

EIP Vigilate

Work Breakdown Structure (WBS)

15 février 2015

Résumé du document

Ce document va d'abord parler de notre projet, un outil pour être au top de la sécurité, puis va expliquer par un schéma et un dictionnaire les différentes parties de Vigilate. Dans un premier temps, nous avons essayé de réfléchir à comment on pourrait faire et aux contraintes qui pouvaient nous toucher. Pour l'aspect technique, nous nous sommes dit que faire un site web en Django était la solution la plus pratique, car le backend sera en python (langage le plus maîtrisé par les membres concernés de l'équipe). Il va aussi falloir réfléchir au design et à l'ergonomie, car nous voulons un outil simple et rapide à utiliser, sans toutefois oublier le serveur qui se devra d'être le plus sécurisé possible. Notre outil se présentera sous forme d'un site internet pour une utilisation facile, il faudra donc d'une part s'occuper du design du site et d'autre part du coeur de l'outil en lui-même (le backend). Toutefois, pour les clients ayant besoin d'un niveau de sécurité supérieur, nous allons mettre en place une machine virtuelle. Cela permettra de contenir les informations sensibles au sein de leur réseau informatique.

Une quatrième partie sera de créer un scanner de programmes, permettant à l'utilisateur d'envoyer automatiquement la liste de ses services à surveiller, ce qui permettra de tenir notre base de données à jour.

Description du document

Titre	[2017][Vigilate][WBS]
Date	15/02/2015
Auteur	Kévin SOULES
Responsable	Kévin SOULES
Email	vigilate_2017@labeip.epitech.eu
Sujet	Work Breakdown Structure
Mots clés	WBS, sécurité, vulnérabilités, architecture, dictionnaire
Version du modèle	1.1

Tableau des révisions

Date	Auteur	Section(s)	Commentaires
09/02/15	Tous	Diagramme (3.1) + Contexte (2)	Premières réflexions sur l'architecture et le contexte
15/02/15	Kevin Soules	Principe de base (1.3)	Début rédaction
14/02/15	Prune Budowski	Diagramme (3.1)	Mise en forme du diagramme complet
15/02/15	Prune Budowski	Résumé du document	Rédaction du résumé
15/02/15	Manuel Poncet & Kevin Soules	Contexte (2.1/2.2) et Résumé	Amélioration
15/02/15	Kevin Soules	Toutes	Intégration des différentes parties dans le document + rédaction glossaire & dico

Table des matières

1	Rappel de l'EIP	1
1.1	Qu'est-ce qu'un EIP et Epitech	1
1.2	Sujet de votre EIP	1
1.3	Principe de base du système futur	1
1.4	Glossaire	2
2	Contexte	4
2.1	Hypothèses	4
2.2	Contraintes	4
3	WBS	5
3.1	Représentation du WBS	5
3.2	Dictionnaire du WBS	6

Chapitre 1

Rappel de l'EIP

1.1. Qu'est-ce qu'un EIP et Epitech

Epitech, école de l'innovation et de l'expertise informatique propose un cursus en 5 années, basé sur une pédagogie par projet (à réaliser seul ou en groupe). L'un de ces projets, l'EIP (Epitech Innovating Project), réalisé par groupes de 6 à 15 étudiants, démarre au cours de la 3^e année et se déroule sur 2 ans et demi. Ce projet doit être particulièrement innovant, car son objectif est d'être commercialisable à la fin de la 5^e année d'Epitech.

1.2. Sujet de votre EIP

Le but de Vigilate est d'avertir les utilisateurs des services obsolètes ou potentiellement vulnérables affectant en particulier leur infrastructure (sites web, réseau d'entreprises, logiciels), dans le but de les informer des risques techniques encourus et des éventuelles mises à jour ou corrections à appliquer. Cela sans effectuer de scan de vulnérabilité. Nous proposons cependant un outil de scan de programme qui envoie la liste sur notre plateforme web.

1.3. Principe de base du système futur

Vigilate sera composé de quatre grands blocs fonctionnels : site web, programme de scan, backend et vm.

La partie frontend du site web communiquera via l'api avec la partie backend.

