



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

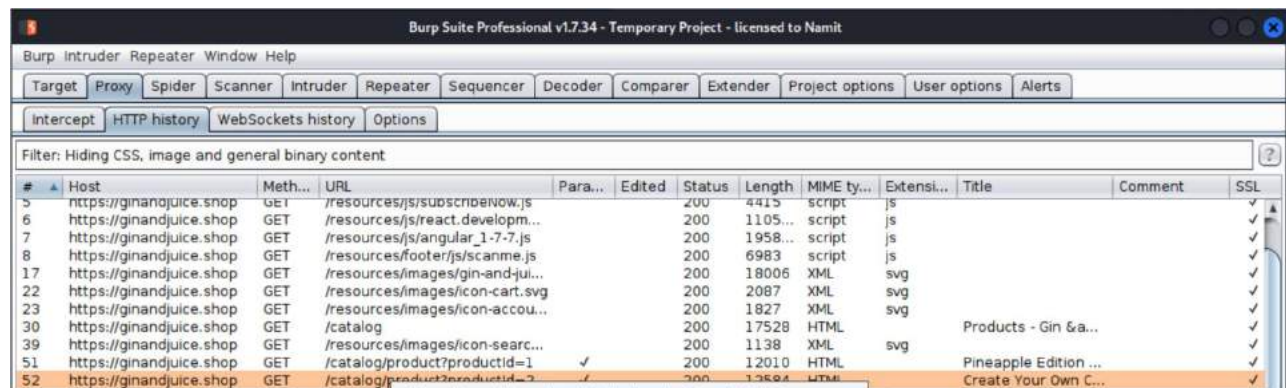
After documenting each feature of the tool and explaining its functionality in detail under experiment 3, provide a minimum of 6 to 10 scenarios (use case) where those functionality's will be used to solve the problems in Information Security Management along with detailed steps for each scenario.

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.



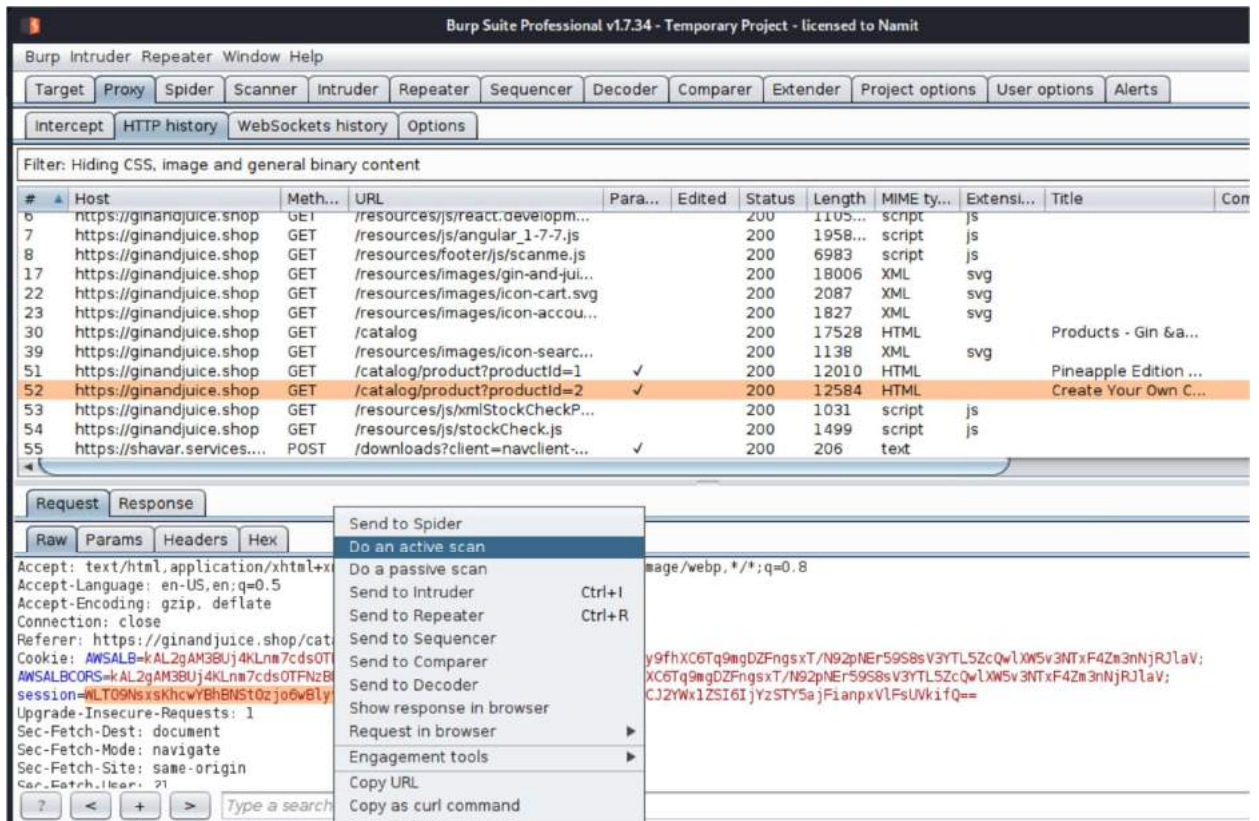
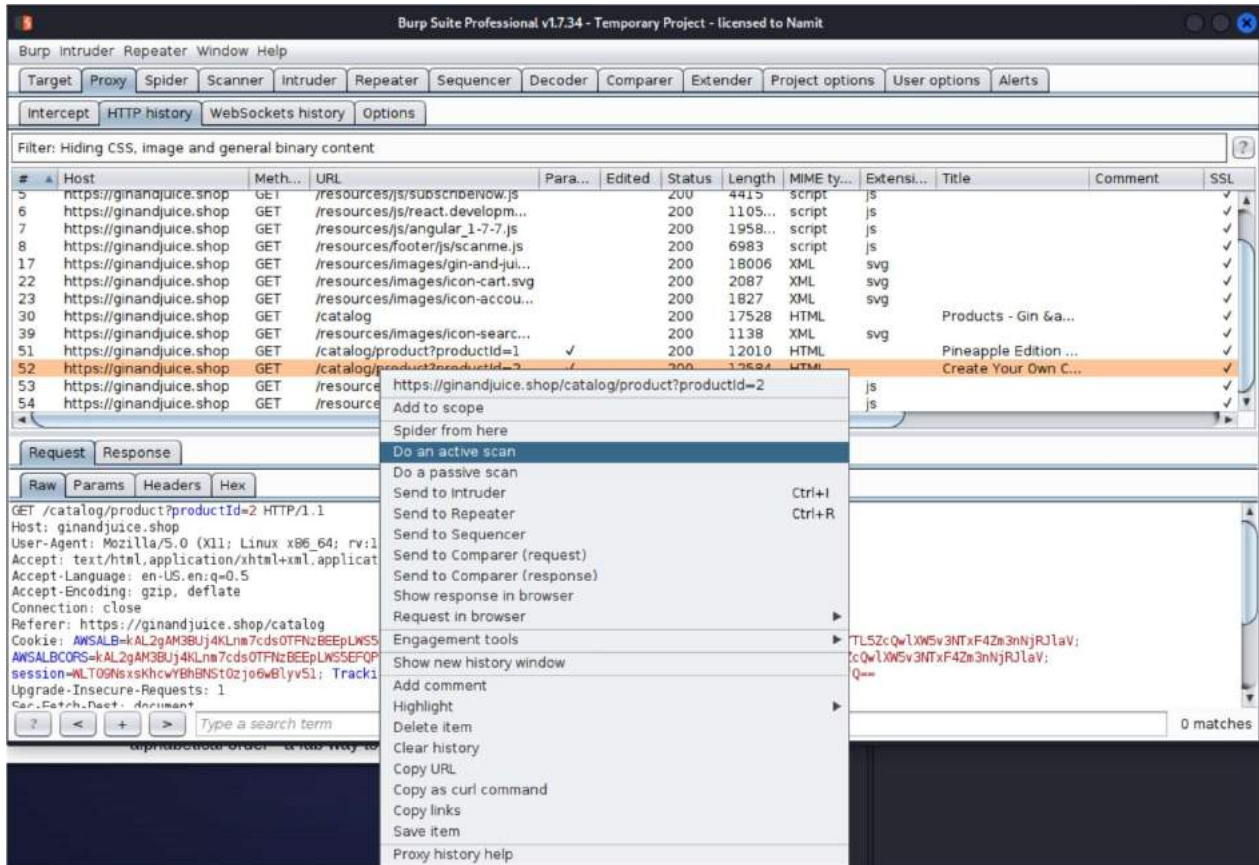
3. Proxy Tab: In Burp Suite, navigate to the Proxy tab and observe the requests/responses.



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.

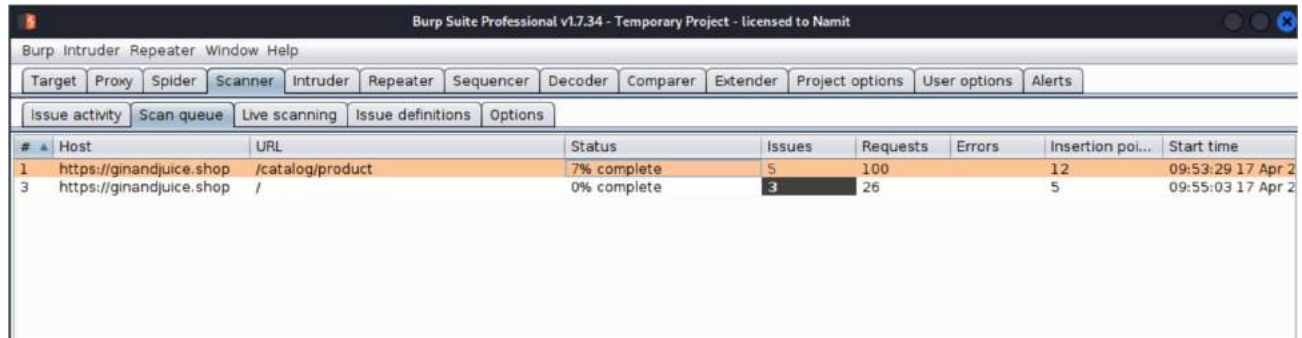


Table with 8 columns: #, Host, URL, Status, Issues, Requests, Errors, Insertion poi..., Start time.

#	Host	URL	Status	Issues	Requests	Errors	Insertion poi...	Start time
1	https://ginandjuice.shop	/catalog/product	7% complete	5	100	12		09:53:29 17 Apr 2
3	https://ginandjuice.shop	/	0% complete	3	26	5		09:55:03 17 Apr 2

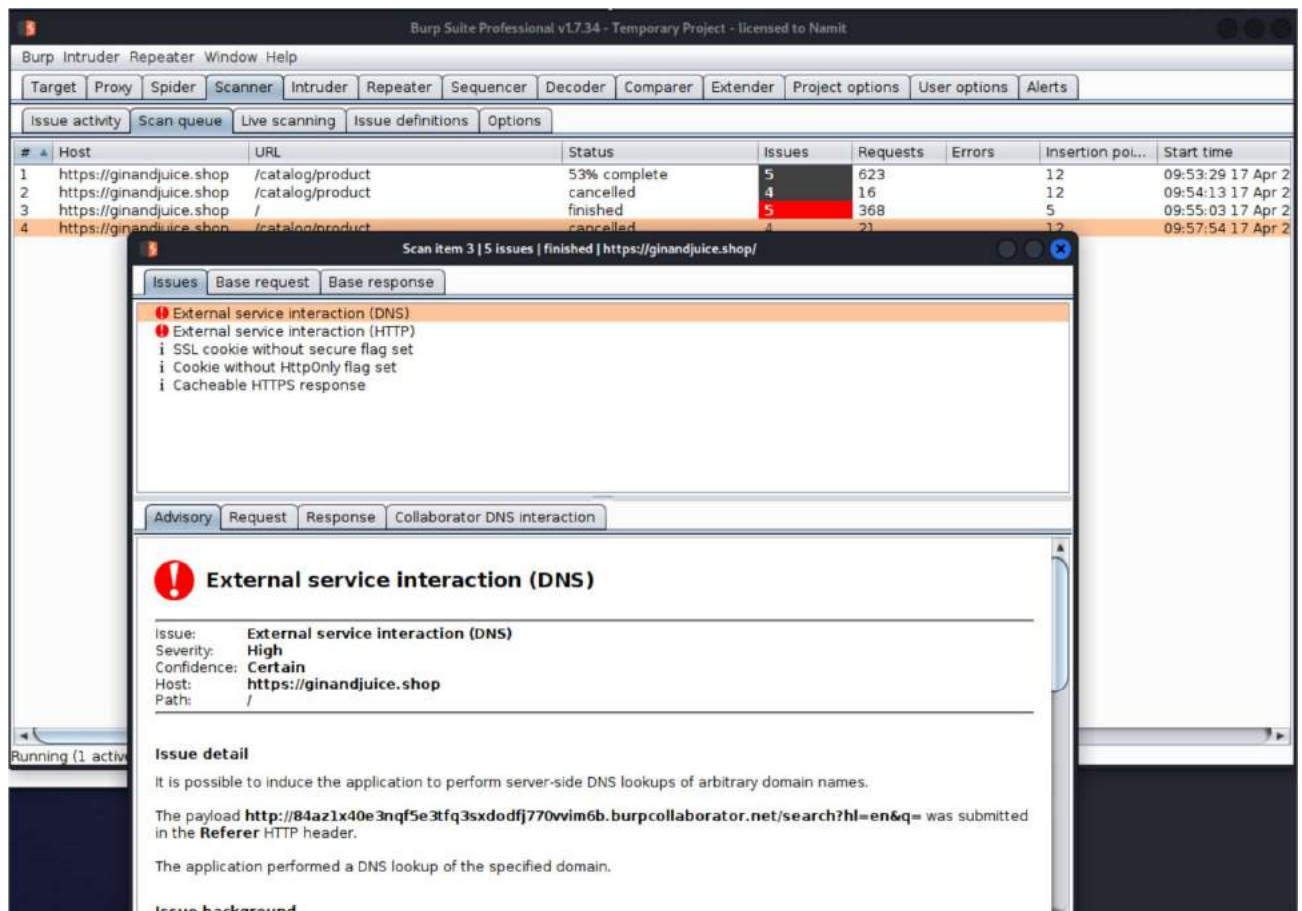


Table with 8 columns: #, Host, URL, Status, Issues, Requests, Errors, Insertion poi..., Start time.

