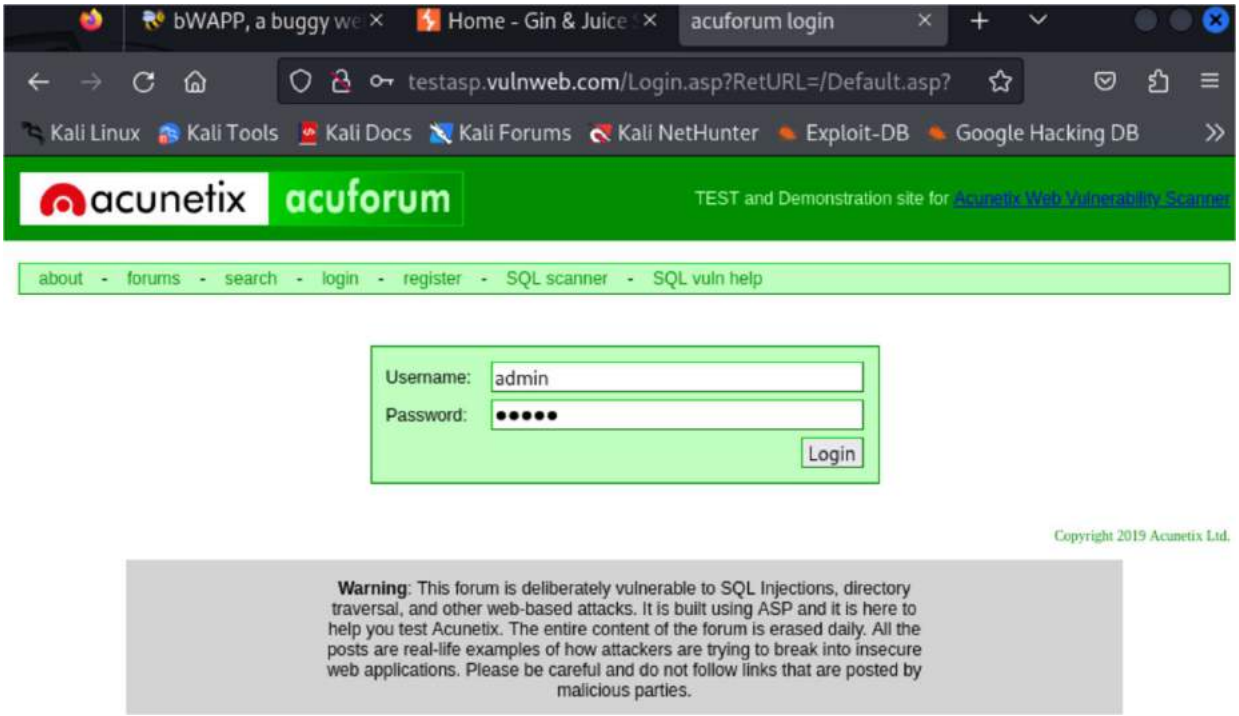
Steps:

1. **Interception:**

- Launch Burp Suite and configure it as a proxy.
- Navigate to testasp.vulnweb.com/Login.asp?RetURL=/Default.asp using your browser.
- Use Burp's Proxy tool to intercept traffic while browsing the application.

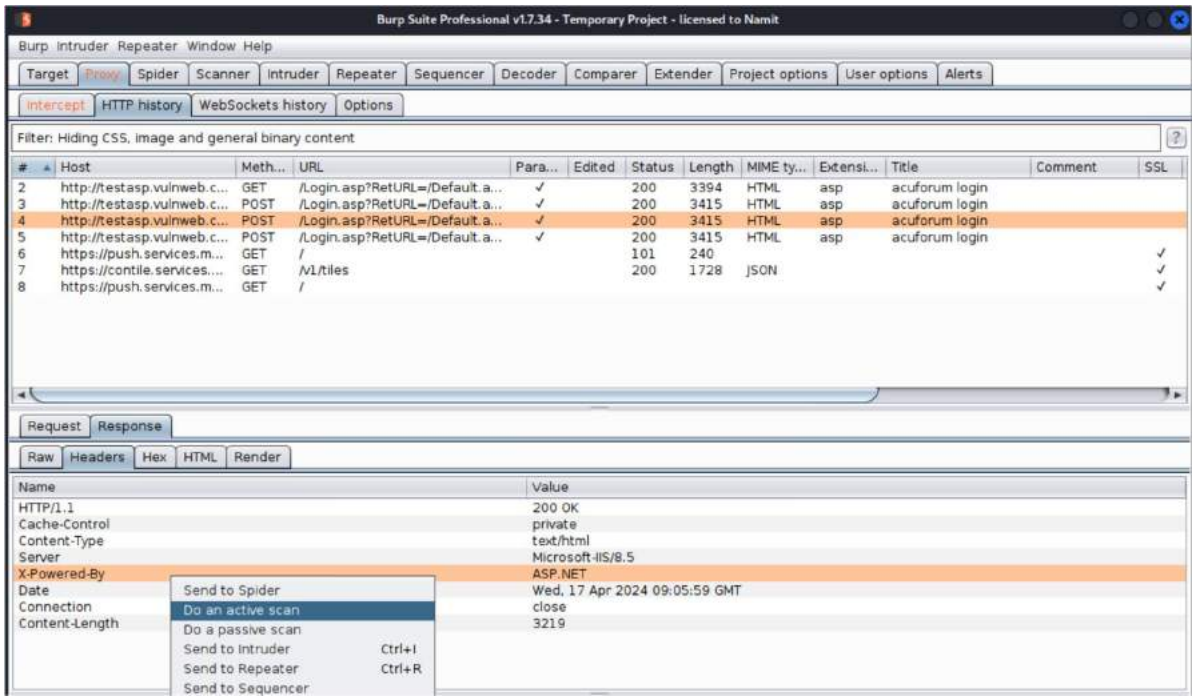


2. **Response Analysis**: Look for responses containing sensitive data such as passwords, API keys, or personal information.



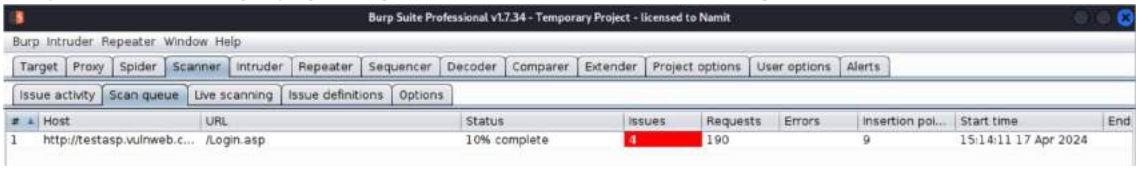
3. **Header Inspection:**

- Inspect the server's response headers for security-related headers such as X-Frame-Options or Content-Security-Policy.
- Look for proper security headers that help prevent sensitive data exposure, such as Content-Security-Policy directives restricting resource loading.

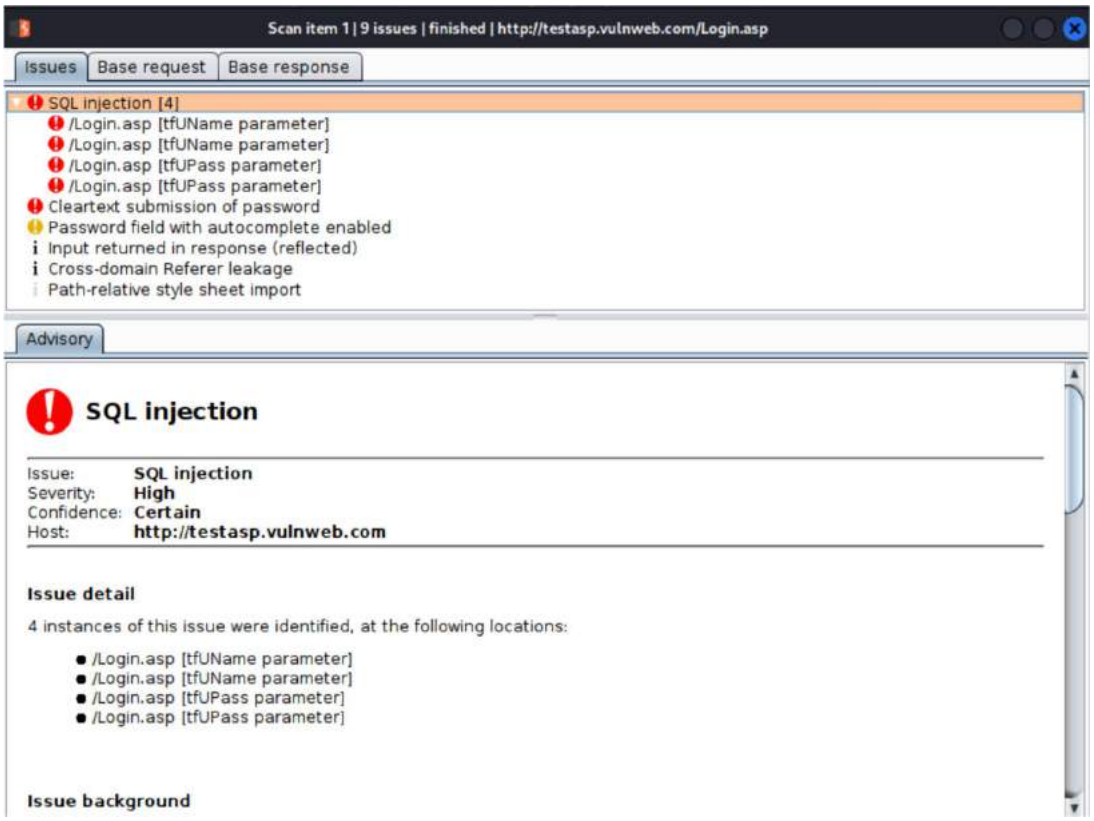


4. **Scanner:**

- Utilize Burp's Scanner to automate the detection of sensitive data exposure vulnerabilities.
- Configure the scanner to target areas of the application where sensitive information may be exposed, such as login pages, registration forms, or account management sections.



5. **Reporting**: Report any findings and suggest remediation measures to the development team.



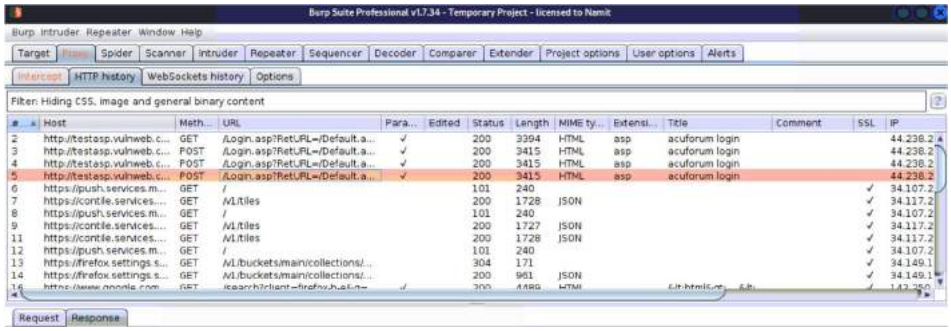
6. CSRF (Cross-Site Request Forgery) Vulnerabilities:

Scenario Question: Can you demonstrate how Burp Suite can be used to assess if the web application http://testasp.vulnweb.com/Login.asp?RetURL=/Default.asp is vulnerable to CSRF attacks?

