

TP Offuscation Guillaume Sanchez

Utilisation de Ghidra pour décompiler et analyser le code de deux programmes afin de récupérer des flags.

Prog2.exe

Une fois décompilé, on peut voir dans la fonction “main” qu'une fonction qui se nomme “verifier_cle” est appelée. Elle a en second argument “compare”

```
int __cdecl main(int _Argc, char **_Argv, char **_Env)
{
    int iVar1;

    __main();
    iVar1 = verifier_cle(_Argv[1], "compare");
    if (iVar1 == 0) {
        printf("Incorrecte.\n");
    }
    else {
        printf("Correcte, bravo !\n");
    }
    return 0;
}
```

Quand on essaye “compare” comme flag, le programme nous retourne correct :

```
C:\Users\Nk\Downloads>prog2.exe compare
Correcte, bravo !
```

Prog3.exe

Une fois décompilé, on peut voir dans la fonction “main” plusieurs chose intéressante.

- Dans un premier temps, nous avons la chaine de caractère “pvdsux”.

- Dans un second temps, nous avons la fonction “deobfusquer” qui attend 3 arguments.

- Dans un troisième temps, la fonction “verifier_cle” qui attend 2 arguments.

Si on regarde de plus près “deobfusquer”:

```
1
2 int __cdecl main(int _Argc, char **_Argv, char **_Env)
3
4 {
5     bool bVar1;
6     undefined7 extraout_var;
7     char local_78 [104];
8     char local_10 [8];
9
10    __main();
11    builtin_strncpy(local_10, "pvdsux", 7);
12    local_10[7] = 1;
13    deobfusquer((longlong)local_78, local_10, 1);
14    bVar1 = verifier_cle(_Argv[1], local_78);
15    if ((int)CONCAT71(extraout_var, bVar1) == 0) {
16        printf("Incorrecte.\n");
17    }
18    else {
19        printf("Correcte, bravo !\n");
20    }
21    return 0;
22 }
23
```

```
1
2 void deobfusquer(longlong param_1, char *param_2, byte param_3)
3
4 {
5     size_t sVar1;
6     int local_c;
7
8     for (local_c = 0; param_2[local_c] != '\0'; local_c = local_c + 1) {
9         *(byte *)(param_1 + local_c) = param_2[local_c] ^ param_3;
10    }
11    sVar1 = strlen(param_2);
12    *(undefined1 *)(sVar1 + param_1) = 0;
13    return;
14 }
15
```

On remarque une boucle qui s’exécutera tant que “param_2” à un caractère différent de “rien” (“\0”). Après plusieurs recherches, j’ai compris qu’il s’agissait de XOR obfusqué avec une clé de 1 (le 3ème argument entré dans la fonction “deobfusquer”). Autrement dit, on suit l’algorithme de XOR pour pouvoir retrouver notre flag.

Si on prend converti “pvdsux” en ASCII, cela donne 112, 118, 100, 115, 117 et 120.
En donnant à un convertisseur les ASCII et la clé 1, j’ai pu trouver “qwerty”.

```
C:\Users\Nk\Downloads>prog3.exe qwerty  
Correcte, bravo !
```