Plan de développement

Version	1.0
Date	13/12/2013
Rédigé par	Pascal Edouard
Relu par	Julien Legras & William Boisseleau
Approuvé par	Ayoub Otmani

MISES À JOUR

Version	Date	Modifications réalisées
0.1	14/12/2013	Création du document
1.0	14/12/2013	Première version
1.1	22/1/2014	Modification du plan de développement

Table des matières

1 Objet

Le projet s'inscrit dans un cadre relatif aux faits d'actualités autour de la sécurité, notamment concernant la crédibilité des fonctions de génération de nombres aléatoires ou encore sur l'application des standards dans les fonctions d'OpenSSL.

RSA et DSA peuvent échouer lamentablement lorsqu'ils sont utilisés par un mauvais générateur de nombres aléatoires mais combien de cas comme celui ci peut-il y avoir sur le web ? La première partie de ce projet essaiera de répondre à cette question en faisant un état des lieux des serveurs TLS et SSH et présentera des preuves montrant que des clefs vulnérables peuvent exister.

La deuxième partie de ce projet consistera à étudier les fonctions d'OpenSSL, en particulier sa partie développant la génération d'éléments aléatoires et ses primitives cryptographiques. Chaque fonction sera comparée avec les standards actuels (cf. PKCS) afin de déterminer si elle est efficace.

Enfin, la troisième partie de ce projet consistera à développer une application tournant sur serveur sécurisé afin d'analyser chaque machine cliente se connectant à celui-ci. L'application possédera une liste de critères et de menaces existantes. Par exemple on peut considérer la validité du certificat du client présenté lors de la connexion.

Ce projet est réalisé dans le cadre de l'enseignement de deuxième année de master Sécurité des Systèmes Informatiques. Il sera réalisé par un groupe de cinq étudiants. La première partie du projet consiste à rédiger les documents qui nous permettent de mieux définir le sujet, les objectifs, les risques et l'organisation du projet. Suite à cela, le développement des différentes applications durera 6 semaines.

Documents de référence :

- Mining Your Ps and Qs : Detection of Widespread Weak Keys in Network Devices
  by Nadia Heninger, Zakir Durumeric, Eric Wustrow, J. Alex Halderman
- ZMAP : Fast Internet-Wide Scanning and its Security Applications
  by Zakir Durumeric, Eric Wustrow, J. Alex Halderman

2 Terminologie et sigles utilisés

- **RSA/DSA :** Advanced Encryption Standard et Digital Signature Algorithm, sont deux algorithmes de chiffrement à bi-clef (publique et privée)
- **Appli RC :** Application de Récupération des certificats
- **Appli F :** Application de Factorisation
- **Certificat :** Document électronique utilisant une signature digitale afin de lier une clef publique à une identité

3 Méthodologie de développement

Ce projet reposera sur la méthode agile en prenant en compte ses valeurs culturelles et principes en l'appliquant dans une méthodologie SCRUM afin d'apporter une discipline de développement et de délivrer les résultats dans les meilleures conditions.

Pourquoi Scrum ? : Les méthodes agiles ont fait leur preuve dans leur efficacité et leur qualité de développement. De plus, Scrum permet un suivi et une transparence totale avec le client. Le découpage en sprint est adapté à notre projet puisqu'il se déroule en différente partie et sur une période relativement courte.

Voici les valeurs et les principes de la méthodologie Scrum :

