Recon:

We run our nmap scan with the command `nmap -sV -sC -T4 10.10.70.161` in which our results come back fairly quickly, which shows us that port 22 and port 80 are open:

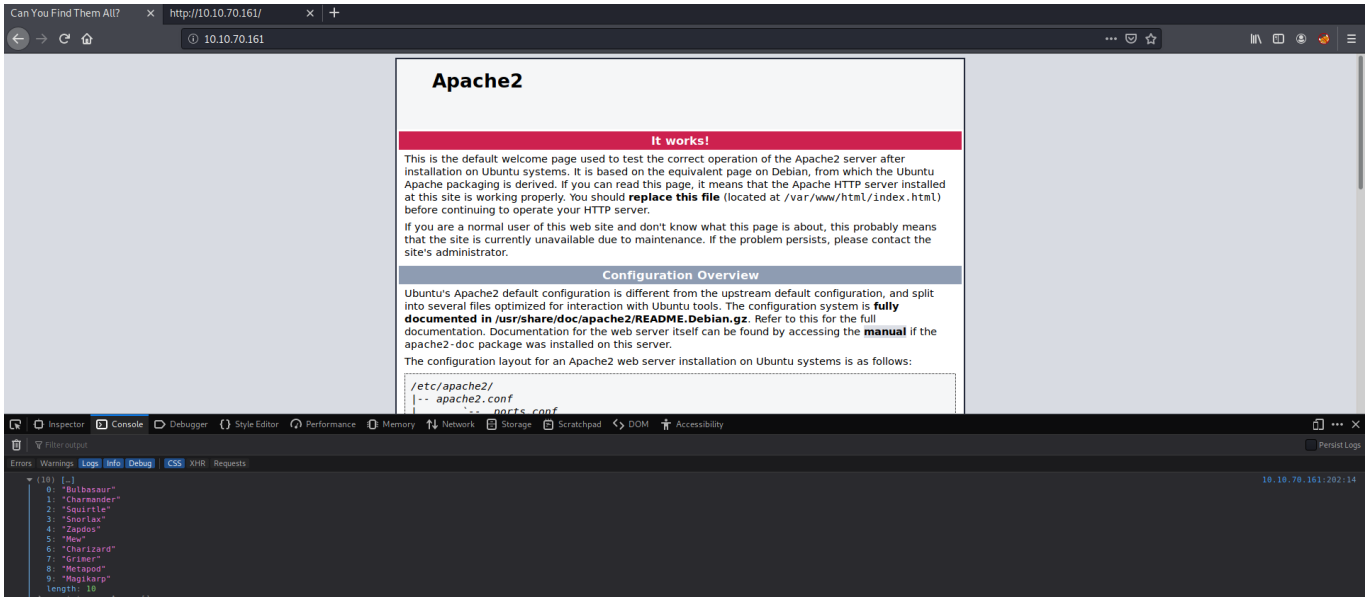
```
blackout@kali:~$ nmap -sV -sC -T4 10.10.70.161
Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-07 21:49 UTC
Nmap scan report for 10.10.70.161
Host is up (0.028s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 58:14:75:69:1e:a9:59:5f:b2:3a:69:1c:6c:78:5c:27 (RSA)
|   256 23:f5:fb:e7:57:c2:a5:3e:c2:26:29:0e:74:db:37:c2 (ECDSA)
|_  256 f1:9b:b5:8a:b9:29:aa:b6:aa:a2:52:4a:6e:65:95:c5 (ED25519)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Can You Find Them All?
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.41 seconds
blackout@kali:~$
```

Discovery:

Lets check out port 80 and see what we can find - We find an apache webpage which nothing looks out of place until we check the source of the page and see possibly the credentials for SSH with the user being `Pokemon` and the password being `hack_the_pokemon` but we also see a HTML comment saying to check the

console for an extar suprise, which is just a bunch of pokemon:

