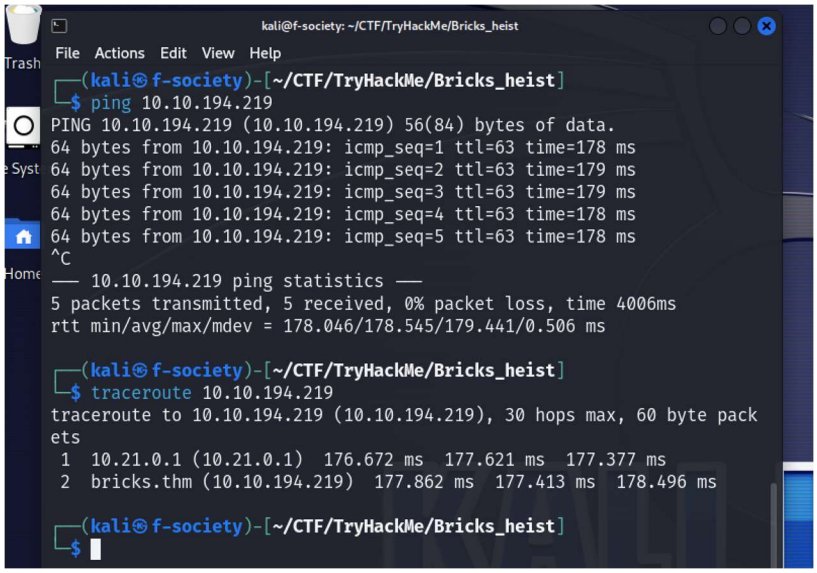
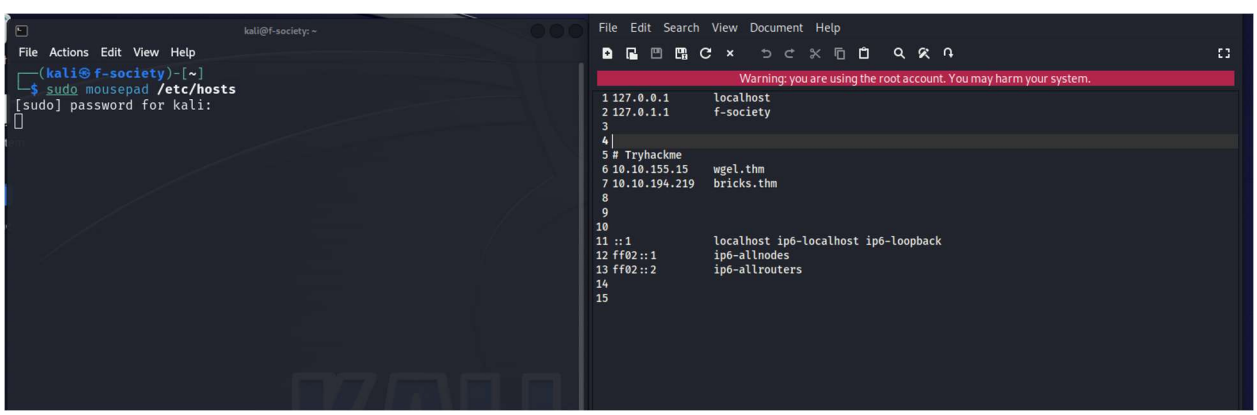
BRICKS HEIST TRYHACKME

To hack this machine, we do the following:

1. As always, first we connect our vpn
2. Next, we check if our machine is online by pinging



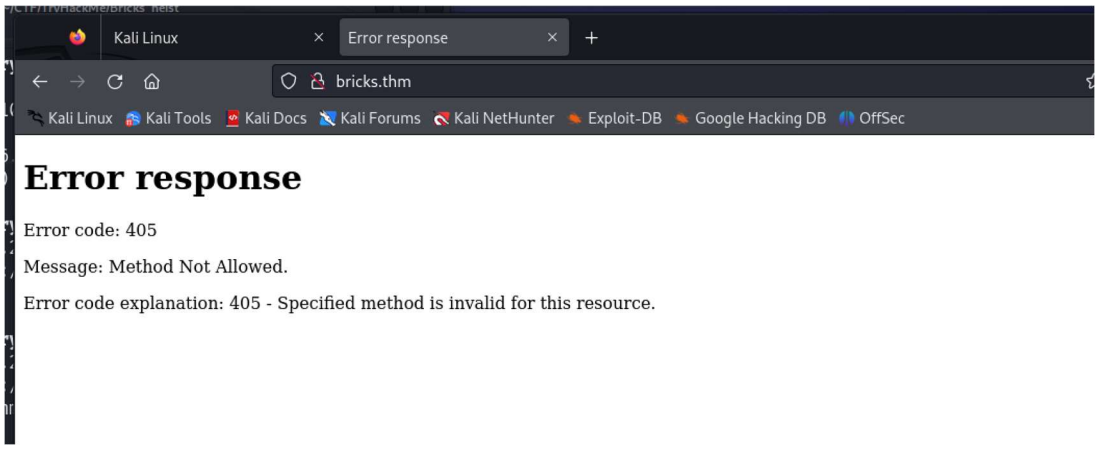
3. Before we start our enumeration, we are told to add the target ip to our **/etc/hosts** file. So, navigate to the file and add the ip:
    - sudo mousepad /etc/hosts