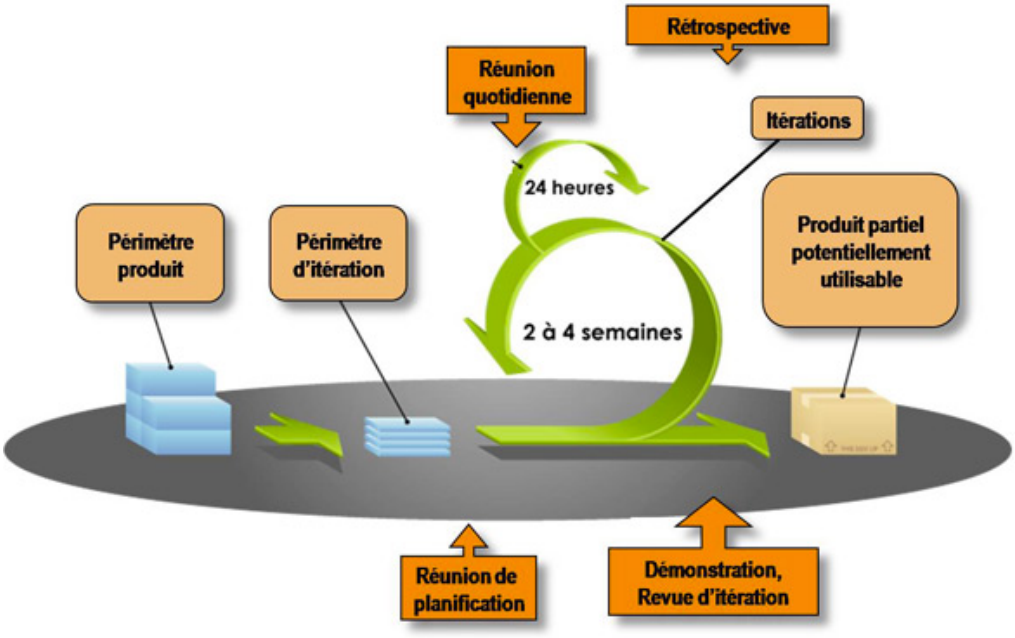
Les individus et leurs interactions plus que les processus et les outils : La méthodologie Scrum correspond à la communication entre les collaborateurs à tous les niveaux (client/fournisseurs, testeurs/programmeurs, ...) afin de ne pas perdre de temps ni d'énergie avec des malentendus ou de l'incompréhension.

La collaboration avec les clients plus que la négociation contractuelle : Une approche directe avec le client qui se sent beaucoup plus impliqué dans le projet afin qu'il puisse apporter ses avis et remarques.

L'adaptation au changement plus que le suivi d'un plan : Être capable de s'adapter lorsqu'une modification importante est nécessaire.

Ainsi, une grande priorité sera de satisfaire les demandes du client en livrant régulièrement des fonctionnalités, testées préalablement par des outils de tests développés, et les faire valider par le client. Chaque fonctionnalité validée sera intégrée au projet.

La méthodologie de Scrum représente correctement cette approche dans le cadre de notre projet. Il est rythmé par un ensemble de réunions clairement définies et strictement limitées dans le temps :



— **Planification du Sprint** (Sprint = itération) : au cours de cette réunion, l'équipe de développement sélectionne les éléments prioritaires du « Product Backlog » (liste ordonnancée des exigences fonctionnelles et non fonctionnelles du projet) qu'elle pense pouvoir réaliser au cours du sprint (en accord avec le « Product Owner »).
— **Revue de Sprint** : au cours de cette réunion qui a lieu à la fin du sprint, l'équipe de développement présente les fonctionnalités terminées au cours du sprint et recueille les feedbacks du Product Owner et des utilisateurs finaux. C'est également le moment d'anticiper le périmètre des prochains sprints et d'ajuster au besoin la planification de release (nombre de sprints restants).
— **Rétrospective de Sprint** : la rétrospective qui a généralement lieu après la revue de sprint est l'occasion de s'améliorer (productivité, qualité, efficacité, conditions de travail, etc) à la lueur du « vécu » sur le sprint écoulé (principe d'amélioration continue).
— **Mêlée quotidienne** : il s'agit d'une réunion de synchronisation de l'équipe de développement qui se fait debout (elle est aussi appelée « stand up meeting ») en 15 minutes maximum au cours de laquelle chacun répond principalement à 3 questions : « Qu'est ce que j'ai terminé depuis la dernière mêlée ? Qu'est ce que j'aurai terminé d'ici la prochaine mêlée ? Quels obstacles me retardent ? »

4 Organisation et responsabilités :

Lors de la première et la troisième partie du projet, qui consistent au développement de différentes applications, chaque membre de l'équipe sera assigné un rôle particulier :