#	Host	URL	Status	Issues	Requests	Errors	Insertion poi...	Start time
1	https://ginandjuice.shop	/catalog/product	53% complete	5	623	12		09:53:29 17 Apr 2
2	https://ginandjuice.shop	/catalog/product	cancelled	4	16	12		09:54:13 17 Apr 2
3	https://ginandjuice.shop	/	finished	5	368	5		09:55:03 17 Apr 2
4	https://ginandjuice.shop	/catalog/product	cancelled	4	21	12		09:57:54 17 Apr 2

Scan Item 3 | 5 Issues | finished | https://ginandjuice.shop/

Issues Base request Base response

- External service interaction (DNS)
- External service interaction (HTTP)
 - SSL cookie without secure flag set
 - Cookie without HttpOnly flag set
 - Cacheable HTTPS response

Advisory Request Response Collaborator DNS interaction

External service interaction (DNS)

Issue: External service interaction (DNS)
Severity: High
Confidence: Certain
Host: https://ginandjuice.shop
Path: /

Issue detail

It is possible to induce the application to perform server-side DNS lookups of arbitrary domain names.

The payload `http://84az1x40e3nqf5e3tfq3sxdodfj770wvim6b.burpcollaborator.net/search?hl=en&q=` was submitted in the **Referer** HTTP header.

The application performed a DNS lookup of the specified domain.

Issue background

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.

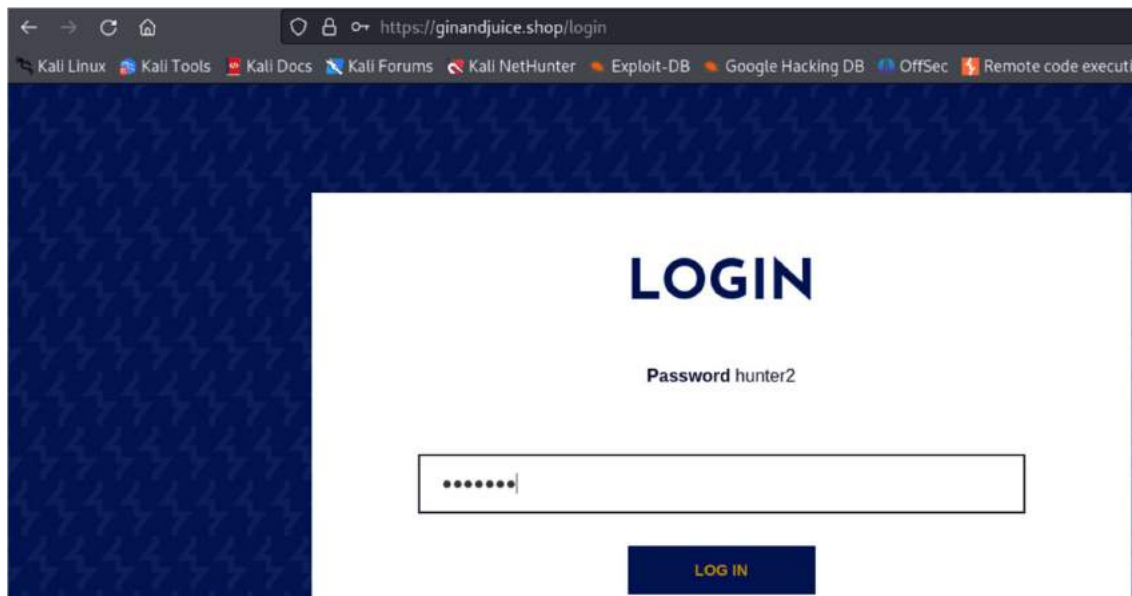
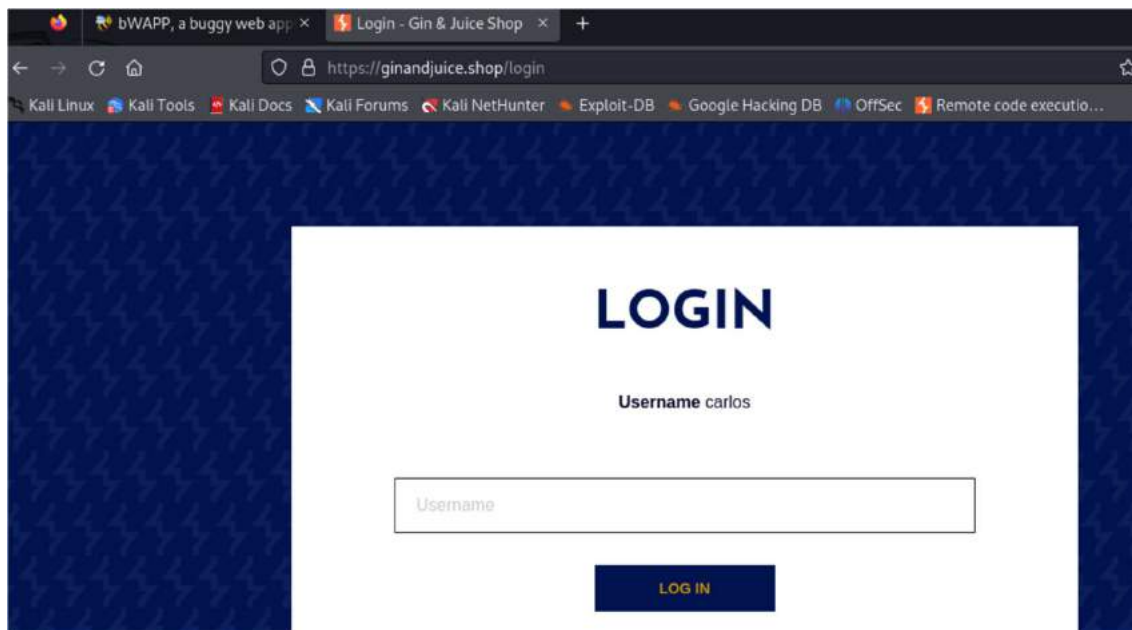
Scenario: We evaluate the effectiveness of authentication and session management mechanisms on the website <https://ginandjuice.shop/> to protect user accounts and sensitive data.

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

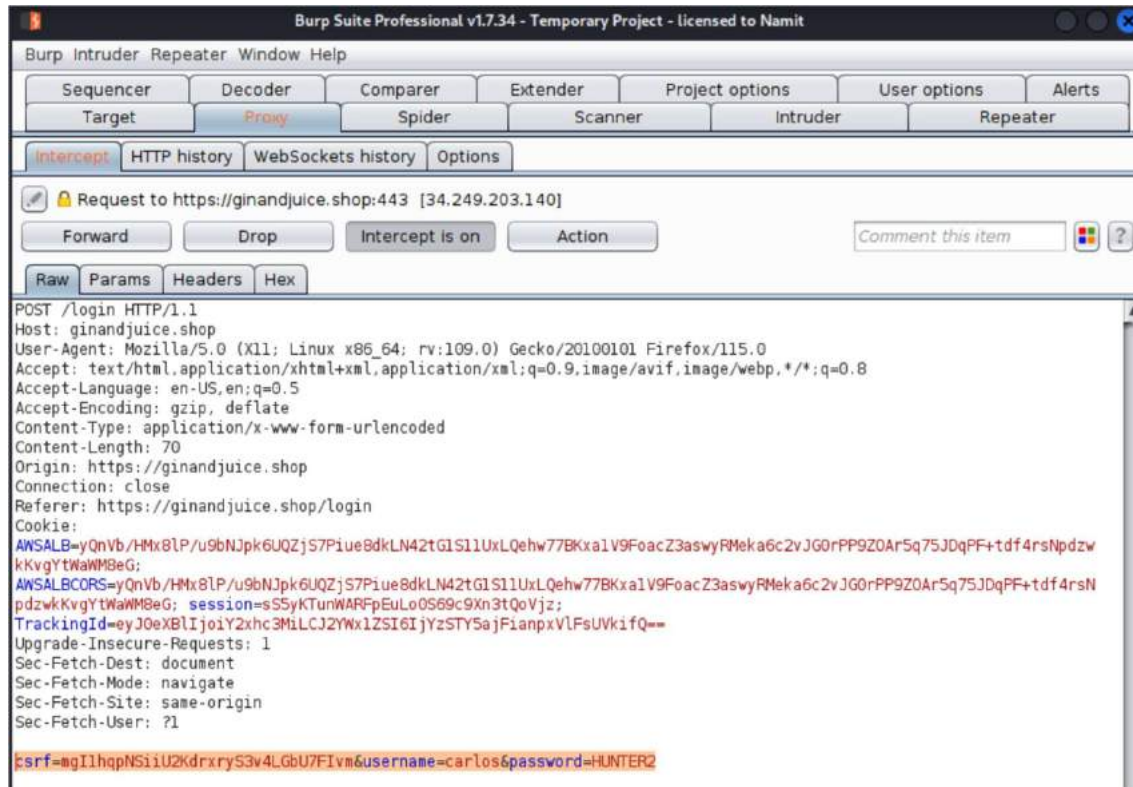
Steps:

1. Capture Login Requests:

- Launch Burp Suite and configure it as a proxy.
- Navigate to <https://ginandjuice.shop/> and initiate the login process.

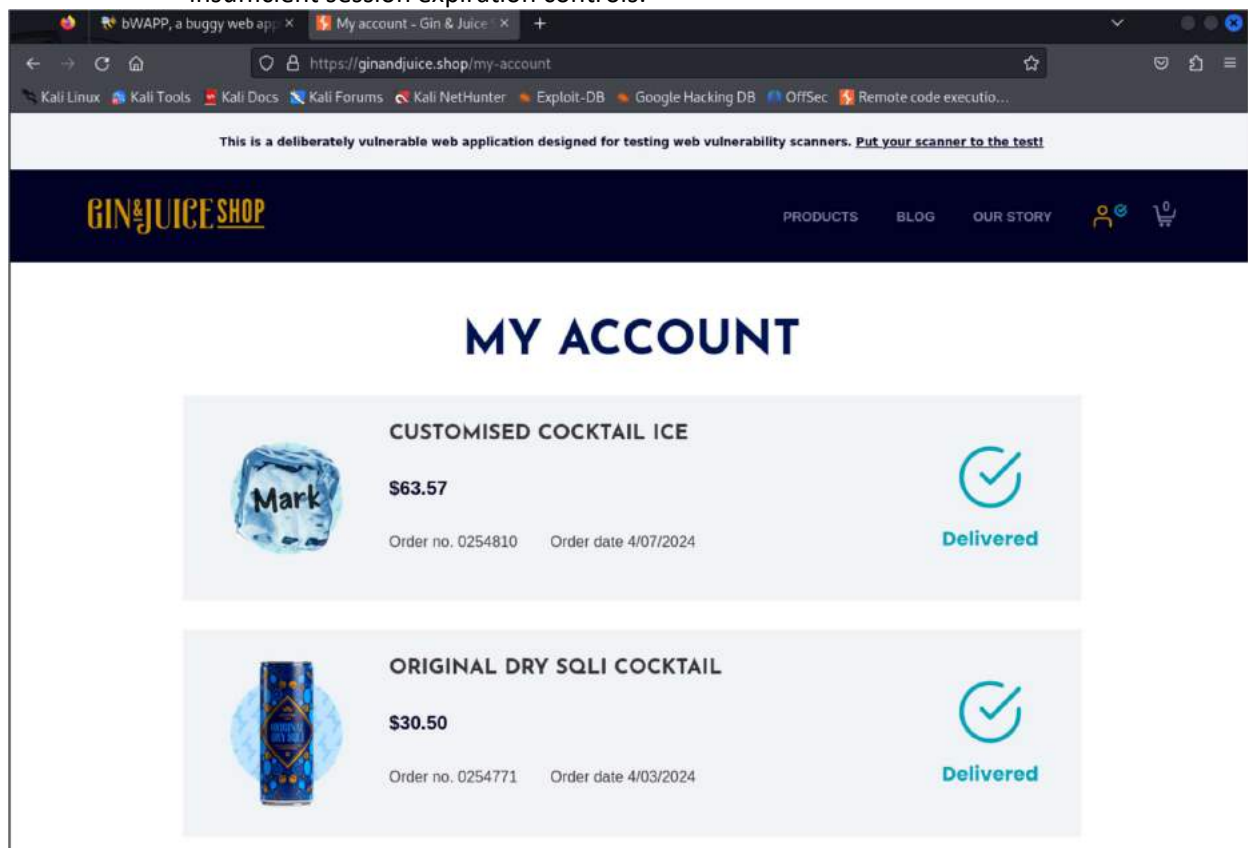


- Use Burp Suite's "Proxy" tool to capture the login requests and subsequent authenticated sessions.



