

Pokemon THM

Port scanning

```
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-title: Can You Find Them All?
|_ http-server-header: Apache/2.4.18 (Ubuntu)
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 8.71 seconds

Checking port 80 content

right button to see more options.

Apache2

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   `-- ports.conf
|-- mods-enabled
|   |-- *.load
|   `-- *.conf
|-- conf-enabled
|   `-- *.conf
|-- sites-enabled
|   `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/`

Something doesn't seem right, after trying some VHOST enumeration and content discovery i curled the page and saw this

```
</div>
<pokemon>:<hack_the_pokemon>
      <!--(Check console for extra surprise!)-->
</div>
```

Because of the format (username:password) i tried to login as pokemon as it worked

```
pokemon@root:~$ whoami
pokemon
```

Found the first flag at Desktop folder, after unzip the file in there

```
pokemon@root:~$ cd Desktop
pokemon@root:~/Desktop$ ls
P0kEm0n.zip
pokemon@root:~/Desktop$ unzip P0kEm0n.zip
Archive:  P0kEm0n.zip
  creating: P0kEm0n/
  inflating: P0kEm0n/grass-type.txt
pokemon@root:~/Desktop$ ls
P0kEm0n  P0kEm0n.zip
pokemon@root:~/Desktop$ cd P0kEm0n/
pokemon@root:~/Desktop/P0kEm0n$ ls
grass-type.txt
pokemon@root:~/Desktop/P0kEm0n$ cat grass-type.txt
50 6f 4b 65 4d 6f 4e 7b 42 75 6c 62 61 73 61 75 72 7d
```

It's cleary in hex format, after decrypt it i got the flag in plain text

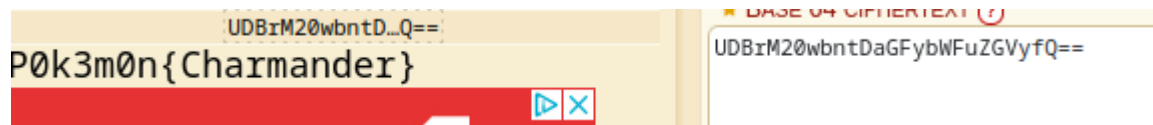
Hex	▼	To	Text	▼
50 6f 4b 65 4d 6f 4e 7b 42 75 6c 62 61 73 61 75 72 7d		PoKeMoN{Bulbasaur}		

Finding more pokemon flags

```
pokemon@root:~$ find / -type f -name "*-type.txt" 2>/dev/null
/var/www/html/water-type.txt
/etc/why_am_i_here?/fire-type.txt
/home/pokemon/Desktop/P0kEm0n/grass-type.txt
```

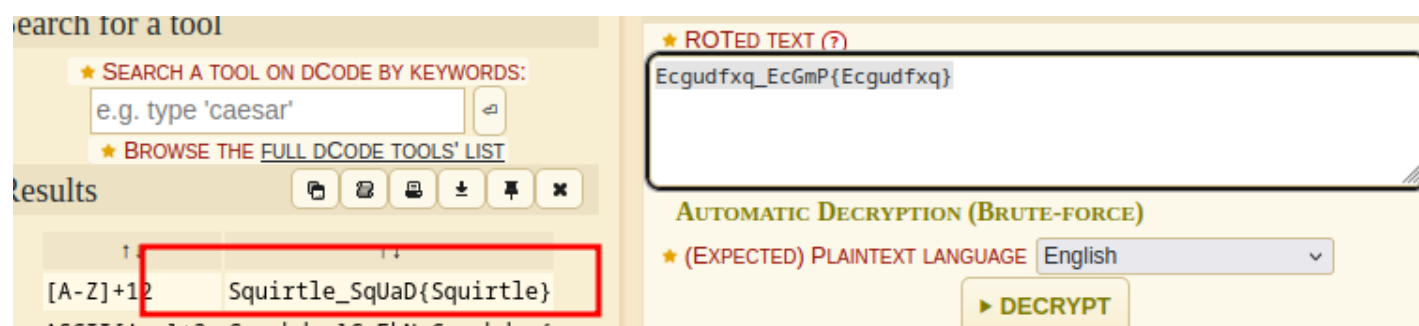
Fire pokemon flag is clearly in base64

```
pokemon@root:/etc/why_am_i_here?$ ls
fire-type.txt
pokemon@root:/etc/why_am_i_here?$ cat fire-type.txt
UDBrM20wbntDaGFybnRFuZGVyfQ==pokemon@root:/etc/why_am_i_here?$
```



Water pokemon flag looks like a ROT cipher, so let's try it out

```
pokemon@root:/$ cd /var/www/html/
pokemon@root:/var/www/html$ ls
index.html  water-type.txt
pokemon@root:/var/www/html$ cat water-type.txt
Ecgudfxq_EcGmP{Ecgudfxq}pokemon@root:/var/www/html$
```



After searching for a while i found a cpp file with interesting information

```
pokemon@root:~/Videos/Gotta/Catch/Them/ALL!$ ls
Could_this_be_what_Im_looking_for?.cplusplus
pokemon@root:~/Videos/Gotta/Catch/Them/ALL!$ _
```

```
# include <iostream>

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

Changing user to ash and then to root and obtaining the root's pokemon flag

```
pokemon@root:~/Videos/Gotta/Catch/Them/ALL!$ su ash
Password:
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

bash: /home/ash/.bashrc: Permission denied
ash@root:/home/pokemon/Videos/Gotta/Catch/Them/ALL!$ sudo su
root@root:/home/pokemon/Videos/Gotta/Catch/Them/ALL!#
```

```
ash pokemon roots-pokemon.txt
root@root:/home# cat roots-pokemon.txt
Pikachu!root@root:/home# Connection to
Connection to pokemon.thm closed.
```