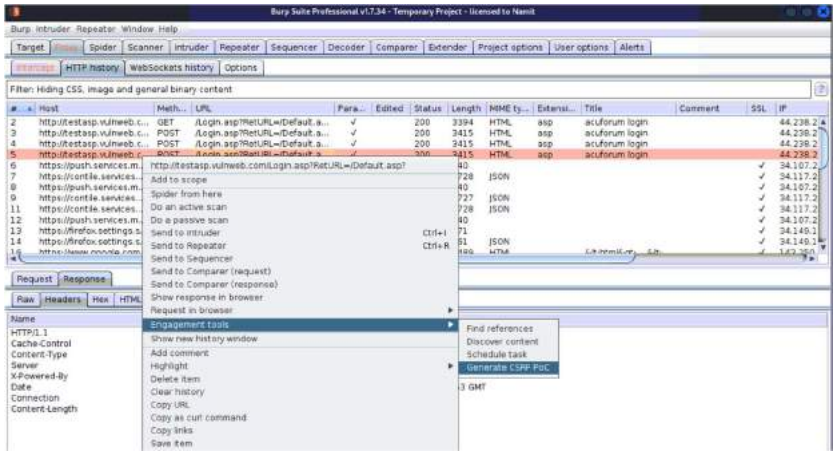
Scenario: You want to assess if the web application http://testasp.vulnweb.com/Login.asp?RetURL=/Default.asp is vulnerable to CSRF attacks.

- **Steps**:

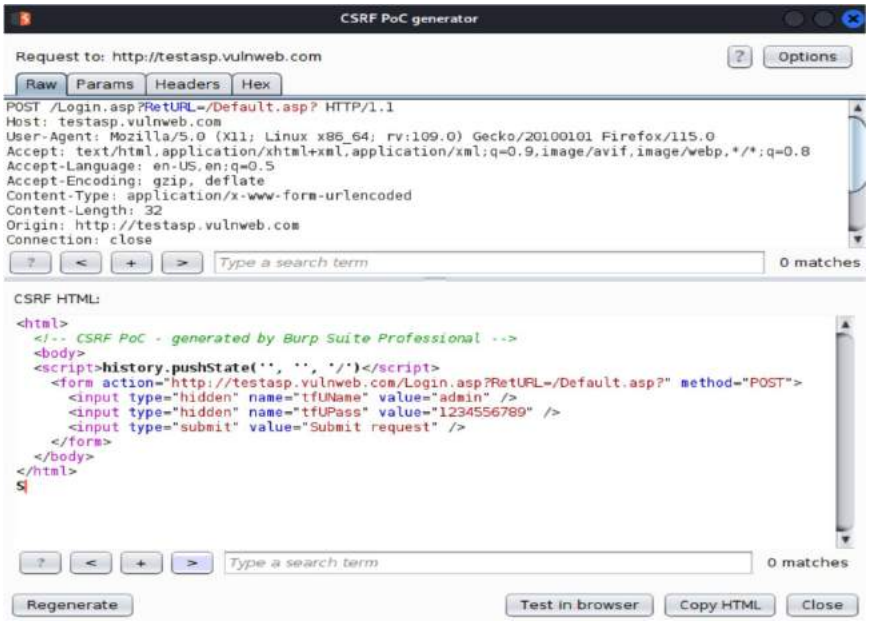
1. **Interception**: Use Burp's Proxy to intercept and modify requests while interacting with the application.



2. **Payload Crafting**: Craft a malicious HTML page containing CSRF payloads targeting the application's functionalities.



3. **Host the Page**: Host the malicious page and trick a logged-in user into visiting it by copying the HTML.



- Now paste html code in Vscode or any browser and host it or go live.
- Trick a logged-in user into visiting the malicious page by sending them a crafted link or embedding it within a legitimate website.

```html
hi.html > ...
<html>
  <!-- CSRF PoC - generated by Burp Suite Professional -->
  <body>
  <script>history.pushState('', '', '/')</script>
    <form action="http://testasp.vulnweb.com/Login.asp?RetURL=/Default.asp?" method="POST">
      <input type="hidden" name="tfUName" value="admin" />
      <input type="hidden" name="tfUPass" value="1234556789" />
      <input type="submit" value="Submit request" />
    </form>
  </body>
</html>
```

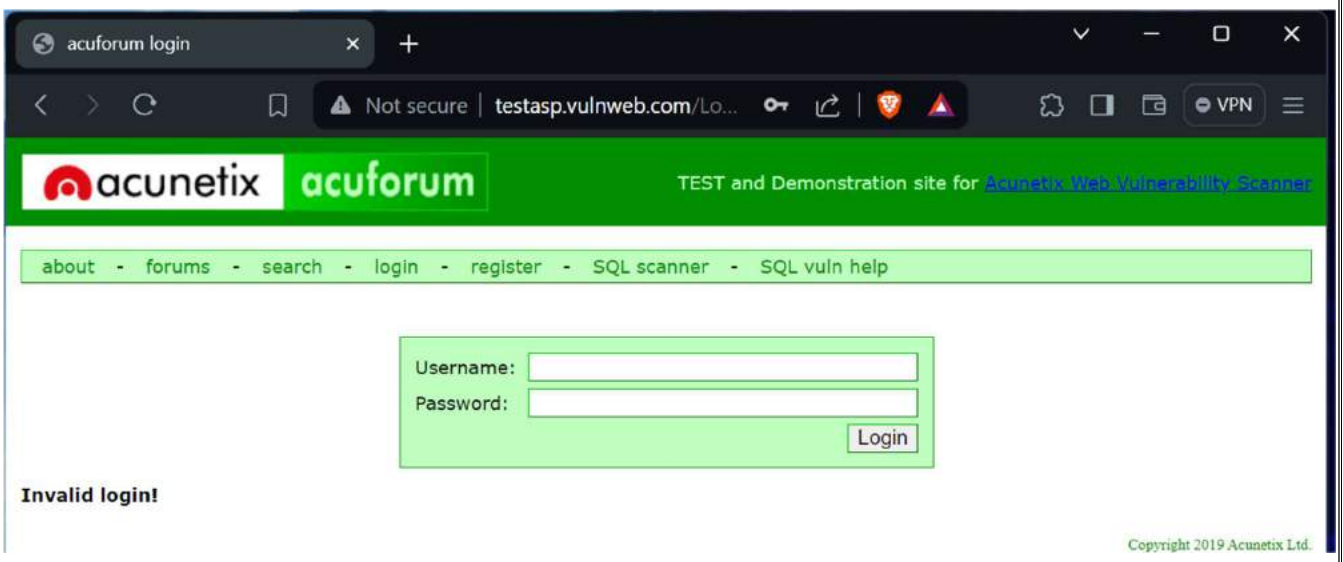
Ln 12, Col 1 Spaces: 4 UTF-8 CRLF HTML Go Live Prettier

4. **Monitoring**: Monitor the intercepted requests in Burp Suite to see if the CSRF attack is successful.

- Analyze the requests generated by the victim user's interaction with the malicious HTML page.
- submit the request

5. **Assessment**: Assess the impact of the attack and recommend mitigations such as CSRF tokens.

- Copy the HTML of the malicious page and test it by pasting it into a web browser or an editor like VSCode, then go live to observe the impact of the attack firsthand.



7. Traffic analysis Testing:

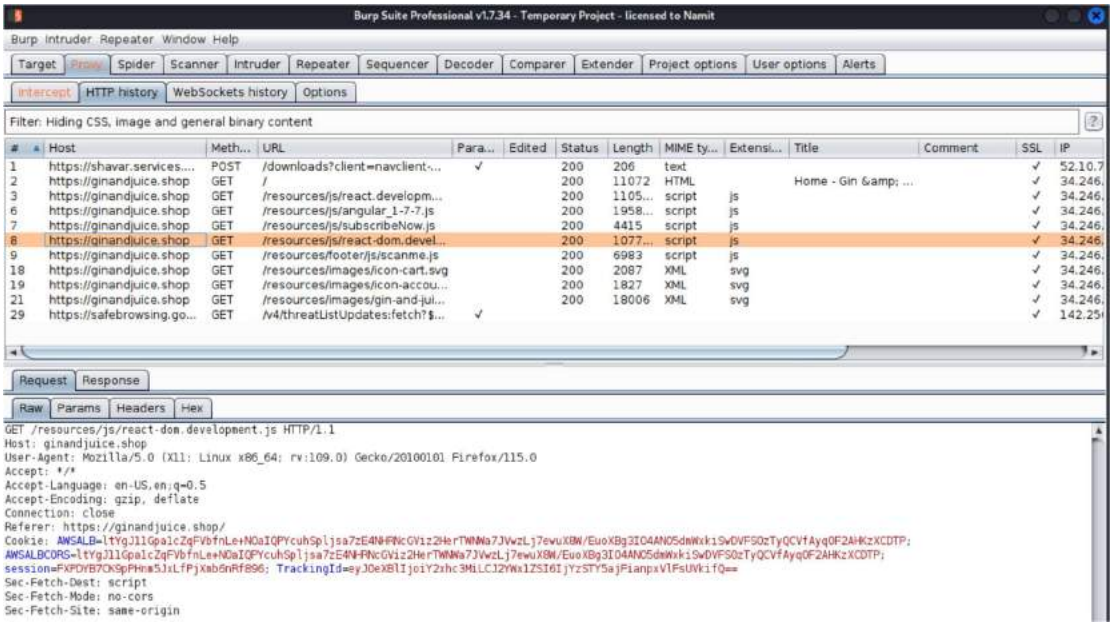
Scenario Question: How would you utilize Burp Suite to detect and mitigate suspicious or malicious activity in the web application https://ginandjuice.shop/ through traffic analysis?

Scenario: We suspect that the web application https://ginandjuice.shop/ may be under attack or compromised, and you want to perform traffic analysis to detect any suspicious or malicious activity.

Steps:

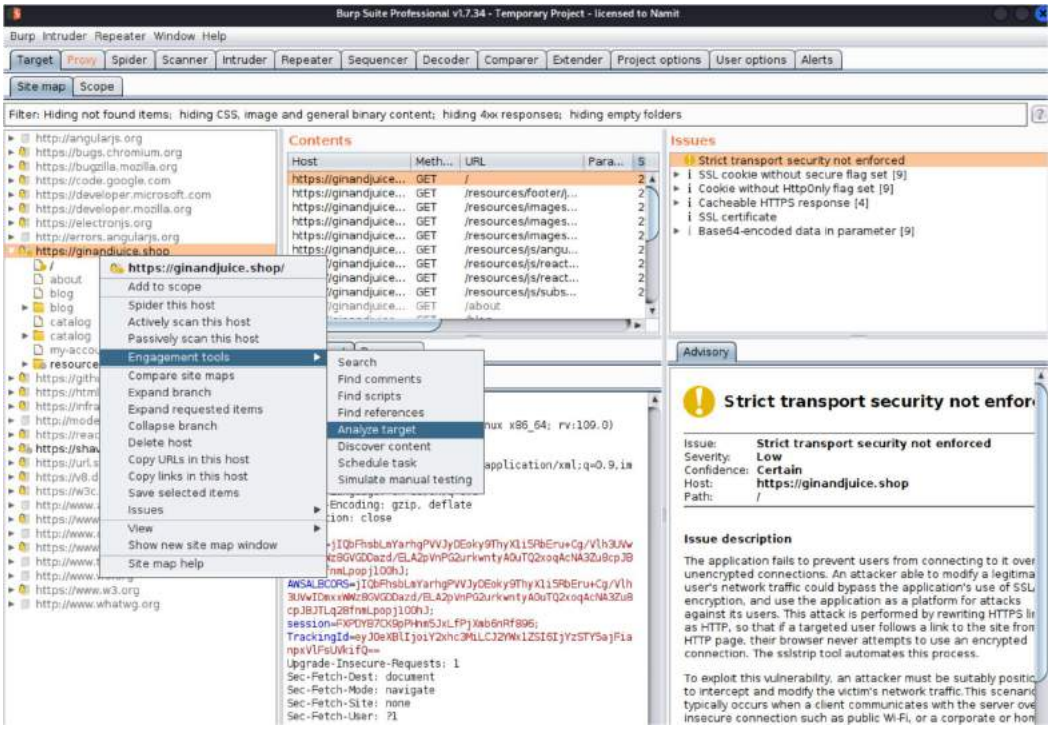
1. Proxy Setup: Configure Burp Suite as a proxy and ensure interception is enabled.

