

Curso CTF Competitivo

Presentación

Índice

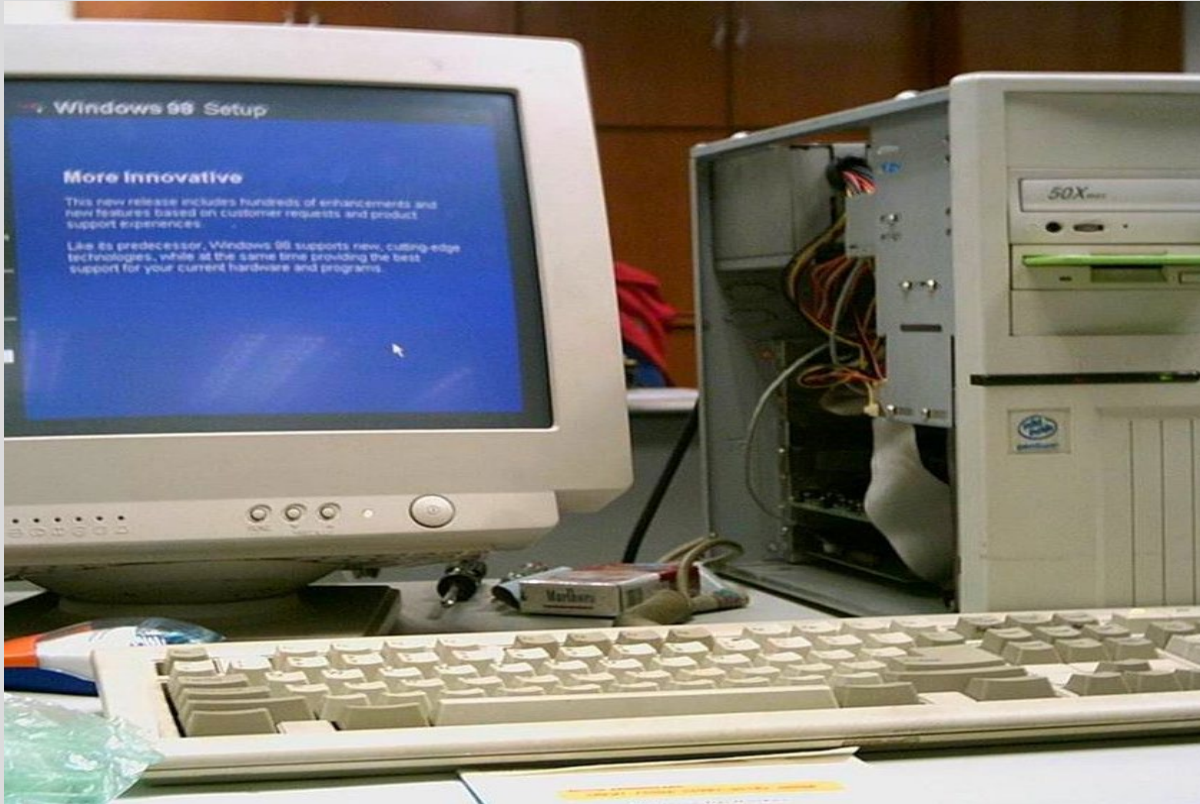
1. Horario de clase
2. Material básico necesario
3. Módulos del curso
4. Plataformas que vamos a utilizar

Horario y aula de clase



- 1 Octubre – 17 Diciembre
- Todos los viernes de **17:00 a 19:00**
- **Presencial:** Se anunciará el aula por correo cada semana
- **Online:** Sesión de Teams cada viernes en el mismo horario
- Información: <https://urjc-ctf.github.io/web/>
- Recomendado asistir a todas las sesiones (**en cualquier modalidad**)

Material básico necesario

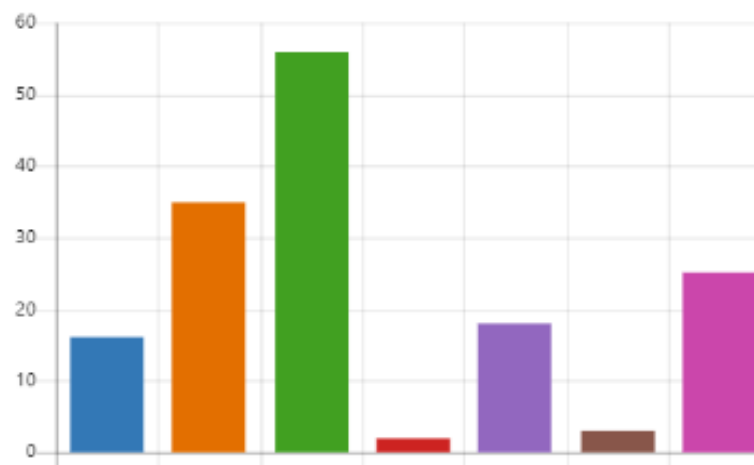
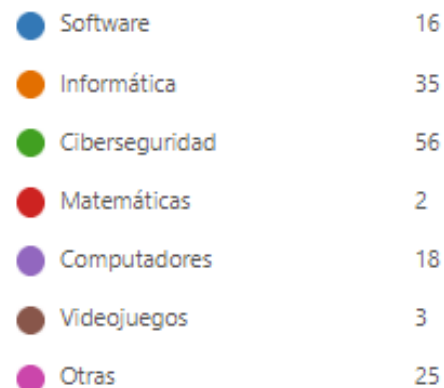