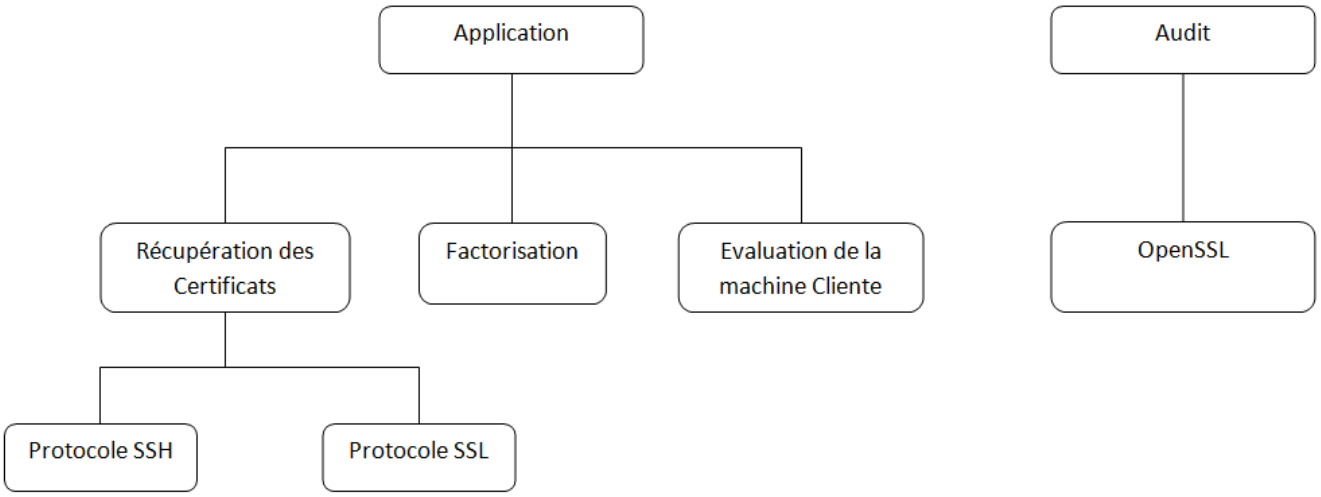
Développeur : responsables de la production du code. Ils aident aussi à la ré-estimation de la charge de travail en fonction de l'avancement du projet. Ces rôles sont tenus par Julien LEGRAS et Pascal EDOUARD.

Client : notamment le «Product Owner», qui porte vision du produit à réaliser et représente généralement le client afin d'assurer l'intermédiaire avec l'équipe. Ce rôle est tenu par Claire SMETS.

Testeur : responsable de la création des procédures de tests afin de valider le bon fonctionnement des applications développées. Ce rôle est tenu par Mathieu LATIMIER et William BOISSELEAU.

Coach : notamment le «Srum Master», reste le garant de l'application de la méthodologie Scrum et de l'équipe en termes de communication et le fonctionnement. Ce rôle est tenu par Pascal EDOUARD.

5 Organigramme des tâches :



6 Evaluation du projet :

L'approche Scrum commence par lister les exigences du client afin de produire le « Product Backlog ». On définit une unité de coût (en terme de complexité) avec la colonne Estimation. Elle permet de faciliter l'ordonnancement du Product Backlog, la planification des sprints et des releases.

7 Dimensionnement des moyens :

Nous avons besoin, pour ce projet, d'une bonne connexion internet afin de récupérer le maximum de certificats avec le minimum de perte de connexion possible. Une machine moyenne (normale) en termes de performance est nécessaire pour cette partie. Cependant, l'application de factorisation des moduli nécessite quelque chose de beaucoup plus puissant en termes de disque, mémoire et calcul de processeur. Nos possibilités d'obtenir un serveur de cette taille peuvent varier comme suit :

1. Serveur local de l'université,
2. Serveur de calcul au CRIHAN,
3. Serveur Amazon à un prix raisonnable.

Le code source ainsi que les différents documents produits sont stockés sur le serveur git : (https ://github.com/RandomGuys)

8 Description des tâches :

		Tâche	Description
Sprint 1	**App. RC**	1	Récupération de la liste des adresses IP des serveurs
		2	Mise en place des procédures de test
		3	Mise en place d'un client SSH simple
		4	Mise en place d'un client SSL/TLS simple
		5	Récupération des certificats SSL/TLS
		6	Stockage des certificats SSL/TLS
	App. F	7	Extraction des clefs des certificats
		8	Développement des arbres de produits
		9	Développement des arbres de restes
		10	Mise en évidence des facteurs communs
		11	Tests
		12	Débogage
		13	Première livraison cliente
Sprint 2	**OpenSSL**	14	Lister les fonctions d'OpenSSL à auditer
		15	Lister les standards utilisés par chaque fonction à auditer
		16	Mettre en place la page web
		17	Auditer les fonctions d'OpenSSL listées
		18	Développer les tests des fonctions auditées
		19	Présenter les clefs communs sur le site
		20	Présenter le résultat de l'audit sur le site
		21	Deuxième livraison cliente
Sprint 3	**App. Eva**	22	Lister les critères à vérifier sur une machine cliente
		23	Développer la page de connexion client
		24	Récupération des données du navigateur
		25	Évaluer chaque critère de la machine cliente
		26	Présenter les résultats de l'évaluation sur la page web
		27	Tests et débeugage
		28	Troisième livraison cliente