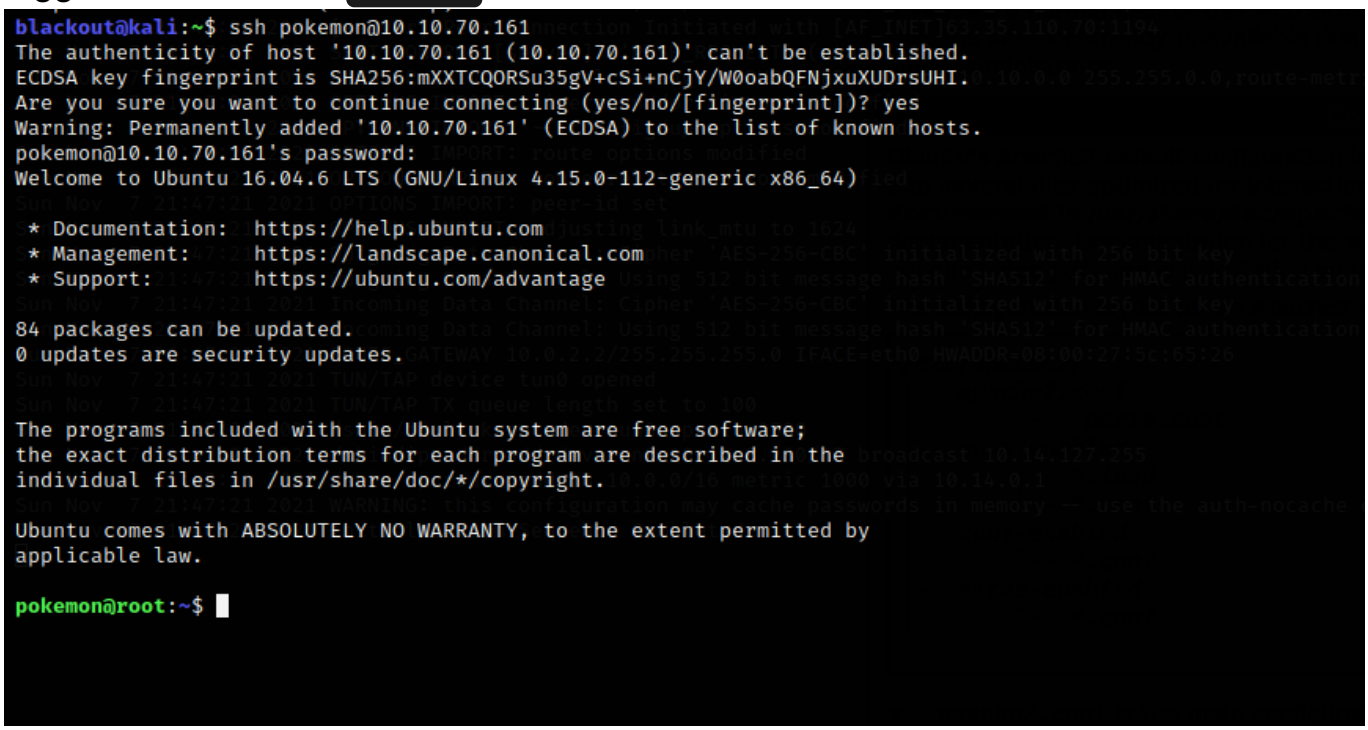
```
<div class="content_section_text">
    <p>
        By default, Ubuntu does not allow access through the web browser to
        <em>any</em> file apart of those located in <tt>/var/www</tt>,
        <a href="http://httpd.apache.org/docs/2.4/mod/mod_userdir.html">public_html</a>
        directories (when enabled) and <tt>/usr/share</tt> (for web
        applications). If your site is using a web document root
        located elsewhere (such as in <tt>/srv</tt>) you may need to whitelist your
        document root directory in <tt>/etc/apache2/apache2.conf</tt>.
    </p>
    <p>
        The default Ubuntu document root is <tt>/var/www/html</tt>. You
        can make your own virtual hosts under /var/www. This is different
        to previous releases which provides better security out of the box.
    </p>
</div>

<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>
```



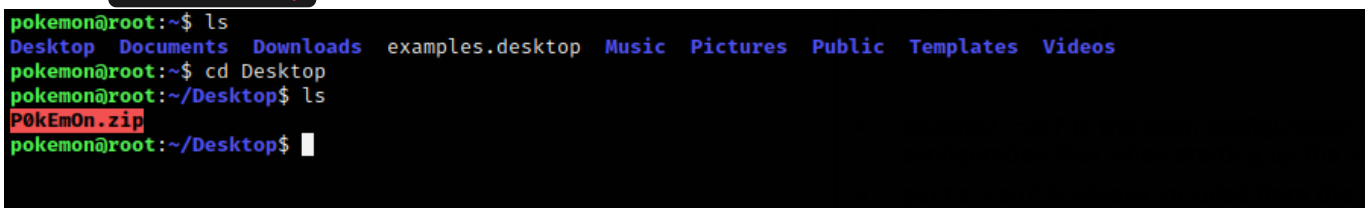
Foothold:

Now we try and login with the credentials we found earlier and we successfuly

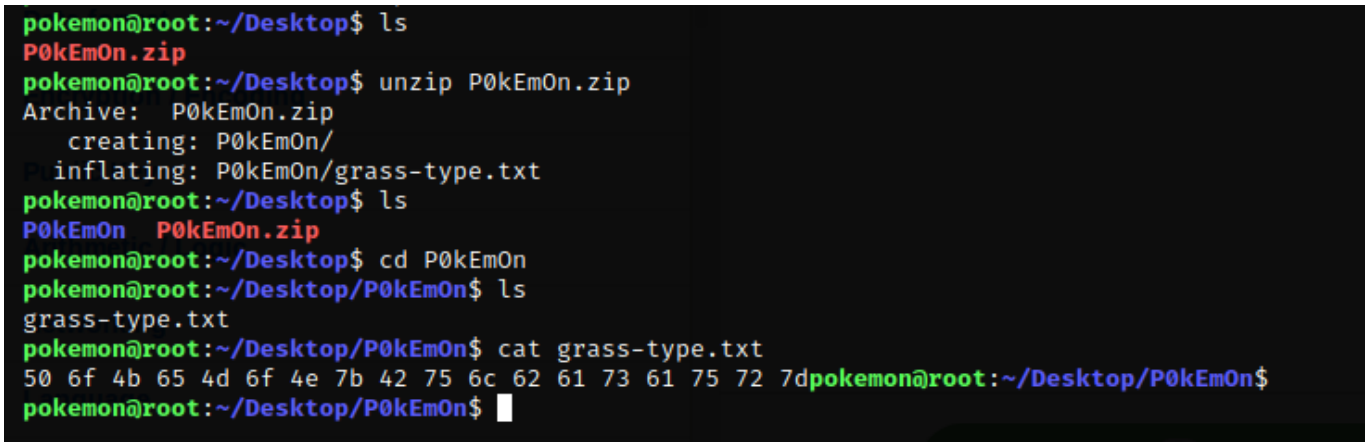
logged in to the user `Pokemon`:



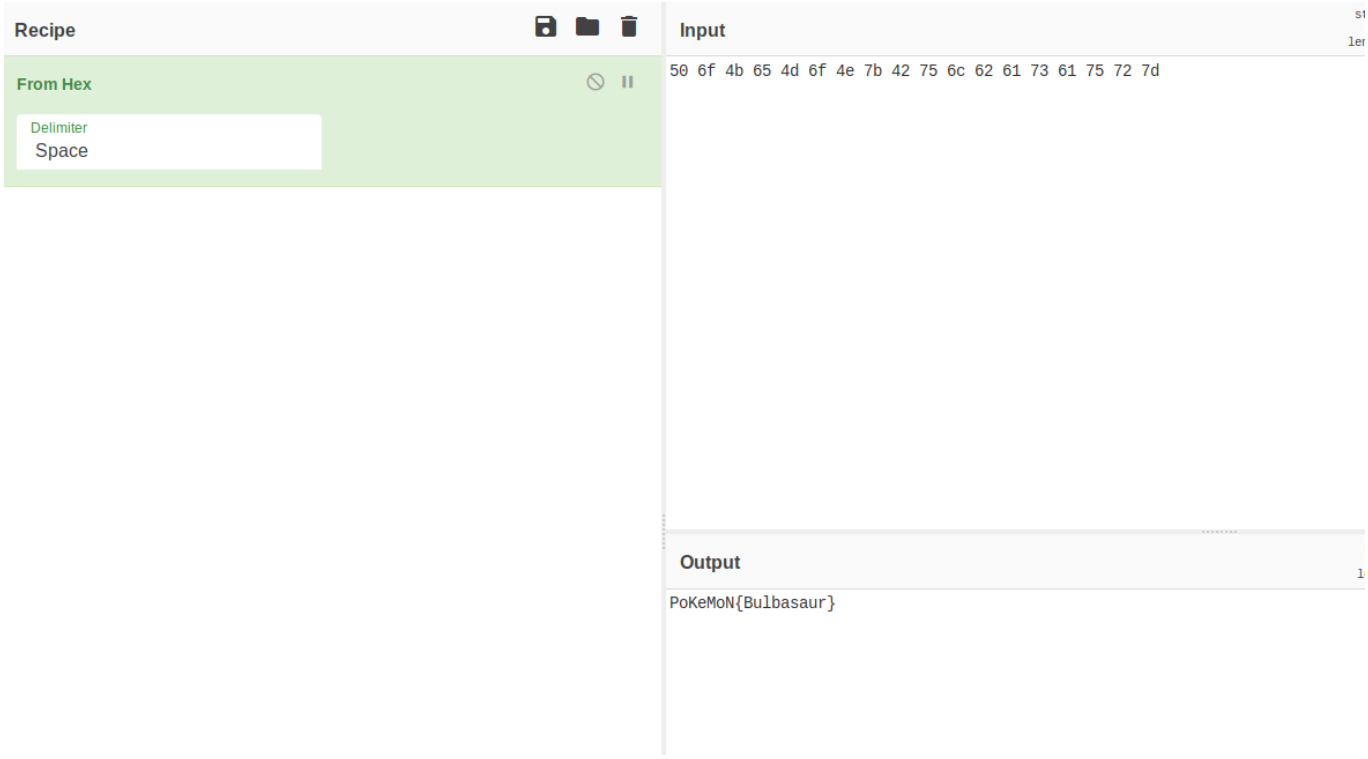
Using the command `ls` to list the folders/files in the current directory, we see a bunch of folders which we `cd` into the `desktop` directory and find an interesting file called `P0kEmOn.zip`:



Which we unzip this by using the command `unzip P0kEmOn.zip` - This creates another directory called `P0kEmOn` which we enter this directory as well and find a txt file called `graa-type.txt`, which we cat this to read and see what's inside the file - It apperas to be hex:

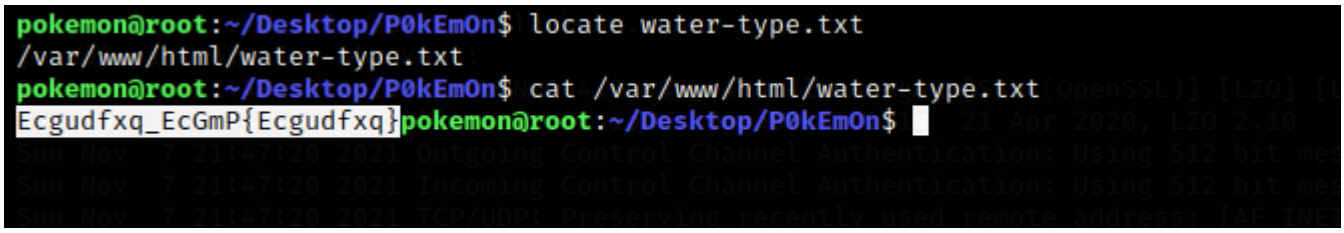


We then put this into cyberchef, which automatically decodes this for us and gives us our first flag:



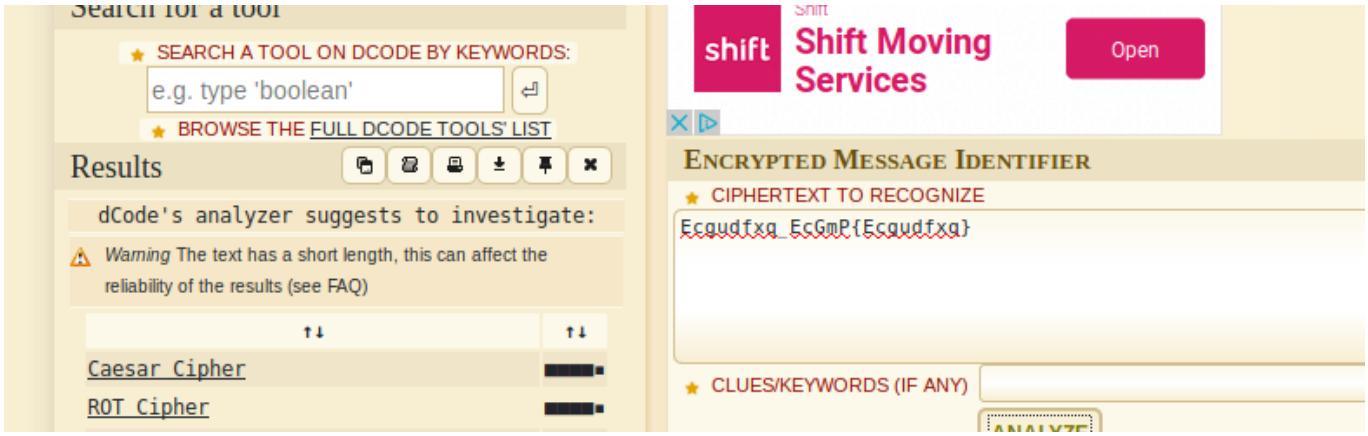
Find the Grass-Type Pokemon:
PoKeMoN{Bulbasaur}