2. Intercept Traffic: Browse the target web application https://ginandjuice.shop/ using your browser allowing Burp Suite to intercept HTTP(S) requests and responses.



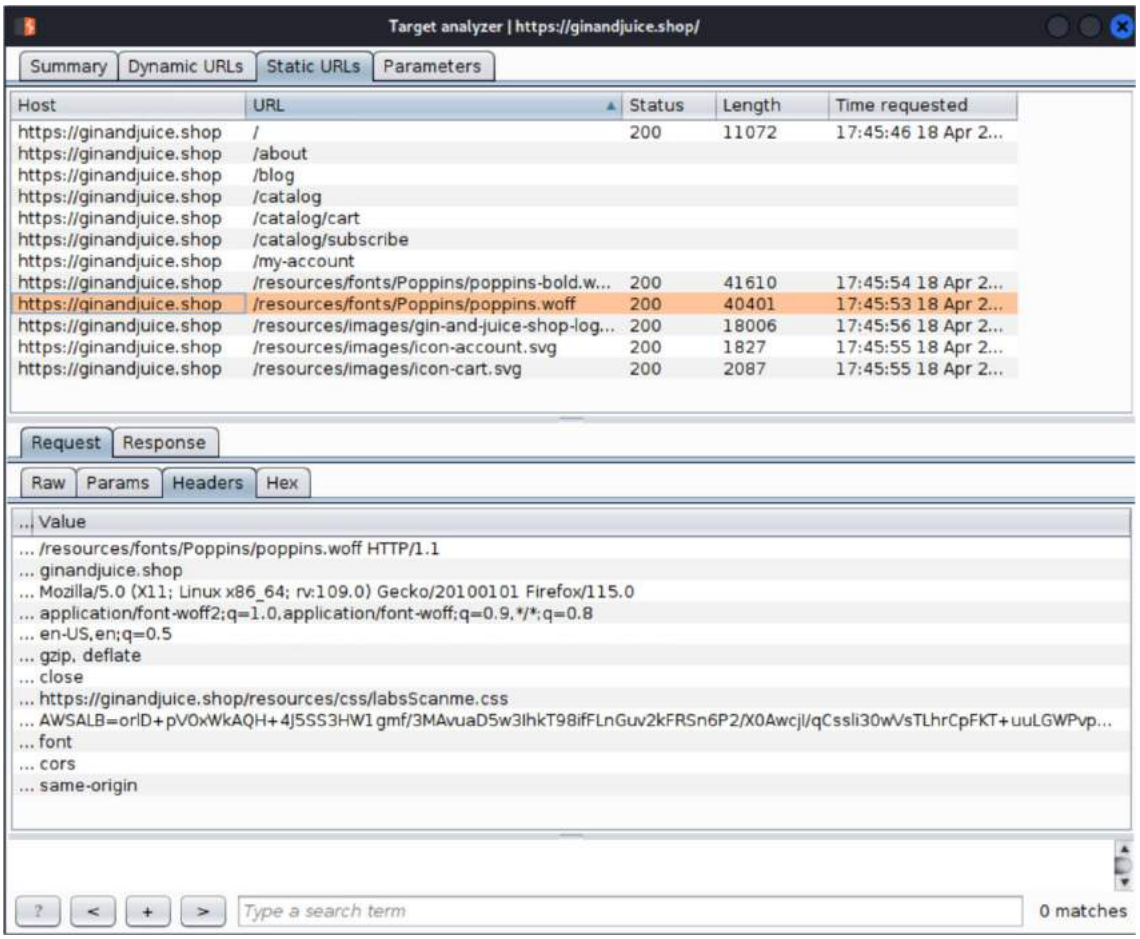
3. Analyze Request Patterns: Review intercepted requests for abnormal patterns or unexpected endpoints.

4. Inspect Response Content: Analyze response content for injected scripts or unexpected files.



5. Check HTTP Headers:

- Inspect the HTTP headers of intercepted requests and responses for suspicious user-agents, unusual cookies, or any other anomalies.
- Look for headers commonly used by attackers to fingerprint or exploit vulnerabilities in web applications.



6. Identify Outbound Connections: Monitor outbound connections for connections to known malicious domains or IP addresses.

- Use tools like Burp Suite's Collaborator or external threat intelligence feeds to identify suspicious outbound connections.

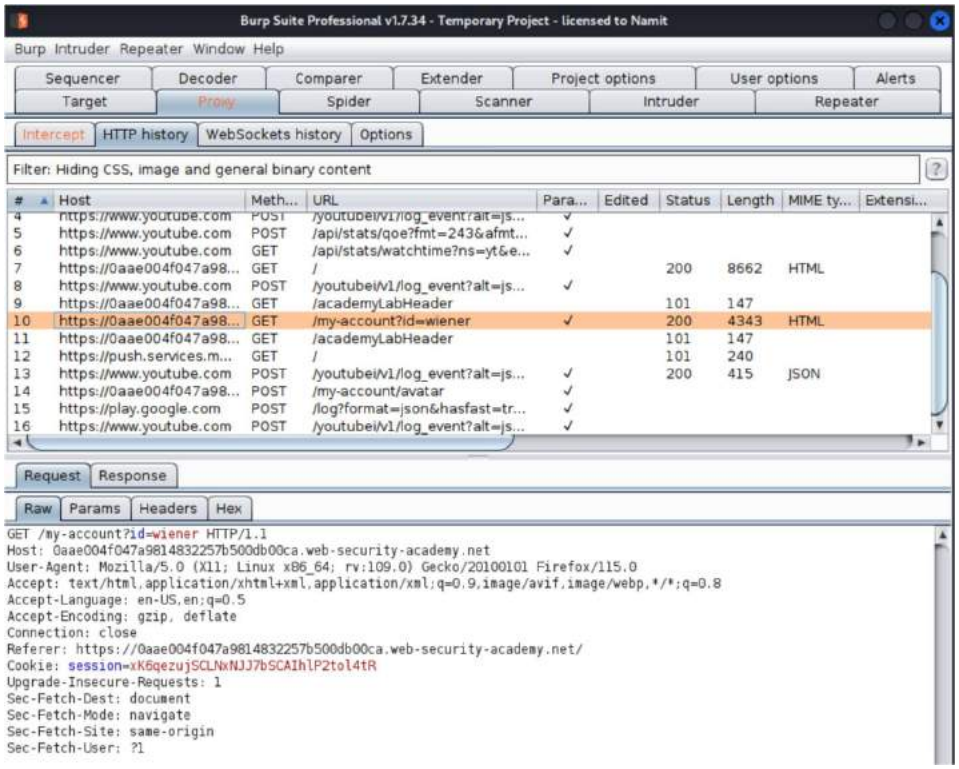
8. **File Upload Vulnerabilities**:

Scenario Question: How can Burp Suite be used to assess the security of a web application's file upload functionality and detect potential vulnerabilities leading to remote code execution?

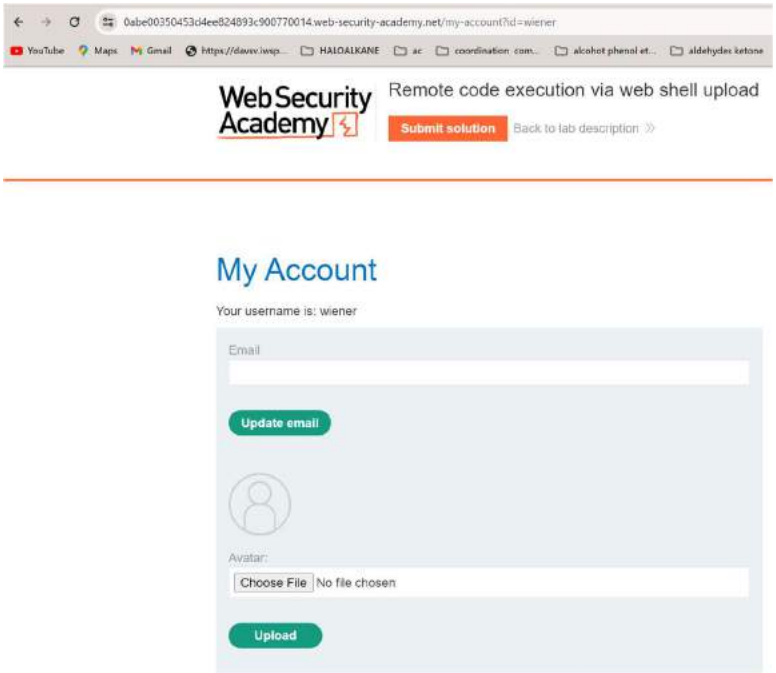
Scenario: We aim to evaluate the security of the file upload functionality on the web application located at https://portswigger.net/web-security/file-upload/lab-file-upload-remote-code-execution-via-web-shell-upload.

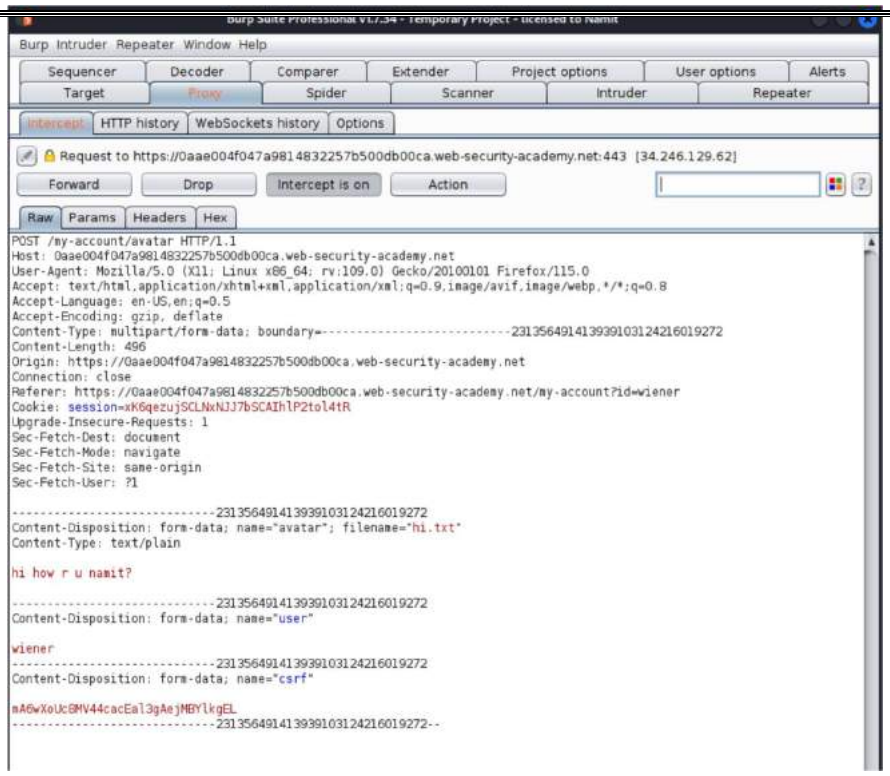
- **Steps**:

1. **Request Interception**: Upload various file types with different extensions using Burp's Repeater tool.