¿Qué es necesario para participar?

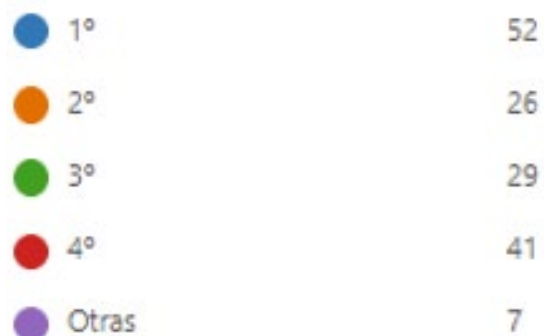
- Un ordenador con conexión a internet (obviamente)
- VirtualBox, para usar la imagen que os vamos a proporcionar
- Ganas de aprender
- Recomendado:
 - Conocimientos básicos en algún lenguaje de programación
 - Conocimientos básicos de Linux y/o terminal
 - Soltura utilizando los buscadores (Google, Bing, DuckDuckGo...)

¿Quiénes sois vosotros?

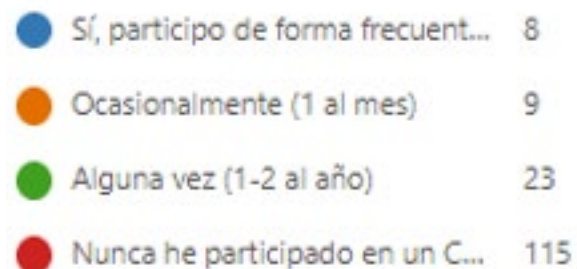
Grado en el que estás matriculado (u otros estudios)



Curso en el que estás matriculado



¿Has participado antes en CTFs?



Módulos del curso

Módulo I: Introducción a retos básicos

Módulo II: OSINT, Forense y Esteganografía

Módulo III: Ataques a servidores y explotación web

Módulo IV: Reversing y explotación de binarios

Plataforma de retos del curso

<https://ctf.numa.host/>

Credenciales en correo de bienvenida

Plataformas recomendadas

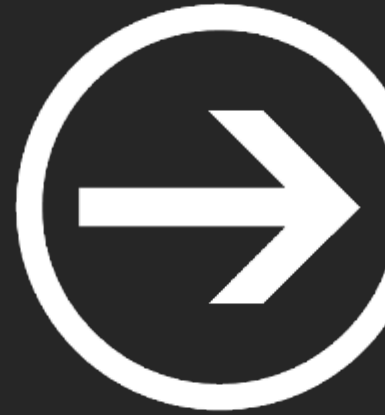
Pico CTF: <https://picoctf.org/>

OverTheWire: <https://overthewire.org/wargames/>

TryHackMe: <https://tryhackme.com/>

HackTheBox: <https://www.hackthebox.eu/>

Atenea: <https://atenea.ccn-cert.cni.es/home>



I. Introducción a los CTF y retos básicos

Alejandro Bermejo, Javier Sánchez García-Ochoa,
Inés Martín, Carlos Alonso y Sergio Pérez

Índice

1. ¿Qué es un CTF?
2. ¿Qué tipos de retos se encuentran en los CTF?
3. Conceptos básicos: encuentra la bandera
4. Criptografía y codificaciones básicas
 - Representación de los datos
 - Codificaciones y cifrados
 - Otros cifrados (XOR, Dcodefr...)
 - Hashes (MD5, SHA1, SHA256)
5. Retos básicos

¿Qué es un CTF?



Universidad
Rey Juan Carlos

¿Qué es un CTF?