I was wondering where the other flag may be and thought maybe it would have a similar name to the last flag so I used the command `locate` to find the flag and as this flag is located to water I tried this command `locate water-type.txt` and I was successful and managed to find the flag - It was in the `/var/www/html` which is the web directory and then we read the file - However it seems that the flag is encrypted:

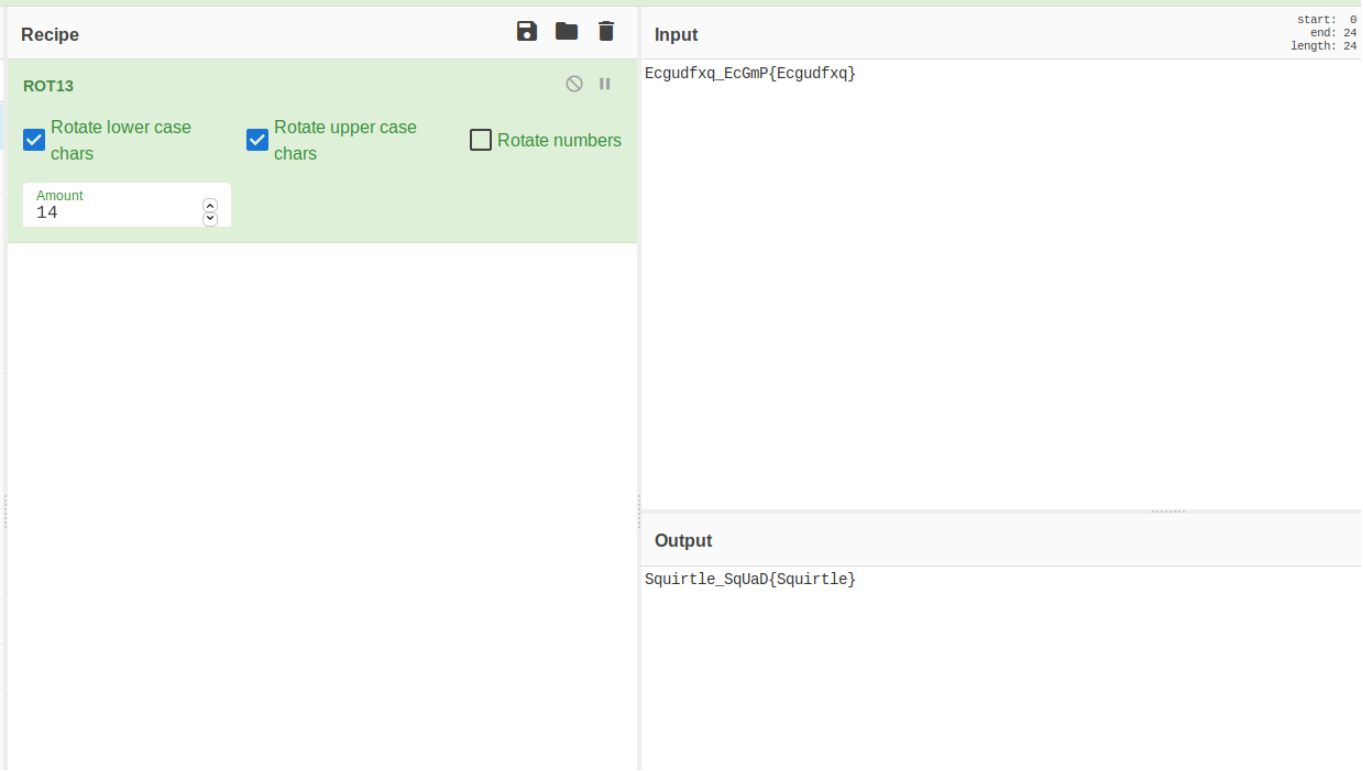


I tried putting it into `hash-identifier` but it said no hash found:

I put it into cyberchef but this time it couldn't automatically find the correct decoder/decipher then I realised that it wasn't a hash but a cipher so I searched up on Google cipher identifier and found out that it was ROT13:



Then I wnet back to cyberchef and used the recipe `ROT13` with the amount being `14` and we got oure second flag:



Find the Water-Type Pokemon:
Squirtle_SqUaD{Squirtle}

To find the next flag I do the exact same thing I did last time to find it by using the `locate` command - This type was fire so I used `locate fire-type.txt` and found it in an interesting directory `/etc/why_am_i_here?/fire-type.txt` - I then read the file and it seems that the flag is encoded with base64 - It was easy to identify that it was base64 due to the two `==` at the end of the encoding:

```
pokemon@root:/var/www/html$ locate fire-type.txt
/etc/why_am_i_here?/fire-type.txt
pokemon@root:/var/www/html$ cd /etc/why_am_i_here?/fire-type.txt
-bash: cd: /etc/why_am_i_here?/fire-type.txt: Not a directory
pokemon@root:/var/www/html$ cd /etc/why_am_i_here?
pokemon@root:/etc/why_am_i_here?$ ls
fire-type.txt
pokemon@root:/etc/why_am_i_here?$ cat fire-type.txt
UDBrM20wbntDaGFybWFuZGVyfQ==pokemon@root:/etc/why_am_i_here?$
```

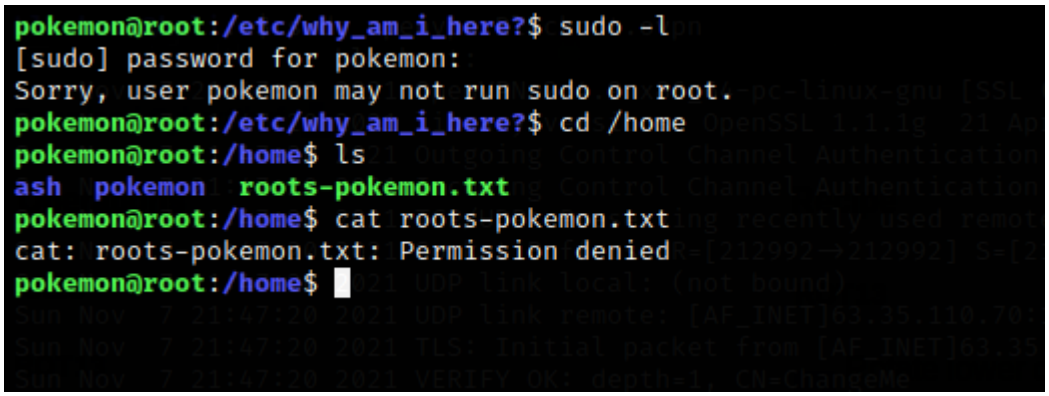
We then can use the command `cat fire-type.txt | base64 -d` which we'll decode the text and make us get our third flag:

```
pokemon@root:/etc/why_am_i_here?$ cat fire-type.txt
UDBrM20wbntDaGFybWFuZGVyfQ==pokemon@root:/etc/why_am_i_here?$ cat fire-type.txt | base64 -d
P0k3m0n{Charmander}pokemon@root:/etc/why_am_i_here?$
```