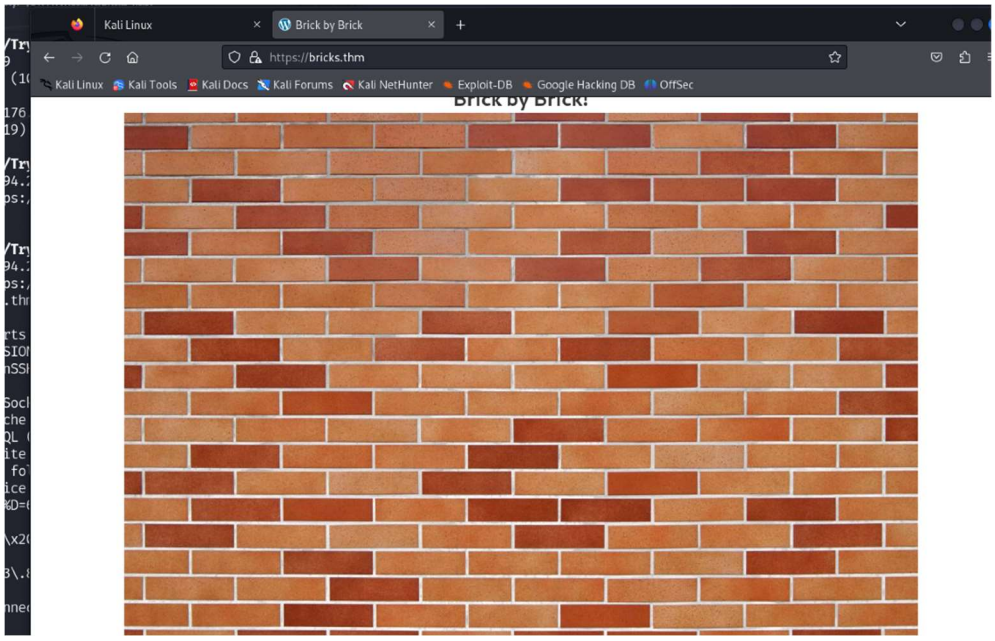
4. Now let's scan our target ip using nmap and find the open ports.
    - nmap -F -Pn -sV {target_ip} -o {file_name}
        - -F = fast mode
        - -Pn = no ping

```
┌──(kali㉿f-society)-[~/CTF/TryHackMe/Bricks_heist]
└─$ nmap -F -Pn -sV 10.10.194.219 -o nmap_results
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-14 16:29 EDT
Nmap scan report for bricks.thm (10.10.194.219)
Host is up (0.18s latency).
Not shown: 96 closed tcp ports (conn-refused)
PORT     STATE SERVICE  VERSION
22/tcp   open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.11 (Ubuntu Linux
; protocol 2.0)
80/tcp   open  http     WebSockify Python/3.8.10
443/tcp  open  ssl/http Apache httpd
3306/tcp open  mysql    MySQL (unauthorized)
1 service unrecognized despite returning data. If you know the service
/version, please submit the following fingerprint at https://nmap.org/
cgi-bin/submit.cgi?new-service :
SF-Port80-TCP:V=7.94SVN%I=7%D=6/14%Time=666CA829%P=x86_64-pc-linux-gnu
%r(G
SF:etRequest,291,"HTTP/1\.1\x20405\x20Method\x20Not\x20Allowed\r\nServ
er:\
SF:x20WebSockify\x20Python/3\.8\.10\r\nDate:\x20Fri,\x2014\x20Jun\x202
024\
SF:x2020:29:29\x20GMT\r\nConnection:\x20close\r\nContent-Type:\x20text
/htm
```

5. Since we found that http and https are open, let's check the website on our browser. But unfortunately, http is is displaying 405 error (method not allowed).

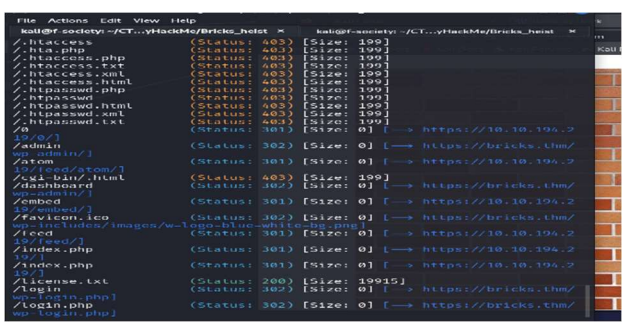


- So, we use https to access the website and it works.



6. Now let's view the source page of the website by right clicking and opening source page and we find that the website is made using wordpress.

7. So, let's run our gobuster to check for subdomains if there are any.
    - gobuster dir -u https://bricks.thm -w /usr/share/dirb/wordlists/common.txt



❖ Even though we found some subdomains, there is nothing there it is a dead end.

