Project SECURITY

Snow Crash

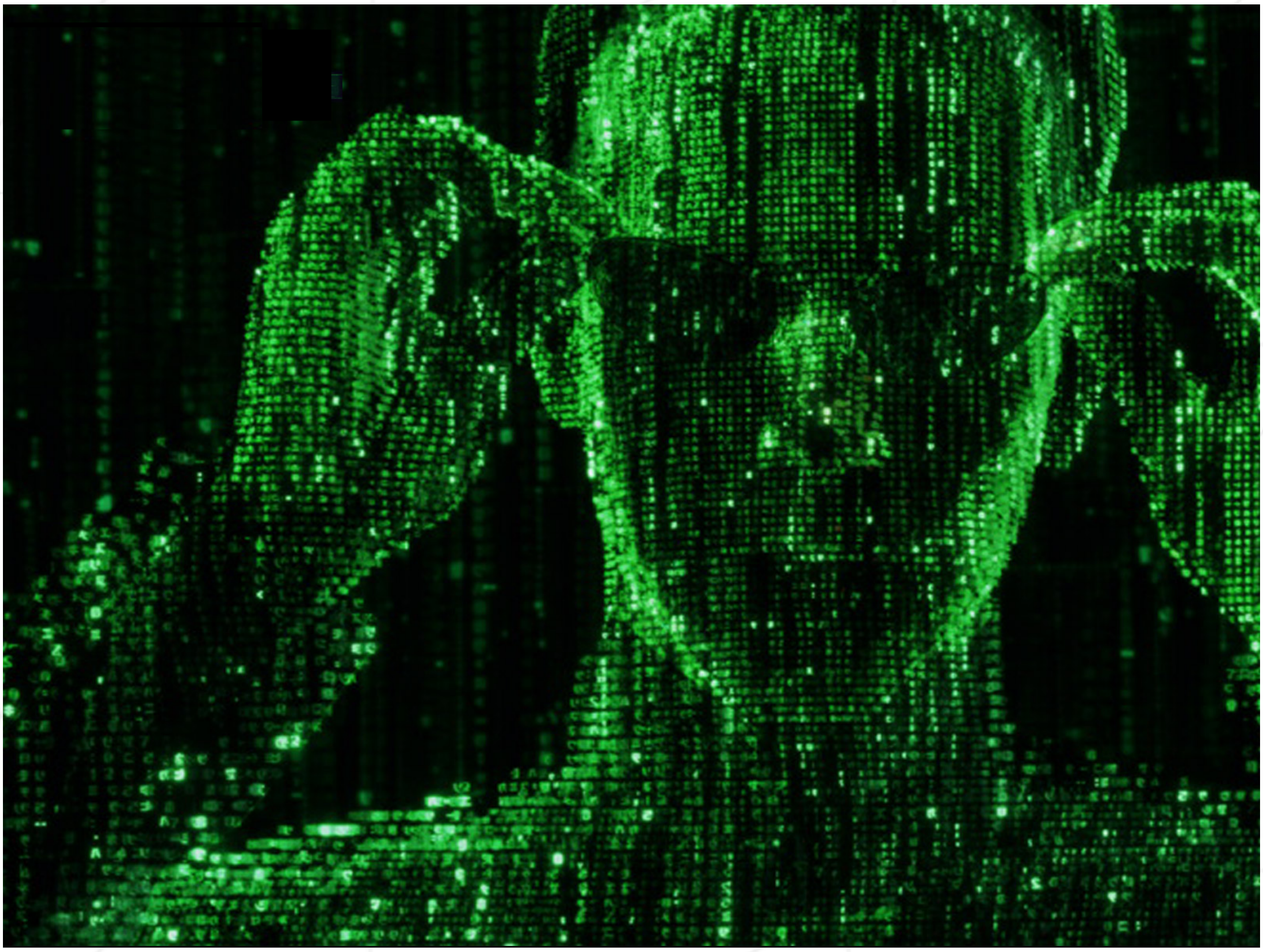
42 Staff pedago@staff.42.fr

Summary: This project will be an introduction to cyber security.

