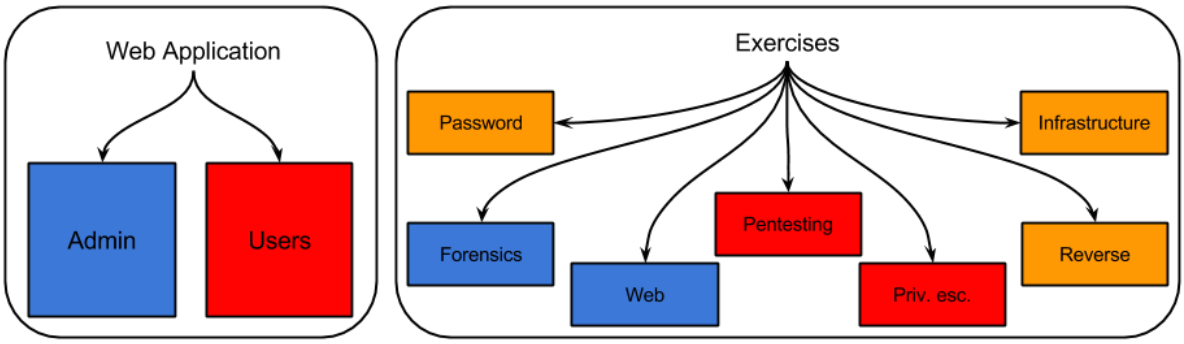
Individual report

by Alexis Semren

1 - Introduction

This document will sum up what I have done in this project. In order to split the work between me and my workmate, we have split the different tasks at the beginning of the project and regularly reviewed our work. On the following image, you can see how the work was split.



Alexis (aggs3)
Steven(sm861)
Both of us

In order to test our work, I have also tested my friend's work (Virtual Machines and files). The test phase was done in real condition, indeed we knew the vulnerabilities but when we were working on them, we were very careful to not speak too much about the implementation and where wheres precisely situated the vulnerabilities or how to fix them.

2 - The website

The website was achieved by both of us. We splited the development in two parts. Steven MARIEN made the front part, i.e. the design for the all the user part and the flag verification system.

As regards me, I coded the back-end, which includes the design of the admin part, the login system, the creation/update of the exercises and the categories via the form, or other actions, such as deleting and displaying in a table. I also made the admin system part, such as the application deployment on a server.

3 - The virtual machines

In this project, I worked on two Virtual Machines. To do so, I used the VMWare software and when the machine was ready, I exported it in OVA extension.

3 - 1 First Virtual Machine

My first virtual machine was for the "Infrastructure Penetration Testing" category. I implemented a vulnerable version of proftpd (version 3.3c) on a Debian 7. To do so, I searched a vulnerable version on exploit-db.com, and installed from the source, while fixing the different problems of dependency (required to install it) at the same time.

3 - 2 Second Virtual Machine

My second virtual machine was the "Web" category. In this machine, I installed an apache2 web server and php5. Then, I created a mini-website on which we can find many exercises:

- Obfuscation (2 levels)
- Sql Injection (2 levels)
- Cookie parameters (1 level)
- Form Upload (1level)
- XSS (4 levels)
- Url parameters (1 level)
- Brute Force (1 level)
- Wordpress

For all the web exercises, I used the Apache2 web server, and PHP5 to do the exercises. On a general point of view, every exercise is built according to the same pattern. It is a webpage with the vulnerability directly in the page. For the SQL Injection level2, I made a mini-website with a fake shop to try to be realistic. In order to perform these exercises and to test them, I used different software, such as SQLMap, Hydra, WPScan, and of course Kali. I learnt how to use them at the same time.

The Wordpress is a fake blog with vulnerable plugins installed. To perform it, I created a small blog about Travels with some posts to be as realistic as possible. The vulnerable plugins come from the exploit-db.com website. I made some research on this website to find good examples of plugins. Then, when the plugin seemed good, I had to download it and install it, do some fake configuration and usage with it. If the plugin was not good enough or not really working, I did not keep it.

4 - Reverse engineering

I also made one buffer overflow exercise. It consisted of writing a C software with vulnerability. The goal of this exercise was to display the flags. The exercise attempts to try to display the flag by passing a condition.

To compile this flag, I had to disable the randomise_va_space and use some flags to remove the protection against the buffer overflow. Finally, the only difficult part was to learn how to use EDB-Debugger in order to find the address of the assembly instructions and change the value.

5 - Work Time

During the development, we worked together in order to be reactive to each other's problems, and also to discuss about what to do and how to do it. It was a good thing to work by pair. It was not the first project we performed together.