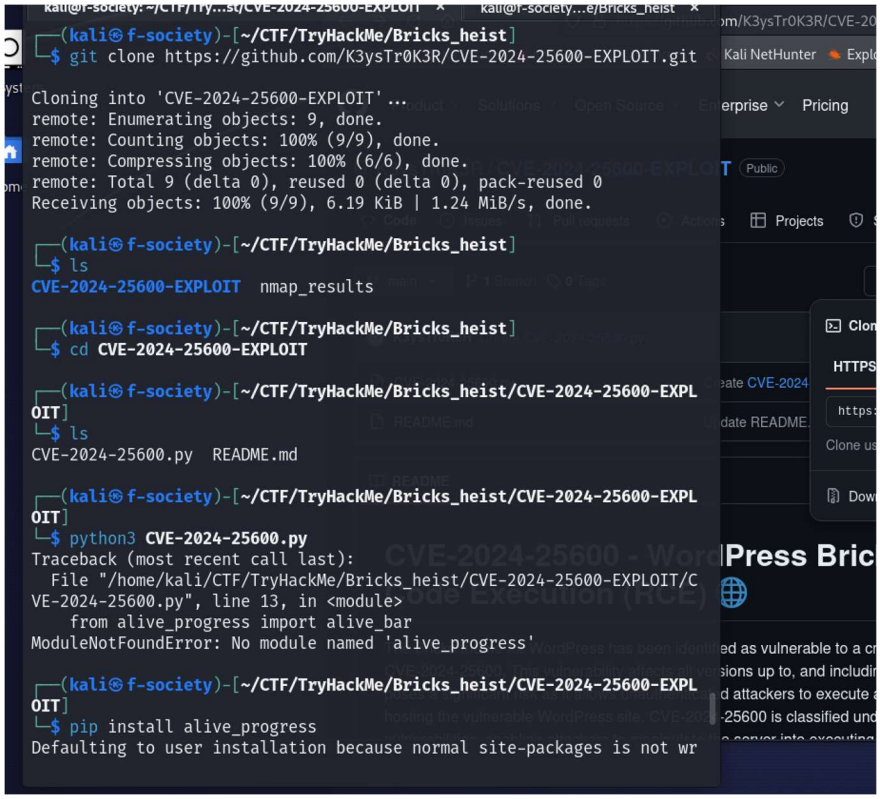
8. Now let's use a tool called **wpscan** which is used to scan for vulnerabilities in wordpress sites and it is built-in in kali.
    - wpscan - - url https://bricks.thm

```
[+] WordPress theme in use: bricks
 | Location: https://bricks.thm/wp-content/themes/bricks/
 | Readme: https://bricks.thm/wp-content/themes/bricks/readme.txt
 | Style URL: https://bricks.thm/wp-content/themes/bricks/style.css
 | Style Name: Bricks
 | Style URI: https://bricksbuilder.io/
 | Description: Visual website builder for WordPress....
 | Author: Bricks
 | Author URI: https://bricksbuilder.io/
 |
 | Found By: Urls In Homepage (Passive Detection)
 | Confirmed By: Urls In 404 Page (Passive Detection)
 |
 | Version: 1.9.5 (80% confidence)
 | Found By: Style (Passive Detection)
 |  - https://bricks.thm/wp-content/themes/bricks/style.css, Match: 'V
ersion: 1.9.5'

[+] Enumerating All Plugins (via Passive Methods)

[i] No plugins Found.

[+] Enumerating Config Backups (via Passive and Aggressive Methods)
 Checking Config Backups - Time: 00:00:00 <> (0 / 137)  0.00%  ETA: ??
 Checking Config Backups - Time: 00:00:00 <> (1 / 137)  0.72%  ETA: 00
 Checking Config Backups - Time: 00:00:00 <> (2 / 137)  1.45%  ETA: 00
 Checking Config Backups - Time: 00:00:00 <> (3 / 137)  2.18%  ETA: 00
```

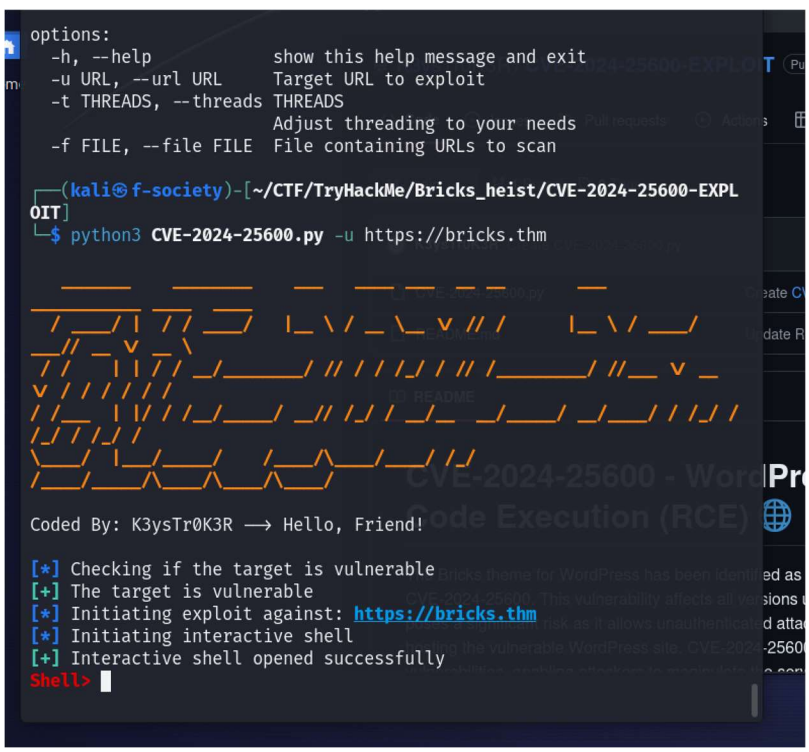
❖ Here we find some juicy informations and one of them is the version of the wordpress.
    - Version: 1.9.5