Capture the flag experience



CTF = Capture The Flag (Captura la Bandera)

- Competición de **hacking**, en la que ponemos a prueba nuestras **habilidades resolviendo retos de ciberseguridad** en un **tiempo limitado**.
- Por **equipos o individual**

[illegible]

```
$ ./decrypt.py
onk.onk.onk.onk.onk.
onkHackOn{G00se_Game
}onk.onk.onkonk.onk.
onkonk.onk
```

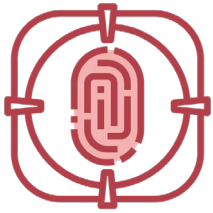
Tipos de CTF

Existen **3 tipos principales** de competiciones CTF:

1. **Jeopardy**: retos de **distintas categorías** a resolver en un **tiempo limitado**.
2. **Ataque – Defensa**: **2 equipos, 2 redes y servicios vulnerables** en cada red. Ambos equipos deben **atacar a los servicios del contrincante** a la vez que **defienden los suyos**.
3. **Boot2root**: **máquinas** creadas **con fallos** de seguridad que se deben **explotar** para convertirse en **superusuario (root)**.
4. **Mezcla**

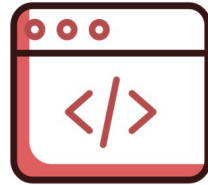


Categorías



Forense

Investigaciones sobre incidentes informáticos



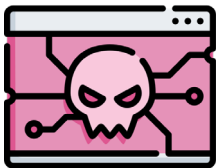
Web

Búsqueda y explotación de vulnerabilidades en aplicaciones web



Criptografía

Descifrado de mensajes ilegibles a simple vista



Reversing

Análisis del código de programas y ejecutables



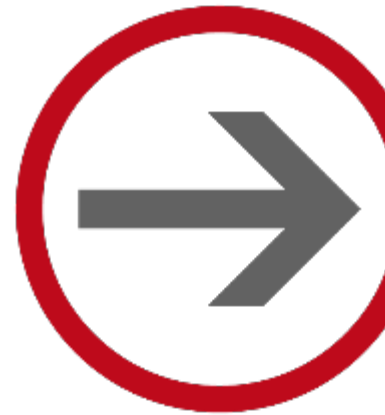
OSINT

Recolección de datos a través de fuentes públicas de información



Esteganografía

Técnica que oculta mensajes o archivos dentro de otros



CRIPTOGRAFÍA BÁSICA

Criptografía - Representación de los datos

Es esencial entender que **nos podemos encontrar los datos con diferentes formatos**. Sin embargo, **su significado será el mismo**. Las formas más comunes son:

ASCII

Relaciona caracteres con números. A cada carácter le corresponde un valor de la tabla ASCII.

Hexadecimal

Utiliza base 16 como representación de los datos. En caracteres toma como referencia el valor ASCII

ASCII	Símbolo
96	,
97	a
98	b
99	c
100	d
101	e
102	f
103	g
104	h
105	i
106	j
107	k
108	l
109	m
110	n
111	o

Criptografía - Representación de los datos

Es esencial entender que **nos podemos encontrar los datos con diferentes formatos**. Sin embargo, **su significado será el mismo**. Las formas más comunes son:

Binario

Es la representación más básica. Tan solo utiliza dos valores: 1 y 0.

DECIMAL	BINARIO	HEXADECIMAL
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Palabra

CTF{Bienvenidos}

ASCII

067 084 070 123
066 105 101 110
118 101 110 105
100 111 115 125

Binario

01000011 01010100 01000110
01111011 01000010 01101001
01100101 01101110 01110110
01100101 01101110 01101001
01100100 01101111 01110011
01111101

Hexadecimal

43 54 46 7b 42 69 65 6e
76 65 6e 69 64 6f 73 7d e2
80 8b



CyberChef: <https://gchq.github.io/CyberChef/>
Dcode.fr: <https://www.dcode.fr/>



Javier Sánchez, Alejandro Bermejo, Inés Martín, Carlos Alonso y Sergio Pérez

Criptografía - Codificaciones y cifrados

Algunas de las maneras más comunes de ocultar información son mediante **codificaciones y cifrados**. Esto consiste en utilizar una única clave para cifrar y descifrar la información. Por lo tanto, siendo el cifrado **reversible**.

