



# Project SECURITE

Rain Fall

42 Staff [pedago@staff.42.fr](mailto:pedago@staff.42.fr)

*Résumé: Ce projet est une introduction à l'exploitation de binaire (type elf).*

# Table des matières

|            |                                 |           |
|------------|---------------------------------|-----------|
| <b>I</b>   | <b>Préambule</b>                | <b>2</b>  |
| <b>II</b>  | <b>Introduction</b>             | <b>3</b>  |
| <b>III</b> | <b>Objectifs</b>                | <b>4</b>  |
| <b>IV</b>  | <b>Consignes générales</b>      | <b>5</b>  |
| <b>V</b>   | <b>Partie obligatoire</b>       | <b>7</b>  |
| <b>VI</b>  | <b>Partie bonus</b>             | <b>9</b>  |
| <b>VII</b> | <b>Rendu et peer-évaluation</b> | <b>10</b> |

# Chapitre I

## Préambule

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 concevoir des programmes plus ou moins complexes, mais vous avez rarement pris en compte l'aspect sécurité.

Par ce projet, vous allez vite vous rendre compte que vous avez probablement introduit dans vos programmes un certain nombre de failles facilement exploitables - la bonne nouvelle, c'est que ces failles sont aussi facilement évitables !

A la fin de ce projet, vous saurez non seulement éviter ces écueils, mais vous aurez également acquis une compréhension de la mémoire plus précise qui vous aidera à concevoir des programmes sans bug.

# Chapitre III

## Objectifs

Ce projet a pour but de vous faire découvrir, via plusieurs petits challenges, l'exploitation 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 résultants de mauvaises pratiques de 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 réflexes 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.

# Consignes générales

- ```

  -----  _-
  |  _ \   ( )   |  _ _ _ |  |  |
  | |_) |  |  _ _ _ _ _ |  |  _ _ |  |  |
  | _ /   _ |  |  _ \ |  _/_   |  |  |
  | | \ \ ( |  |  |  |  |  |  |  |  |  |
  | _ \ \ _ , _ |  |  |  |  |  |  |  |

```
- Good luck & Have fun
- To start, ssh with level0/level0 on 172.16.39.128:4242  
 level0@172.16.39.128's password:



- Ce fichier **.pass** est situé à la racine du home de chaque utilisateur (level0 exclu).

- Bien entendu, une fois level9 vous devez logiquement aller vers l'utilisateur bonus0.
- Voici un exemple de session :

```
level0@RainFall:~$ ./level0 $(exploit)
$ cat /home/user/level1/.pass
????????????????????
$ exit
level0@RainFall:~$ su level1
Password:
level1@RainFall:~$ _
```

- 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 level0
drwxr-xr-x 2 root root 4096 Dec 3 XX:XX level1
drwxr-xr-x 2 root root 4096 Dec 3 XX:XX level2
drwxr-xr-x 2 root root 4096 Dec 3 XX:XX level3
[.]
$> ls -alR level0
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
-rw-r--r-- 1 root root 50 Dec 3 15:22 walkthrough
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
$> cat level0/flag | cat -e
XXXXXXXXXXXXXXXXXXXXXXXXX$
$> nl level0/source
1  #include <stdio.h>
2  int
3  main(void) {
4  printf("Code, source!\n");
5      return (0x0);
6  }
$> _
```

- Vous placerez tout ce dont vous aurez besoin pour prouver votre résolution en soutenance dans le dossier **Ressources**. Il est possible que le fichier **flag** soit vide, mais une justification sera alors demandée.
- Le fichier **source** doit contenir le binaire exploité sous sa forme compréhensible pour un développeur. Le langage n'est pas imposé.
- Le fichier **walkthrough** devra contenir les étapes qui ont permis la résolution de



l'épreuve.



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 :
  - level0.
  - level1.
  - level2.
  - level3.
  - level4.
  - level5.
  - level6.
  - level7.
  - level8.
  - level9.
- 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



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, vous pouvez compléter la liste de niveaux suivante :

- bonu0
- bonus1
- bonus2
- bonus3



Le dernier utilisateur est "end".



Devenir root est considéré ici comme de la triche.

# Chapitre VII

## Rendu et peer-évaluation

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



Exceptionnellement, ce projet sera corrigé par un membre de l'équipe pédago. Lorsque vous avez close votre projet et que vous voulez le faire corriger, contactez Wandre (wil sur le Slack) pour organiser votre soutenance.