gitlab.enssat.fr

Alexandre DOYEN / CTF Logiciels sécurisés · GitLab

3-4 minutes

My First Serial

1) Description des outils permettant la comprehension du fonctionnement de l'application

En utilisant le logiciel Ghidra, qui fait office de décompilateur et de désassembleur, il est possible de comprendre en profondeur le fonctionnement du logiciel à analyser. Ceci est une méthode d'analyse statique.

En complément de celui-ci, dans le cadre d'une analyse dynamique, nous avons utilisé le logiciel "GDB" (GNU DeBugger), muni de l'extension "Gef" qui permet d'améliorer son utilisation en affichant bon nombre d'informations utiles, telles que le contenu des registres, ou encore le sommet de la pile, et ce, après chaque point d'arrêt défini.

2) Analyse du code décompilé

Tout ce qui suit a été déduit à partir de la décompilation[1]. En effet, dans le binaire fourni, il n'y a aucun symbole.

Cette fonction correspond au point d'entrée du programme. Celà se déduit assez facilement en inspectant les paramètres qu'elle prend : Un entier, et un taleau de chaînes de caractères. Ceci se confirme en mettant un point d'arrêt sous GDB à l'adresse `0x08049827` : On remarque que dans la pile, on trouve le premier paramètres, qui est le nombre de paramètres passés au programme, et le second, qui correspond au tableau de chaînes de caractères qui contient les paramètres eux-mêmes (Le chemin vers le binaire lui-même, puis le numéro de série passé en paramètre par l'utilisateur).

Lorsque le programme est appelé avec un argument, est appelée la fonction nommée "checkToken" qui vérifie que le numéro de série est valide selon le patron suivant : 8 section séparées par le caractère "_".

Si le numéro de série est sous le bon format, alors est générée une clé pour chaque section selon la formule suivante : "$z = \oplus_{i=0}^{n} a_i$", où z est la clé générée, a_i le $i$ème caractère de la section courante, et n le nombre de caractères de la section courante.

Ainsi, la vérification finale consiste en une vérification de chaque segment selon la formule $a_i \oplus z = b_i$, où a_i correspond au caractère i du segment courant, z à la clé générée précédemment, et b_i à la valeur de référence.

Si tous les blocs sont conformes, alors le numéro de série entré est valide !

3) Comment avons-nous trouvé le numéro de série ?

À l'aide des informations trouvées précédemment, il devient aisé de trouver la clé z par force brute, et ce, huit fois. Ainsi, la clé était `F1nd_th3_seri4l_number_is_so_easy_:)`.[2]

4) Notes

1. Les codes décompilés sont fournis en annexe de ce Writeup. ↩
2. Le script Python ayant servi à réussir le challenge est également fourni en annexe du Writeup. ↩