BreakingLAB 

Plateforme d’e-learning informatique et pentesting via des dockers

Livrable 1

Version 1.0

Réalisé par Adil AHMED, Gaëtan GIANQUINTIERI, Moustapha BARI,

Rémi BILLY et Patience EBONGUE

Encadré par Daniel RANC

19 AVRIL 2020

TABLE DES MATIERES

I. Cahier des charges

a. Description du projet

Nom	**BreakingLab**
Description	Plateforme d'apprentissage de l'informatique et du pentest avec des dockers
Date	24/04/2020
Domaine	Informatique, Cybersécurité, Pédagogie
Module	Projet Informatique
Gitlab Project	PRO-8445-20-RAN-BreakingLab
Equipe	Adil AHMED Moustapha BARI Rémi BILLY Gaëtan GIANQUINTIERI Patience Noé EBONGUE NJOE

b. Objectif du projet

Notre projet a pour objectif de permettre aux gens d'apprendre l'informatique et la Cybersécurité de manière très simple. Nous souhaitons mettre à la disposition des gens une plateforme d'e-learning sur l'informatique et le pentest.

La grande majorité des personnes de notre société n'ont aucune notion en informatique et encore moins en cybersécurité bien qu'elles utilisent des objets numériques tous les jours dans leur quotidien.

Notre projet a pour but de les sensibiliser à l'informatique et à la Cybersécurité de manière simple afin qu'ils découvrent un nouveau domaine et pour qu'ils adoptent les bons gestes de cybersécurité dans leur quotidien.

Il est important de noter que notre objectif n'est pas de former des gens à devenirs des pirates informatiques mais à les sensibiliser à la cybersécurité et d'initier, pourquoi pas, une vocation dans la sécurité informatique.

c. Composantes du projet

Développement Web	Tutoriel Informatique	Programmation des failles
Site web principal sur le serveur Page de login avec base de données en POSTGRESQL Page sommaire et webs Mise en place des DOCKERS	Langage Bash (bases) Langage Python Outils de scans Outils de brute-force Exploitation des failles web	Injection SQL, HTML, PHP Reverse Shell Session Hijacking Local File Inclusion Injection HTTP

d. Exigences

Notre projet doit permettre à nos utilisateurs de :

- Apprendre l'informatique ainsi que la cybersécurité
- Pratiquer directement sur notre site ce qu'ils ont appris via des dockers
- Faire du pentest dans un container spécialement conçu pour l'utilisateur
- Avoir un suivi de ses cours

e. Public ciblé

Nous visons les personnes de tout horizon.

Cependant, nous pensons qu'un tel projet serait un très bon outil pour introduire/former les gens à la cybersécurité. Ainsi nous pourrions mettre cet outil à disposition des clubs de cybersécurité afin de former leurs nouveaux arrivants.

De plus, conscient de la difficulté de l'apprentissage des langages de programmation à l'IMTBS, notre projet pourrait leur permettre d'apprendre facilement les bases d'un langage.

f. Compétences mobilisées

- Langages de programmation : Bash, Java, Python, SQL
- Développement web : HTML, CSS3, PHP et DJANGO
- Base de données : PostgreSQL
- Connaissances sur les Dockers et les serveurs web
- Connaissances en Cybersécurité : injections SQL/PHP/HTML, reverse shell, session hijacking, LFI, brute-force, outils de scan, utilisation de proxy pour du Man In The Middle, web shell…
- Gestion de projet et d'équipe

g. Outils à notre disposition

- Google drive pour la gestion des documents de projet et documents techniques
- Gitlab de TSP pour une gestion intelligente de nos codes
- OpenProject pour la gestion de notre projet
- Messenger, Discord ou Skype pour des réunions visio à distance
- Serveur web

II. Description des fonctionnalités

a. Fonctionnalités détaillées de notre projet

- Création d'un site web avec DJANGO et de son serveur pour l'interface utilisateur.

- Gestion des utilisateurs sur le site et création d'une base de données (users, date, ip, etc.)