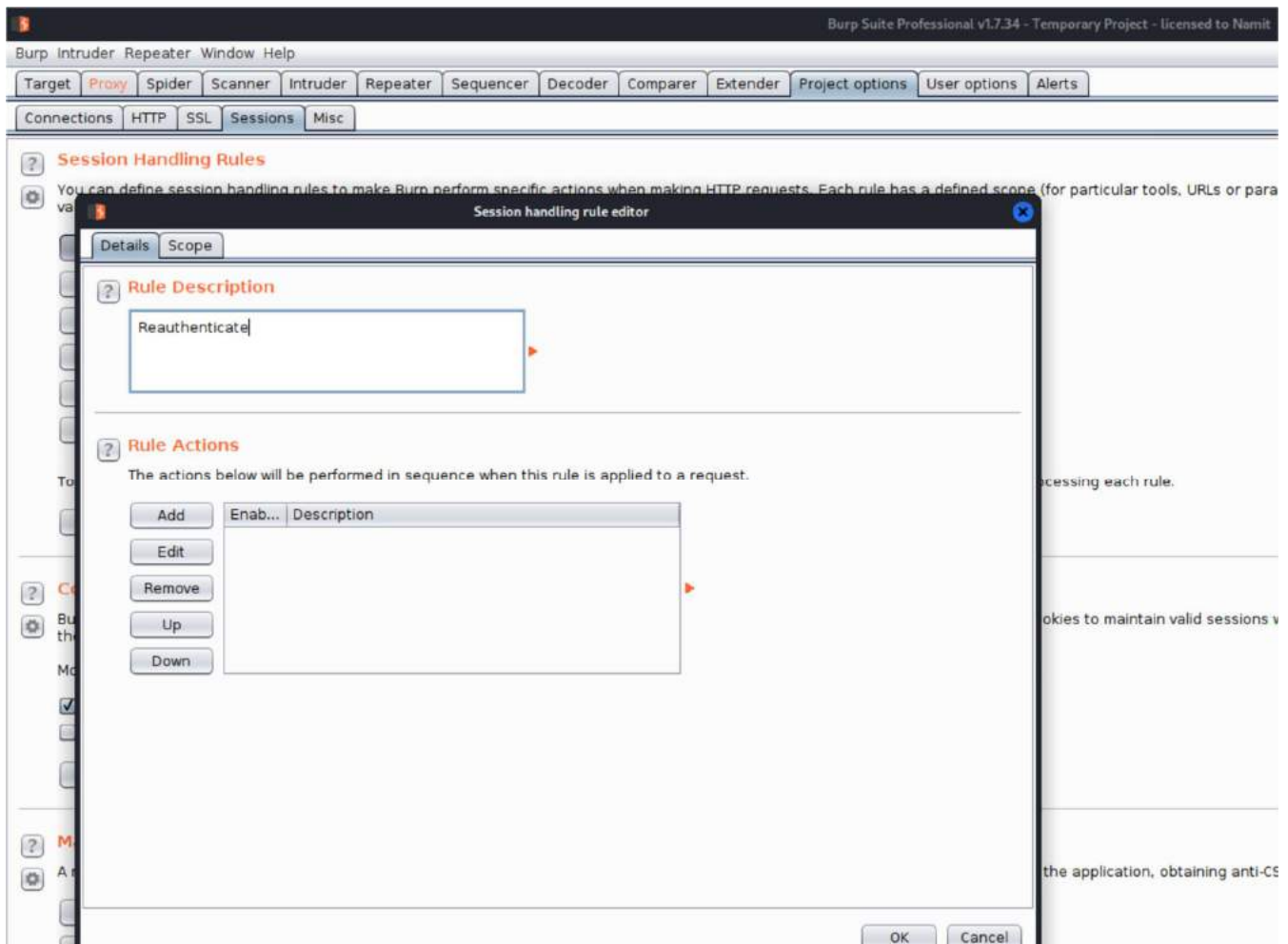
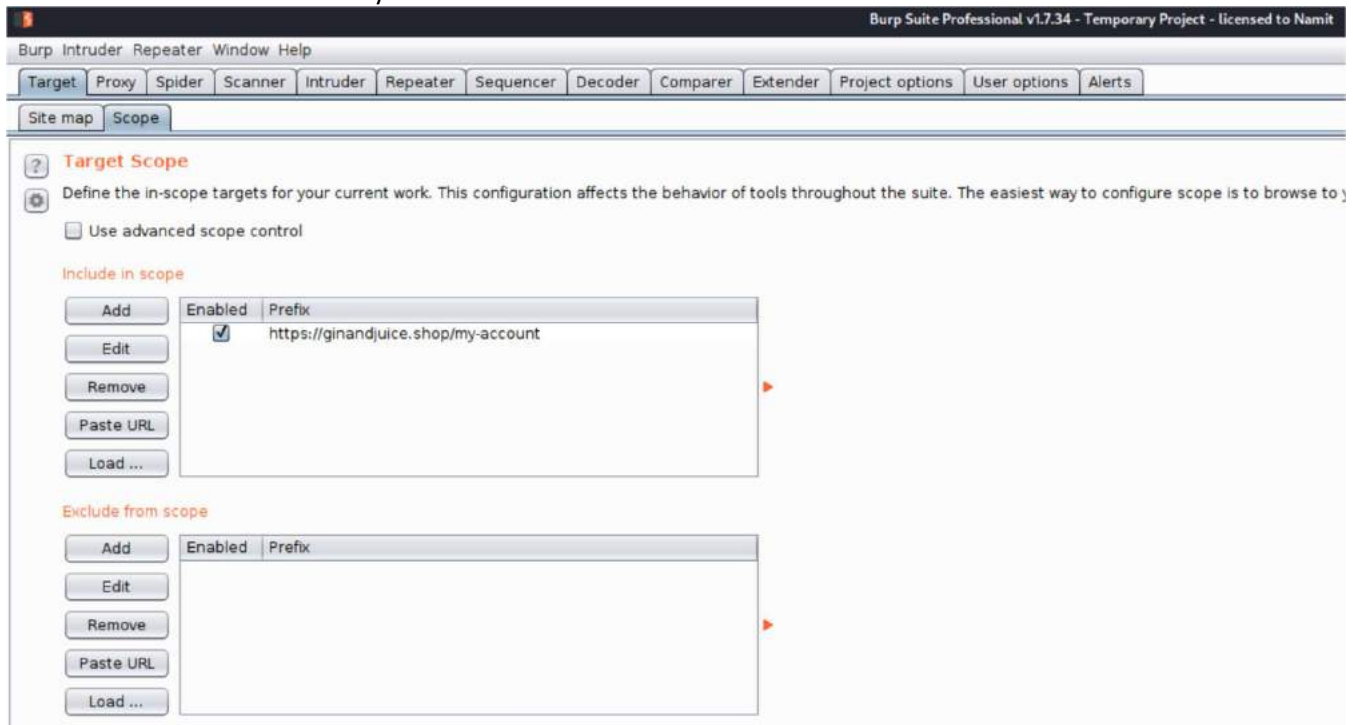
2. Analyze Authentication Process:

- 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.

☒ Target

☒ Scanner

☒ Repeater

☒ Spider

☒ Intruder

☒ 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 ☒ 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:

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

☒ 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.

☒ 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 CSRF 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=f1KVBHUP3ajaghcWDBXvYGFbHETOfzkfcVXqL4/Y5oYwUoI6xMuKAZSHpJtmqRgvBvcPOS4WwdGINLmwOt4wQR8k6vggSP19HdNji7rtzcK0
Cy50t+tccthlV30;
AWSALB=...
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

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=gVpXglIWxerlFfQ+D8C/z214bRtj5VHsAQu8NvgbGzAc4tzsG17fSBdQ2dv3Qm1jDCFYyUy+Dmq6TTDI1P221OrZc99eXbwPurLwsVKY7wLQsaiJ/kbYfw1ev;
AWSALBCORS=gVpXglIWxerlFfQ+D8C/z214bRtj5VHsAQu8NvgbGzAc4tzsG17fSBdQ2dv3Qm1jDCFYyUy+Dmq6TTDI1P221OrZc99eXbwPurLwsVKY7wLQsaiJ/kbYfw1ev;
Type a search term 0 matches

Configure item

Move up

Move down

Remove item

Re-record macro

Re-analyze macro

Test macro

OK Cancel

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

1 Burp Intruder Repeater Window Help

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

Connections HTTP SSL Sessions Misc

? Session Handling Rules

⚙️ You can define session handling rules to make Burp perform specific actions when making HTTP requests. Each rule has a defined scope (for particular tools, URLs or parts of the request). Before each request is issued, Burp applies in sequence each of the rules that are in-scope for the request.

Enabled	Description	Tools
<input checked="" type="checkbox"/>	Use cookies from Burp's cookie jar	Spider and Scanner
<input checked="" type="checkbox"/>	Reauthenticate	Target, Spider, Scanner, Intruder, Repeater

Buttons: Add, Edit, Remove, Duplicate, Up, Down

To monitor or troubleshoot the behavior of your session handling rules, you can use the sessions tracer to view in detail the results of processing each rule.

Open sessions tracer

4. Manipulate Parameters:

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

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

1 Burp Intruder Repeater Window Help

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

1 ...

Go Cancel < | * > | *

Request

Raw Params Headers Hex

```
GET /my-account 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/login
Connection: close
Cookie:
ANSALB=A54LtbHn8qg6cquUixiPyL15BoMIzq2myGLEYmH8vXmHSoRETijbo8IIJ2UHv+qLB+hzAaFA/ybpf1MCMXQfRi9JlueLhxECci8urN/aENvmYZPdU8+RXauL4Yl0;
ANSALBCORS=A54LtbHn8qg6cquUixiPyL15BoMIzq2myGLEYmH8vXmHSoRETijbo8IIJ2UHv+qLB+hzAaFA/ybpf1MCMXQfRi9JlueLhxECci8urN/aENvmYZPdU8+RXauL4Yl0;
session=L0LytcUuxEjdvdELI3MDaSn3Krk0K9b;
TrackingId=eyJ0eXB1Ijo1Y2xhc3M6LCJ2YX1ZSI6IjYzSTY5ajFianpxVlFsUVkiQ==
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
```

Response

Raw

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

Target: [ginandjuice.shop](#) Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts

Go Cancel < >

Request

Raw Params Headers Hex

```
GET /my-account 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/login
Connection: close
Cookie: AWSALB=15r42PQ906pD/T1M85jC0PwH8LzrAmoeG5vU3QYk3Bx1Q2XctnRf/dy529WkC3QpK3QpM/r990G/vhs8LaruzD8Nxbjknob3D3Pah/f1CQWwIn; AWSALBCORS=15r42PQ906pD/T1M85jC0PwH8LzrAmoeG5vU3QYk3Bx1Q2XctnRf/dy529WkC3QpK3QpM/r990G/vhs8LaruzD8Nxbjknob3D3Pah/f1CQWwIn; session=QZC0QpK3QpM/r990G/vhs8LaruzD8Nxbjknob3D3Pah/f1CQWwIn; TrackingId=keyJ0nX8LIj0iY2zhc3MLCJ2YK1Z5T0IjY2Y5aJPaapxVfPaUWifo=
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
```

Response

Raw Headers Hex HTML Render

```
HTTP/1.1 200 OK
Date: Fri, 19 Apr 2024 07:46:17 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 12597
Connection: close
Set-Cookie: AWSALB=15r42PQ906pD/T1M85jC0PwH8LzrAmoeG5vU3QYk3Bx1Q2XctnRf/dy529WkC3QpK3QpM/r990G/vhs8LaruzD8Nxbjknob3D3Pah/f1CQWwIn; Expires: 07:46:17 GMT; Path=/; SameSite=None; Secure
Cache-Control: no-cache
X-Backend: 3f8362d1-2f49-463e-96f3-ab5fd8da8530
X-Frame-Options: SAMEORIGIN

<!DOCTYPE html>
<html>
  <head>
    <link href=/resources/labheader/css/scanHeader.css rel=stylesheet>
    <link href=/resources/css/labsScanee.css rel=stylesheet>
    <meta name=viewport content=width=device-width, user-scalable=no>
    <script src=/resources/js/react-devs.development.js></script>
    <script type=text/javascript src=/resources/js/angular_1-7-7.js></script>
    <title>My account - Gin & Juice Shop</title>
  </head>
  <body ng-app>
    <div id=scanHeader>
      <section class=header-description>
        <p>
          This is a deliberately vulnerable web application designed for testing web vulnerability scanners. Put your scanner to the test!
        </p>
      </section>
      <section class=scanHeader>
        <div class=container>
          <a href=/>
            <div class=scanee-logo></div>
          </a>
          <div class=title-container>
            <nav>
              <ul class=navigation-header-links primary-links>
                <li>
                  <a class=button href=/catalog>Products</a>
                </li>
                <li>
                  <a class=button href=/blog>Blog</a>
                </li>
              </ul>
            </nav>
          </div>
        </div>
      </section>
    </div>
  </body>
</html>
```

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.

Response


Raw Headers Hex HTML Render

This is a deliberately vulnerable web application designed for testing web vulnerability scanners. Put your scanner to the test!

- Products
- Blog
- Our story
- Log out
- My account

0

My Account



13,239 bytes | 337 millis

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.

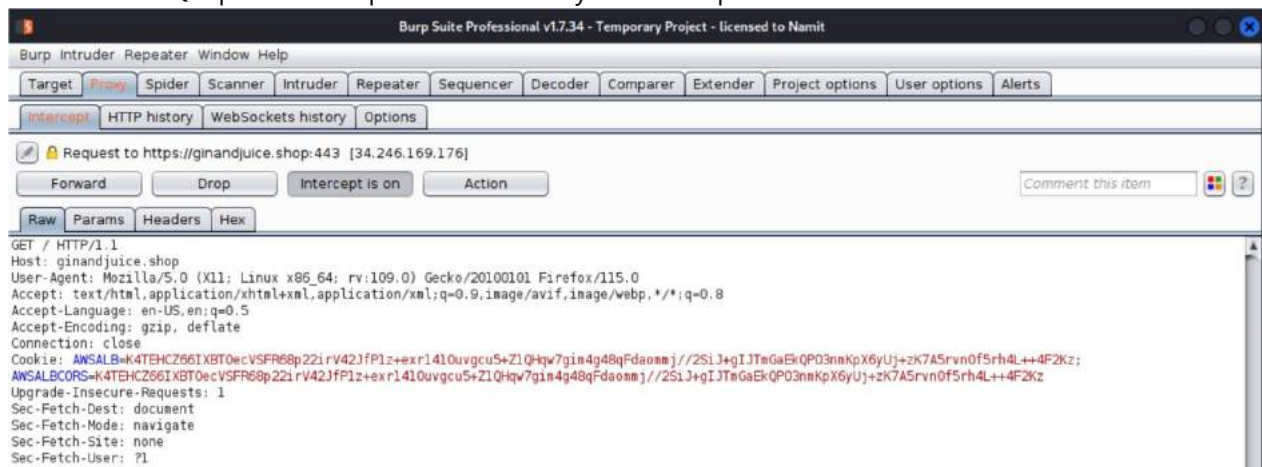
Steps:

1. Proxy Setup:

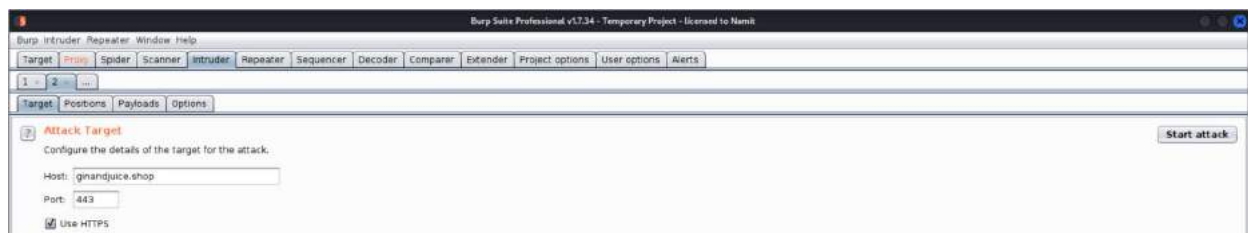
- Configure Burp Suite as a proxy and ensure interception is enabled.
- Navigate to <https://ginandjuice.shop/> using your browser.

2. Interception:

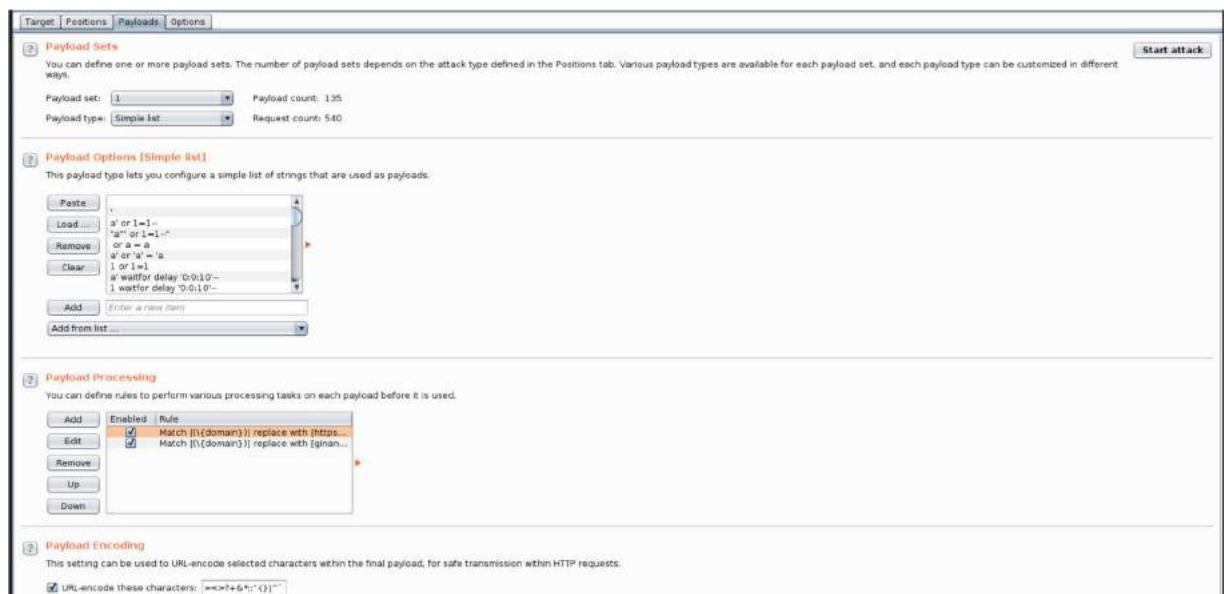
- Interact with input fields on the ginandjuice.shop website, such as search or login forms.
- Observe SQL queries in requests in the Proxy tab of Burp Suite.



