Document d'architecture logicielle

Version	1.2
Date	22/01/2014
Rédigé par	Julien Legras
Relu par	William Boisseleau
Approuvé par	Ayoub Otmani

MISES À JOUR

Version	Date	Modifications réalisées
0.1	12/12/2013	Création du document
1.0	18/12/2013	Première version approuvée
1.1	18/01/2014	Ajout du schéma général Justification de la configuration
1.2	22/01/2014	Suppression de la partie SSH

Table des matières

1 Objet

Ce document présente l'architecture utilisée pour réaliser les outils nécessaires à notre audit SSL :
— Récupération des certificats : Application RC ;
— Factorisation : Application F.

2 Documents applicables et de référence

— STB (Spécification Technique des Besoins) ;
— N. Heninger, Z. Durumeric, E. Wustrow, J. A. Halderman. Mining Your Ps and Qs : Detection of Widespread Weak Keys in Network Devices. Proceedings of the 21st USENIX Security Symposium. 2012.

3 Terminologie et sigles utilisés

— **Appli RC** : Application de Récupération des Certificats
— **Appli F** : Application de Factorisation
— **Machine D** : Machine Distante
— **Machine L** : Machine Locale
— **Certificat** : Un certificat électronique (aussi appelé certificat numérique ou certificat de clé publique) peut être vu comme une carte d'identité numérique. Il est utilisé principalement pour identifier une entité physique ou morale, mais aussi pour chiffrer des échanges.
Il est signé par un tiers de confiance qui atteste du lien entre l'identité physique et l'entité numérique (Virtuel).
— **Audit** : Procédure consistant à s'assurer du caractère complet, sincère et régulier des comptes d'une entreprise, à s'en porter garant auprès des divers partenaires intéressés de la firme et, plus généralement, à porter un jugement sur la qualité et la rigueur de sa gestion. Ici ce n'est pas une entreprise qui en sera la cible mais un programme : OpenSSL.
— **Modulus** : En mathématiques et plus précisément en théorie algébrique des nombres, l'arithmétique modulaire est un ensemble de méthodes permettant la résolution de problèmes sur les nombres entiers. Ces méthodes dérivent de l'étude du reste obtenu par une division euclidienne.

4 Configuration requise

4.1 Machine de récupération de certificats (Instance micro Amazon Web Services EC2)

— accès au port 443 en destination
— bande passante importante
— espace disque pour stockage des certificats : 20 Go
— système : Ubuntu Server 12.04 LTS (Dernière version LTS)

4.2 Machine de factorisation

Recommandations de l'équipe de chercheurs ayant développé l'outil fastgcd :
— au moins 30 Go de RAM (stockage en RAM de très grands entiers)
— espace disque pour stockage : 150 Go (stockage sur disque des calculs intermédiaires)

5 Architecture statique

5.1 Structure

Les principales parties à développer sont :

- Application récupération de certificats
- Application de factorisation de grands entiers RSA
- Données :
  - adresses IP fournies par ZMap
  - clefs publiques RSA récupérées par l'application RC

Figure 1 – Schéma du traitement du Sprint 1

5.2 Description des constituants

	Application RC SSL/TLS
Rôle	Récupération et stockage des certificats SSL/TLS
Propriétés et attributs de caractérisation	Permet d'obtenir des certificats SSL/TLS en tentant une connexion
Dépendances avec d'autres constituants	ZMap : liste des adresses IP ayant le port 443 ouvert
Langages de programmation	Scripts Perl avec commandes openssl
Procédé de développement	1. lancement de la commande openssl s_client 2. extraire le certificat de la session (openssl sess_id) et le stocker 3. itérer les connexions pour toutes les adresses IP données par ZMap
Taille complexité	20% du projet

	Application F
Rôle	Trouver des facteurs communs des clefs récupérées
Propriétés et attributs de caractérisation	GMP, cmake
Dépendances avec d'autres constituants	Application RC SSL/TLS
Langages de programmation	C
Procédé de développement	1. Arbre des produits 2. Arbre des restes 3. Exploitation des résultats
Fonction à développer	buildProductTree() : construit l'abre des produits grâce à l'algorithme 1 buildRemainderTree() : construit l'arbre des restes grâce à l'algorithme 2
Taille complexité	20% du projet