9. Now we go to our friend "google" and search for an exploit if there is any for wordpress version 1.9.5.

10. We found an exploit with the name **CVE-2024-25600** on github and it is a python script to get access to the shell. So, let's clone it:
    - git clone {github_link}

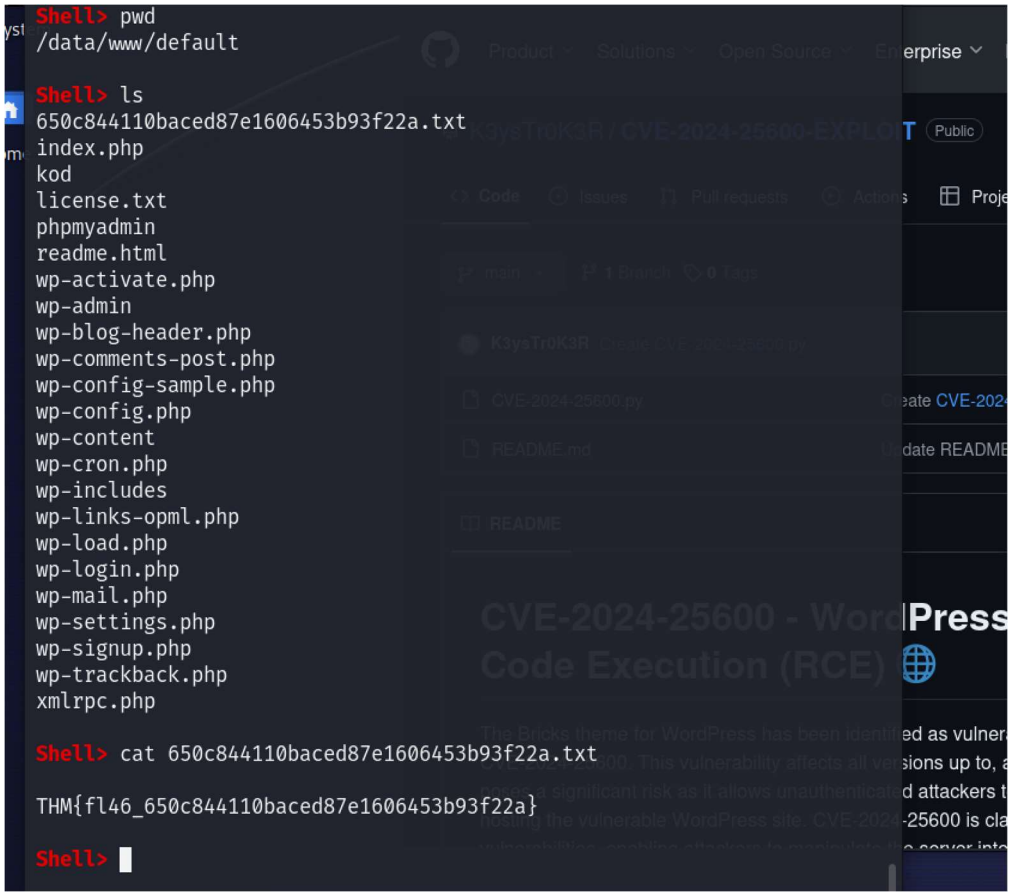


- Now after cloning the python script let's run our script to get access.
  - -u = url of target



- **BOOM SHAKALAKA** we got access.

11. Now on our shell let's try to see the file by using file listing command.
    - **"ls"** and we find files and one is text file with numbers.
    - So, we use the command "**cat"** to display the text file.



❖ **BOOYA** here is our hidden text.

12. Next, we find the services that are running using the command **systemctl** and try to find a suspicious process.
    - systemctl list-units - - type=service - - state=running
        - -- type = show what is the process
        - -- state = show in what state the process is, is it running or not.