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.

SQL injection [4]

- ! /Login.asp [tfUName parameter]
- ! /Login.asp [tfUName parameter]
- ! /Login.asp [tfUPass parameter]
- ! /Login.asp [tfUPass parameter]
- ! Cleartext submission of password
- ! Password field with autocomplete enabled
- ! Input returned in response (reflected)
- ! Cross-domain Referer leakage
- ! Path-relative style sheet import

Advisory

SQL injection

Issue: **SQL injection**
Severity: **High**
Confidence: **Certain**

Summary Audit items **Issues** Event log Logger Audit log Live crawl view

Filter High Medium Low Info Certain Firm Tentative BCheck generated Scan checks Extensions

Time	Source	Issue type	Host	Path
12:02:35 29 Nov 2023	Task 3	External service interaction (DNS)	https://ginandjuice.shop	/catalog
12:01:33 29 Nov 2023	Task 3	SQL injection	https://ginandjuice.shop	/catalog
11:58:48 29 Nov 2023	Task 3	Client-side template injection	https://ginandjuice.shop	/catalog
11:58:34 29 Nov 2023	Task 3	Cross-site scripting (reflected)	https://ginandjuice.shop	/catalog/subscribe

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

Intruder attack 1

Attack Save Columns

Results Target Positions Payloads Options

Filter: Showing all items

Requ...	Position	Payload	Status	Error	Time...	Length	Comment
490	4	exec xp	200			11072	
491	4	exec sp	200			11072	
492	4	'; exec master..xp_cmds...	200			11072	
493	4	'; exec xp_regread	200			11072	
494	4	t'exec master..xp_cmds...	200			11072	
495	4	--sp_password	200			11072	
496	4	\\x27UNION SELECT	200			11072	
497	4	' UNION SELECT	200			11072	
498	4	' UNION ALL SELECT	200			11072	
499	4	' or (EXISTS)	200			11072	
500	4	' (select top 1	200			11072	
501	4	' JUTL_HTTP.REQUEST	200			11072	
502	4	1;SELECT%20*	200			11072	
503	4	to timestamp...	200			11072	

Request Response

Raw Headers Hex HTML Render

Name	Value
HTTP/1.1	200 OK
Date	Wed, 17 Apr 2024 08:04:12 GMT
Content-Type	text/html; charset=utf-8
Content-Length	10445
Connection	close
Set-Cookie	AWSALB=SS/Rn3pkfWPrb7EeY5DCAZ0lqBPfIQbjqVK5Q3WxWBnJ9u6BbWkpWtHJC31d+sPOePf7Mkk+a1RrgGakU5/F14...
Set-Cookie	AWSALBCORS=SS/Rn3pkfWPrb7EeY5DCAZ0lqBPfIQbjqVK5Q3WxWBnJ9u6BbWkpWtHJC31d+sPOePf7Mkk+a1RrgGakU...
X-Backend	c7018540-747f-4a46-af86-545733e9231e
X-Frame-Options	SAMEORIGIN

<!DOCTYPE html>
<html>
 <head>
 <link href=/resources/labheader/css/scanMeHeader.css rel=stylesheet>
 <link href=/resources/css/labsScanner.css rel=stylesheet>
 <meta name="viewport" content="width=device-width, user-scalable=no">
 <script src=/resources/js/react.development.js></script>
 <script src=/resources/js/react-dom.development.js></script>
 <script type="text/javascript" src=/resources/js/angular_1-7-7.js></script>
 <title>Home - Gin & Juice Shop</title>
 </head>

0 matches

Finished

- 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.

Intruder attack 1

Attack Save Columns

Results Target Positions Payloads Options

Filter: Showing all items


Requ...	Position	Payload	Status	Error	Time...	Length	Comment
490	4	exec xp	200			11072	
491	4	exec sp	200			11072	
492	4	'; exec master..xp_cmds...	200			11072	
493	4	'; exec xp_regread	200			11072	
494	4	t'exec master..xp_cmds...	200			11072	
495	4	--sp_password	200			11072	
496	4	\x27UNION SELECT	200			11072	
497	4	' UNION SELECT	200			11072	
498	4	' UNION ALL SELECT	200			11072	
499	4	' or (EXISTS)	200			11072	
500	4	' (select top 1	200			11072	
501	4	' UTL_HTTP.REQUEST	200			11072	
502	4	1;SELECT%20*	200			11072	

Request Response

Raw Headers Hex HTML Render

[View all products](#)

Created in 2022 by the man Distiller's World has called "the evil genius of gin", Gin & Juice Shop is open 24/7 to satisfy all of your web vulnerability scanner evaluation needs.



Finished

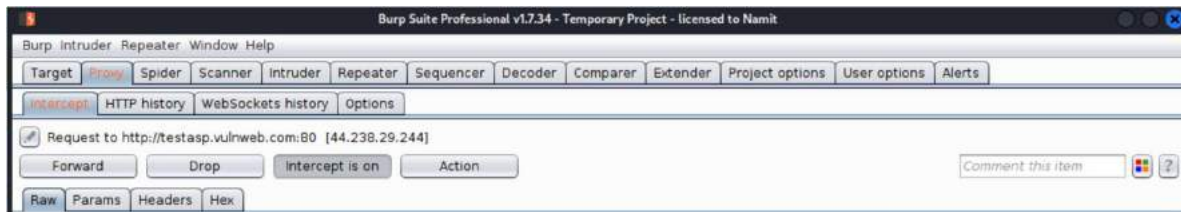
4. Session Hijacking:

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?

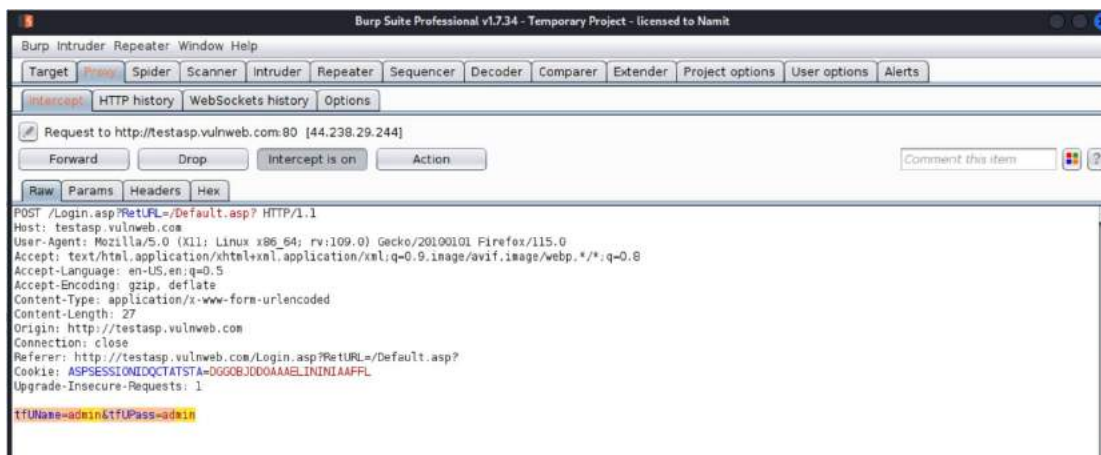
Scenario: We want to test the security of <https://testasp.vulnweb.com/Login.asp?RetURL=/Default.asp?> against session hijacking attacks.

- **Steps:**

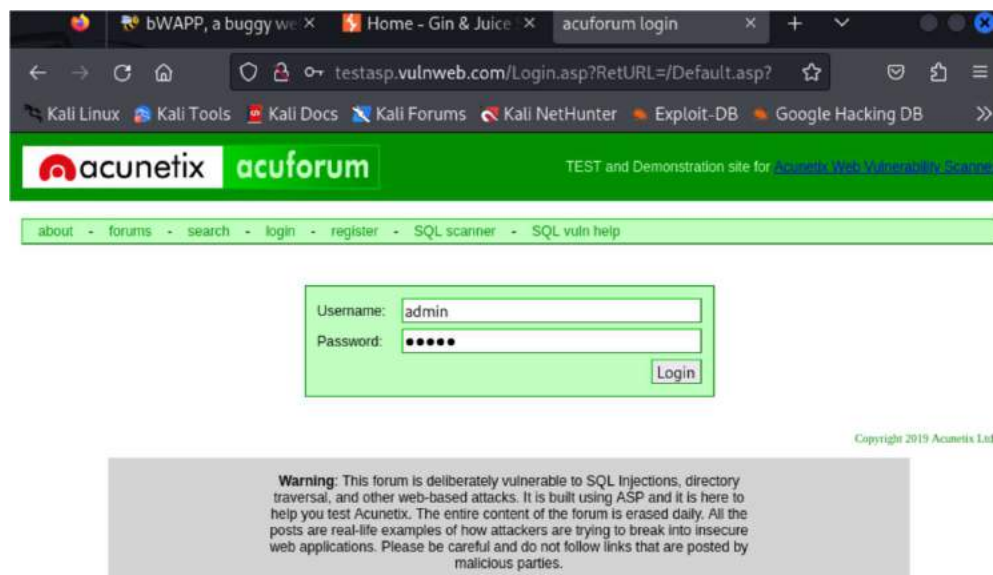
1. **Interception:** Intercept a user's session token using Burp's Proxy.



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:

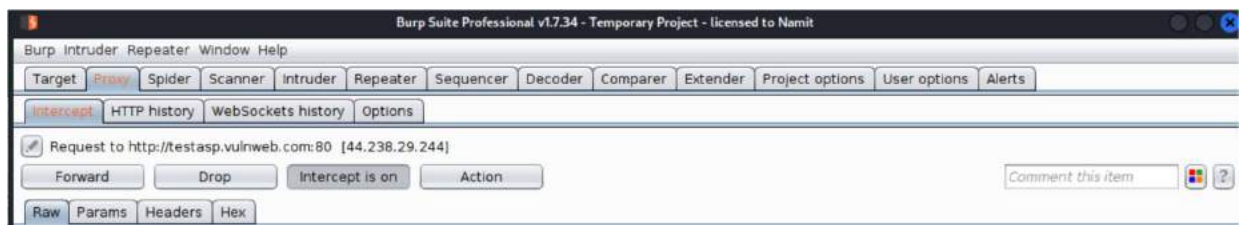
Scenario Question: Can you demonstrate how Burp Suite can be used to identify and mitigate sensitive data exposure vulnerabilities in the web application <http://testasp.vulnweb.com/Login.asp?RetURL=/Default.asp?>

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

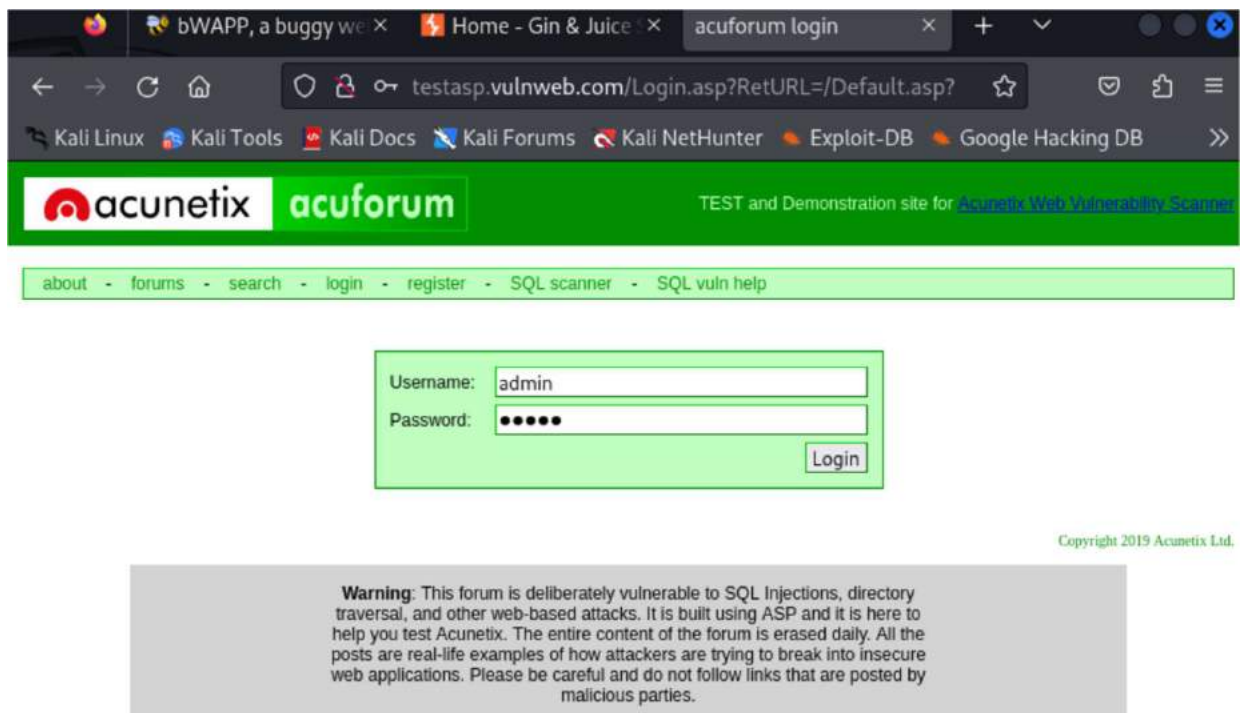
Steps:

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.

