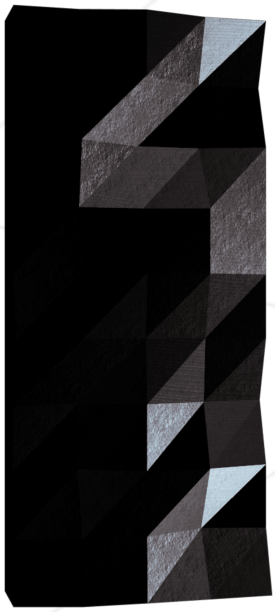


SECURITY

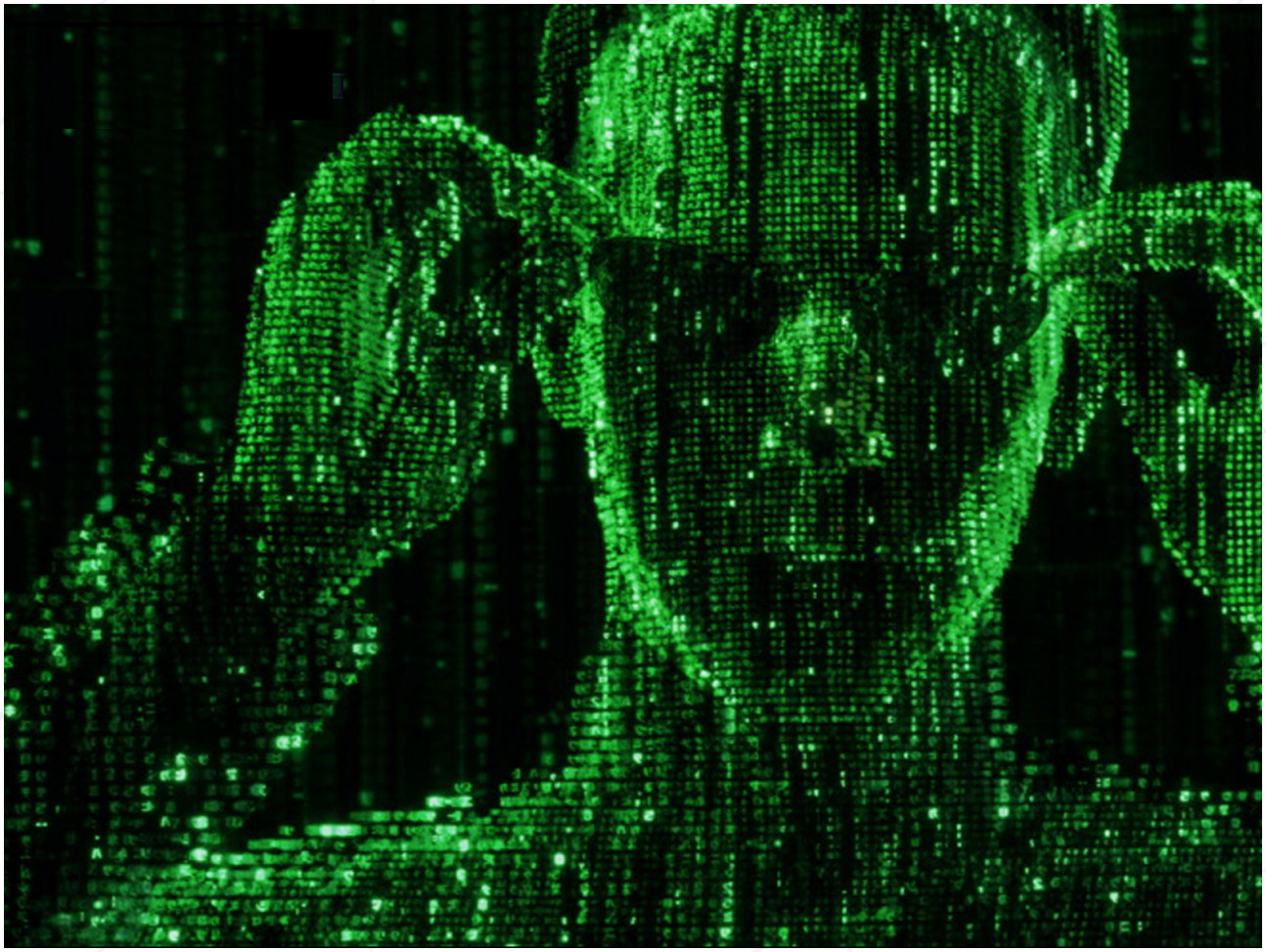
Snow Crash

Summary: This project is an introduction to cyber security

Contents

Chapter I

Préambule



There is something wrong..

Chapter II

Introduction

As a coder, you will most probable work in your career on some programs that will be used by hundreds of people.

If your code has vulnerabilities, the people using it will also be vulnerable and exposed with a system that can be exploited.

It is your responsibility to understand the techniques used to exploit these vulnerabilities so that you can detect and avoid them.

This project is therefore a short introduction to the vast universe that is cyber security, where the right to error cannot exist.

Chapter III

Objectives

This project's goal is to make you discover, through a few small challenges, cyber security is a few various fields.

