MOTA Yuri

CTF Night Hack: Documentation de conception des challenges personnels

/!\

Ce document décrit les étapes de conception de chaque challenges.

Tous les challenges présentés ici ont été conçus et rédigés par moi-même.

/!\

INTRODUCTION

Nom : ROT

Type : Cryptographie

Ce challenge a été construit en convertissant la valeur du flag en ROT47.

Nom : Fingerprint

Type : Forensic

Pour réaliser ce challenge j'ai créé:

- Un dossier "sk" contenant le fichier PNG avec le flag, "flag.png".
- Une image servant à cacher le flag, "basic-image.jpg".
- Un dossier ZIP avec le dossier sk à l'intérieur.

Nom : Pastebin

Type : OSINT

J'ai ici créé le pastebin contenant le flag en ajoutant les tags nécessaires.

Nom : Hello world !

Type : Reverse

J'ai créé un fichier C en mettant le flag dans un char* et non un tableau pour qu'il soit moins apparent à la décompilation par le challenger, puis je l'ai compilé:

```c
#include <stdio.h>


int main() {
```

```
    char *flag = "NHM2I{simpl3_introduction_to_r3v3rs3}";

    printf("Hello,"" World!");

    return 0;

}
```

FACILE

Nom : XOR

Type : Cryptographie

Input : 674

```
# Made by Yur0k


# key is a precomposed character with 3 compositions (unicode).

# first_composition = '0x6f' (U+006F)

# To determine second_composition, add '0x293' to first_composition value.

# To determine third_composition, add '0x1' to second_composition value.


encrypted = 'ỷ□ảụẞẬắẸấẈẾẤẈạỤấẮẈẨẤỤằẨẻẈẾẹẈẪấẮầảẸẪ'


compositions_sum = input('Entrez la somme des 3 compositions : ')

hex_to_add_to_compositions = '0x1863'


key = chr(int(compositions_sum, 16) + int(hex_to_add_to_compositions, 16))

print('key :', key)


decrypted = ''.join([chr(ord(x) ^ ord(key)) for x in encrypted])

print(decrypted)
```

Entièrement créé par ma part, j'utilise un chiffrement XOR pour déchiffrer mon message avec la clé qui doit être trouvée par le challenger.

Nom : Data stream CAPture

Type : Forensic

J'ai ici utilisé une capture de paquets située dans un fichier PCAP.

J'ai converti le flag présent dans le flux 5 d'un paquet (TCP) en base64 (ajouté avec "hexeditor").

Nom : Overpass

Type : OSINT

L'idée de ce challenge m'est venue d'une conférence (SeaSea 2k22) dans laquelle l'outil Overpass et l'API utilisée ont été présentés.

J'ai donc repéré l'endroit où j'allais utiliser le code me servant de flag. Je me suis renseigné pour savoir comment créer la requête nécessaire pour le trouver.

Puis j'ai ajouté la photo et les éléments, docs etcetera dans un document de prise en main du challenge.

/!\

Le challenge qui suit a été annulé car initialement prévu pour un seul conteneur pour tous les challengers.

Le risque étant de donner l'accès au fichier /etc/passwd qui permettrait de changer le mot de passe root avec n'importe quel binaire avec le SUID actif, ayant comme propriétaire root et d'avoir un accès complet au conteneur commun (compromission, évasion).

/!\

Nom : UNIX command abusing SUID

Type : PWN

Ici j'ai monté un conteneur avec l'utilisateur root et user1.

J'ai activé le SUID sur la commande "cp", mis les bonnes permissions pour chaque fichier donc le flag et tester le bon déroulement du challenge pour devenir root.

Le challenge suivant vient remplacer le précédent :

Nom : Wrap de valeur, sauce overflow

Type : PWN

Le script est déjà détaillé dans le walkthrough (montrant comment le challenger l'aurait analysé pour réussir le challenge).

Nom : Some base

Type : Reverse

Pour le principe voir le walkthrough dédié.

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


void some_base_to_string(uint8_t* msg, size_t msg_sz, uint8_t* some_base,
size_t some_base_sz)
{
   memset(msg, '\0', msg_sz);
   if (some_base_sz % 2 != 0 || some_base_sz/2 >= msg_sz)
      return;


   for (int i = 0; i < some_base_sz; i+=2)
   {
      uint8_t msb = (some_base[i+0] <= '9' ? some_base[i+0] - '0' :
(some_base[i+0] & 0x5F) - 'A' + 10);
      uint8_t lsb = (some_base[i+1] <= '9' ? some_base[i+1] - '0' :
(some_base[i+1] & 0x5F) - 'A' + 10);
      msg[i / 2] = (msb << 4) | lsb;
   }
}


int main() {
   int res;
```

```
   printf("Flag:\n");
   scanf("%x", &res);

   if (res == 0x466C6167)
   {
      uint8_t some_base_flag[] =
"4e484d32497b746869735f69735f615f67723361745f72337633727333217d";
      uint8_t flag[31];

      some_base_to_string(flag, 38, some_base_flag, 62);
      printf("Le flag est:\n");
      printf(flag);
   }

   return 0;
}
```

Ici la fonction “some_base_to_string” utilise deux bases pour la conversion vers un “string”.

Les 4 paramètres sont la taille de la variable d’entrée, sa valeur et de même pour la variable de sortie.

J’ai ensuite utilisé un algorithme préexistant que j’ai adopté (avec un décalage bit à bit).

Nom : Flag Master Chief

Type : Web

Ici j’ai créé mon site (statique) et mis en place sur le port 15003 (port de la plage qui m’est attribué).