Le programme de scan communiquera également via l'api avec le backend pour mettre à jour la liste de logiciel que veut surveiller l'utilisateur, l'objectif étant de l'informer le plus rapidement possible et d'être simple d'utilisation.

La machine virtuelle sera une sorte de copie locale de ce qu'il y a sur nos serveurs. Elle permettra aux entreprises qui le souhaitent d'avoir un niveau de sécurité encore plus important.

Cet outil est adapté à plusieurs types de clients : Aussi bien un webmaster souhaitant garder un oeil sur le CMS de son site web, qu'une entreprise voulant veiller à la sûreté de ses services

informatiques.

1.4. Glossaire

- A -

API (Application Programming Interface) : un ensemble de classes, de méthodes ou de fonction qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels.

- B -

Backend : un back-end est un terme désignant un étage de sortie d'un logiciel devant produire un résultat. On l'oppose au front-end (aussi appelé un frontal) qui lui est la partie visible de l'iceberg.

- C -

CMS (Content Management System) : c'est une famille de solution destinés à la conception et à la mise à jour dynamique de sites Web. Ils permettent de s'intéresser qu'au contenu à publier.

CVE (Common Vulnerabilities and Exposures) : dictionnaire d'informations publiques relatives aux vulnérabilités informatiques. Par métonymie, on emploie souvent le terme CVE à la place de CVE ID (ou identifiant CVE), qui lui désigne le numéro qui renvoie à la fiche descriptive complète de cette vulnérabilité. Exemple : CVE-2013-4343.

- F -

Frontend : La partie frontend sera la partie avec laquelle l'utilisateur va interagir. Le serveur frontal intercepte les requêtes utilisateur et les ré-envoie vers le serveur backend.

- V -

VM (Machine Virtuelle) : une machine virtuelle est une illusion d'un appareil informatique créée par un logiciel d'émulation. Le logiciel d'émulation simule la présence de ressources matérielles et logicielles telles que la mémoire, le processeur, le disque dur, voire le système d'exploitation et les pilotes, permettant d'exécuter des programmes dans les mêmes conditions que celles de la machine simulée.

Chapitre 2

Contexte

2.1. Hypothèses

La partie backend sera développée en python. Un langage simple, rapide à écrire et performant.

Le site web utilisera le framework django et son frontend sera développé avec angular js

Le programme de scan sera développé en python ce qui permettra une portabilité aisée.

La distribution présente sur la vm sera une debian. Ce qui permettra d'avoir une stabilité reconnue dans le temps.

2.2. Contraintes

L'interface utilisateur devra être intuitive et facile d'utilisation pour l'utilisateur lambda, dans le but de favoriser un maximum l'accès à nos services en toute simplicité.

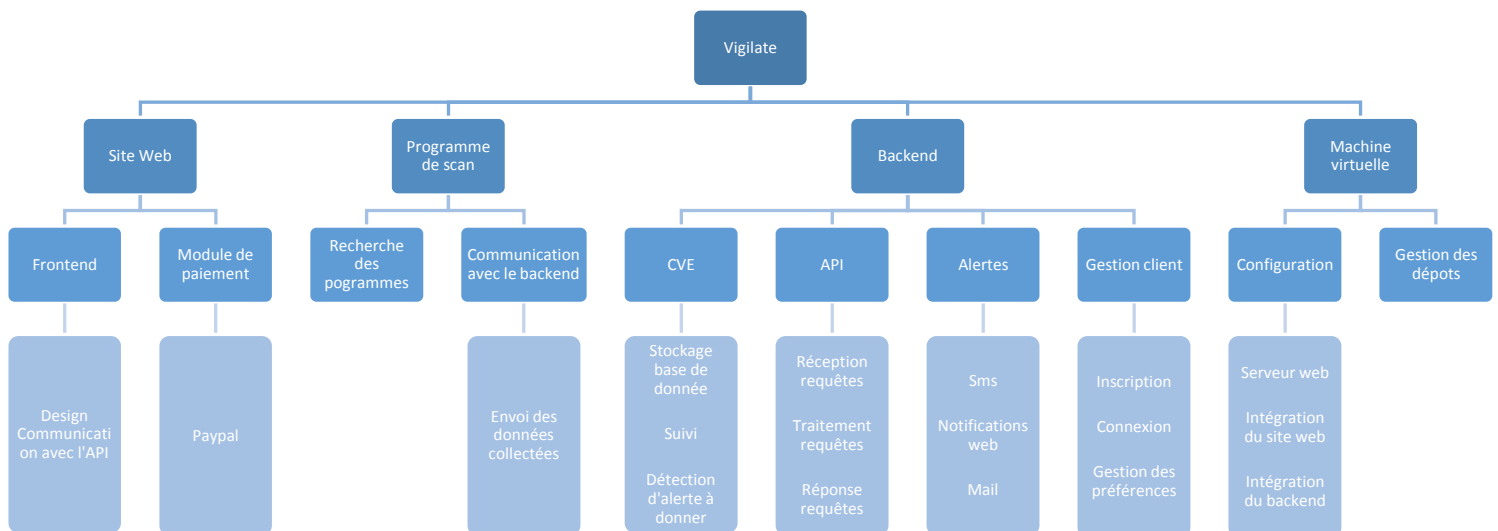
La réactivité sera un des points principaux à respecter afin de ne laisser s'échapper qu'un minimum de temps entre la diffusion d'une CVE et sa transmission aux clients concernés.

