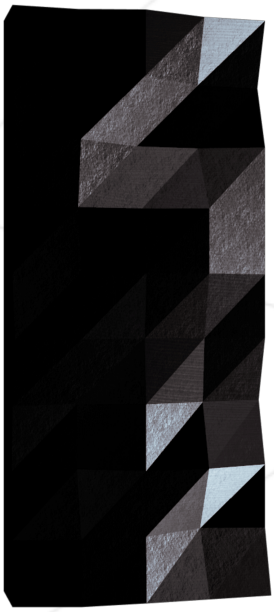

Project SECURITE

Over Ride

42 Staff pedago@staff.42.fr

Résumé: Ce projet est la suite de RainFall dans le but d'apprendre l'éxploitation de binaire (type elf).

Version:

Table des matières

Chapitre I

Preamble

There is something wrong...

Chapitre II

Introduction

En tant que développeur, vous risquez dans votre carrière de travailler sur des logiciels qui vont être utilisés par des centaines de personnes.

Vous avez appris à faire des programmes plus ou moins complexes sans prendre en compte le côté sécurité.

Par ce projet vous allez assez vite vous rendre compte de la facilitée à pouvoir exploiter des soucis assez simplement évitable.

Lorsque vous aurez terminé ce projet vous aurez alors une compréhension de la mémoire vraiment plus claire ce qui va vous aider à concevoir des programmes sans bug !

Chapitre III

Objectifs

Ce projet a pour but d'améliorer vos connaissance dans le monde de l'éxploitation de binaire de type elf dans un système i386.

Les méthodes que vous allez utiliser, plus ou moins complexes, vous feront voir différemment l'informatique en général et surtout prendre conscience des problèmes découlant de mauvaise pratique dans la programmation.

Durant ce projet, vous allez surement rencontrer des difficultés : soyons clairs, ces difficultés, il faut que vous les dépassiez de vous-même. Il faut que votre approche des différentes épreuves vienne vraiment et uniquement de VOUS. L'intérêt ici est de vous faire développer une certaine logique ainsi acquérir des reflexs qui vont vous suivre par la suite. Avant de demander de l'aide, demandez-vous bien si vous avez vraiment réfléchi à toutes les possibilités.

Chapitre IV

Consignes générales

- Ce projet ne sera corrigé que par des humains.
- Vous pouvez être amené, durant votre soutenance, à prouver vos résultats. Il faut vous y préparer.
- Vous allez devoir utiliser une machine virtuelle (64 bits) pour faire ce projet. Une fois votre machine lancée avec l'ISO fourni avec le sujet, si tout est bien configuré, vous aurez un simple prompt avec une IP :

```
      ____                ____ _     __
     / __ \_  _____ _____/ __ \(_)___/ /__
    / / / / |/ / _ \/ ___/ /_/ / / __  / _ \
   / /_/ /|  / __/ / / _, _/ / /_/ / __/
   \____/ |___/\___/_/ /_/ |_/_/\__,_/\___/

                  Good luck & Have fun

  To start, ssh with level00/level00 on 192.168.1.3:4242
level00@192.168.1.3's password: _
```

Si l'adresse ip n'est pas visible vous pourrez la récupérer une fois connecté par la commande ifconfig.

- A ce moment-là, vous aurez la possibilité de vous connecter en utilisant le couple de `login:password` suivant : `level00:level00`.

  Je vous conseille vivement d'utiliser la connexion SSH disponible sur le port 4242 :

  ```
  $> ssh level00@192.168.1.13 -p 4242
  ```

- Une fois connecté, vous allez devoir trouver le moyen permettant de lire le fichier ".pass" avec le compte "du prochaine niveau level0X" (X = numéro du niveau suivant)
- Ce fichier ".pass" est situé à la racine du home de chaque utilisateur (level00 exclu)

- Bien entendu une fois level09 vous devez logiquement aller vers l'utilisateur end
- Voici un exemple de session :

```
level00@OverRide:~$ ./level00 $(exploit)
$ cat /home/user/level01/.pass
???????????????????
$ exit
level00@OverRide:~$ su level01
Password:
level01@OverRide:~$ _
```

- Rien n'est laissé au hasard. En cas de problème, demandez-vous avant tout s'il n'y a pas un souci de votre côté.
- Tout usage d'outil d'automatisation sera considéré comme un cas de triche et sera donc sanctionné d'un -42
- Evidemment, en cas de bug avéré, prévenez la pedago !
- Vous pouvez poser vos questions sur le forum, sur jabber, IRC, slack...