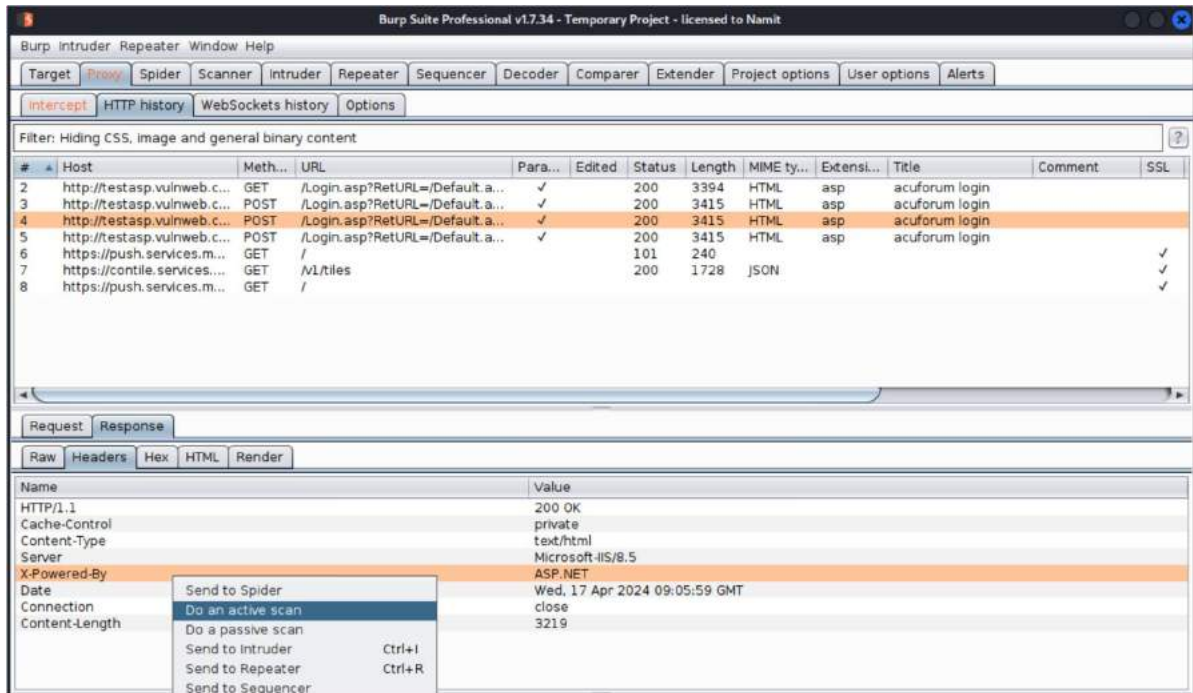


- ### 2. Response Analysis:
- Look for responses containing sensitive data such as passwords, API keys, or personal information.



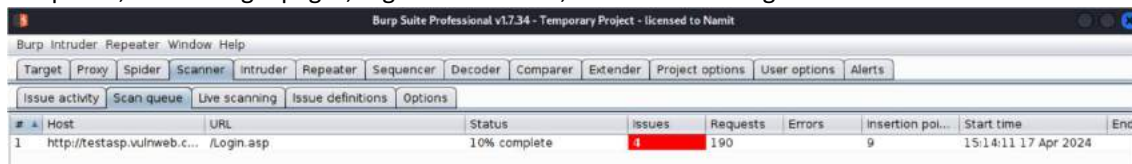
3. Header Inspection:

- 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.

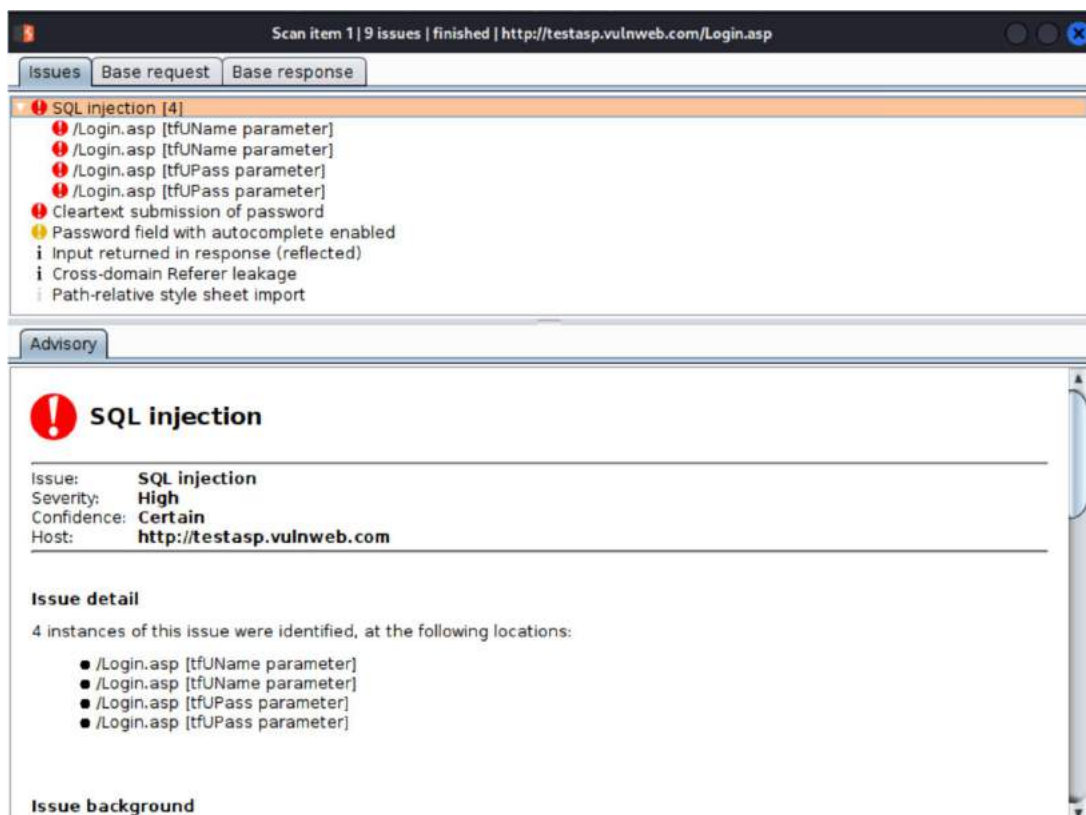


4. Scanner:

- 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.



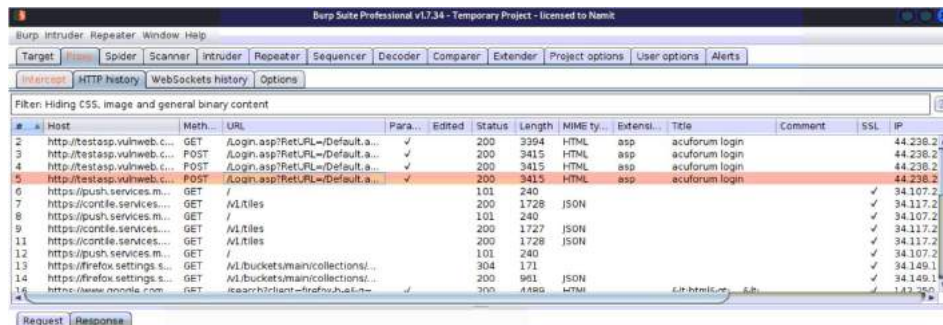
6. CSRF (Cross-Site Request Forgery) Vulnerabilities:

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?

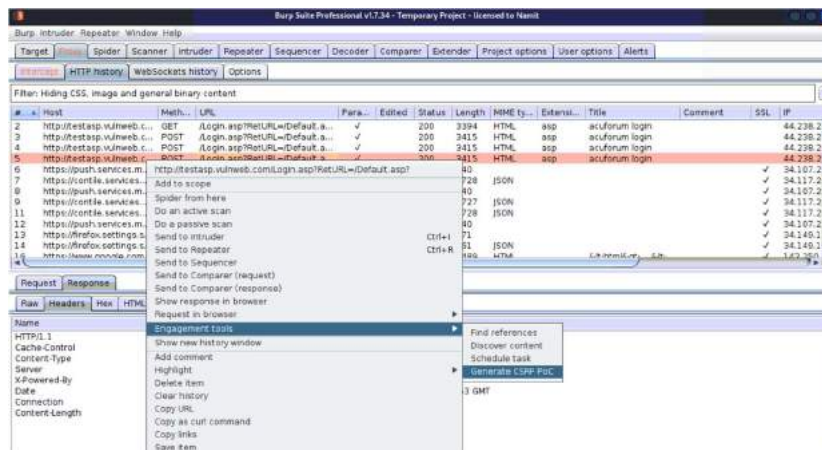
Scenario: You want to assess if the web application <http://testasp.vulnweb.com/Login.asp?RetURL=/Default.asp> is vulnerable to CSRF attacks.

• Steps:

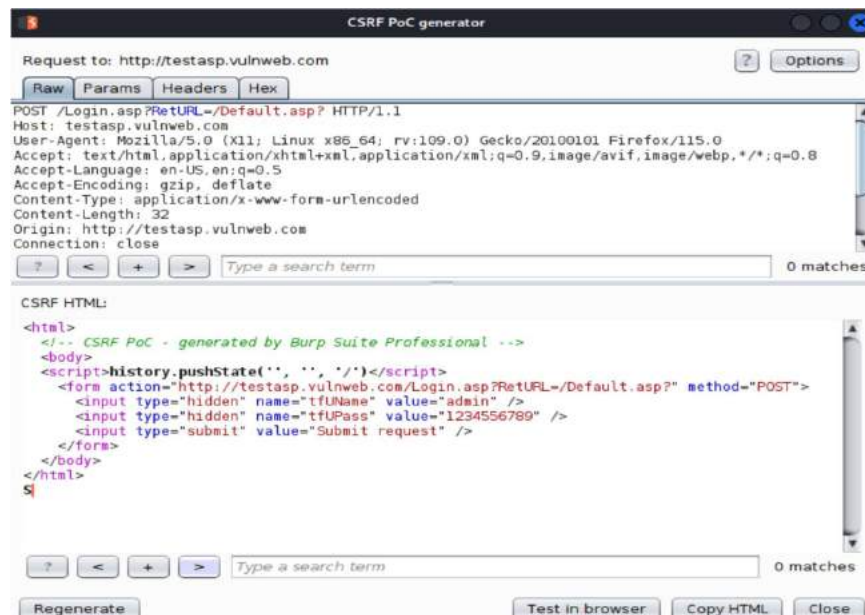
1. **Interception:** Use Burp's Proxy to intercept and modify requests while interacting with the application.



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.

```

1 <html>
2 <!-- CSRF PoC - generated by Burp Suite Professional -->
3 <body>
4 <script>history.pushState('', '', '/')</script>
5 <form action="http://testasp.vulnweb.com/Login.asp?RetURL=/Default.asp" method="POST">
6   <input type="hidden" name="tfUName" value="admin" />
7   <input type="hidden" name="tfUPass" value="1234556789" />
8   <input type="submit" value="Submit request" />
9 </form>
10 </body>
11 </html>
12

```

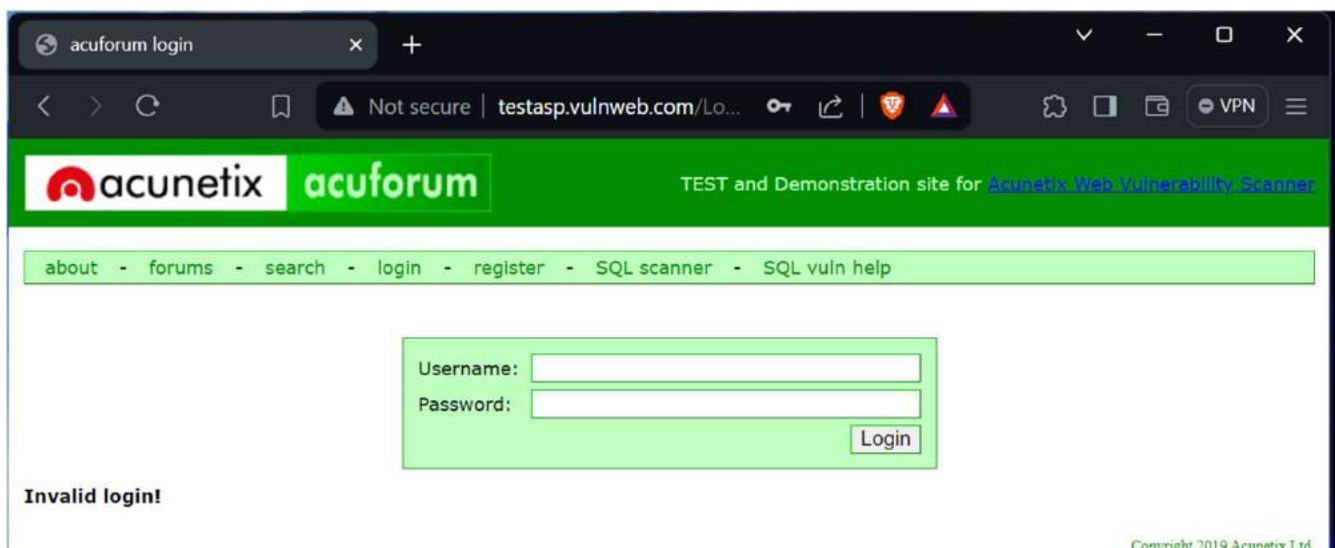
4. **Monitoring:** Monitor the intercepted requests in Burp Suite to see if the CSRF attack is successful.

- 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.



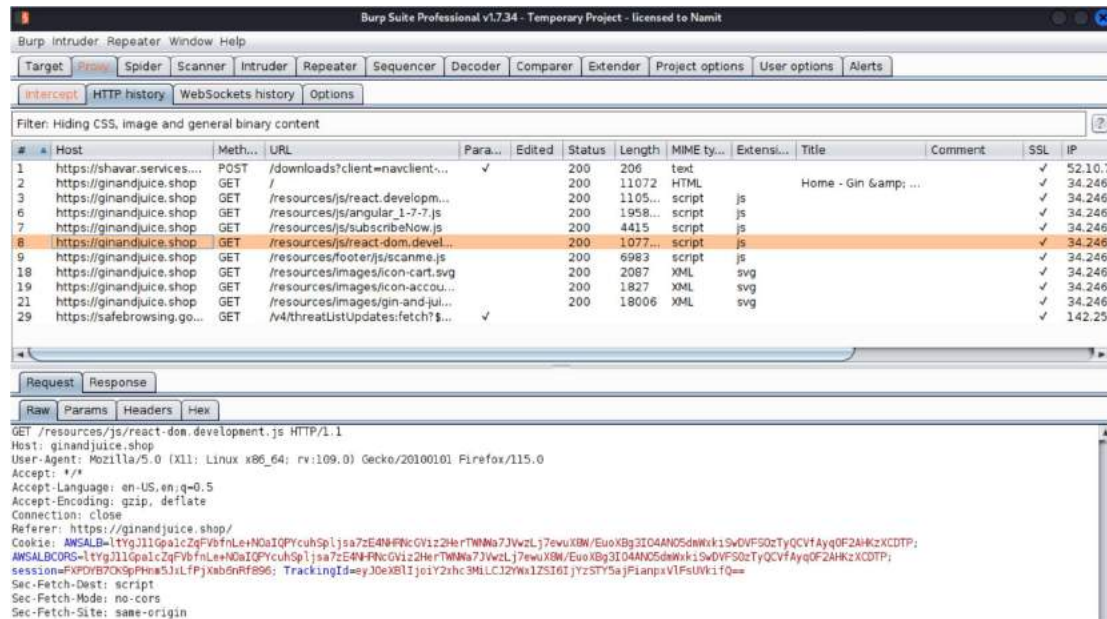
7. Traffic analysis Testing:

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.