Codificaciones

- Representan la misma información de diferentes maneras.
- Es reversible
- Algunos ejemplos son Base64, Base32 o ASCII

Cifrados

- Ocultan la información mediante claves, normalmente secretas, y un conjunto de operaciones.
- Es reversible
- Algunos ejemplos son ROT-N/César o Vigenère

Criptografía - Codificaciones y cifrados

BASE64

Es un sistema de **numeración posicional** que usa 64 caracteres como base. Sirve para representar cualquier información en binario como texto. **Se suele identificar rápidamente** por su estructura (en general, suelen acabar en ==)

Texto original

CTF{Esto es un texto en Base64. También existen otras como Base32, Base58 o Base85, por ejemplo}

Texto en Base64

QIRGe0VzdG8gZXMgdW4gdGV4dG8gZW4gYXNINjQulFRhbWJp6W4gZXhpc3RlbiBvdHJhcyBjb2IvIEJhc2UzMiwgQmFzZTU4IG8gQmFzZTglLCBwb3lgZWplbXBsb30=

Criptografía - Codificaciones y cifrados

ROT-N

Es un tipo particular de cifrado en el que los caracteres se desplazan N posiciones. Por ello, N será nuestra clave secreta que ayudará a cifrar y descifrar el texto. Además de conocer la clave, deberemos conocer el diccionario que se usa.

Texto original

CTF{El rot solo va a
modificar las letras, pero no
las llaves}

abcdefghijklmnopqrstuvwxyz

Texto en ROT 13

PGS{Ry ebg fbyb in n
zbqvsvpne ynf yrgenf, creb ab
ynf yynirf}

nopqrstuvwxyzabcdefghijklm

Cifrado de
César



Criptografía - Codificaciones y cifrados

Vigenère

Se basa en una **tabla con dos entradas**. Una será **la clave** y la otra **el texto a cifrar**. Iremos sustituyendo en el texto carácter a carácter con ayuda de la tabla y la clave. La clave será la misma para cifrar y descifrar.

Texto original

CTF{Mi clave de cifrado es
Chachipiruli}

Texto en Vigenère

EAF{Op kaimy om epfthld mj
Wsieoirpzjtz}

Criptografía - Codificaciones y cifrados

ENTRADA TEXTO PLANO		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
ENTRADA CLAVE	A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
	B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
	C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
	D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
	E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
	F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
	G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
	H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
	I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
	J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
	K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
	L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
	M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
	N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
	O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
	P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
	R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
	S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
	T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
	U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
	V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
	W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
	X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
	Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
	Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

CTF{Mi clave de cifrado es Chachipiruli}
 CHA{chhipiruliChachipiruliChachipiruli}
 EAF{Opkaimyom epfthldmjWsieoirpzjtz}

Texto original

CTF{Mi clave de cifrado es
Chachipiruli}

Texto en Vigenère

EAF{Op kaimy om epfthld mj
 Vsieoirpzjtz}

Criptografía - Otros cifrados

XOR

Consiste en cifrar siguiendo unas **reglas matemáticas** y una **clave secreta**. Como la **longitud** de la **clave** suele ser **menor al texto**, se repetirá **cíclicamente**. Todos los caracteres se pasarán a binario y se operará con ellos. Reglas:

1. Conmutativa: $A \text{ xor } B = B \text{ xor } A$

2. Asociativa: $(A \text{ xor } B) \text{ xor } C = A \text{ xor } (B \text{ xor } C)$

3. Autoinversa: $(A \text{ xor } B) \text{ xor } B = A$

<i>A</i>	<i>B</i>	XOR
0	0	0
0	1	1
1	0	1
1	1	0

Criptografía - Otras codificaciones

Tic-Tac-Toe

Texto original

CTF{Hay cifrados de todo
tipo}

Texto en Tic-Tac-Toe

L	⊠	□	□	└	◡	L	┐
□	⊗	└	□	⊗	⊠	□	□
⊠	⊗	□	⊗	⊠	┐	⊗	⊗

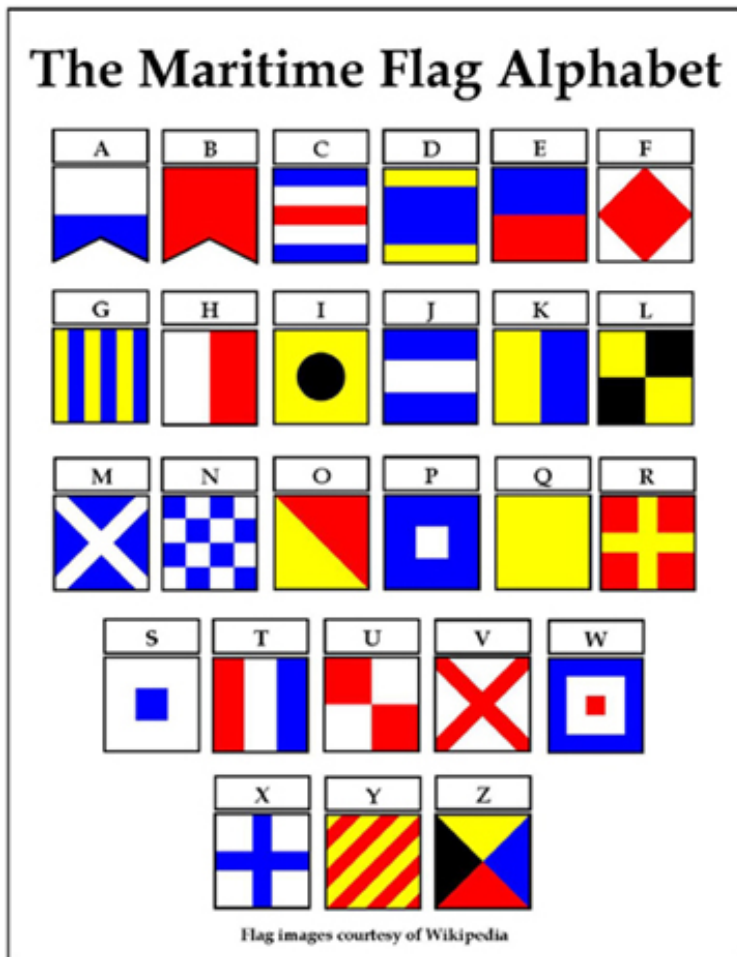


[DCode.fr: https://www.dcode.fr/chiffre-tic-tac-toe](https://www.dcode.fr/chiffre-tic-tac-toe)

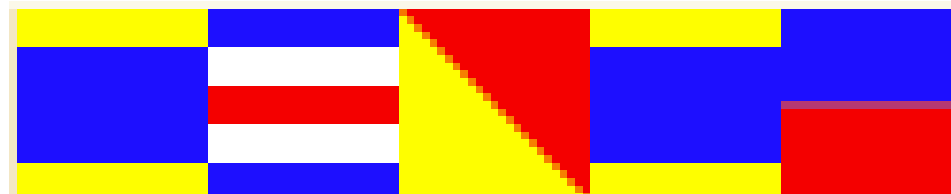


Criptografía – Otras codificaciones

Sustitución con Banderas marítimas



Texto original: DCODE



<https://www.dcode.fr/maritime-signals-code>



Operations

Search...

Favourites

Data format

To Hexdump

From Hexdump

To Hex

From Hex

To Charcode

From Charcode

To Decimal

From Decimal

To Binary

From Binary

To Octal

Recipe

From Binary

Delimiter

Space

Byte Length

8

From Base32

Alphabet

A-Z2-7=

☒ Remove non-alphabet chars

From Base64

Alphabet

A-Za-z0-9+/=

☒ Remove non-alphabet chars

From Hex

Delimiter

Auto

Input

01001001 01010100 01001100 01001000 01001100 01001010 01001100
01001010 01000001 01010111 01001111 01010100 01010100 01001011
01010011 01000110 01001101 00110101 01001000 01001000 01010101
01000111 01010111 01011010 00110010 01001111 01010000 01001010
01001001 01010110 01010100 01010101 00110100 00110011 01001011
01010010 01001100 01001000 01001010 01010110 01000100 01010101
01000011 01010111 01001111 01010100 01010011 01001000 01001011
01011010 01001101 00110101 01000111 01010111 01010101 01010001
01010011 01011010 00110010 01001111 01001110 01001010 01010110
01001110 01010100 01010101 00110100 00110011 01001011 01011010
01001100 01001000 01001010 01010110 01000100 01010101 01001011
01010111 01001111 01010100 01010011 01010101 01001010 01010110
01001101 00110101 01000111 01010111 01010101 01010001 01001100
01011010 00110010 01001111 01001110 01001010 01000011 01010111
01010100 01010101 00110100 00110010 01010100 01001100 01001101
01001000 01001100 01001010 01001011 01000101 01001011 01011010
01001111 01010100 01001100 01001011 01001001 01010110 01010100
00110101 01001000 01000110 01001001 01010110 01001100 01001000
00110010 01001111 01010000 01001010 01001011 01010111 01001111
01010101 00110100 00110110 01010011 01010110 01001101 00110101