❖ Among the services that are running we see there is a description that says **TRYHACK3M** so that got be the suspicious process.

```
  httpd.service                                      loaded active running LSB: starts Apache Web Server
  irqbalance.service                                 loaded active running irqbalance daemon
  kerneloops.service                                 loaded active running Tool to automatically collect and submit kerne
  lightdm.service                                    loaded active running Light Display Manager
  ModemManager.service                               loaded active running Modem Manager
  multipathd.service                                 loaded active running Device-Mapper Multipath Device Controller
  mysqld.service                                     loaded active running LSB: start and stop MySQL
  networkd-dispatcher.service                        loaded active running Dispatcher daemon for systemd-networkd
  NetworkManager.service                             loaded active running Network Manager
  polkit.service                                     loaded active running Authorization Manager
  rsyslog.service                                    loaded active running System Logging Service
  rtkit-daemon.service                               loaded active running RealtimeKit Scheduling Policy Service
  serial-getty@ttyS0.service                         loaded active running Serial Getty on ttyS0
  snap.amazon-ssm-agent.amazon-ssm-agent.service loaded active running Service for snap application amazon-ssm-agent.a
  snapd.service                                      loaded active running Snap Daemon
  ssh.service                                        loaded active running OpenBSD Secure Shell server
  switcheroo-control.service                         loaded active running Switcheroo Control Proxy service
  systemd-journald.service                           loaded active running Journal Service
  systemd-logind.service                             loaded active running Login Service
  systemd-networkd.service                           loaded active running Network Service
  systemd-resolved.service                           loaded active running Network Name Resolution
  systemd-timesyncd.service                          loaded active running Network Time Synchronization
  systemd-udevd.service                              loaded active running udev Kernel Device Manager
  ubuntu.service                                     loaded active running TRYHACK3M
  udisks2.service                                    loaded active running Disk Manager
  unattended-upgrades.service                        loaded active running Unattended Upgrades Shutdown
  upower.service                                     loaded active running Daemon for power management
  user@1000.service                                  loaded active running User Manager for UID 1000
  user@114.service                                   loaded active running User Manager for UID 114
  whoopsie.service                                   loaded active running crash report submission daemon
  wpa_supplicant.service                             loaded active running WPA supplicant

LOAD   = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB    = The low-level unit activation state, values depend on unit type.

40 loaded units listed.

Shell>
```

- When we view the service by its name, which is **ubuntu.service** we find the name of the process which is **nm-inet-dialog**.

```
Shell> systemctl cat ubuntu.service
# /etc/systemd/system/ubuntu.service
[Unit]
Description=TRYHACK3M

[Service]
Type=simple
ExecStart=/lib/NetworkManager/nm-inet-dialog
Restart=on-failure

[Install]
WantedBy=multi-user.target

Shell>
```

- **BOOM** we just answered two questions at once, now let's continue to our next question.

13. Since we know the path of the suspicious process, to find the log file we search in the directory where the process is.
    - ➢ ls /lib/NetworkManager

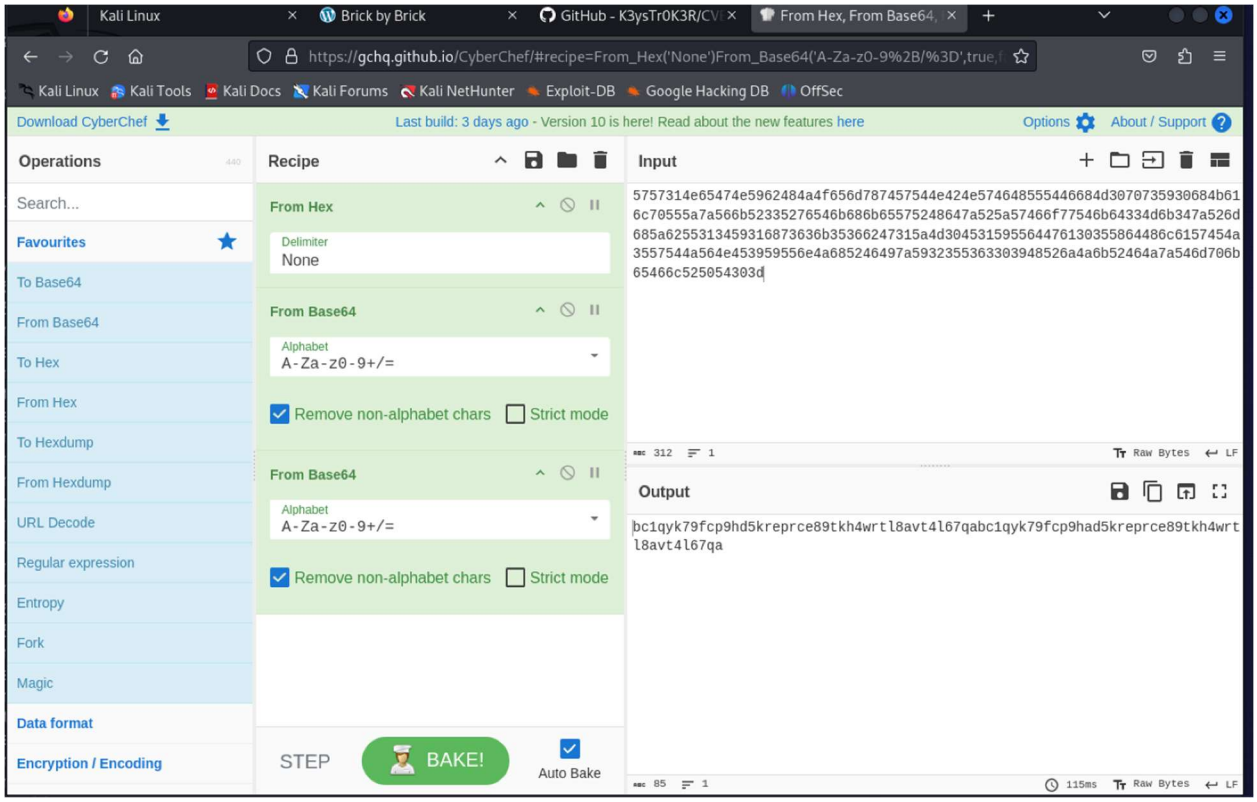
```
Shell> ls /lib/NetworkManager
VPN
conf.d
dispatcher.d
inet.conf
nm-dhcp-helper
nm-dispatcher
nm-iface-helper
nm-inet-dialog
nm-initrd-generator
nm-openvpn-auth-dialog
nm-openvpn-service
nm-openvpn-service-openvpn-helper
nm-pptp-auth-dialog
nm-pptp-service
system-connections

Shell>
```