Contents

Chapter I

Preamble



There is something wrong..

Chapter II

Introduction

As a developer, you may have to work on softwares that will be used by hundreds of persons in your career.

If your software shows some weaknesses, these weaknesses will expose the users through your software.

It is your duty to understand the different techniques used to exploit these weaknesses in order to spot them and avoid them.

This project is a modest introduction to the wide world of cyber security. A world where you'll have no margin for errors.

Chapter III

Objectives

This project aims to make you discover, through several little challenges, cyber security in various fields.

You will use more or less complex methods that will give you a new perspective on IT in general.

You will reach some stalemate during this project. You will have to surpass them yourself. You must be the one and ONLY key to the locked doors you will face. This project aims to develop some logic thinking you will learn for good and use in the future. Before asking for help, ask yourself if you've really explored all the possibilities.

Chapter IV

General instructions

- This project will be evaluated by humans.

- You might have to prove your results during your evaluation. Be ready to do so.

- To make this project, you will have to use a VM(64 bits). Once you have started your machine with the ISO provided with this subject, if your configuration is right, you will get a simple prompt with an IP:

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

 Good luck & Have fun

    192.168.16.128

  SnowCrash login: _
```

If the IP address is not visible, you will get it with the command `ifconfig` once you're connected.

- Then, you will be able to register using the following couple of `login:password`: `level00:level00`.

  You really should use the SSH connection available on port 4242:

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

- Once registered, you're gonna have to find the password that will log you in with the "flagXX" account(XX = current level number).

Once logged to the "flagXX" account, launch the "getflag" command. It will give you the password to connect to the next level (You may not be able to connect to a "flagXX" account - in this case, you will have to find an alternative method, like a command injection on the program depending on its rights, for instance!).

- Here is a session example:

```
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:~$ _
```

- To help you with some levels, you're gonna have to use external softwares. You should learn to use the SCP command.

/tmp/ and /var/tmp/ folders have limited rights and will be reset from time to time. You should not work directly on the machine.

- Nothing is left to chance. If there is a problem, start wondering if your code is not the cause.
- Of course, in case of a true bug, run to the educational team!
- You can post your questions on the forum, Jabber, IRC, Slack...

Chapter V

Mandatory part

- Your repo must include anything that helped you solve each validated test.
- Your repository will look like this:

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

- You will keep everything you need to prove your results during the evaluation in the Resource folder. The `flag` file may be empty, but you may have to explain why.

`WARNING: You must be able to clearly and precisely explain anything that is included in the folder. The folder mustn't include ANY binary.`

- If you need to use a specific file that's included on the project's ISO, you must download it during the evaluation. You must put it in your repo under no circumstances.

- If you plan to use a specific external software, you must set up a specific environment (VM, docker, Vagrant).
- You're invited to create scripts that will make you stall, but you will have to explain them during the evaluation.
- For the mandatory part, you must complete the following list of levels:
  - level00.
  - level01.
  - level02.
  - level03.
  - level04.
  - level05.
  - level06.
  - level07.
  - level08.
  - level09.

> ⚠ `Hey, smarty (or not so smarty) pants! You cannot bruteforce the ssh flags. This would be useless anyway, since you will have to justify your solution during the evaluation.`

Chapter VI

Bonus part

```
Bonus will be taken into account only if the mandatory part is
PERFECT. PERFECT meaning it is completed, that its behavior cannot be
faulted, even because of the slightest mistake, improper use, etc...
Practically, it means that if the mandatory part is not validated,
none of the bonus will be taken in consideration.
```

For the bonus part, you can complete the following list of levels:

- level10
- level11
- level12
- level13
- level14

Chapter VII

Turn-in and peer-evaluation

As usual, turn in your work on your repo `GiT`. Only the work included on your repo will be reviewed during the evaluation.