Être sécurisé afin d'éviter impérativement une potentielle fuite d'informations relative aux clients ou à notre infrastructure.

Chapitre 3

WBS

3.1. Représentation du WBS



3.2. Dictionnaire du WBS

ID	Fonctionnalité	Description	% réalisé
1	Site Web	Réaliser le site web	0%
1.1	Frontend	Réaliser la partie frontal	0%
1.1.1	Design	Réaliser le design du site	0%
1.1.2	Communication avec l'API	Réaliser la partie qui communique avec l'API	0%
1.2	Module de paiement	Intégration d'un module de paiement	0%
1.2.1	Paypal	Intégration de paypal	0%
2	Programme de scan	Réaliser un programme de scan	0%
2.1	Recherche des programmes	Réaliser la partie recherche de programme	0%
2.2	Communication avec le backend	Réaliser la partie qui communique avec le backend	0%
2.2.1	Envoi des données collectées	Envoi des données au backend	0%
3	Backend	Réaliser un backend	0%
3.1	CVE	Réaliser la partie qui gère les CVE	0%
3.1.1	Stockage en bdd	Stockage en bdd des CVE reçus	0%
3.1.1	Suivi	Suivre en permanence la sortie des nouvelles CVE	0%
3.1.1	Détection d'alerte à donner	Réaliser une partie qui va être lancé à chaque nouvelle CVE afin de détecter à qui il faut envoyer une alerte	0%
3.2	API	Réaliser l'API	0%
3.2.1	Réception requêtes	Réaliser la partie qui reçoit les requêtes	0%
3.2.2	Traitement des requêtes	Réaliser la partie qui traite les requêtes	0%
3.2.3	Réponse au requêtes	Réaliser la partie qui répond au requêtes	0%
3.3	Alertes	Réaliser la partie qui gère les alertes	0%
3.3.1	SMS	Réaliser un module d'alerte SMS	0%
3.3.2	Notification web	Réaliser un module d'alerte via l'interface du site web	0%
3.3.3	Mail	Réaliser un module d'alerte SMS	0%
3.4	Gestion utilisateurs	Réaliser la partie qui gère les utilisateurs	0%
3.4.1	Inscription	Réaliser une partie inscription	0%
3.4.2	Connexion	Réaliser une partie connexion	0%

3.4.3	Gestion préférences	Réaliser une partie de gestion de préférence	0%
4	Machine virtuelle	Réaliser une vm	0%
4.1	Configuration	Configurer la vm	0%
4.1.1	Serveur web	Configurer un serveur web	0%
4.1.2	Intégration site web	Intégrer le site web sur la vm	0%
4.1.3	Intégration backend	Intégrer le backend sur la vm	0%
4.2	Dépôts	Gestion des dépôts	0%