- ❖ As you can see the log file name is **inet.conf** which we will prove on the next screenshot.

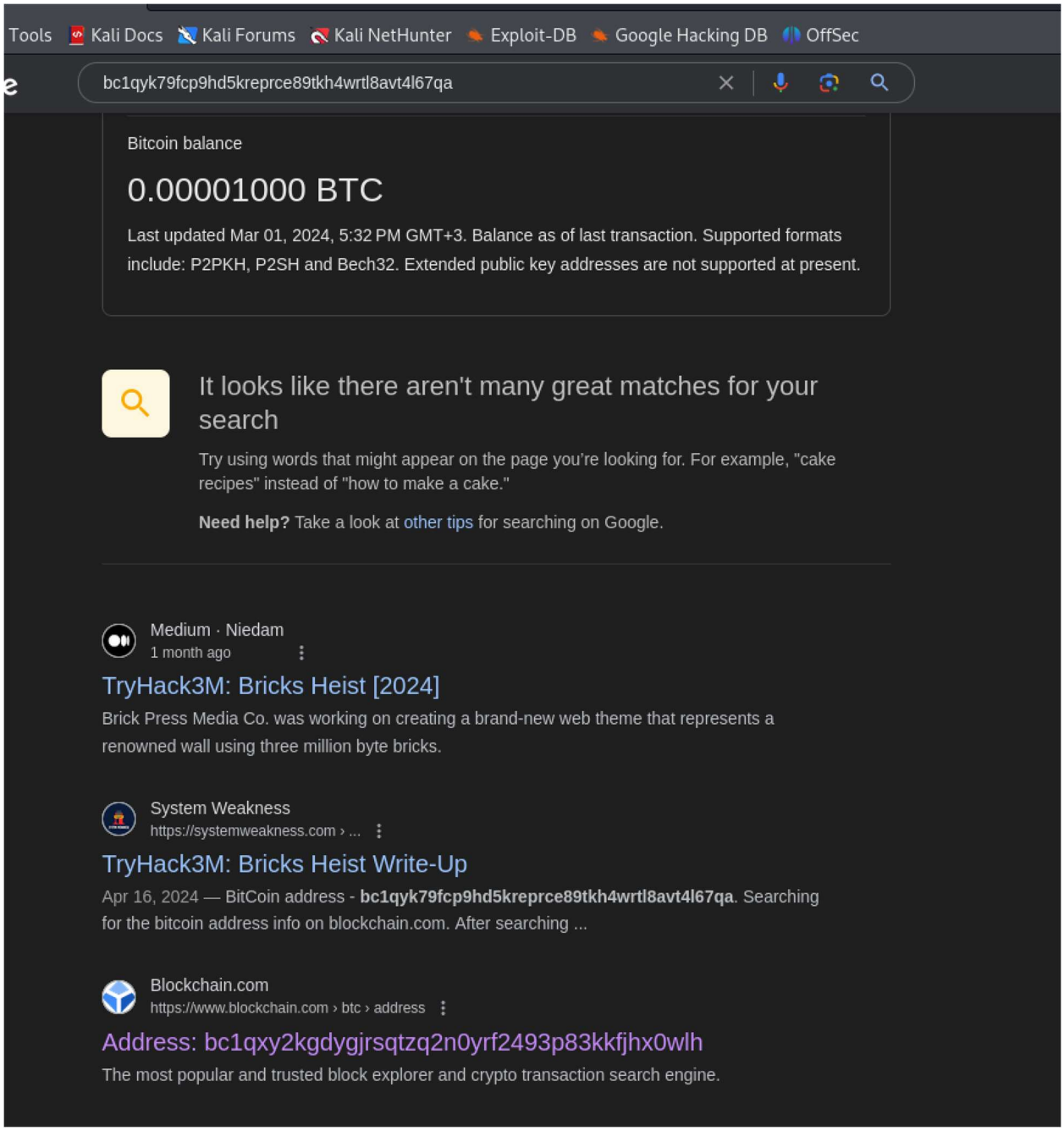
14. Running the log file, we will find what it does and an id which is ecoded.
    - The process is a miner
    - To find the Id we must decode it.
    - cat /lib/NetworkManager/inet.conf

```
Shell> cat /lib/NetworkManager/inet.conf
ID: 5757314e65474e596248 4a4f656d787457544e424e5746485554466 84d3070735930684b616c70555a7a566b5233527654 6b686b65575248647a525a57466f77546b64334d6b347a526d685a6255313459316
873636b35366247315a4d304531595564476130355864486c6157454a3557544a564e453959556e4a685246497a593235536330394852 6a4a6b52464a7a546d706b65466c525054303d
2024-04-08 10:46:04,743 [*] confbak: Ready!
2024-04-08 10:46:04,743 [*] Status: Mining!
2024-04-08 10:46:08,745 [*] Miner()
2024-04-08 10:46:08,745 [*] Bitcoin Miner Thread Started
2024-04-08 10:46:08,745 [*] Status: Mining!
2024-04-08 10:46:10,747 [*] Miner()
2024-04-08 10:46:12,748 [*] Miner()
2024-04-08 10:46:14,751 [*] Miner()
2024-04-08 10:46:16,753 [*] Miner()
2024-04-08 10:46:18,755 [*] Miner()
2024-04-08 10:46:20,757 [*] Miner()
2024-04-08 10:46:22,760 [*] Miner()
2024-04-08 10:46:24,762 [*] Miner()
ID: 5757314e65474e596248 4a4f656d787457544e424e5746485554466 84d3070735930684b616c70555a7a566b5233527654 6b686b65575248647a525a57466f77546b64334d6b347a526d685a6255313459316
873636b35366247315a4d304531595564476130355864486c6157454a3557544a564e453959556e4a685246497a593235536330394852 6a4a6b52464a7a546d706b65466c525054303d
2024-04-08 10:48:04,647 [*] confbak: Ready!
2024-04-08 10:48:04,648 [*] Status: Mining!
2024-04-08 10:48:08,649 [*] Miner()
2024-04-08 10:48:08,649 [*] Bitcoin Miner Thread Started
2024-04-08 10:48:08,649 [*] Status: Mining!
2024-04-08 10:48:10,651 [*] Miner()
2024-04-08 10:48:12,653 [*] Miner()
2024-04-08 10:48:14,656 [*] Miner()
2024-04-08 10:48:16,656 [*] Miner()
2024-04-08 10:48:18,659 [*] Miner()
2024-04-08 10:48:20,660 [*] Miner()
```