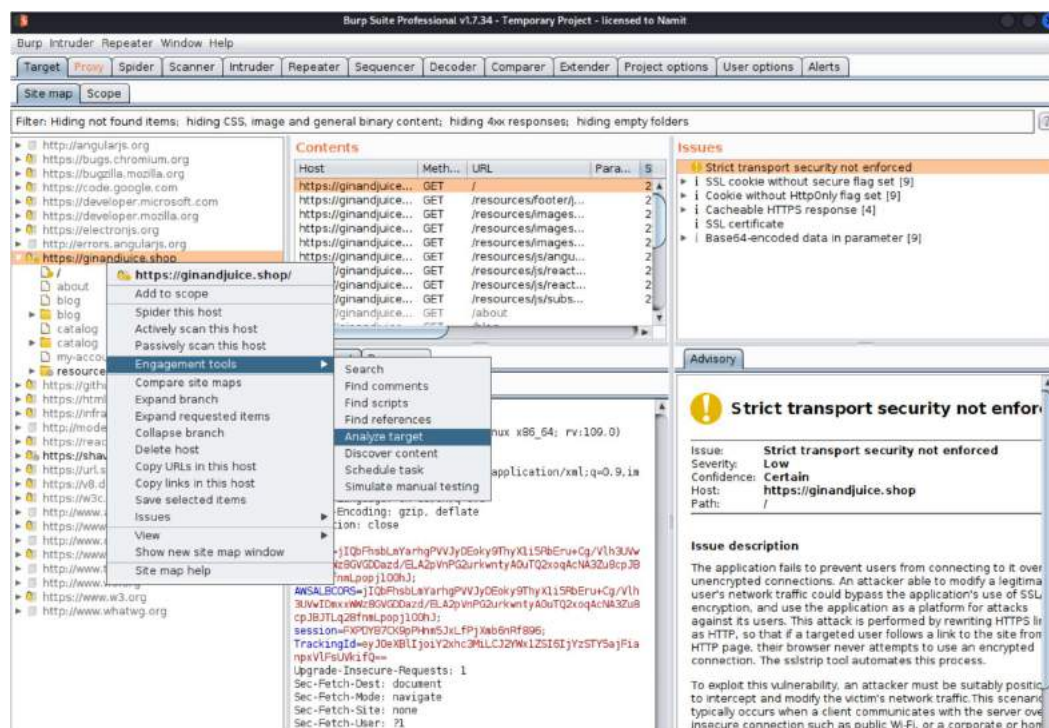
Steps:

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.

The screenshot shows the Burp Suite Target Analyzer interface for the target `https://ginandjuice.shop/`. The 'Summary' tab is active, displaying a table of requests. The selected request is for `/resources/fonts/Poppins/poppins.woff`.

Host	URL	Status	Length	Time requested
https://ginandjuice.shop	/	200	11072	17:45:46 18 Apr 2...
https://ginandjuice.shop	/about			
https://ginandjuice.shop	/blog			
https://ginandjuice.shop	/catalog			
https://ginandjuice.shop	/catalog/cart			
https://ginandjuice.shop	/catalog/subscribe			
https://ginandjuice.shop	/my-account			
https://ginandjuice.shop	/resources/fonts/Poppins/poppins-bold.w...	200	41610	17:45:54 18 Apr 2...
https://ginandjuice.shop	/resources/fonts/Poppins/poppins.woff	200	40401	17:45:53 18 Apr 2...
https://ginandjuice.shop	/resources/images/gin-and-juice-shop-log...	200	18006	17:45:56 18 Apr 2...
https://ginandjuice.shop	/resources/images/icon-account.svg	200	1827	17:45:55 18 Apr 2...
https://ginandjuice.shop	/resources/images/icon-cart.svg	200	2087	17:45:55 18 Apr 2...

The 'Request' tab is active, showing the raw HTTP request details:

```
... Value
... /resources/fonts/Poppins/poppins.woff HTTP/1.1
... ginandjuice.shop
... Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
... application/font-woff2;q=1.0,application/font-woff;q=0.9,*/*;q=0.8
... en-US,en;q=0.5
... gzip, deflate
... close
... https://ginandjuice.shop/resources/css/labsScanme.css
... AWSALB=orID+pV0xWkAQH+4J5SS3HW1gmf/3MAvuaD5w3lhkT98iffLnGuv2kFR5n6P2/X0Awcj/qCssl30wVsTLhrCpFKT+uuLGWPvp...
... font
... cors
... same-origin
```

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.

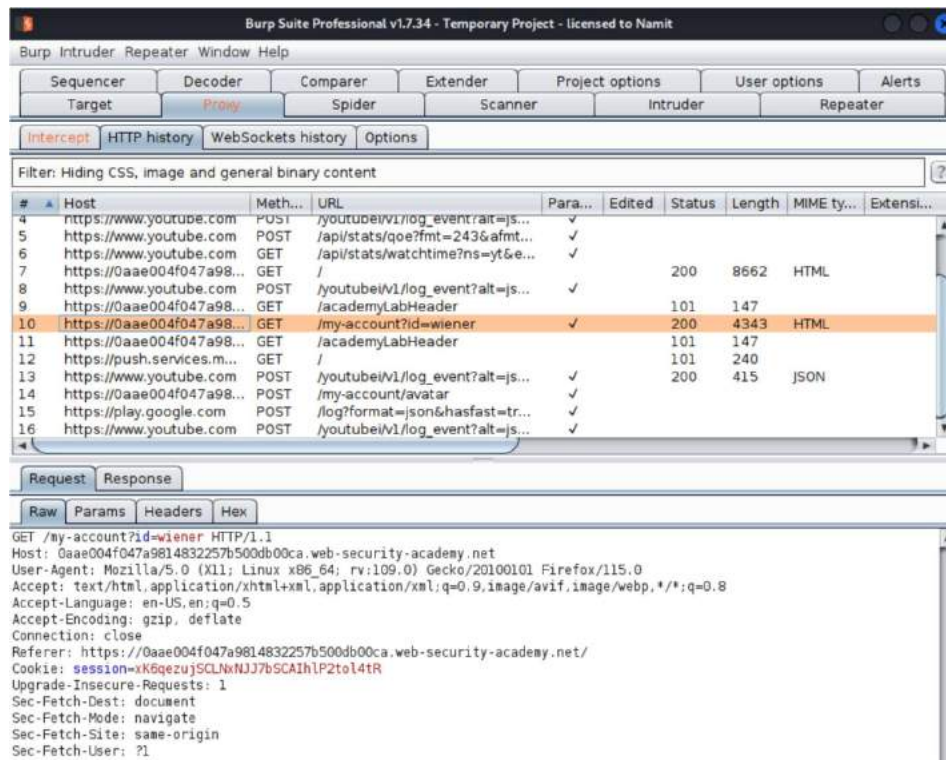
8. File Upload Vulnerabilities:

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?

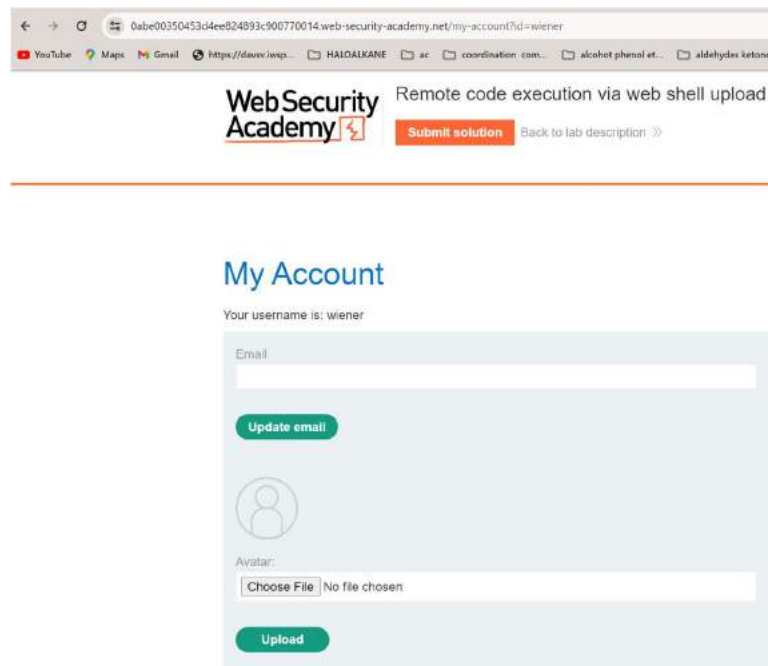
Scenario: We aim to evaluate the security of the file upload functionality on the web application located at <https://portswigger.net/web-security/file-upload/lab-file-upload-remote-code-execution-via-web-shell-upload>.

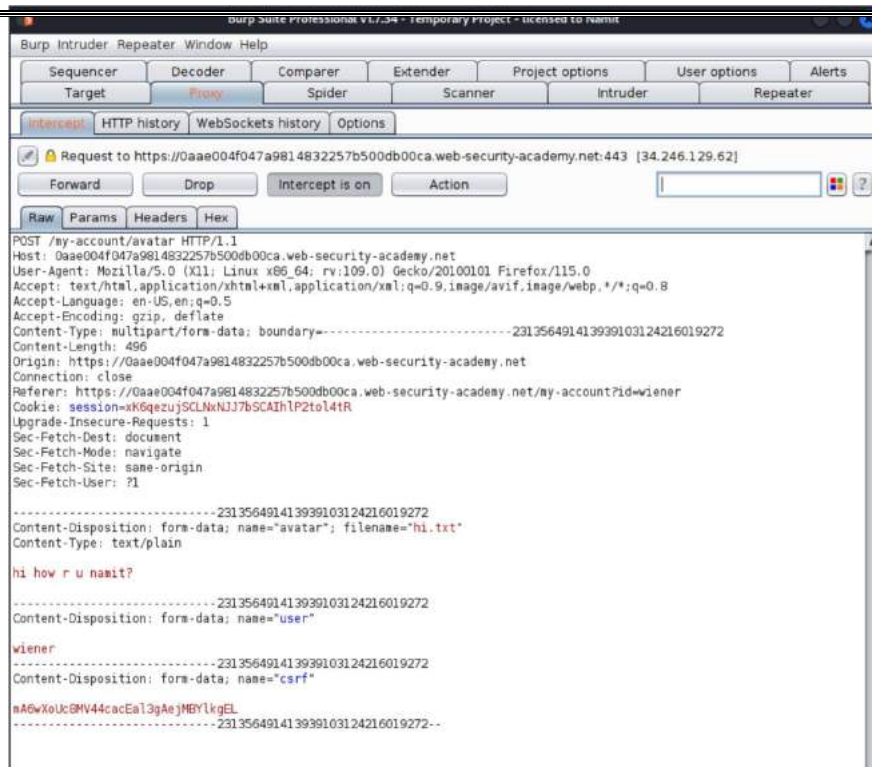
- **Steps:**

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



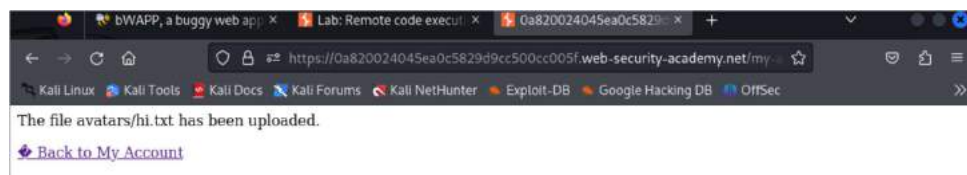
2. **Payload Modification:** Modify the file content to include malicious scripts or executable code.