Output

¿Estáis listos chicos?
¡Sí capitán!
¡No oigo!
¡Sí capitán!
Uuuuuh

Ej 1: VVJKQ3tNdXkgYmllbiwgdMvvlHFI ZSBzYVWJlcyBpZGVudGlmaWNhciBI biBiYXNINjR9

Ej 2: GXJ{Rs xshsw psw VSX wsr 57}

Ej 3: DIETr0RtqzlwMKZtoT9mVUEyrUEiplOyp3EuovOwnJMLJEiplOgLKZtMTHtqJ5uVUMyra0=

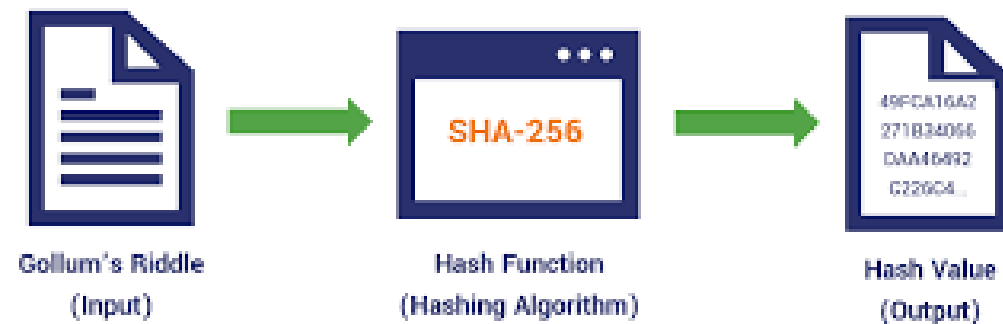
Ej 4: -.-. - ..- / - --- / / -.-. --- -... ---. --- / -- --- .-. /

Ejercicios propuestos

¿Qué es un hash?

- Es una **función matemática o criptográfica**, resume la información
- Da como **resultado** una cadena de caracteres de longitud fija (**digest**), **independientemente** de la longitud entrada
- Es **irreversible**. Una vez aplicada **no se puede obtener el valor inicial**.

How Hashing Works

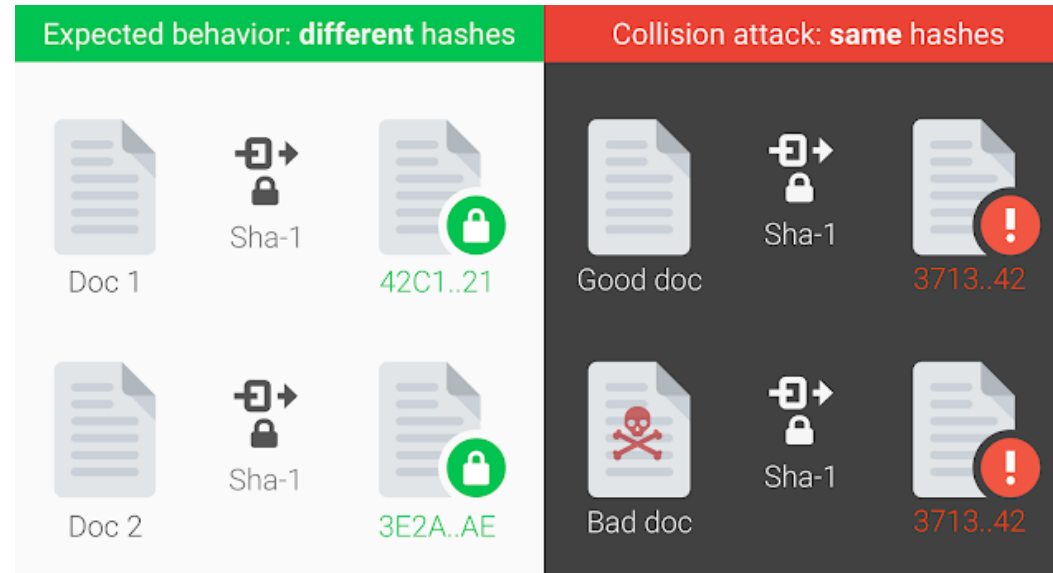


Criptografía - Hashes

- Lo que sí puede hacerse es **pre-computar** cadenas típicas, dado que una función hash devolverá el mismo resultado para la misma cadena (es determinista)
- **Conociendo la función utilizada** podemos realizar ataques de **fuerza bruta** sobre los hashes, de forma que, si en nuestro **diccionario** se encuentra la palabra *hasheada*, **sabremos qué esconde el hash**
- Es importante destacar que esto **NO ES LO MISMO QUE REVERTIR EL CÁLCULO**
- **Intentar adivinar un hash** de una palabra de longitud mayor que 8 es **computacionalmente muy costoso**

Criptografía - Hashes

- Existen determinadas funciones hash cuyo uso **no se recomienda**
 - **MD5**
 - **SHA1**
- Aunque la probabilidad es muy baja, podrían existir **colisiones**