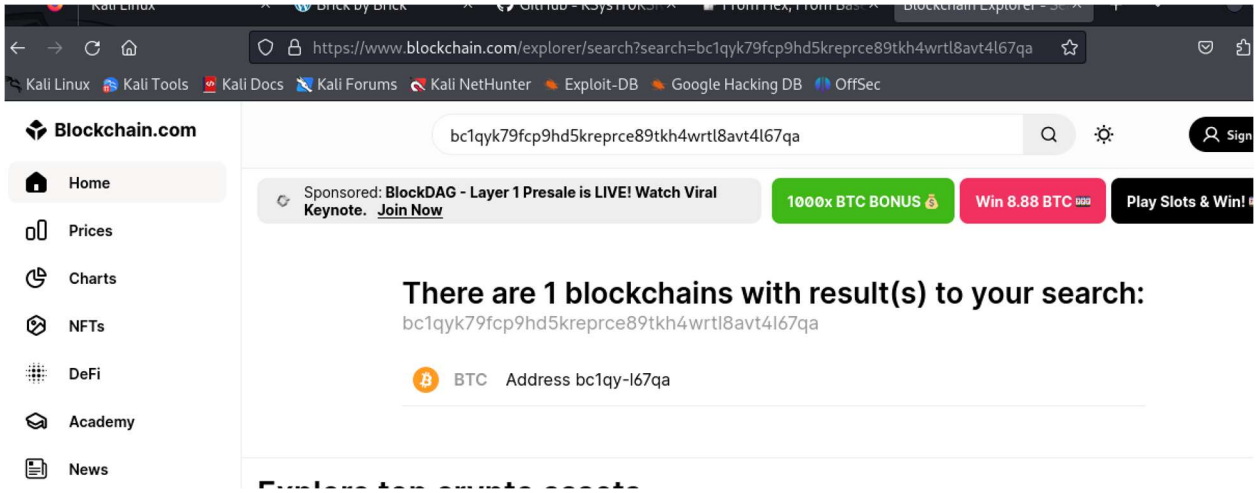
15. To decode the id, we will use cyberchef website
    - On decoding it we found two duplicate keys with one letter (a) added to the second key.



