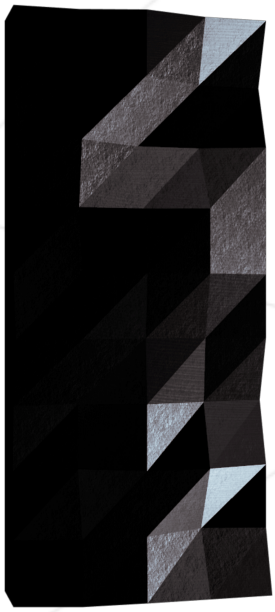

Cybersecurity Piscine

Reverse me i'm famous!

Summary: The reverse passion.

Version: 1.00

Contents

Chapter I

Introduction

This project aims to discovery the art of Reversing Engineering.

You will learn to understand how several programs work.

For the more adventurous, you will be able to patch these files so that they can validate all the entries you want!

Chapter II

Mandatory part

This project involves using a machine with linux. You can use a VM or whatever you would like to do this project.

- You have 3 programs available on your intra in this project page. There are 3 different difficulties, your goal is just to understand each binary and then find a password to validate each of them.

- You must from this moment understand how theses program works using what you would need (I suggest you to understand how gdb works here).

- You have to write a program in C that will simply be a copy of the basic program.

- Your submission folder must contain at least what is listed below:
  - A folder named by the difficulty of the program.
  - A file password container the password to spend the crackme in this folder.
  - A file with source.c name containing a representation of C program in this folder.

Your rendering will be of the form:

```
$> ls -al
[..]
drwxr-xr-x 2 wil wil 4096 Dec 3 XX:42 level1
[..]
$> ls -alR level1
easy:
total 16
drwxr-xr-x 3 wil wil 4096 Dec 3 XX:42 .
drwxr-xr-x 6 wil wil 4096 Dec 3 XX:42 ..
-rw-r--r-- 1 wil wil XXXX Dec 3 XX:42 password
-rw-r--r-- 1 wil wil XXXX Dec 3 XX:42 source.c
```

```
$> cat level1/password | cat -e
XXXXXXXXXXXXXXXXXXXXXXXXXXXX$
$>
```

To validate the mandatory part at the minimal, you only need to reverse the first two programs.

A programm is considered valid if a C source representing the algorithmic part of the binary and a valid password to solve this crackme is present in your repository.

It is possible to find several passwords for each binary that is intentional.

Chapter III

Bonus Part

You must add for each program a patch allowing to validate each program with any password.

WARNING: Each modified binary should be correctly explain!

You will have to add everything you need to patch the programs on your computer. It is of course mandatory to justify and explain your method for each case.

It is forbidden to override the functions we would like to be able to execute the binary without adding anything during its execution. So it is not possible to override the library with tricks like LD_PRELOAD.

The bonus part will only be assessed if the mandatory part is PERFECT. Perfect means the mandatory part has been integrally done and works without malfunctioning. If you have not passed ALL the mandatory requirements, your bonus part will not be evaluated at all.

Chapter IV

Submission and peer-evaluation

Turn in your assignment in your `Git` repository as usual. Only the work inside your repository will be evaluated during the defense. Don't hesitate to double check the names of your folders and files to ensure they are correct.