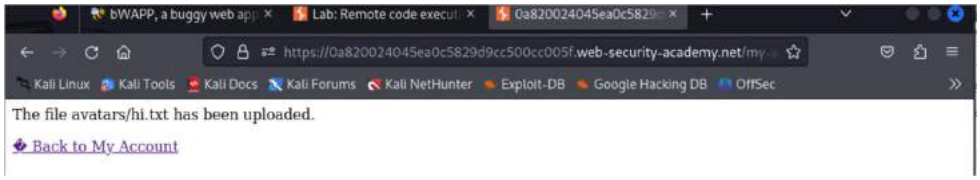
2. **Payload Modification**: Modify the file content to include malicious scripts or executable code.



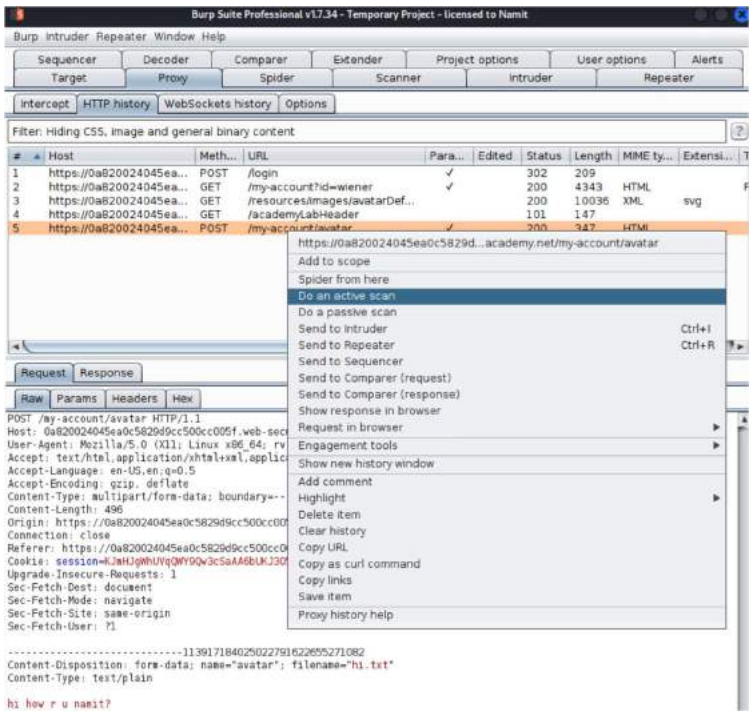


3. **Response Analysis**:

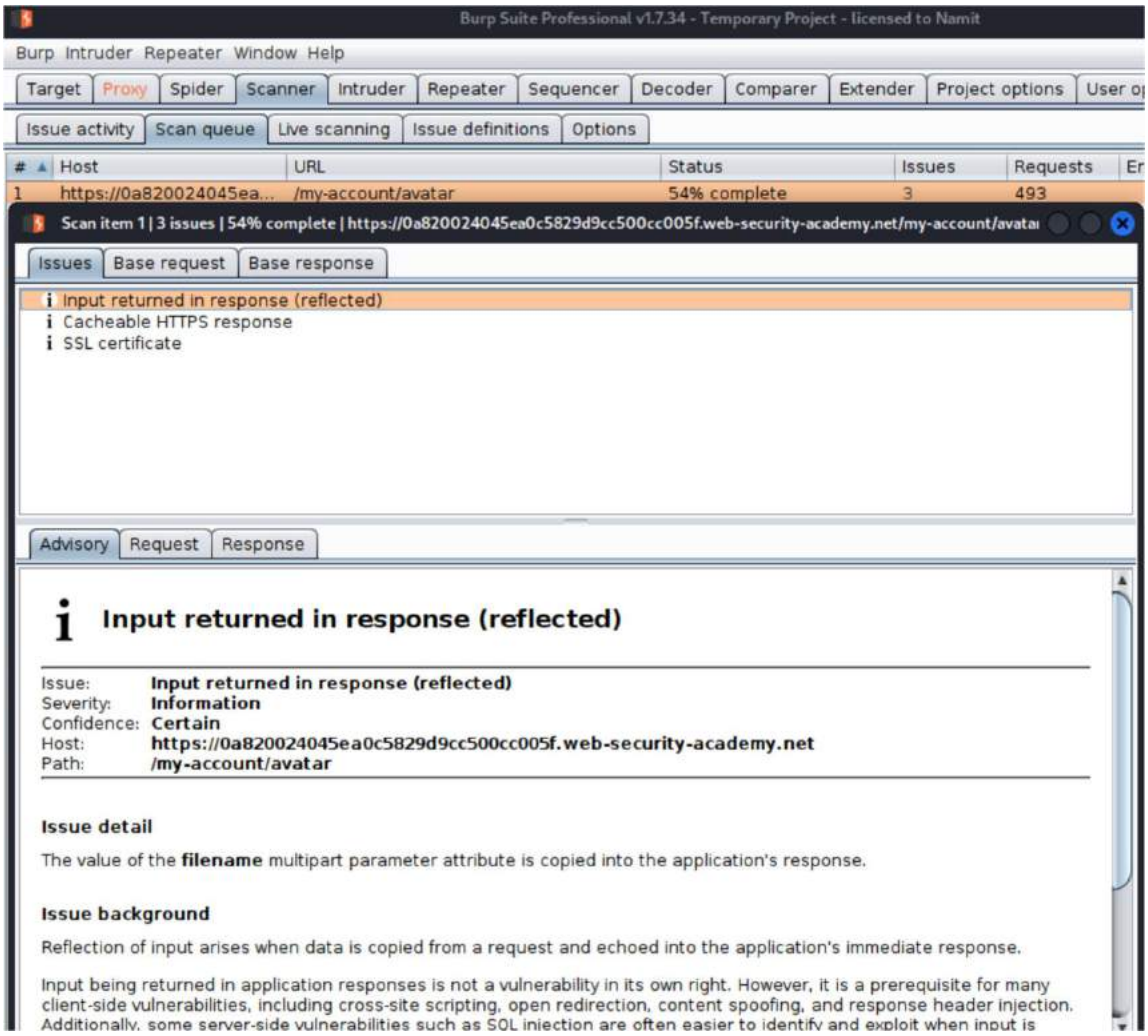
- Analyze the responses from the server to the file upload requests for any indications of validation errors or unexpected behaviors.
- Look for responses that may suggest successful execution of the injected payloads or bypassing of file type restrictions.



4. **Scanner Usage**: Use Burp's Scanner to automate file upload vulnerability testing.

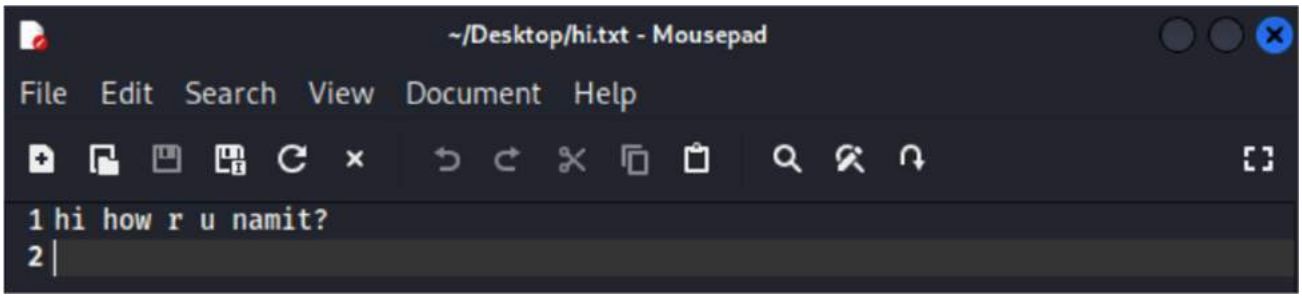


- Configure the scanner to target the file upload functionality and perform thorough testing to identify any potential security weaknesses.



5. **Analyzes**: Analyze the findings and the changes made.

- Evaluate the impact and severity of any discovered vulnerabilities, particularly those leading to remote code execution.
- Document the changes made during the testing process and provide recommendations for remediation to enhance the security of the file upload functionality.



Burp Suite Professional v1.7.34 - Temporary Project - licensed to Namit

Burp Intruder Repeater Window Help

Target | Proxy | Spider | Scanner | Intruder | Repeater | Sequencer | Decoder | Comparer | Extender | Project options | User options | Alerts

1 ...

Go | Cancel | < | > | Target: https://0a820024045ea0c5829d9cc500cc005f.web-security-academy.net

Request

Raw | Params | Headers | Hex

```
POST /my-account/avatar HTTP/1.1
Host: 0a820024045ea0c5829d9cc500cc005f.web-security-academy.net
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101
Firefox/115.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/we
bp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data;
boundary=---------------------------281583080716475047321166121 65
Content-Length: 497
Origin: https://0a820024045ea0c5829d9cc500cc005f.web-security-academy.net
Connection: close
Referer:
https://0a820024045ea0c5829d9cc500cc005f.web-security-academy.net/my-acco
unt
Cookie: session=KJmHJgWhUVqQWY9Qw3cSaAA6bUKJ3O5W
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1

-----------------------------281583080716475047321166121 65
Content-Disposition: form-data; name="avatar"; filename="hi.txt"
Content-Type: text/plain

hi how are you garvit.

-----------------------------281583080716475047321166121 65
Content-Disposition: form-data; name="user"

wiener
-----------------------------281583080716475047321166121 65
Content-Disposition: form-data; name="csrf"

PiGIRZD5Qdz9IxWi2SjN9SNzKTCpl6i0
-----------------------------281583080716475047321166121 65--
```

Response

Raw | Headers | Hex | HTML | Render

```
HTTP/1.1 200 OK
Date: Thu, 18 Apr 2024 18:06:24 GMT
Server: Apache/2.4.41 (Ubuntu)
Vary: Accept-Encoding
Connection: close
Content-Type: text/html; charset=UTF-8
X-Frame-Options: SAMEORIGIN
Content-Length: 127

The file avatars/hi.txt has been uploaded.<p><a href="/my-account"
title="Return to previous page">« Back to My Account</a></p>
```

bWAPP, a buggy web app | Lab: Remote code execut | Remote code execution v | 0a820024045ea0c5829d

https://0a820024045ea0c5829d9cc500cc005f.web-security-academy.net/files/avatars/hi.txt

Kali Linux | Kali Tools | Kali Docs | Kali Forums | Kali NetHunter | Exploit-DB | Google Hacking DB | OffSec | Remote code exec

```
hi how are you garvit.
```

9. **Information Disclosure via Error Messages**:

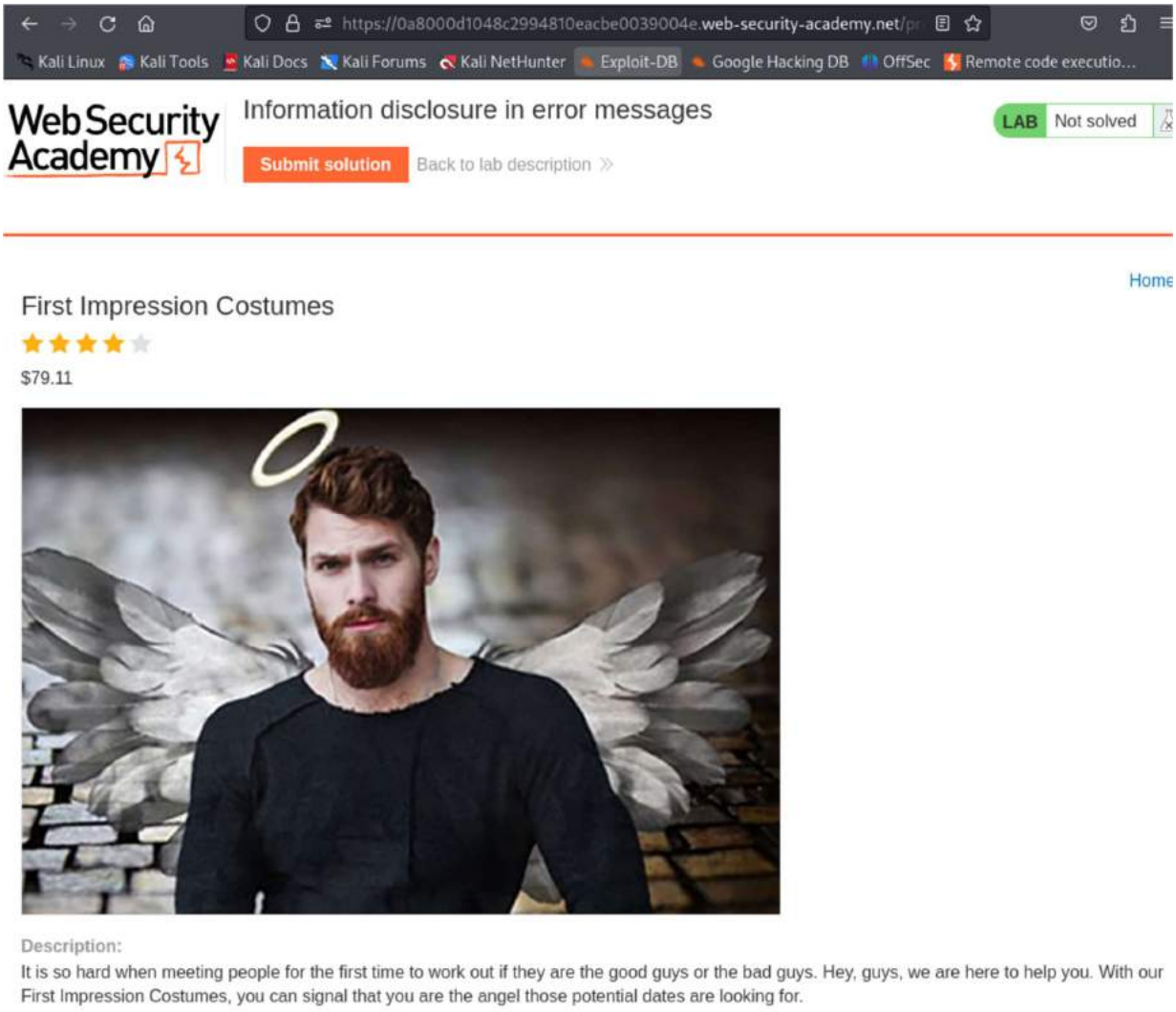
Scenario Question: How can Burp Suite be utilized to investigate and mitigate potential information disclosure vulnerabilities through error messages in a web application?

Scenario: We aim to assess the security of error messages in the web application located at https://portswigger.net/web-security/information-disclosure/exploiting/lab-infoleak-in-error-messages

- **Steps**:

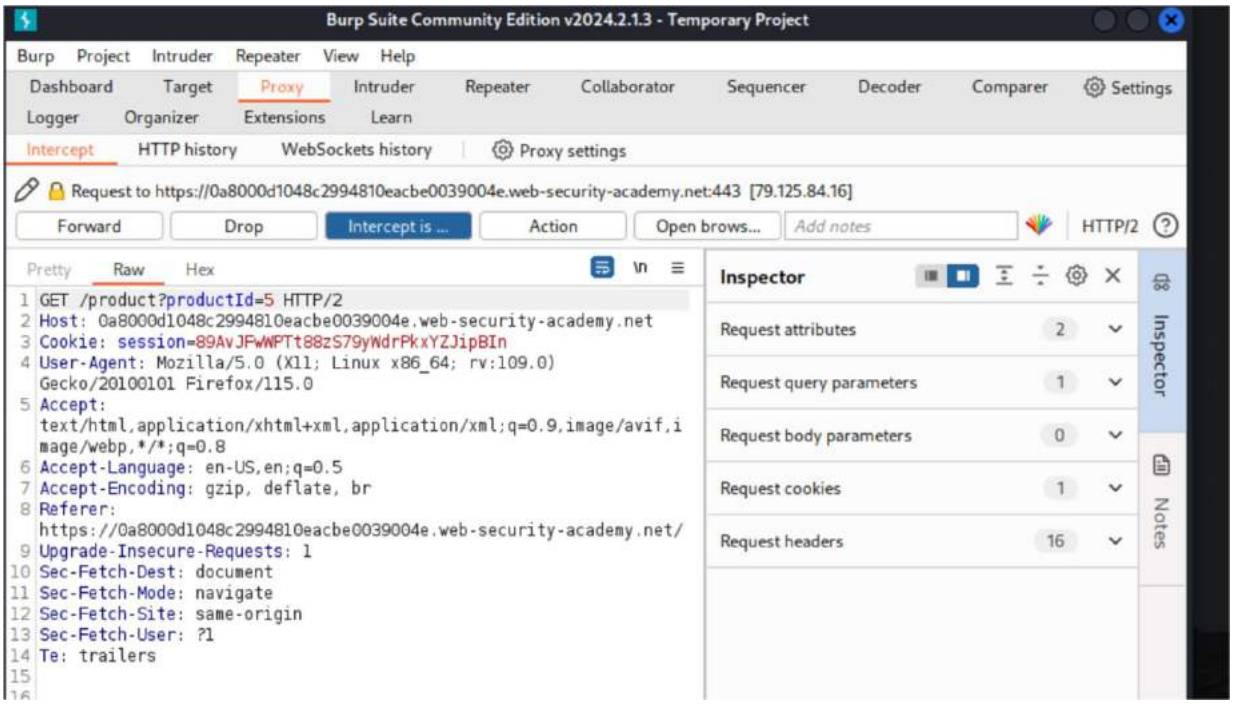
1. **Error Triggering**: Trigger various error conditions within the application.

- Navigate through various functionalities of the web application and intentionally trigger error conditions.
- Manipulate input fields or perform actions that may lead to the generation of error messages by the server.



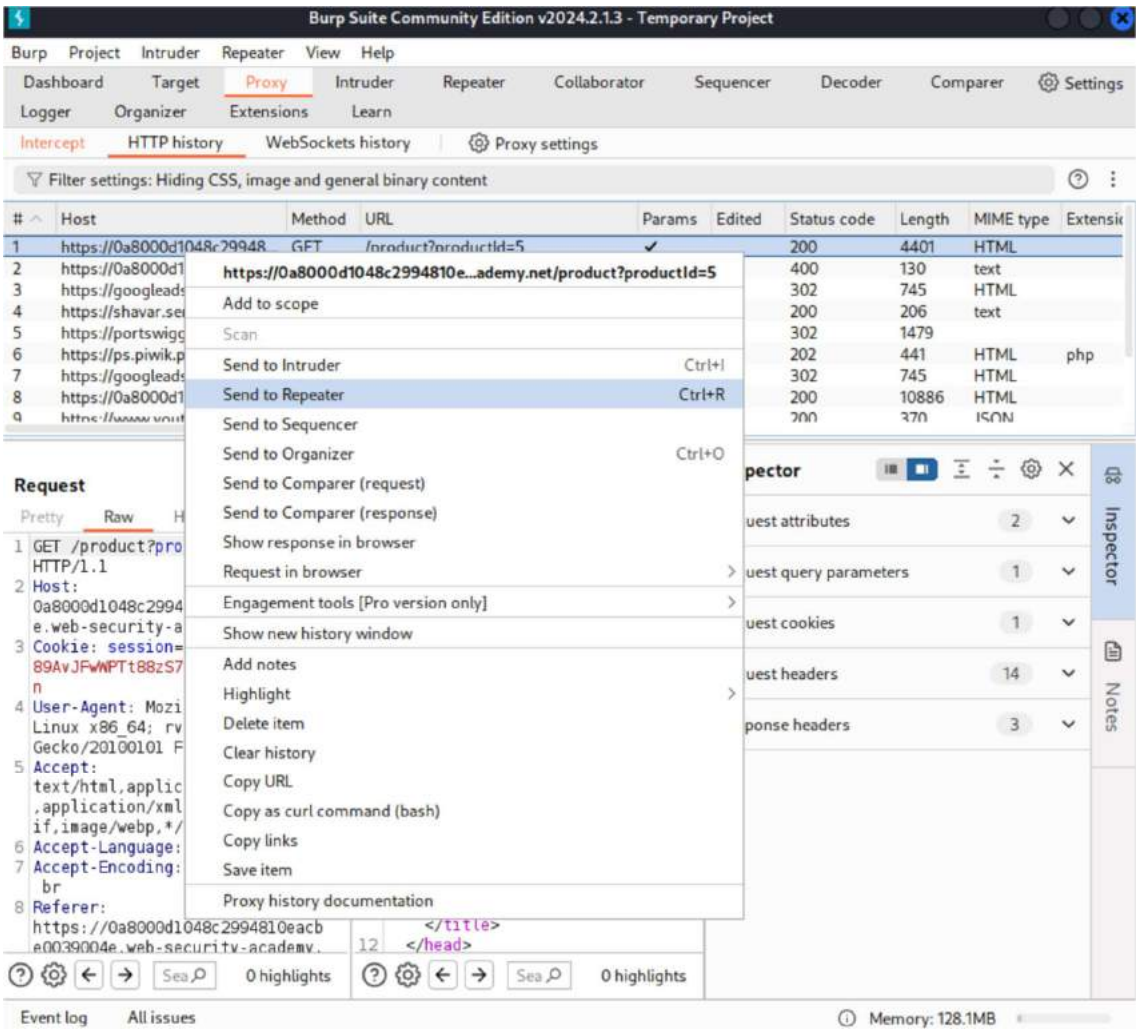
2. **Response Interception**:

- Use Burp's Proxy tool to intercept the server's responses to the error-triggering requests.
- Ensure that Burp Suite is configured to intercept both HTTP and HTTPS traffic.

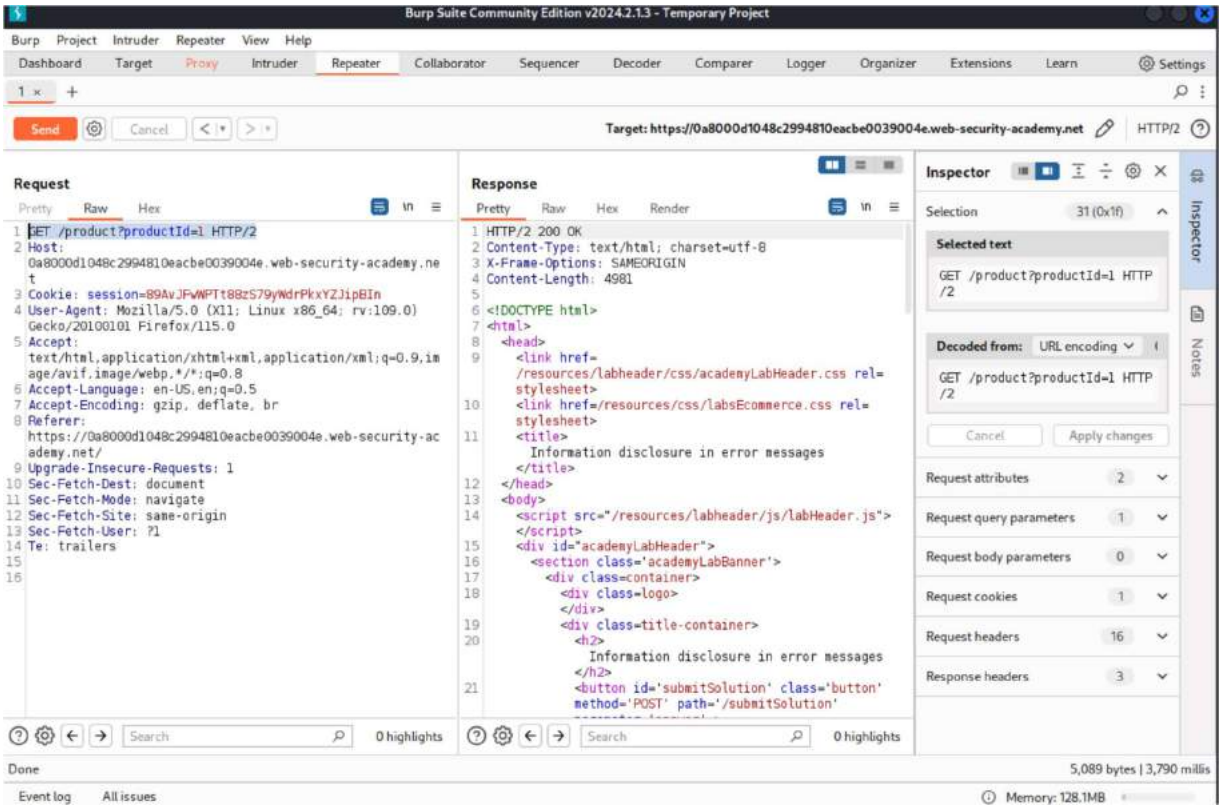


3. **Error Message Analysis**: Analyze the error messages for any hints or direct exposure of sensitive information.