Chapitre V

Partie obligatoire

- Votre dossier de rendu ne doit contenir que les choses qui vous ont permises de résoudre chacune des épreuves validées.

- Votre rendu sera de la forme :

```
$> ls -al
[..]
drwxr-xr-x 2 root root 4096 Dec 3 XX:XX level00
drwxr-xr-x 2 root root 4096 Dec 3 XX:XX level01
drwxr-xr-x 2 root root 4096 Dec 3 XX:XX level02
drwxr-xr-x 2 root root 4096 Dec 3 XX:XX level03
[..]
$> ls -alR level00
level0:
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 flag
-rw-r--r-- 1 root root 50 Dec 3 15:22 source
drwxr-xr-x 2 root root 4096 Dec 3 15:22 Ressources

level0/Ressources:
total 8
drwxr-xr-x 2 root root 4096 Dec 3 15:22 .
drwxr-xr-x 3 root root 4096 Dec 3 15:22 ..
-rw-r--r-- 1 root root  0 Dec  3 15:22 whatever.wahtever
$> cat level00/flag | cat -e
XXXXXXXXXXXXXXXXXXXXXXXXXXXX$
$> nl level00/source
1   #include <stdio.h>
2   int
3   main(void) {
4   printf("Code, source!\n");
5       return (0x0);
6   }
$> _
```

- Dans le dossier Ressources vous placerez tout ce dont vous aurez besoin pour prouver votre résolution en soutenance. Il est possible que le fichier flag soit vide mais une justification sera alors demandé.

- Le fichier source doit contenir simplement le binaire exploité sous sa forme compréhensible pour un développeur. Le langage nést pas imposé.

ATTENTION: Tout ce qui est présent dans ce dossier doit pouvoir être expliqué clairement sans aucune hésitation. AUCUN binaire ne doit être présent dans ce dossier.

- Si vous avez besoin d'utiliser un fichier spécifique présent sur l'ISO du projet, vous devez le télécharger en soutenance. Vous ne devez sous aucun prétexte mettre celui-ci dans votre dépôt.
- Dans le cas d'utilisation d'un logiciel spécifique externe, vous devez préparer un environnement spécifique (VM, docker, Vagrant).
- La création de script dans le but de gagner du temps est encouragée, mais une explication détaillée pourra en être demandée en soutenance.
- Dans le cadre de votre partie obligatoire, vous devez compléter la liste de niveaux suivante :
  - level00.
  - level01.
  - level02.
  - level03.
  - level04.
  - level05.
  - level06.
  - level07.
  - level08.
- Lors de votre soutenance chaque membre du groupe doit pouvoir justifier de chaque challenge résolu.

Pour les malins (ou pas)... Bien sûr vous n'avez pas le droit de bruteforce les flags ssh. Ce serait de toute façon inutile, puisque vous devez justifier votre résolution en soutenance.

Chapitre VI

Partie bonus

Dans le cadre de votre partie bonus, vous pouvez compléter avec ce dernier niveaux :

- level09

Le dernier utilisateur est "end". Devenir root n'est pas un bonus.
Cela est même considérée comme une tricherie.

La partie bonus ne sera évaluée que si la partie obligatoire
est PARFAITE. Parfait signifie que la partie obligatoire a été
intégralement réalisée et qu'elle fonctionne sans dysfonctionnement.
Si vous n'avez pas satisfait à TOUTES les exigences obligatoires,
votre partie bonus ne sera pas évaluée du tout.

Chapitre VII

Rendu et peer-évaluation

Rendez votre travail dans votre dépôt `Git` comme d'habitude. Seul le travail contenu dans votre dépôt sera évalué lors de la soutenance. N'hésitez pas à vérifier les noms de vos dossiers et fichiers pour vous assurer qu'ils sont corrects.