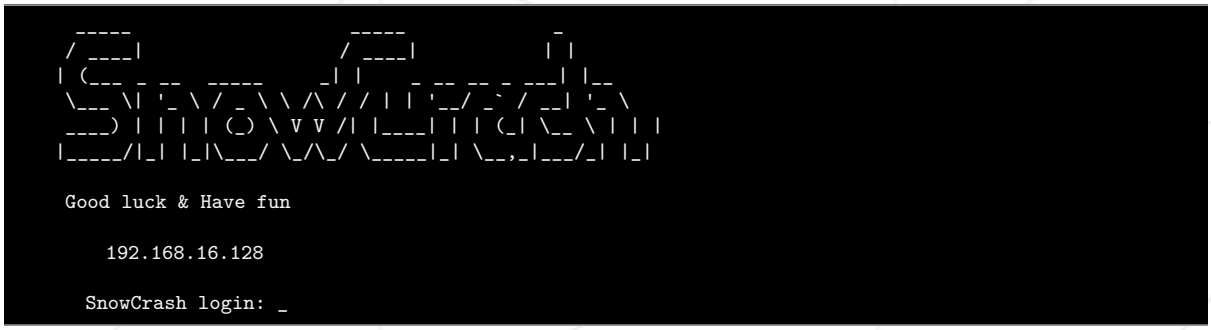
The methods you are about to use, all more or less complicated, will present to you computer science under a different light.

In the course of this project, you will most probably encounter some difficulties: let's be clear, thess difficulties, you will have to overcome them on your own. Your approach to these various tests will have to come from YOU. The point here is to make you develop a certain logic that you will have to follow through. Before asking for help, ask yourself if you have really thought through of everything.

Chapter IV

Consignes générales

- This project will be corrected by humans only.
- You can be brought, during p2p to prove your results. Be prepared.
- You will have to use a virtual machine (64 bits) to create this project. Once the machine starts with the ISO provided, if all is well configured, you will have a simple prompt with an IP:



If the IP address is not visible, you will be able to get it once connected with the ifconfig command.

- At that point, you will have the option to connect using the following `login:password` `level00:level00`.

  I strongly urge you to use the available SSH connection on the port 4242:

```
$> ssh level00@192.168.16.128 -p 4242
```

- Once connected, you will have to find the password allowing you to connect with the account “flagXX”(XX= number of the current level).

When you are connected to the account "flagXX", you will have to launch "getflag", that will give you the password for you to connect to the next level. (It is possible that you won't be able to connect to the "flagXX" - in that case you will need to think of an alternative method, like for example, inject a command on the program depending on the rights of that one!)

- Here is an example of a session:

```
level00@SnowCrash:~$ su flag00
Password:
Don't forget to launch getflag !
flag00@SnowCrash:~$ getflag
Check flag.Here is your token : ????????????????
flag00@SnowCrash:~$ su level01
Password:
level01@SnowCrash:~$ _
```

- For some levels, you will have to use one or many external programs, I therefore invite you to learn how to use the SCP command SCP.

The /tmp/ and /var/tmp/ folders are limited in terms of rights and will be reset from time to time, so it is strongly recommended that you do not work directly on the machine.

- Nothing has been left to chance. In case of a problem, ask yourself if you do not have a problem on your side.
- Naturally in case of a proven bug, alert the pedago team!
- You can ask your questions on the forum, on slack...

Chapter V

Mandatory part

- Your repository folder can only contain things that allowed you to solve each of the validated problems.
- Your repository will be as such:

```
$> 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
level00:
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
drwxr-xr-x 2 root root 4096 Dec 3 15:22 Ressources

level00/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
$> cat level00/flag | cat -e
XXXXXXXXXXXXXXXXXXXXXXXXXXXX$
$>
```

- In the resource folder, you will put everything you will need to prove your resolution during p2p. It is possible for the flag file to be empty but you will then be asked to justify.

WARNING: Everything present in that folder will have to be explained clearly and without hesitation. NO binary will be allowed in this folder.

- If you need to use a specific file present in the ISO of the project, you will have to download it during p2p. You cannot under any circumstances have it in your repository

- In case you are using a specific external program, you will need to prepare a specific environment (VM, docker, Vagrant).
- The creation of a script to save time is strongly encouraged, but a detailed explanation can be asked during p2p.
- Within the frame of the mandatory part, you will have to complete the following list of levels:
  - level00.
  - level01.
  - level02.
  - level03.
  - level04.
  - level05.
  - level06.
  - level07.
  - level08.
  - level09.

For the smarty pants (or not)... Naturally you are not allowed to bruteforce the ssh flags. It would in any case be useless, since you will have to justify during p2p.

Chapter VI

Bonus Part

We will look at your bonus part if and only if your mandatory part is EXCELLENT. This means that your must complete the mandatory part, beginning to end, and your error management needs to be flawless, even in cases of twisted or bad usage. If that's not the case, your bonuses will be totally IGNORED.

As a bonus you can complete the list of following levels:

- level10
- level11
- level12
- level13
- level14

Chapter VII

Turn-in and peer-evaluation

Submit your work on your GiT repository as usual. Only the work on your repository will be graded.