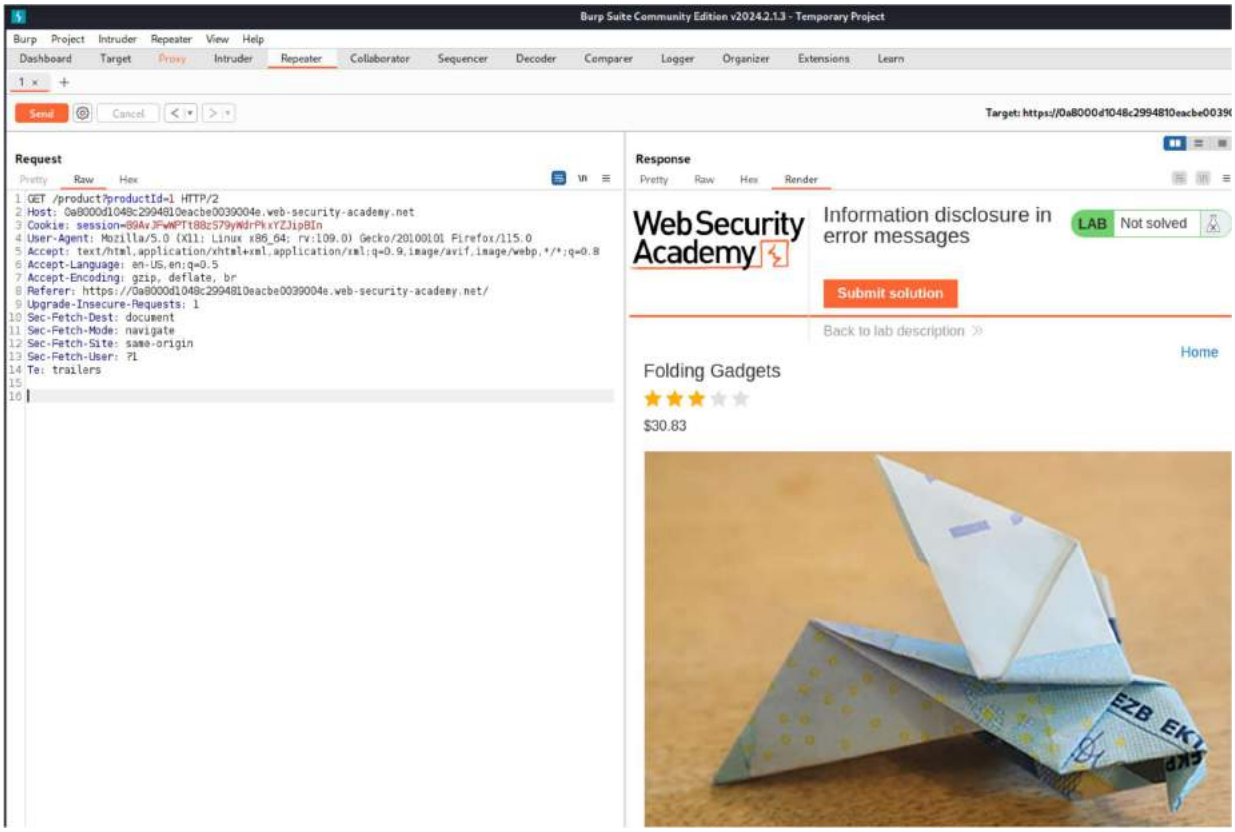
    - Look for details such as stack traces, file paths, database query fragments, or other sensitive data that may be inadvertently disclosed.



4. **Input Testing**: Test different input scenarios to see if error messages change based on the type of input.



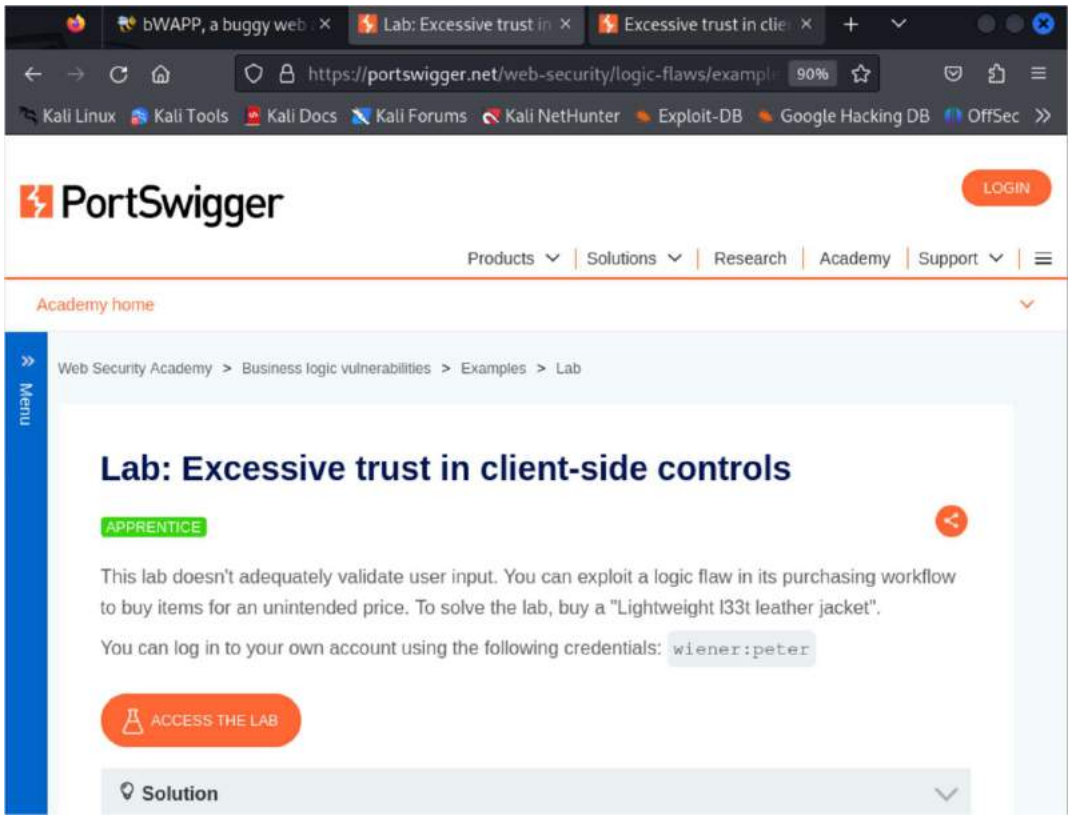
5. **Documentation**: Document any instances of information disclosure and recommend improvements in error handling.

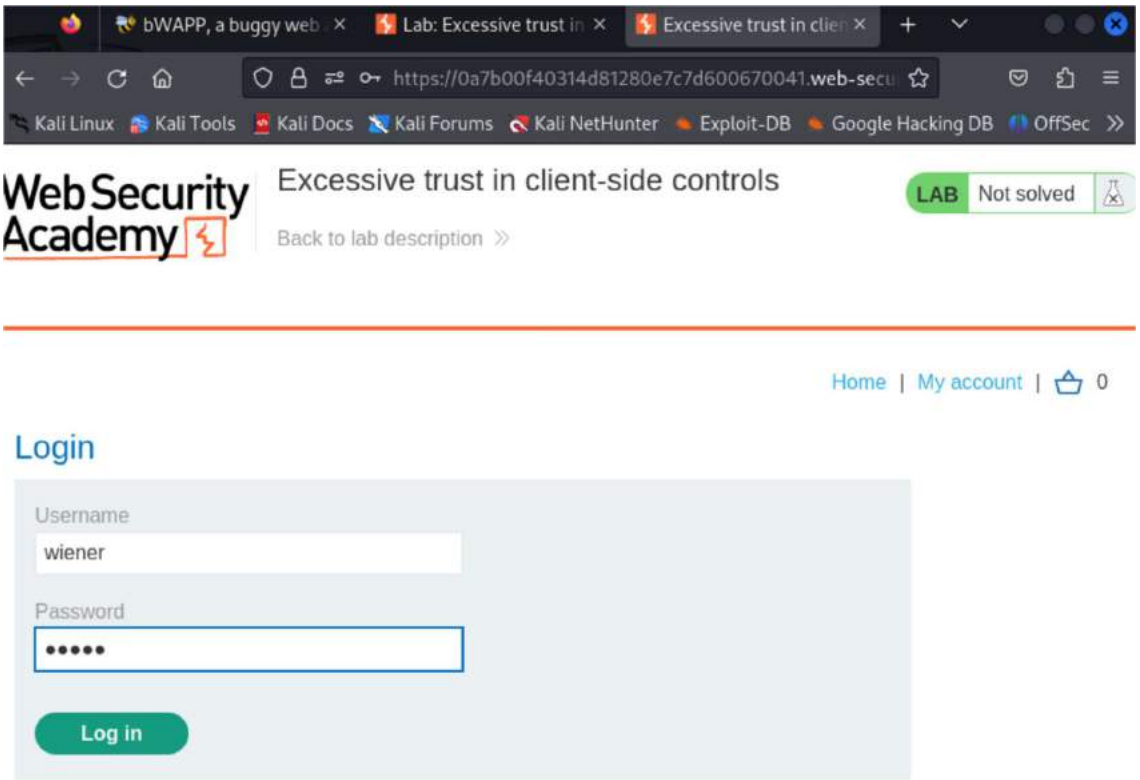


10. **HTTP Header Manipulation**:

Scenario Question: How can Burp Suite be utilized to assess how a web application manages different HTTP headers?