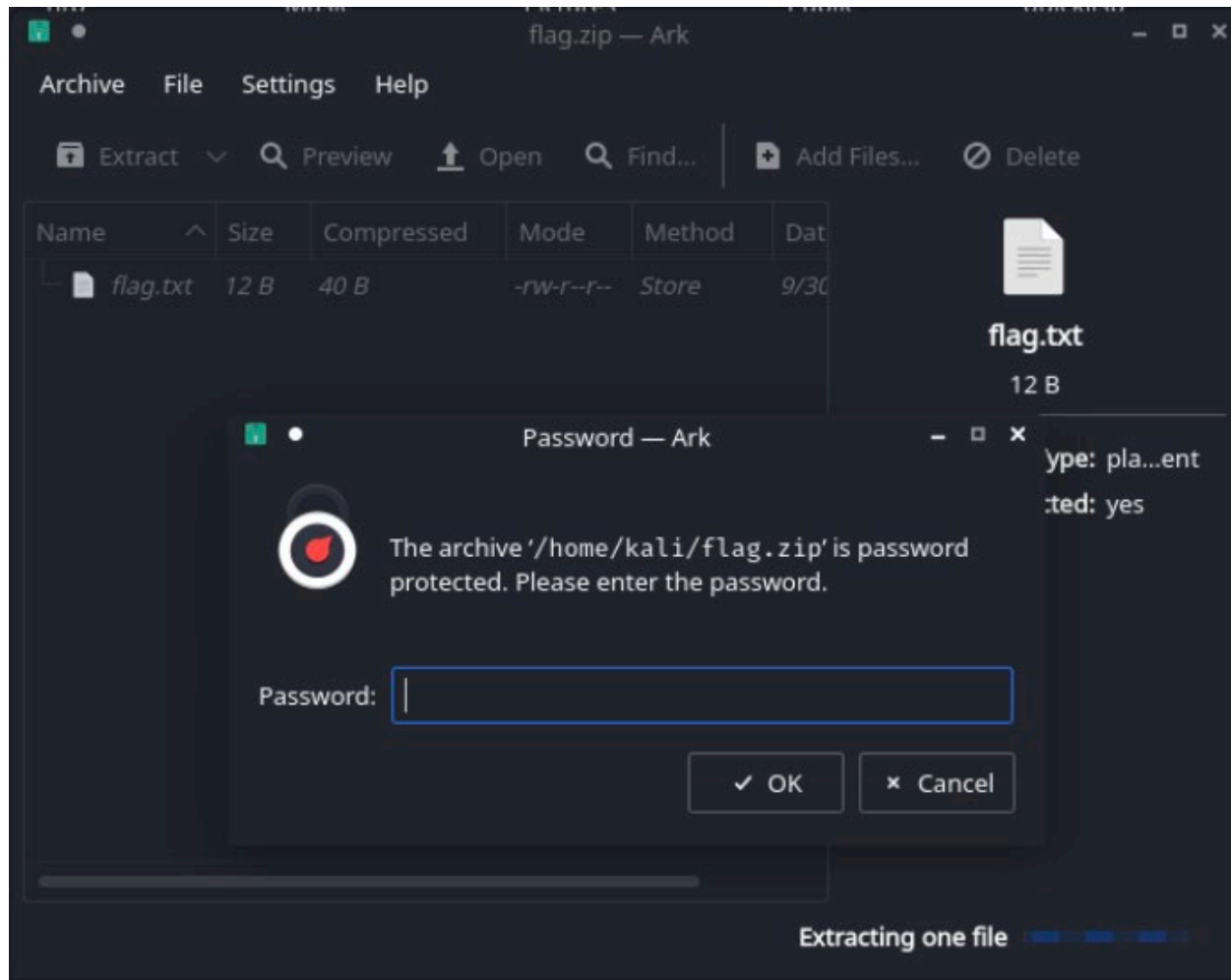


Criptografía - Hashes

- Cada **fichero** se puede resumir con un **valor hash**
- Existen herramientas que, dada una lista de **hashes**, nos automatizan el proceso de obtener un valor que genere dicho hash.
- **Esto permite obtener la contraseña de ficheros cifrados**



Criptografía – Hashes (Ejemplo)



Criptografía – Hashes (Ejemplo)



```
~: zsh — Konsole
File Edit View Bookmarks Plugins Settings Help
New Tab Split View Left/Right Split View Top/Bottom
(kali@kali)-[~]
$ zip2john flag.zip > hashZip
```

```
~: zsh — Konsole
File Edit View Bookmarks Plugins Settings Help
New Tab Split View Left/Right Split View Top/Bottom Load a new tab with layout 2x2 terminals
(kali@kali)-[~]
$ zip2john flag.zip | grep -E -o '(\$pkzip2\$.*\$/pkzip2\$) | (\$zip2\$.*\$/zip2\$)' > zipHash2hashcat
```