- Possibilité pour l'utilisateur de cliquer sur un bouton « Pratiquer » pour s'exercer.
  Au clic, un container est créé à partir d'une image prédéfinie contenant seulement les éléments nécessaires aux failles que l'utilisateur souhaite tester. Ce container est totalement isolé des autres containers et de l'hôte.

- L'utilisateur est libre sur ce container dans la limite du besoin des failles avec possibilité de réinitialisation par l'utilisateur en cas de problème.

- Chaque utilisateur lance une instance de docker différentes des autres utilisateurs, il n'y a pas de conflit entre les dockers.

b. Exemple d'utilisation

L'utilisateur se connecte sur le site et choisi parmi les différentes catégories qui lui sont proposés celle qu'il souhaite travailler ou découvrir. Dans cette catégorie, il choisit ensuite le cours qu'il aimerait apprendre, cours affichés dans l'ordre de difficulté croissante. Un indice / marqueur indiquera la complexité du cours.

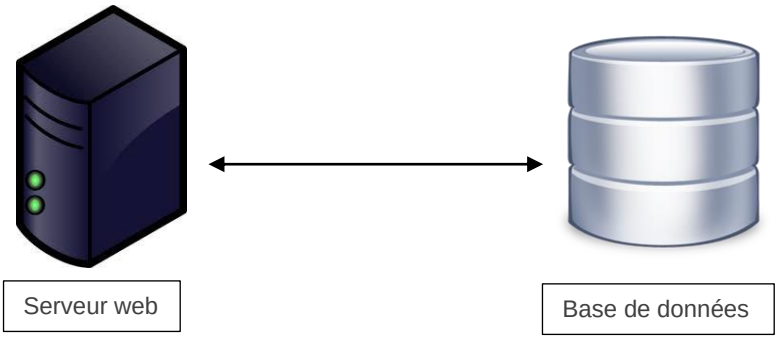
Une fois le cours choisi, l'utilisateur arrive sur une page lui fournissant toutes les informations nécessaires à la compréhension de celui-ci. Ces documents peuvent être par exemple du texte explicatif, des liens vers des documents externes comme des RFC ou des documents de normalisation, des vidéos, des schémas, etc.

Une fois le cours terminé, l'option "pratiquer" sera disponible à l'utilisateur. Cette option, sous la forme d'un bouton, lancera la création d'un container à partir d'une image prédéfinie spécifiquement pour ce cours. Ce container regroupe les logiciels et les configurations nécessaires à l'application du cours vu précédemment.

Les accès à ce container seront différents et dépendant du cours sélectionné, allant de la page web, à l'interface en ligne de commande en passant par le SSH. En cas de problème sur le container, l'utilisateur pourra le réinitialiser en utilisant le bouton cliquable prévu à cet effet sur le site web. Une fois la pratique terminé, l'utilisateur indique s'il a réussi à finir l'activité, et note sa difficulté.

III. Regroupement modulaire des fonctionnalités

Les pages webs de présentation du site, d'inscription, de connexion, d'accueil, d'options, des cours et de déconnexion sont sur notre serveur web principal. Ce même serveur web est associé à une base de données exclusivement réservée à elle. La base de données est sur la même machine que notre serveur et non indépendante.



Les pages de cours ont toutes un bouton « pratiquer » qui lance un container. Ce container regroupe l'ensemble des applications nécessaire pour s'exercer. En l'occurrence, il s'agira généralement de pages web avec des failles. Ces containers sont isolés de notre serveur web principal, les impacts sur ces containers n'affectent pas notre site web principal. Ainsi, nous pouvons regrouper notre travail en 2 parties :

- **Partie 1 :** création du site web principal avec toutes les pages nécessaires et de la base de données.

- **Partie 2 :** création des failles informatiques sur des page webs indépendantes de notre site web principal.

On peut également considérer le lancement du container comme une troisième partie. En effet, ce lancement fait le lien entre le site web principal et les containers avec les pages webs (avec des failles). Il nécessitera un travail particulier pour une bonne mise en correspondance de nos deux parties.

IV. Flux des données entre les modules