Scenario: We aim to evaluate how the web application handles various HTTP headers, specifically focusing on the lab located at https://portswigger.net/web-security/logic-flaws/examples/lab-logic-flaws-excessive-trust-in-client-side-controls





bWAPP, a buggy web × | Lab: Excessive trust in × | Excessive trust in clie ×

https://0a7b00f40314d81280e7c7d600670041.web-security-

Kali Linux | Kali Tools | Kali Docs | Kali Forums | Kali NetHunter | Exploit-DB | Google Hacking DB | OffSec

Web Security Academy

Excessive trust in client-side controls

LAB Not solved

Back to lab description »

Store credit:
$100.00

Home | My account | 0 | Log o

My Account

Your username is: wiener

Email

Update email

https://0a7b00f40314d81280e7c7d600670041.web-security-academy.net

Kali Linux | Kali Tools | Kali Docs | Kali Forums | Kali NetHunter | Exploit-DB | Google Hacking DB | OffSec | Remot

Web Security Academy

Excessive trust in client-side controls

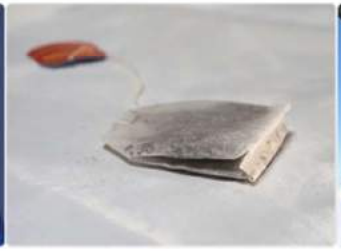
Back to lab description »

Store credit:
$100.00

Home | M

WE LIKE TO SHOP

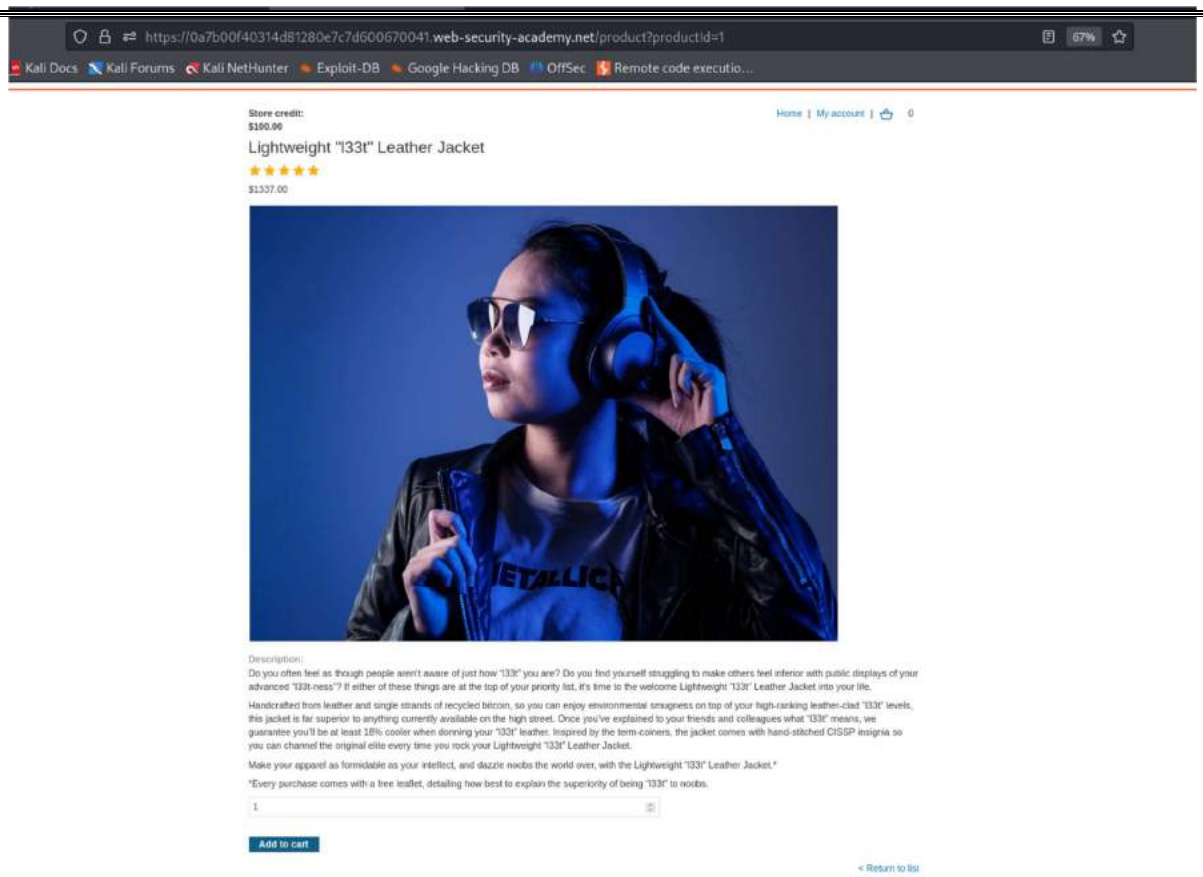
Lightweight "l33t" Leather Jacket
$1337.00
View details



Waterproof Tea Bags
$63.90
View details

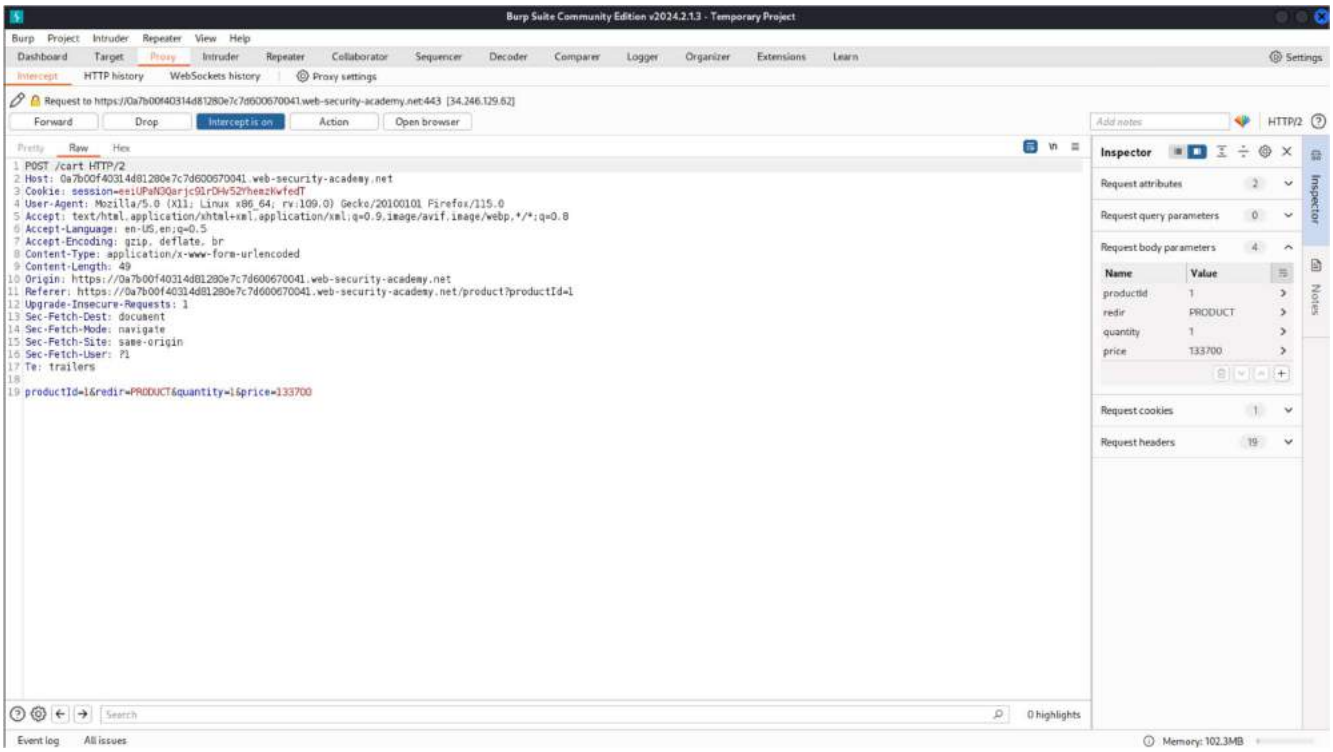
Hitch A Lift
$72.64
View details

Adult Space Hopp
Vie

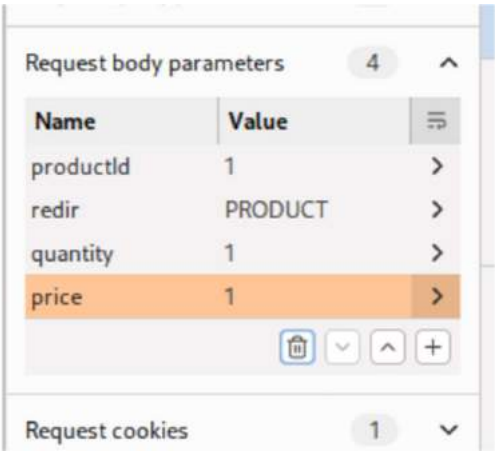


Steps:

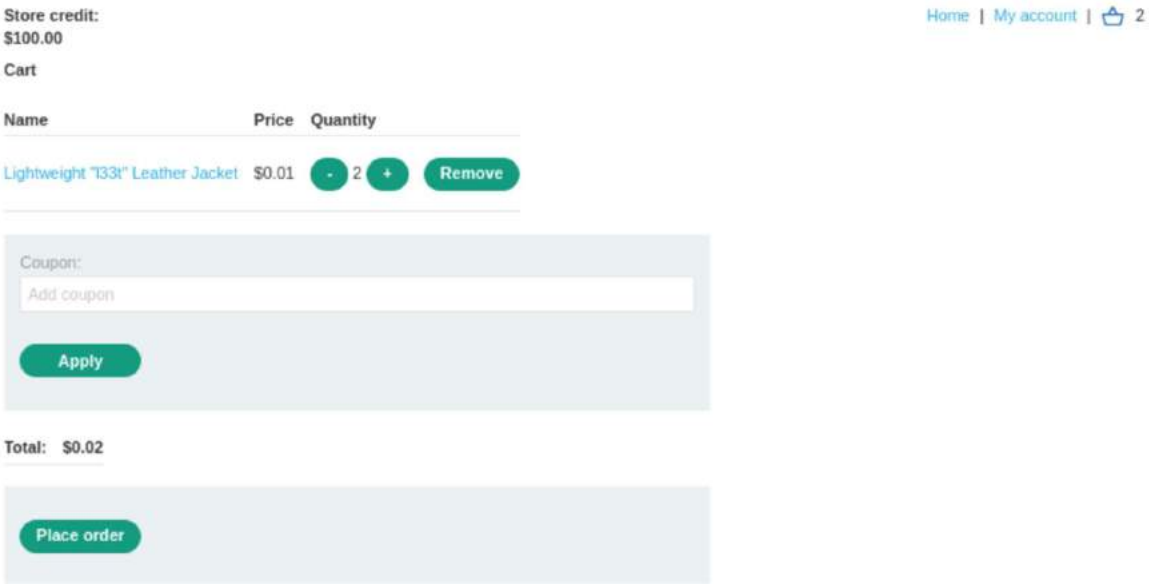
1. **Proxy Setup:**
    - Configure Burp Suite as a proxy and ensure interception is enabled.
    - Set up your browser to use Burp Suite as a proxy to intercept requests and responses.



2. **Interception and Inspector:**
    - Navigate to the target web application's URL provided in the scenario.
    - Perform various actions within the application while Burp Suite intercepts the traffic.
    - Observe the HTTP headers in both requests and responses to understand their handling by the application.



3. **Header Modification:**
    - Use Burp Suite to modify existing HTTP headers or inject new headers into intercepted requests.
    - Experiment with different header values and combinations to assess the application's response.



4. **Behavior Analysis:**
    - Analyze how the application behaves in response to modified HTTP headers.
    - Look for any changes in functionality, security controls, or application behavior triggered by manipulated headers.

5. **Security Mechanisms Testing:**
    - Test for the presence and effectiveness of security mechanisms such as HTTP Strict Transport Security (HSTS) or Content Security Policy (CSP).
    - Manipulate headers related to security controls to evaluate their impact on the application's security posture.
6. **Vulnerability Reporting:**
    - Document any vulnerabilities or weaknesses discovered during the testing process.
    - Provide recommendations for improving the application's handling of HTTP headers to enhance security and mitigate potential risks.