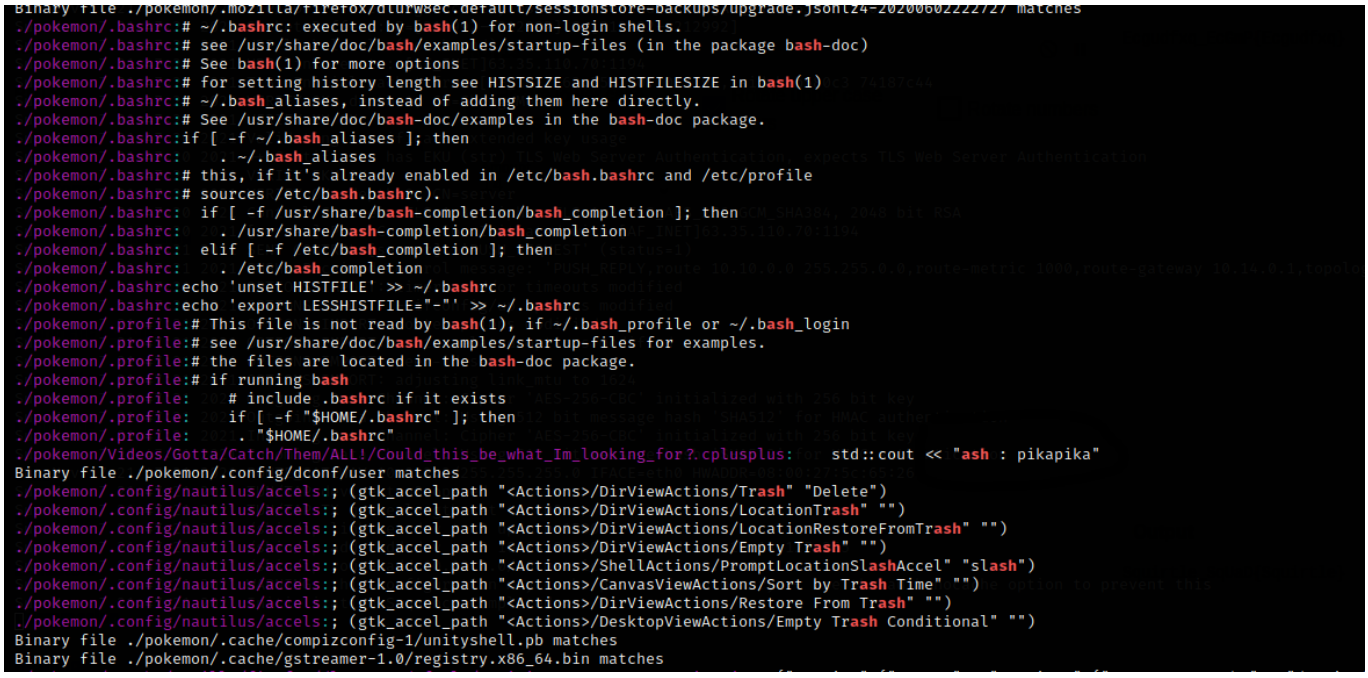
Find the Fire-Type Pokemon:
P0k3m0n{Charmander}

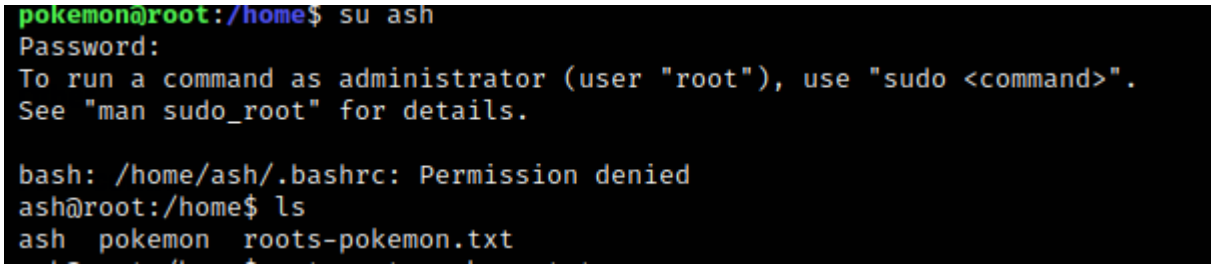
Privilege Escalation:

We now need to find our final flag so we go back to the `/home` directory which we find another user called `ash` and the last flag but however we are not able to read it as we are not root:

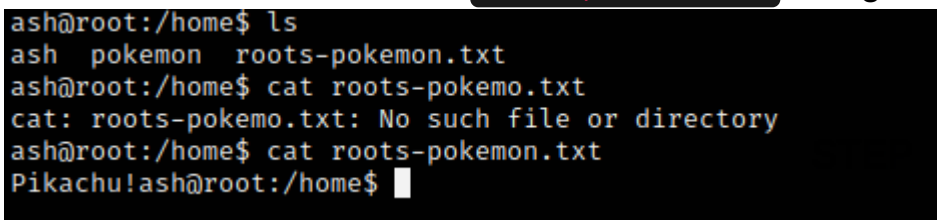


We can not cd into the ash directory either and we also get a `permission denied` when trying to enter the directory - As Ash is only able to reads this file we use the command `grep -r "ash" . 2>/dev/null` to show all his permissions and how we can get access to his account - Which this command successfully shows us his password and we are able to log into ash:





We are then able to read the `roots-pokemon.txt` and get our final flag:



Who is Root's Favorite Pokemon?:

Pikachu!