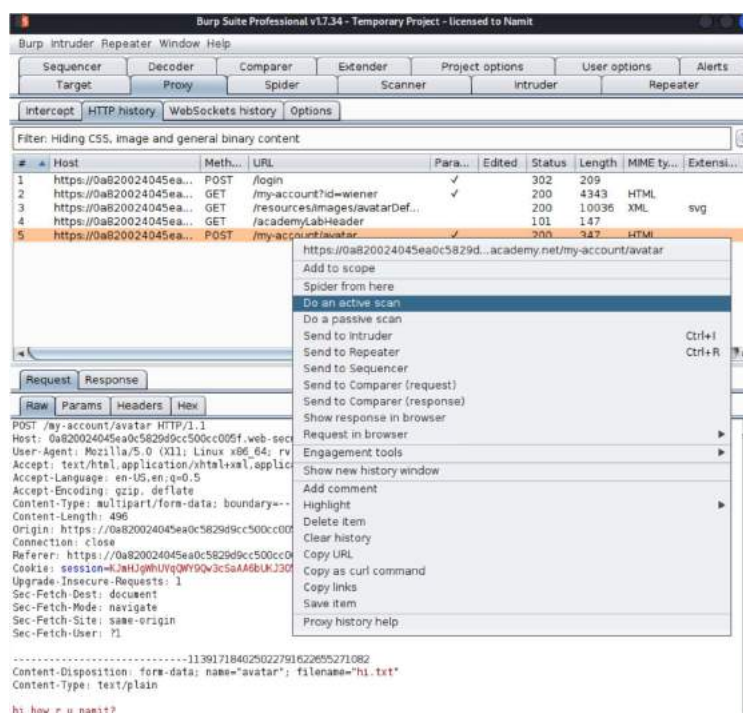


3. Response Analysis:

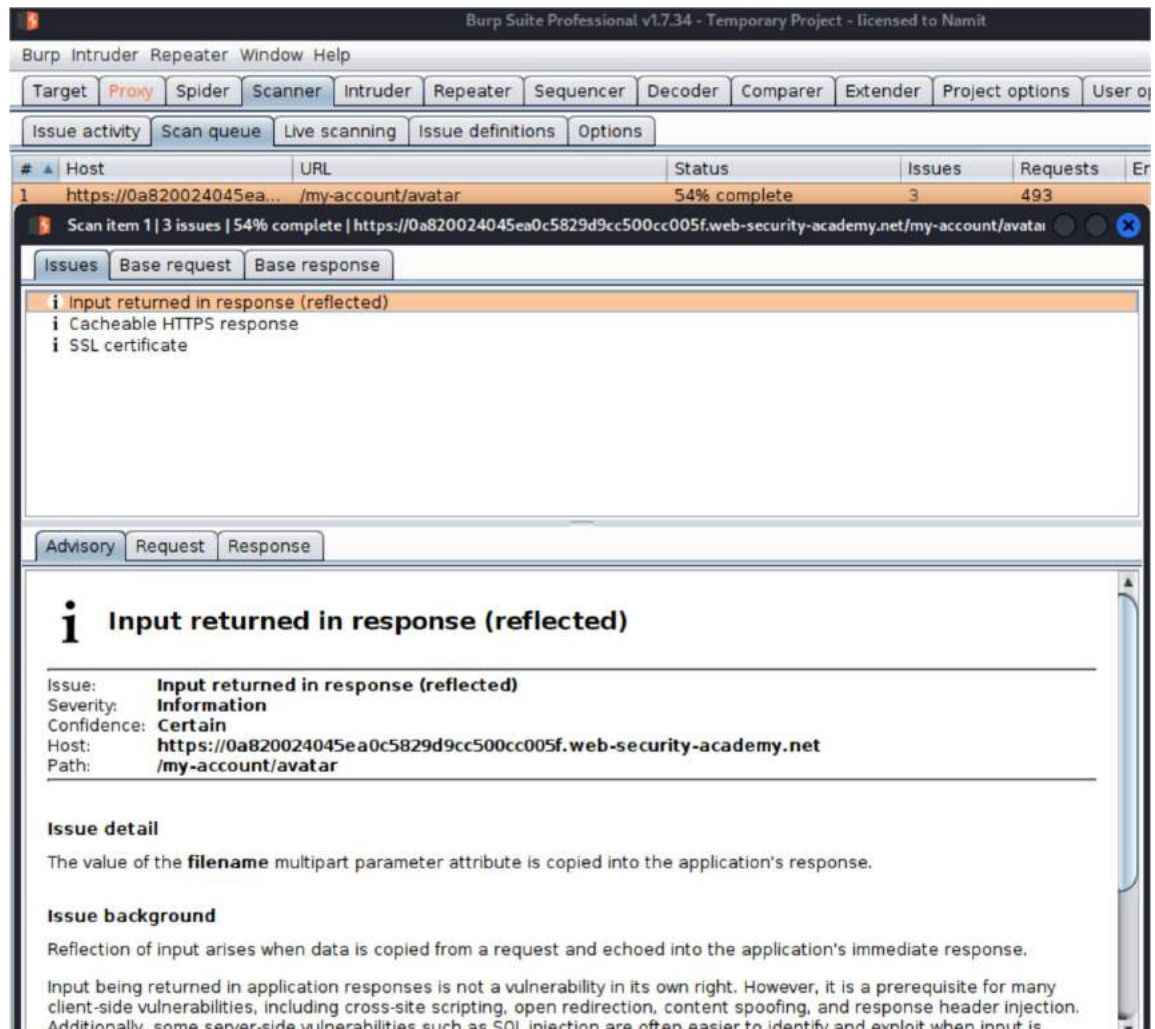
- Analyze the responses from the server to the file upload requests for any indications of validation errors or unexpected behaviors.
- Look for responses that may suggest successful execution of the injected payloads or bypassing of file type restrictions.



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

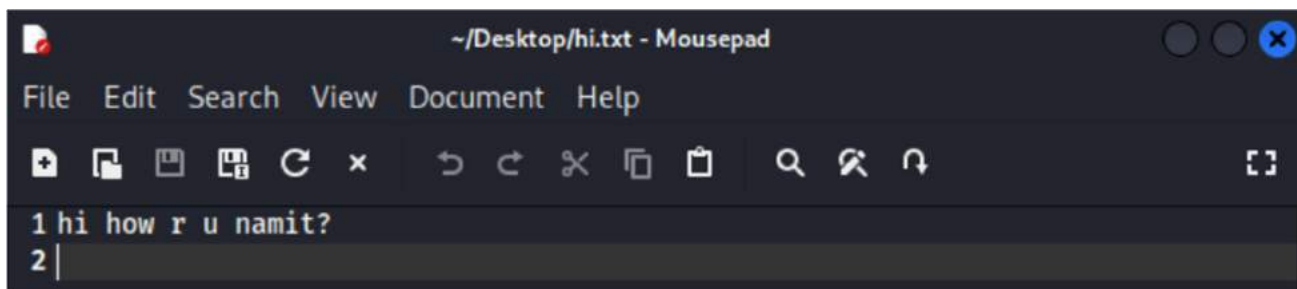


- Configure the scanner to target the file upload functionality and perform thorough testing to identify any potential security weaknesses.



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/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data;
boundary=-----28158308071647504732116612165
Content-Length: 497
Origin: https://0a820024045ea0c5829d9cc500cc005f.web-security-academy.net
Connection: close
Referer: https://0a820024045ea0c5829d9cc500cc005f.web-security-academy.net/my-account
Cookie: session=KJmHJgWhUVqQWY9Qw3cSaAA6bUKJ305W
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1

-----28158308071647504732116612165
Content-Disposition: form-data; name="avatar"; filename="hi.txt"
Content-Type: text/plain
hi how are you garvit.

-----28158308071647504732116612165
Content-Disposition: form-data; name="user"

wiener

-----28158308071647504732116612165
Content-Disposition: form-data; name="csrf"

PiGIRZD5Qdz9IXwI2SjN9SNzKTcpl6i0

-----28158308071647504732116612165--
```

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

hi how are you garvit.

9. Information Disclosure via Error Messages:

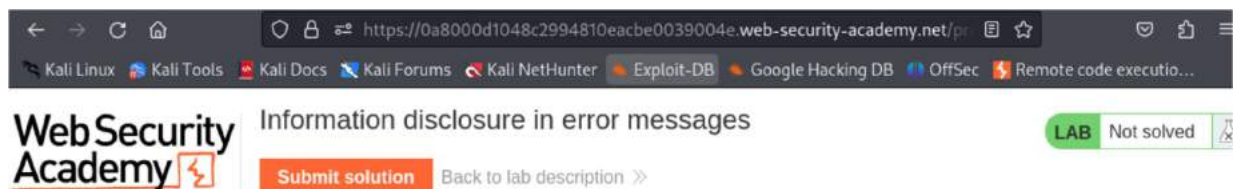
Scenario Question: How can Burp Suite be utilized to investigate and mitigate potential information disclosure vulnerabilities through error messages in a web application?

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.



First Impression Costumes



\$79.11



Description:

It is so hard when meeting people for the first time to work out if they are the good guys or the bad guys. Hey, guys, we are here to help you. With our First Impression Costumes, you can signal that you are the angel those potential dates are looking for.

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.

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

The screenshot shows the Burp Suite interface with a GET request to `/product?productId=1` and its response. The response is an HTTP 200 OK with content type `text/html; charset=utf-8`. The HTML body contains a title `Information disclosure in error messages` and a submit button with `id='submitSolution' class='button' method='POST' path='/submitSolution'`. The Inspector panel on the right shows the selected text `GET /product?productId=1 HTTP/2` and the decoded response body.

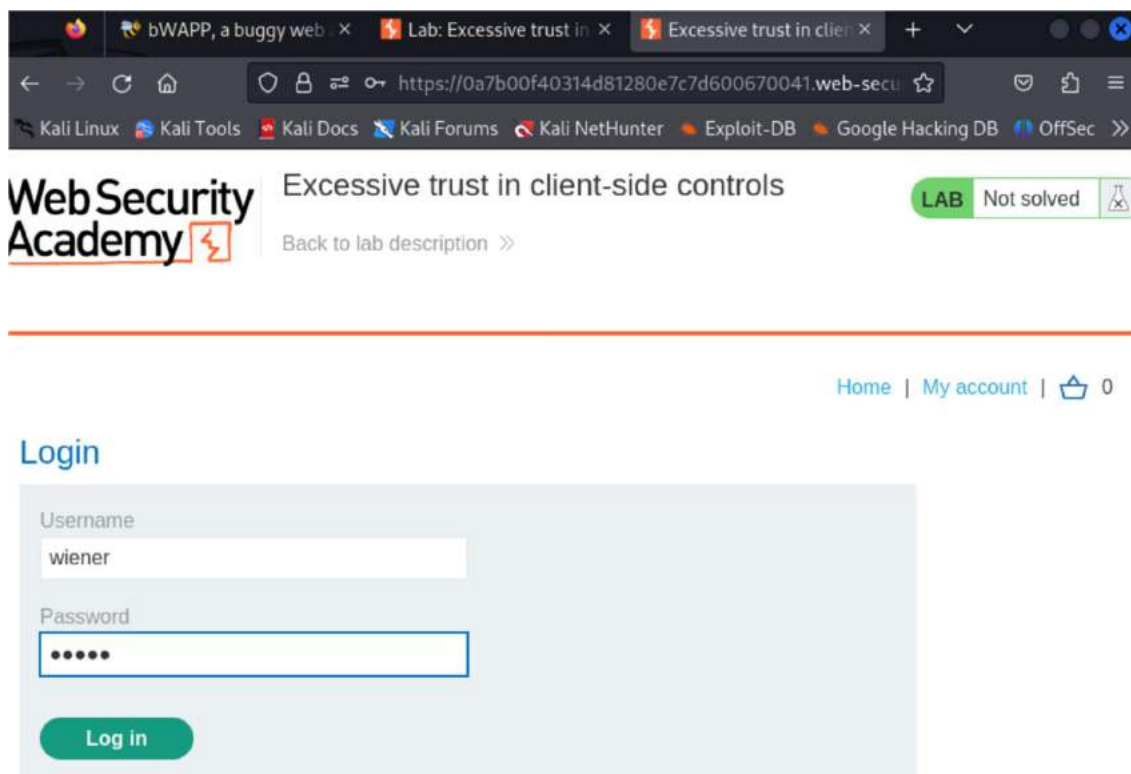
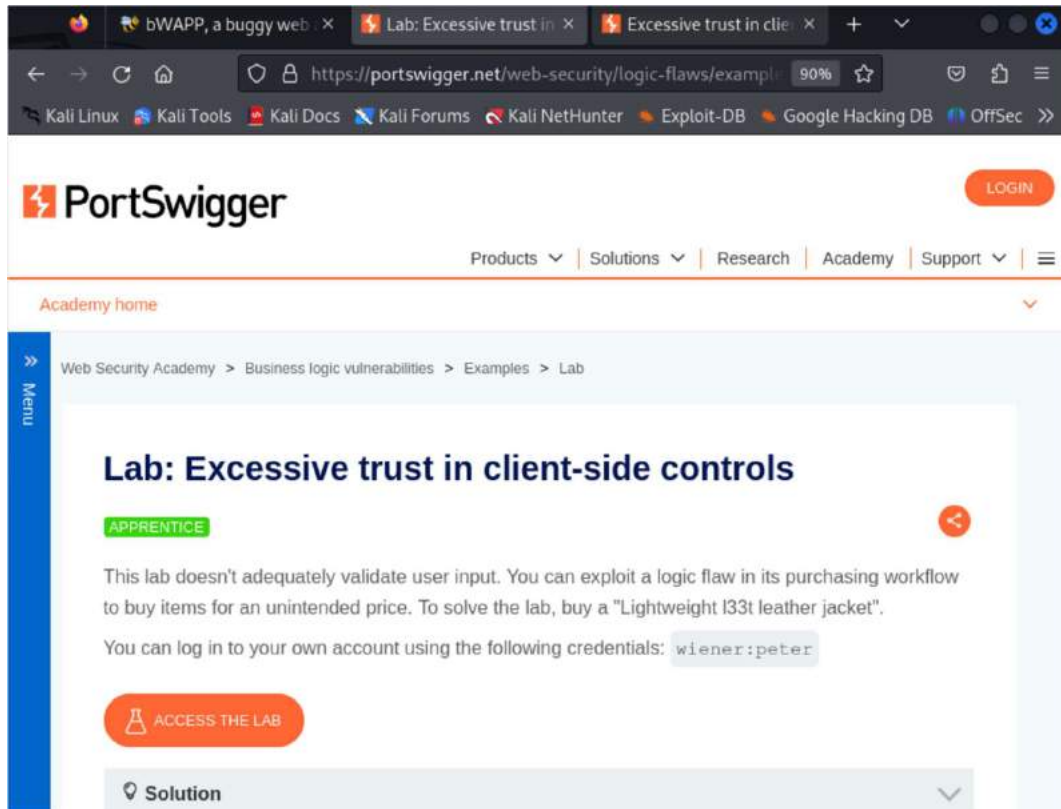
5. **Documentation:** Document any instances of information disclosure and recommend improvements in error handling.

The screenshot shows the Burp Suite interface with a GET request to `/product?productId=1` and its response. The response is an HTTP 200 OK with content type `text/html; charset=utf-8`. The HTML body contains a title `Information disclosure in error messages` and a submit button with `id='submitSolution' class='button' method='POST' path='/submitSolution'`. The Inspector panel on the right shows the selected text `GET /product?productId=1 HTTP/2` and the decoded response body.

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>



Store credit:
\$100.00

[Home](#) | [My account](#) | [0](#) | [Log o](#)

My Account

Your username is: wiener

Email

Update email

Store credit:
\$100.00

[Home](#) | [M](#)

WE LIKE TO
SHOP 



Lightweight "I33t" Leather Jacket
★★★★★ \$1337.00

[View details](#)



Waterproof Tea Bags
★★★★☆ \$63.90

[View details](#)



Hitch A Lift
★★★☆☆ \$72.64

[View details](#)



Adult Space Hop
★★★★☆


[View details](#)

Store credit: \$100.00

Lightweight "133t" Leather Jacket

★★★★★

\$1337.00



Description:

Do you often feel as though people aren't aware of just how "133t" you are? Do you find yourself struggling to make others feel inferior with public displays of your advanced "133t-ness"? If either of these things are at the top of your priority list, it's time to welcome Lightspeed's "133t" Leather Jacket into your life. Handcrafted from leather and single strands of recycled bitcoin, so you can enjoy environmental smugness on top of your high-ranking leather-clad "133t" levels, this jacket is far superior to anything currently available on the high street. Once you've explained to your friends and colleagues what "133t" means, we guarantee you'll be at least 18% cooler when donning your "133t" leather. Inspired by the term-coins, the jacket comes with hand-stitched CHSSP insignia so you can channel the original elite every time you rock your Lightweight "133t" Leather Jacket.

Make your apparel as formidable as your intellect, and dazzle noobs the world over, with the Lightweight "133t" Leather Jacket.

*Every purchase comes with a free booklet, detailing how best to explain the superiority of being "133t" to noobs.

1

Add to cart

< Return to list

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.

Burp Suite Community Edition v2024.2.13 - Temporary Project

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Settings

Intercept HTTP history WebSockets history Proxy settings

Request to https://0a7b00f40314d81280e7c7d600670041.web-security-academy.net:443 [34.246.129.62]

Forward Drop Intercept is on Action Open browser

Pretty Raw Hex