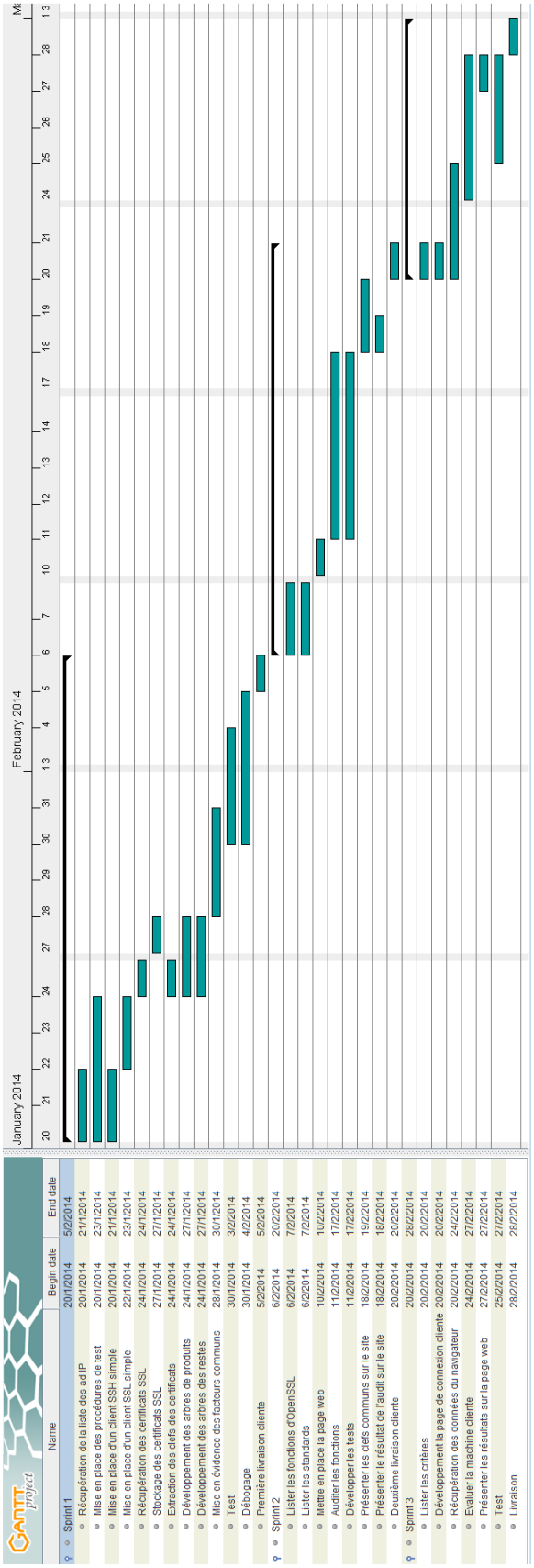
	N° Tâche	Début	Fin	Durée	Effectif	Charge	Ressources
Sprint 1	1	20/1/2014	21/1/2014	2	1	2	Julien
	2	20/1/2014	23/1/2014	4	2	8	William, Mathieu
	3	20/1/2014	21/1/2014	2	2	4	Pascal, Claire
	4	22/1/2014	23/1/2014	2	1	2	Julien
	5	24/1/2014	24/1/2014	1	3	3	Julien, Claire, Pascal
	6	27/1/2014	27/1/2014	1	3	3	Julien, Claire, Pascal
	7	24/1/2014	24/1/2014	1	2	2	Mathieu, William
	8	24/1/2014	27/1/2014	1.5	2	3	Mathieu, Pascal
	9	24/1/2014	27/1/2014	1	2	3	Claire, William
	10	28/1/2014	30/1/2014	2.5	2	5	Claire, William
	11	30/1/2014	3/2/2014	3	2	5	William, Pascal, Mathieu
	12	30/1/2014	4/2/2014	3	2	8	Julien, Claire
	13	5/2/2014	5/2/2014	1	5	–	Tous
Sprint 2	14	6/2/2014	10/2/2014	3	3	9	William, Julien, Mathieu
	15	6/2/2014	7/2/2014	2	2	4	Pascal, Claire
	16	10/2/2014	10/2/2014	1	2	2	Pascal, Claire
	17	11/2/2014	17/2/2014	5	2	10	Claire, Julien
	18	11/2/2014	17/2/2014	5	3	15	Pascal, William, Mathieu
	19	18/2/2014	20/2/2014	3	3	9	Claire, Pascal, Mathieu
	20	18/2/2014	18/2/2014	1	2	2	Julien, William
	21	20/2/2014	20/2/2014	1	5	–	Tous
Sprint 3	22	20/2/2014	20/2/2014	0.5	2	2	Julien, Pascal
	23	20/2/2014	20/2/2014	0.5	1	0.5	Mathieu
	24	20/2/2014	24/2/2014	3	2	6	Julien, Pascal
	25	24/2/2014	27/2/2014	2	3	6	Claire, Pascal, Julien
	26	20/2/2014	21/2/2014	1	2	2	Mathieu, William
	27	25/2/2014	27/2/2014	3	2	6	Mathieu, William
	28	28/2/2014	28/2/2014	1	5	–	Tous

