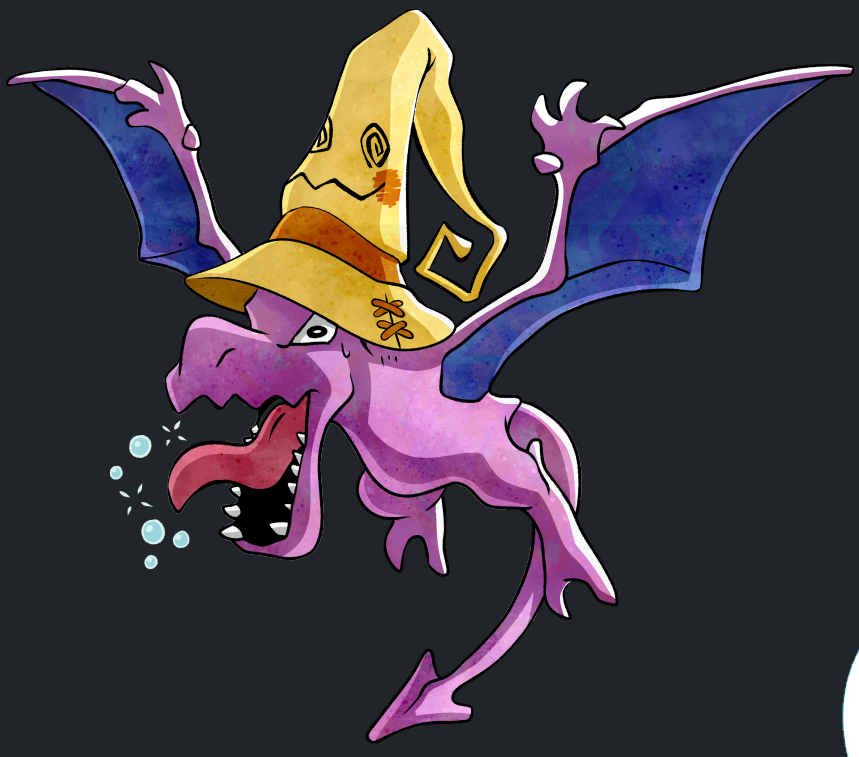
{EPITECH}

G-SEC-210

BOOTSTRAP



G-SEC-210

Bootstrap - Introduction to Application Security

The aim of this bootstrap is to see different type of vulnerabilities, exploit them and find out how to fix them !

You can get all the exercise on the intranet !

Each exercise will have the same workflow :

- ✓ Find and exploit the vulnerability from the code or the binary
- ✓ Use the password you find to unlock the **PDF file**
- ✓ The **PDF file** will explain to you the exploit
  - If we only give you the binary, the **PDF file** will give you the c code
- ✓ Fix the code to make the exploit unusable !

Obviously some of the vulnerable data is already readable from the code, the goal is not to use the data directly.

When you are fixing the code, you need to write with the *Epitech C norm*, it's **mandatory**

To use the binaries, don't forget to ***chmod*** them before.

```
Terminal
~/G-SEC-210> chmod +x a.out
```


{EPITECH}

Exercice 01 - Format String Attack

The idea on this exercise is to recuperate the data string only by using the binary.
This one is really easy, it's just to be started !
You can read the source code of the C file for this one.
But don't take directly the password !

Secondly, find a way to correct this vulnerability in the ex_01_fix.c file.

Exercice 02 - Buffer Overflow

You need to get the admin access and the password, only by using the binary.
To be admin, you need to turn the value is_admin to 1.

Secondly, find a way to correct this vulnerability in the ex_02_fix.c file.

Exercice 03 - Use After Free

The goal here is to get the password, by adding code in between the comments.
You cannot use directly the password (obviously).

Secondly, find a way to correct this vulnerability in the ex_03_fix.c file.

Exercice 04 - Command Injection

Here you need to get the private data who is located in the Private folder in the private.data file, only by using the binary.
You can look at the code but don't look directly to the password in the private data file.

Secondly, find a way to correct this vulnerability in the ex_04_fix.c file.

Exercice 05 - Race Condition

Find a way to exploit the vulnerability in the ex_05.c only by using the binary and some outside manipulation.

Secondly, find a way to correct this vulnerability in the ex_05_fix.c file.

Exercice 06 - Hardcoded Password

Here, the idea is to get the secret password from the code by only using the binary.
You need to find the password to unlock the real password of the PDF.
You can use a super command, called strings, to access some variables you would not get in ordinary time.

Secondly, find a way to correct this vulnerability in the ex_06_fix.c file.

Exercice 07 - Core Dumps

This time it will be a little different, you will try to exploit a crashing function, by using different tools.
Before running the program, you'll need to ensure that your system is configured to generate core dumps. By default, core dumps might be disabled or limited in size.

To enable core dumps:

```
Terminal
~/G-SEC-210> ulimit -c unlimited
```

You can check if core dumps are enabled by running:

```
Terminal
~/G-SEC-210> ulimit -a
```

Look for the line that shows core file size. If it is set to unlimited, core dumps will be generated.

With the core set up now, you just need to use a really useful tool for debugging to exploit them !

The binary that you have is compiled with debugging flags like ggdb ...

Here the idea is to use a new tool, and not to re-use the last method !

Secondly, find a way to correct this vulnerability in the ex_07_fix.c file.

v 1.0