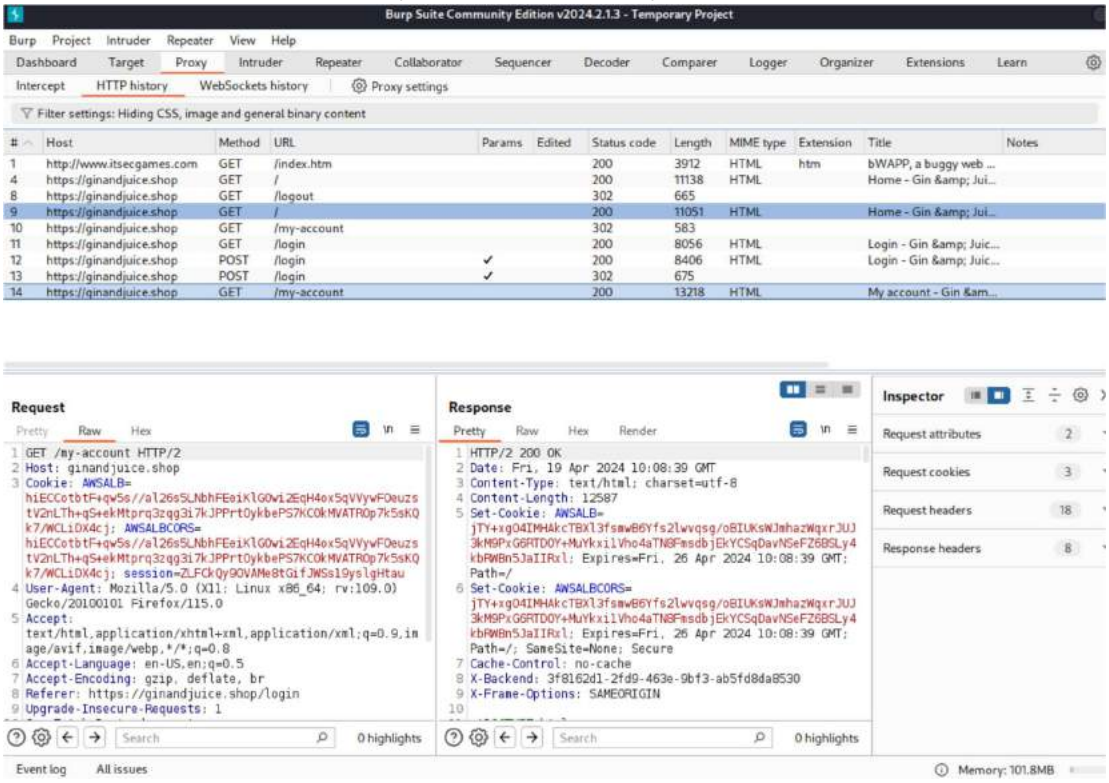
11. Determining the session timeout using Burp Suite

Scenario Question: How can Burp Suite be employed to determine the session timeout of a web application?

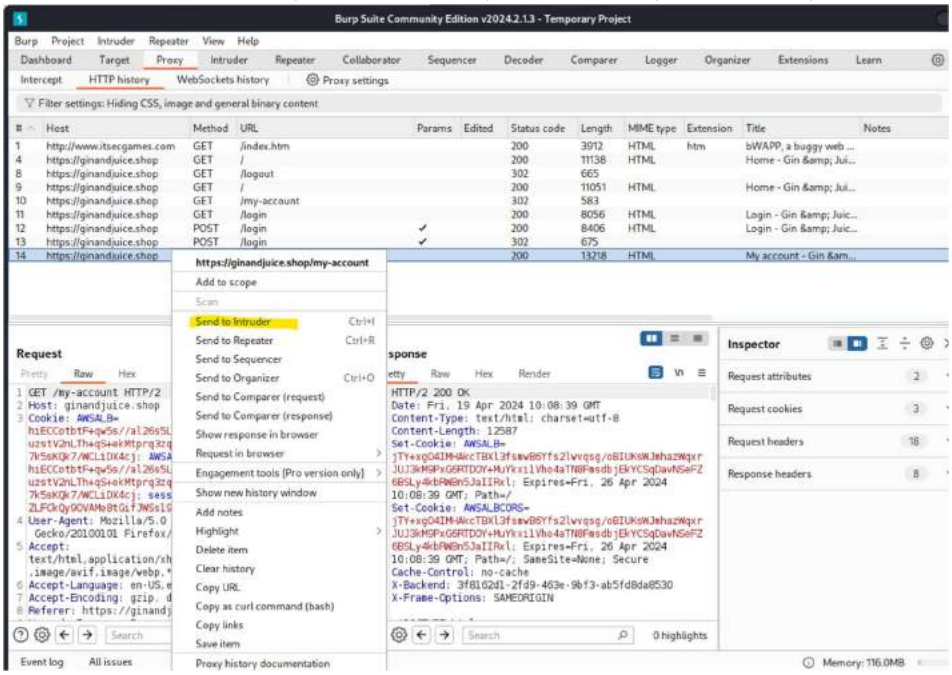
Scenario: We aim to determine the session timeout of the web application located at https://ginandjuice.shop/.

Steps:

1. **Proxy Setup:**
   - Launch Burp Suite and configure it as a proxy.
   - Ensure that interception is enabled in Burp Suite.

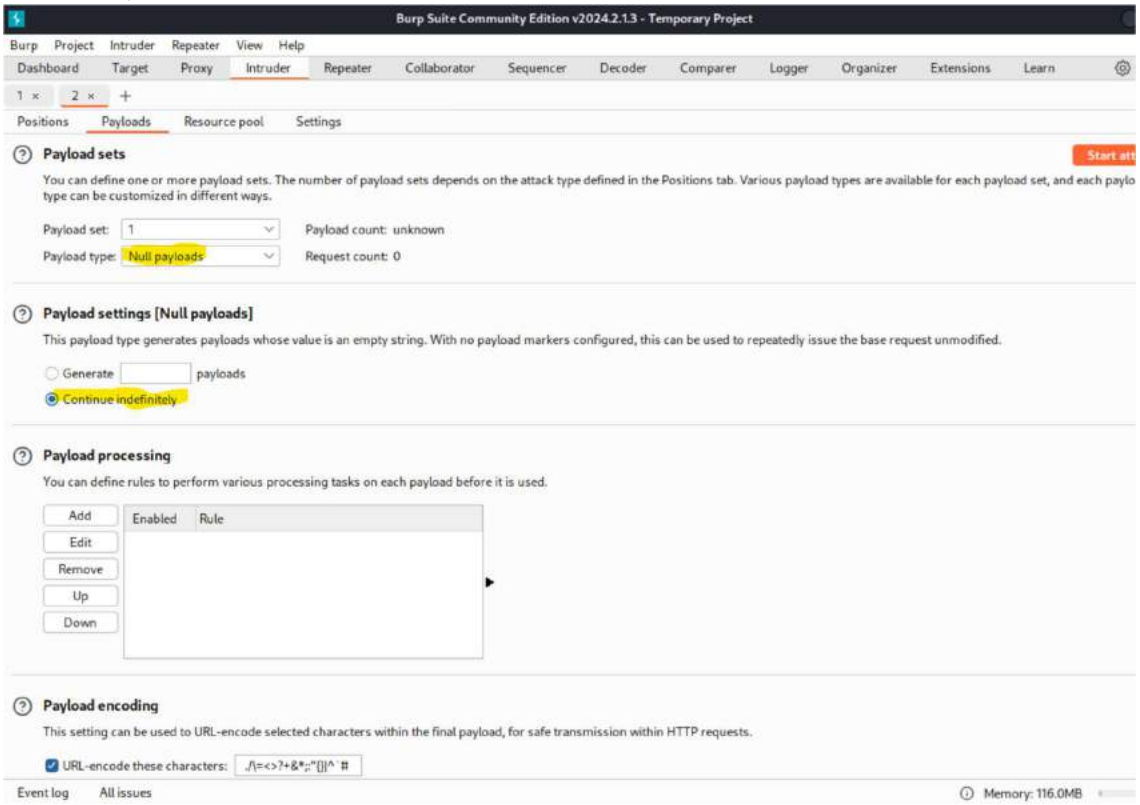


2. **Interception and Observation:**
   - Access the web application at https://ginandjuice.shop/ through your browser.
   - Allow Burp Suite to intercept the HTTP requests and responses.



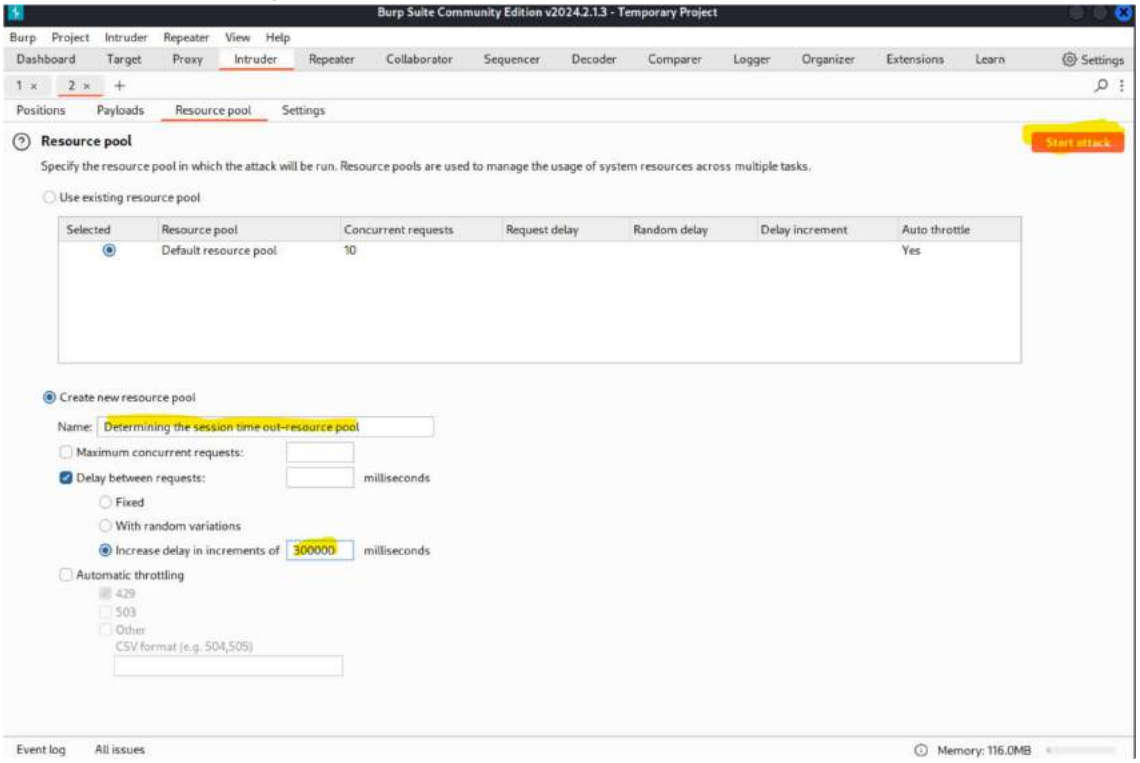
3. **Payload Injection:**

- Utilize Burp Suite's Intruder tool to perform automated session timeout testing with payloads.
- Configure the Intruder tool to send a series of requests with different payloads targeting session-related parameters.



4. **Session Timeout Testing with Payloads:**

- Execute the Intruder attack to send a sequence of requests with payloads designed to manipulate session-related parameters.
- Observe the responses from the web application to determine any changes in session behavior or session timeout settings.



5. **Session Persistence Testing:**

    - Keep the application idle for a period longer than the expected session timeout duration.
    - Observe how the session-related parameters change or if the session is terminated after the timeout duration.

6. **Documentation and Reporting:**

    - Document the observed session timeout behavior, including the duration of inactivity required for session expiration.
    - Provide recommendations for adjusting the session timeout settings, if necessary, to align with security best practices and user experience requirements.