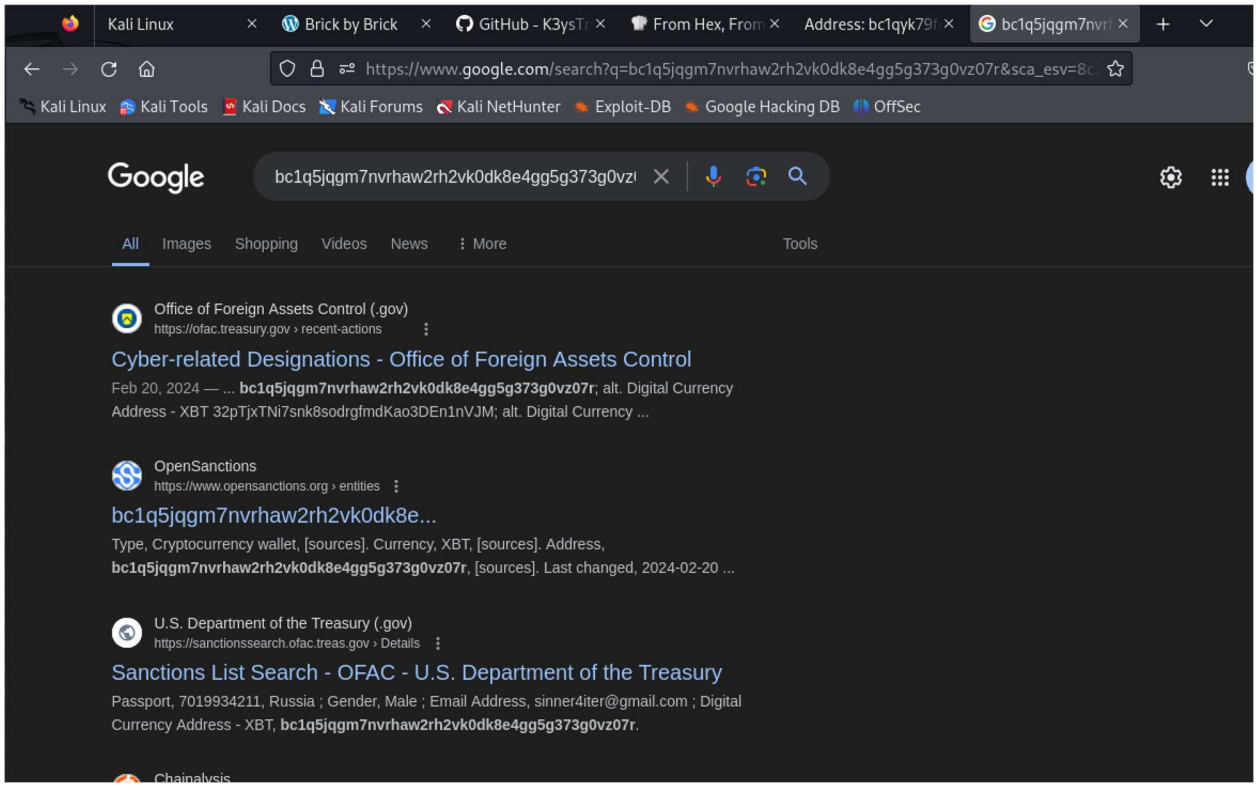
16. When we searched for the id we decoded it is a bitcoin wallet addres and we found the following search results.



- From the search results we see that the blockchain.com website is showing us the address so let's visit the website.
    - On visiting the website and searching for the address we found a match for the wallet address.



- Going to the address we found that there are seven transactions and going to the first transaction we found another address. Searching for this address gave us the following results:



- Opening the first link and going through it we found Russian-based hacker group by the name of **LockBit Ransomware**.

U.S. DEPARTMENT OF THE TREASURY

READ THE LATEST TREASURY NEWS

Office of Foreign Assets Control

A PART OF TREASURY'S OFFICE OF TERRORISM AND FINANCIAL INTELLIGENCE

ABOUT OFAC | RECENT ACTIONS | SANCTIONS LISTS | SANCTIONS PROGRAMS | SUBMIT A REPORT | CONTACT US | SEARCH

HOME > RECENT ACTIONS > CYBER-RELATED DESIGNATIONS

- Specially Designated Nationals List (SDN List)
- Consolidated Sanctions List (Non-SDN Lists)
- Additional Sanctions Lists
- Search OFAC's Sanctions Lists
- Sanctions Programs and Country Information
- Recent Actions
- OFAC License Application Page

Cyber-related Designations

02/20/2024

Press Release Link

United States Sanctions Affiliates of Russia-Based LockBit Ransomware Group

SPECIALLY DESIGNATED NATIONALS LIST UPDATE

The following individuals have been added to OFAC's SDN List:

FILTER BY CATEGORY

- All Recent Actions
- Enforcement Actions
- General Licenses
- Miscellaneous
- Regulations and Guidance
- Sanctions List Updates

17. Finally, here is the last screenshot for you. Thank you.....

Title	Target IP Address	Expires
Brick.By.Brick.v.2.1	10.10.194.219	20min 7s

? | Add 1 hour | Terminate

THM{fl46_650c844110baced87e1606453b93f22a} — Correct Answer

What is the name of the suspicious p

nm-inet-dialog — Correct Answer

What is the service name affiliated w

ubuntu.service — Correct Answer

What is the log file name of the mine

inet.conf — Correct Answer

What is the wallet address of the mi

bc1qyk79fcp9hd5kreprce89tkh4w — Correct Answer

The wallet address used has been in

LockBit — Correct Answer

Congratulations!

You've completed the room! Share this with your friends:

Twitter | Facebook | Linkedin

Leave feedback