Criptografía – Hashes (Ejemplo)



```
~: zsh — Konsole
File Edit View Bookmarks Plugins Settings Help
New Tab Split View Left/Right Split View Top/Bottom
(kali@kali)-[~]
$ zip2john flag.zip > hashZip
```

```
~: zsh — Konsole
File Edit View Bookmarks Plugins Settings Help
New Tab Split View Left/Right Split View Top/Bottom Load a new tab with layout 2x2 terminals
(kali@kali)-[~]
$ cat hashZip | grep -E -o '(\$pkzip2\$.*\$/pkzip2\$)|(\$zip2\$.*\$/zip2\$)' > zipHash2hashcat
```

Criptografía – Hashes (Ejemplo)



```
~: zsh — Konsole
File Edit View Bookmarks Plugins Settings Help

New Tab Split View Left/Right Split View Top/Bottom

(kali@kali)-[~]
$ john hashZip --wordlist=wordlists.txt
Using default input encoding: UTF-8
Loaded 1 password hash (ZIP, WinZip [PBKDF2-SHA1 128/128 SSE2 4x])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 1 candidate left, minimum 16 needed for performance.
hola1234 (flag.zip/flag.txt)
1g 0:00:00:00 DONE (2021-09-30 16:55) 100.0g/s 100.0p/s 100.0c/s 100.0C/s hola1234
Use the "--show" option to display all of the cracked passwords reliably
Session completed

(kali@kali)-[~]
$ john hashZip --show
flag.zip/flag.txt hola1234:flag.txt:flag.zip:flag.zip

1 password hash cracked, 0 left

(kali@kali)-[~]
$
```

```
(kali@kali)-[~]
$ hashcat -m 13600 zipHash2hashcat ./wordlists.txt
hashcat (v6.1.1) starting... (hashcat starts with
```

```
Session.....: hashcat (uncompressed) hash number 17200
Status.....: Cracked
Hash.Name.....: WinZip
Hash.Target.....: $zip2$*0*3*0*f819c01513f1f5018f4e73128d711b52*8d6c* ... /zip2$
Time.Started.....: Thu Sep 30 16:59:35 2021 (0 secs)
Time.Estimated...: Thu Sep 30 16:59:35 2021 (0 secs)
Guess.Base.....: File (./wordlists.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 3 H/s (1.66ms) @ Accel:64 Loops:999 Thr:1 Vec:4
Recovered.....: 1/1 (100.00%) Digests: 1
Progress.....: 1/1 (100.00%)
Rejected.....: 0/1 (0.00%)
Restore.Point...: 0/1 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-999
Candidates.#1....: hola1234 -> hola1234

Started: Thu Sep 30 16:58:55 2021
Stopped: Thu Sep 30 16:59:37 2021

(kali@kali)-[~]
$ hashcat -m 13600 zipHash2hashcat --show
$zip2$*0*3*0*f819c01513f1f5018f4e73128d711b52*8d6c*c*327662bd488eec34fe3ad3fa*4b36073395bdba927dda*$/zip2$:hola1234

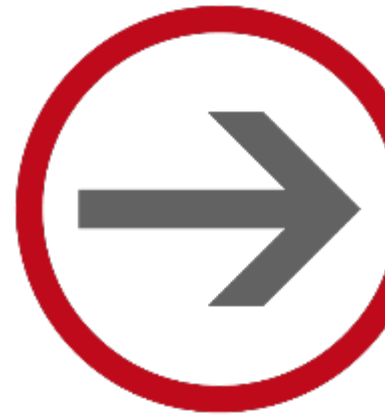
(kali@kali)-[~]
$
```

Criptografía - Hashes

- Como **atacantes**, esto nos viene bastante bien, dado que **podemos intentar encontrar colisiones que nos favorezcan**



- Como **defensores**, debemos utilizar siempre **funciones hash seguras**



RETOS BÁSICOS

Para practicar lo aprendido

- Para **practicar lo que hemos visto hasta ahora**, podéis realizar los **primeros 7 retos** de la categoría ***Básica*** de la plataforma **Atenea**

<https://atenea.ccn-cert.cni.es/challenges>

- Estos retos resumen **lo visto hasta ahora**
- La semana que viene, veremos **criptografía más avanzada**
 - **RSA, AES, etc.**



I. Introducción a los CTF y retos básicos

Alejandro Bermejo, Javier Sánchez García-Ochoa, Inés Martín, Carlos Alonso, Sergio Pérez



Universidad
Rey Juan Carlos