Les sprints 1 et 2 sont les parties indispensable. Lors de la répartition des tâches, nous avons donc opté pour la méthode du Planning Poker le mercredi 15 janvier 2014, pour calculer les charges de chaque tâche en équipe et les effectifs nécessaires correspondants. Nous avons donc pu connaitre le délais nécessaire pour terminer les 2 premiers sprints.

La date de soutenance du projet étant le vendredi 28 février, la dernière semaine sera plutôt dédié à la rédaction du rapport et préparation de la soutenance. Cependant, dans le cas où nous nous trouvons en avance par rapport aux dates fixées des deux premiers sprint, nous pourrons envisager de considerer le troisième sprint [1], et dans ce cas seulement.

1. Ceci explique pourquoi les charges du troisième sprint furent calculées et ajoutées dans ce tableau

9 Plan de développement :



Un outil en ligne, Teambox (récemment renommé en Readbooth) sera utilisé pour gérer la répartition des tâches et suivre leur évolution. Teambox offre une interface permettant de voir ses tâches en cours. La convention utilisée par Teambox, une tâche est assignée à une personne uniquement mais peut avoir plusieurs "watchers" (les personnes qui seront notifiées lors d'un changement ou modification apportées sur la tâche) alors que notre plan de développement précise que plusieurs membres de l'équipe peut travailler sur la même tâche. De ce fait, pour chaque tâche, une personne sera désigné comme responsable de la tache et autres seront donc les "watchers" mais seront toujours capable de modifier la tâche (ajouter des commentaires, modifier son état, ...).

10 Gestion de la documentation :

Les documents de référence pour la gestion de ce projet seront disponibles au format pdf sur le serveur git.

Chaque membre est responsable d'un document livrable, qui évolue au fil du projet et sera incrémenté par un numéro de version :

- La Spécification Technique du Besoin : Claire SMETS
- Le Document d'Architecture Logicielle : Julien LEGRAS
- L'Analyse des Risques : Pascal EDOUARD
- Le Cahier des Recettes : Mathieu LATIMIER et William BOISSELEAU
- Le Plan de Développement : Pascal EDOUARD

L'ensemble du groupe devra se tenir au courant de l'évolution de chaque document. De plus, après chaque réunion avec le client, un compte-rendu sera rédigé et validé par l'équipe avant d'être soumis au client, qui pourra ensuite le valider. Le compte-rendu sera accessible sur le dépôt de documents git.
Ce dépôt permettra à l'équipe de suivre l'évolution des documents (livrables, compte-rendu, ...) ainsi que les fichiers de scripts développés durant le projet. En cas de perte des données, il sera possible de retrouver une ancience version et limiter les pertes.

Une intégration Git - Teambox est également possible afin de suivre la gestion de la documentation et le plan de développement plus facilement. En effet, la fonction de hooks de Git permet de rajouter un commentaire à chaque fois qu'un commit (sauvegarde d'une modification directement sur le dépot) est effectué par l'un des membres de l'équipe.