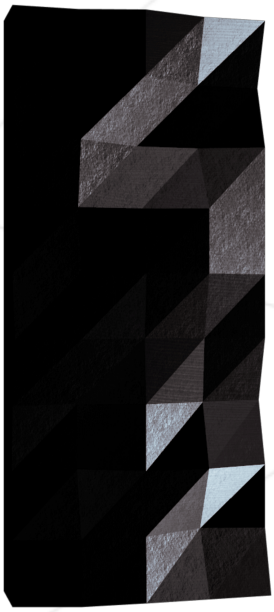

RUSH

Reverse me i'm famous !

wandre wandre@student.42.fr
42 Staff pedago@42.fr

Résumé: Simple rush pour découvrir la passion du reverse !

Table des matières

I	**Préambule**	**2**
II	**Introduction**	**3**
III	**Objectifs**	**4**
IV	**Consignes générales**	**5**
V	**Partie obligatoire**	**6**
VI	**Partie bonus**	**7**
VII	**Rendu et peer-évaluation**	**8**

Chapitre I

Préambule

There is something wrong..

Chapitre II

Introduction

Chapitre III

Objectifs

Ce rush a pour but de vous faire découvrir l'art du reverse engineering.

Vous allez pouvoir comprendre le fonctionnement d'un programme pour reformer sa source avec un simple binaire basique sous une architecture i386 !

Chapitre IV

Consignes générales

- Ce projet ne sera corrigé que par des humains.
- Vous disposez de 3 binaires (i386) executables avec chacun une difficulte différente, votre but sera de simplement reverse chaque binaire pour trouver un password et ainsi débloquer celui-ci.
- Pour se faire vous allez devoir donner les droits d'execution pour chaque binaire téléchargé.
- Vous devez à partir de ce moment comprendre le fonctionnement de votre binaire en utilisant tout ce dont vous aurez besoin (je vous conseille de bien comprendre le fonctionnement de lldb ici).
- Durant votre périple vous allez devoir écrire une source en C qui sera simplement la réplique algorithmique de votre binaire.
- Pour modifier le binaire vous n'avez pas besoin d'un logiciel spécifique vim suffit !
- Vous pouvez être amené, durant votre soutenance, à prouver vos résultats. Il faut vous y préparer.

Chapitre V

Partie obligatoire

- Votre dossier de rendu ne doit contenir que ce qui est listé ci-dessous :
  - - un dossier nommé par la difficulte du crackme.
  - - Un fichier password contenant le password pour passer le crackme dans ce dossier.
  - - Un fichier avec le nom source.c contenant une représentation du programme en C dans ce dossier.
  - - Le binaire modifié si besoin.
- Votre rendu sera de la forme :

```
$> ls -al
[..]
drwxr-xr-x  2 root root 4096 Dec  3 XX:XX {Difficulte}
[..]
$> ls -alR {Difficulte}
{Difficulte}:
total 16
drwxr-xr-x 3 root root 4096 Dec  3 15:22 .
drwxr-xr-x 6 root root 4096 Dec  3 15:20 ..
-rw-r--r-- 1 root root    5 Dec  3 15:22 password
-rw-r--r-- 1 root root    5 Dec  3 15:22 binary
-rw-r--r-- 1 root root    5 Dec  3 15:22 source.c

$> cat {Difficulte}/password | cat -e
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX$
$>
```

- Dans le cadre de votre partie obligatoire, vous devez compléter uniqueument deux crackmes.

- Un crackme est considéré comme valide si une source en C représentant la partie algorithmique du binaire ainsi qu'un mot de passe valide pour résoudre ce crackme est présente dans votre dépot.

Pour les malins (ou pas)... L'utilisation de la technique ayant pour but d'override une fonction est interdite.

Chapitre VI

Partie bonus

Les bonus ne seront comptabilisés que si votre partie obligatoire est PARFAITE. Par PARFAITE, on entend bien évidemment qu'elle est entièrement réalisée, et qu'il n'est pas possible de mettre son comportement en défaut, même en cas d'erreur aussi vicieuse soit-elle, de mauvaise utilisation, etc ... Concrètement, cela signifie que si votre partie obligatoire n'est pas validée, vos bonus seront intégralement IGNORéS.

Dans le cadre de votre partie bonus il sera possible de rendre le binaire modifié dans le but de pouvoir accepter tous les mots de passe que l'on pourrait lui entrer.

ATTENTION: chaque binaire modifié devra être expliqué correctement!

Vous avez la possibilitée aussi de pouvoir rendre les 3 binaires.

Pour avoir la note maximale il faudra alors logiquement rendre les 3 binaires avec les versions modifiées dans chaque cas.

Chapitre VII

Rendu et peer-évaluation

Rendez-votre travail sur votre dépot `GiT` comme d'habitude. Seul le travail présent sur votre dépot sera évalué en soutenance.