```

1 POST /cart HTTP/2
2 Host: 0a7b00f40314d81280e7c7d600670041.web-security-academy.net
3 Cookie: session=eiUPaND0arjc91rDhV52YhezKvfeT
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 49
10 Origin: https://0a7b00f40314d81280e7c7d600670041.web-security-academy.net
11 Referer: https://0a7b00f40314d81280e7c7d600670041.web-security-academy.net/product?productId=1
12 Upgrade-Insecure-Requests: 1
13 Sec-Fetch-Dest: document
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-Site: same-origin
16 Sec-Fetch-User: ?1
17 Te: trailers
18
19 productId=1&redir=PRODUCT&quantity=1&price=133700

```

Inspector

Request attributes

Request query parameters

Request body parameters

Name	Value
productId	1
redir	PRODUCT
quantity	1
price	133700

Request cookies

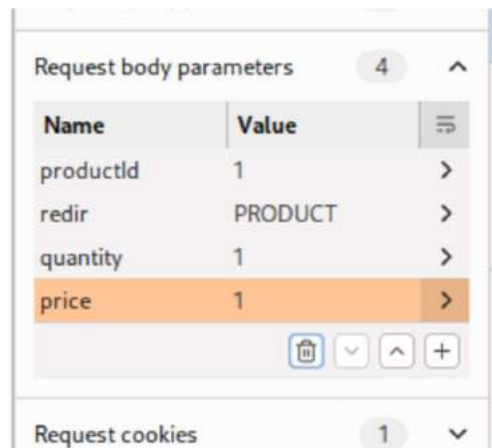
Request headers

Event log All issues

Memory: 102.3MB

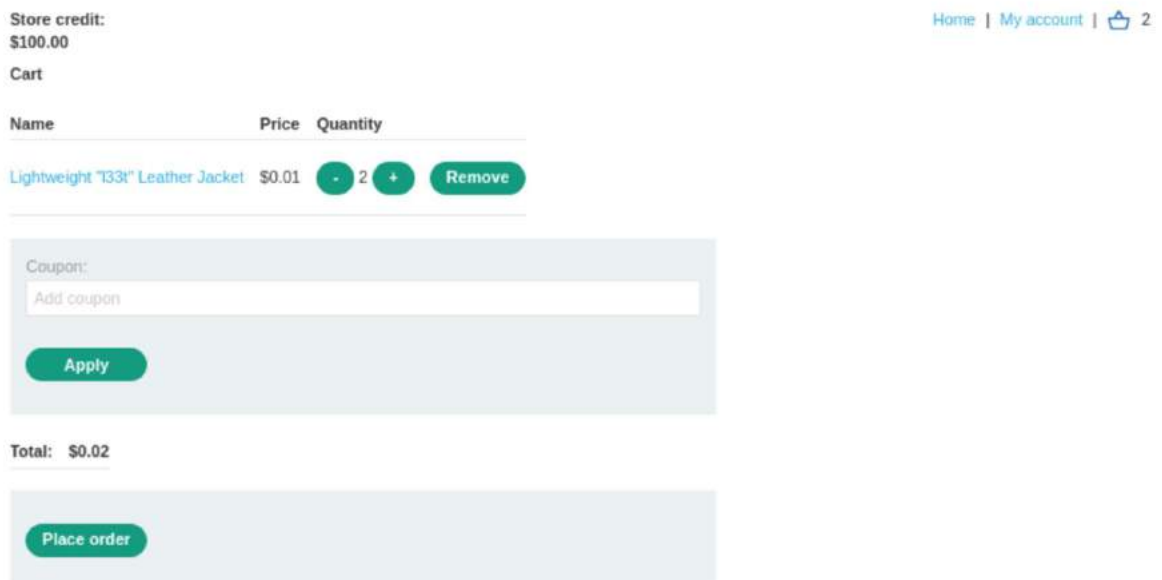
2. Interception and Inspector:

- 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.



Excessive trust in client-side controls

[Back to lab description >>](#)

LAB Solved

Congratulations, you solved the lab!

Share your skills!



[Continue learning >>](#)

Store credit:
\$99.98

[Home](#) | [My account](#) | 0

Your order is on its way!

Name	Price	Quantity
Lightweight "T33t" Leather Jacket	\$1337.00	2

Total: \$0.02

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.

The screenshot shows the Burp Suite Community Edition v2024.2.1.3 interface. The top menu bar includes Burp, Project, Intruder, Repeater, View, and Help. The main toolbar has buttons for Intercept, HTTP history, WebSockets history, and Proxy settings. Below the toolbar is a filter settings bar. The HTTP history table lists several requests to https://ginandjuice.shop, including GET requests for /index.htm, /, /logout, /my-account, /login, and POST requests for /login. The selected request is a GET request to /my-account. The Request tab shows the raw HTTP request, and the Response tab shows the raw HTTP response. The Inspector panel on the right shows the request attributes, cookies, headers, and response headers.

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes
1	http://www.itsecgames.com	GET	/index.htm			200	3912	HTML	htm	bWAPP, a buggy web ...	
4	https://ginandjuice.shop	GET	/			200	11138	HTML		Home - Gin & Juice...	
8	https://ginandjuice.shop	GET	/logout			302	665				
9	https://ginandjuice.shop	GET	/			200	11051	HTML		Home - Gin & Juice...	
10	https://ginandjuice.shop	GET	/my-account			302	583				
11	https://ginandjuice.shop	GET	/login			200	8056	HTML		Login - Gin & Juice...	
12	https://ginandjuice.shop	POST	/login		✓	200	8406	HTML		Login - Gin & Juice...	
13	https://ginandjuice.shop	POST	/login		✓	302	675				
14	https://ginandjuice.shop	GET	/my-account			200	13218	HTML		My account - Gin & Juice...	

Request: GET /my-account HTTP/2
Host: ginandjuice.shop
Cookie: AWSALB=...
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0

Response: HTTP/2 200 OK
Date: Fri, 19 Apr 2024 10:08:39 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 12587
Set-Cookie: AWSALB=...
Set-Cookie: AWSALBCORS=...

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.

The screenshot shows the Burp Suite Community Edition v2024.2.1.3 interface. The top menu bar includes Burp, Project, Intruder, Repeater, View, and Help. The main toolbar has buttons for Intercept, HTTP history, WebSockets history, and Proxy settings. Below the toolbar is a filter settings bar. The HTTP history table lists several requests to https://ginandjuice.shop, including GET requests for /index.htm, /, /logout, /my-account, /login, and POST requests for /login. The selected request is a GET request to /my-account. The Request tab shows the raw HTTP request, and the Response tab shows the raw HTTP response. The Inspector panel on the right shows the request attributes, cookies, headers, and response headers. The 'Send to Intruder' option is highlighted in the context menu.

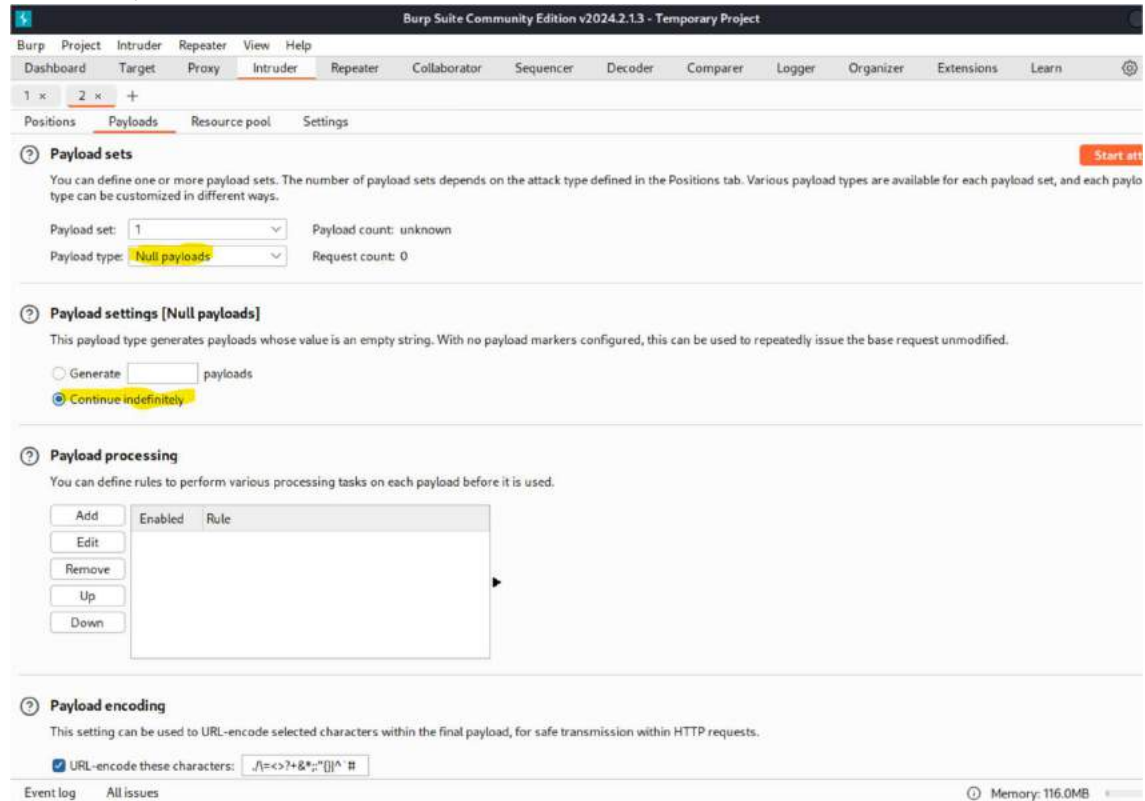
#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes
1	http://www.itsecgames.com	GET	/index.htm			200	3912	HTML	htm	bWAPP, a buggy web ...	
4	https://ginandjuice.shop	GET	/			200	11138	HTML		Home - Gin & Juice...	
8	https://ginandjuice.shop	GET	/logout			302	665				
9	https://ginandjuice.shop	GET	/			200	11051	HTML		Home - Gin & Juice...	
10	https://ginandjuice.shop	GET	/my-account			302	583				
11	https://ginandjuice.shop	GET	/login			200	8056	HTML		Login - Gin & Juice...	
12	https://ginandjuice.shop	POST	/login		✓	200	8406	HTML		Login - Gin & Juice...	
13	https://ginandjuice.shop	POST	/login		✓	302	675				
14	https://ginandjuice.shop	GET	/my-account			200	13218	HTML		My account - Gin & Juice...	

Request: GET /my-account HTTP/2
Host: ginandjuice.shop
Cookie: AWSALB=...
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0

Response: HTTP/2 200 OK
Date: Fri, 19 Apr 2024 10:08:39 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 12587
Set-Cookie: AWSALB=...
Set-Cookie: AWSALBCORS=...

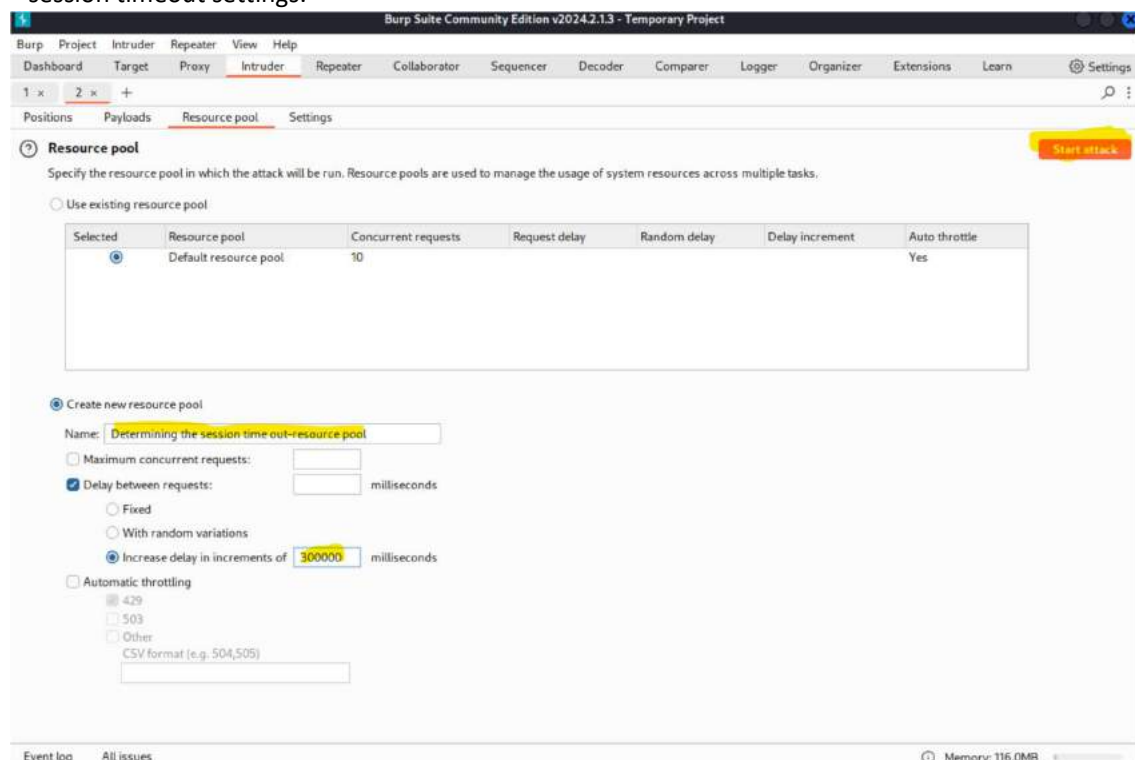
3. Payload Injection:

- 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.

The screenshot shows the Burp Suite interface during an intruder attack on <https://ginandjuice.shop>. The top bar indicates the target URL and provides buttons for 'Attack', 'Save', and a refresh icon. Below the bar, the 'Results' tab is active, showing a table of requests. The table has columns for Request, Payload, Status code, Time of day, Response received, Error, Timeout, Length, and Comment. The selected request (index 2) shows a status code of 200 and a response received of 531. Below the table, the 'Request' tab is active, displaying the raw HTTP request details.

Request	Payload	Status code	Time of day	Response received	Error	Timeout	Length	Comment
0		200	15:42:31 19 Apr 2024	374			13218	
1	null	200	15:47:31 19 Apr 2024	531			13218	
2	null	200	15:57:31 19 Apr 2024	241			13218	
3	null			0				

Request Details:

```
1 GET /my-account HTTP/2
2 Host: ginandjuice.shop
3 Cookie: AWSALB=H5CCc1btfqg5s//a126sLmNfEeik100vi2EgHox5qVvywF0uzstV2hLThqSsekHtrq3cgg3i7k3PPr10ykbPSTKCKHvATRQp7k5sKQK7/NGL1DX4cJ; AWSALBCORS=H5CCc1btfqg5s//a126sLmNfEeik100vi2EgHox5qVvywF0uzstV2hLThqSsekHtrq3cgg3i7k3PPr10ykbPSTKCKHvATRQp7k5sKQK7/NGL1DX4cJ; session=L_FCKQy90VAM8tG1f3W5s19ys1gHtau
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.3
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://ginandjuice.shop/login
9 Upgrade-Insecure-Requests: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-User: ?1
14 Te: trailers
15 Connection: keep-alive
16
17
```