	Serveur web
Rôle	Affiche les résultats de l'audit
Propriétés et attributs de caractérisation	Pages statiques et dynamiques qui affichent sous forme graphique les résultats de l'audit
Dépendances avec d'autres constituants	Appli RC, F
Langages de programmation	HTML, PHP, JS
Procédé de développement	1. Affichage clefs récupérées 2. Affichage des facteurs communs 3. Documentation audit OpenSSL
Taille complexité	10% du projet

	Évaluateur niveau de sécurité
Rôle	Évaluer le niveau de sécurité d'un navigateur web
Propriétés et attributs de caractérisation	Programme qui établie la connexion SSL/TLS avec un navigateur et qui évalue le niveau de sécurité de la connexion
Dépendances avec d'autres constituants	Serveur web
Langages de programmation	C
Procédé de développement	1. Mise en place du contexte SSL/TLS 2. Récupérer les informations du hello client 3. Calculer le score associé au niveau de sécurité 4. Renvoyer une page html avec le score
Fonctions à développer	analyseHelloClient() : analyse le paquet HELLO du client (algorithmes supportés, certificat du navigateur etc.) computeScore() : calculer un score à partir de l'analyse et de critères d'évaluation fixés lors en début de sprint
Taille complexité	20% du projet

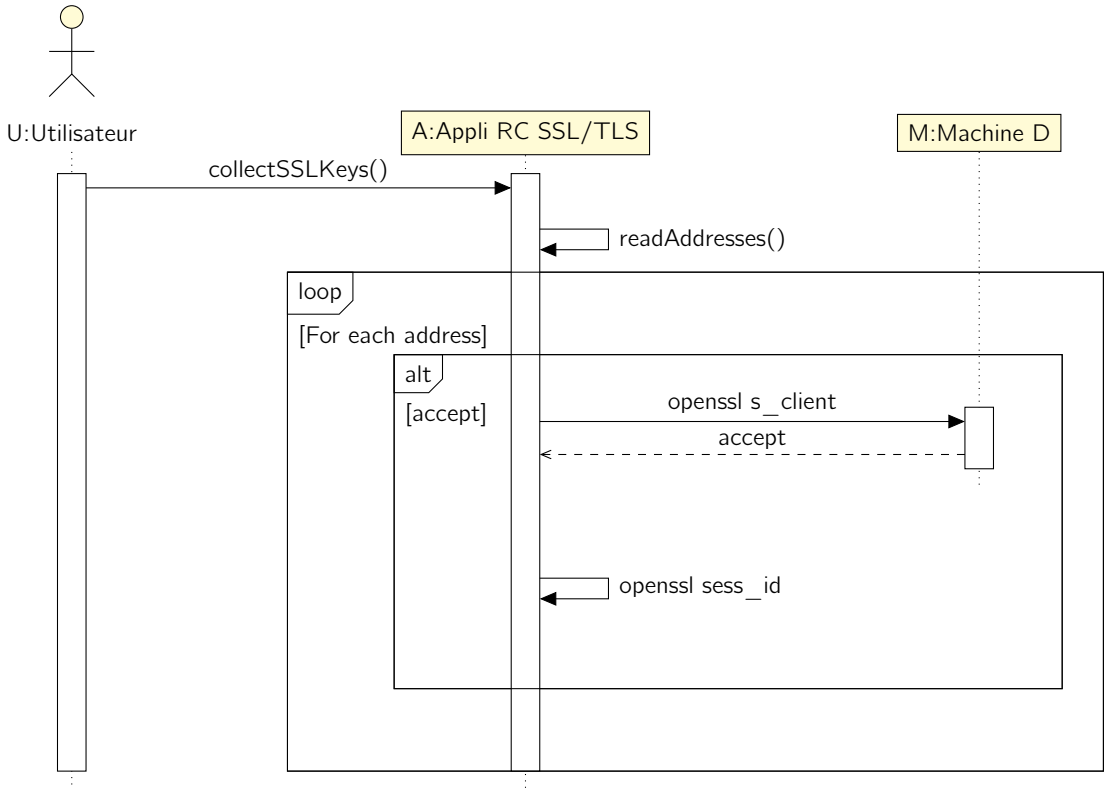
6 Fonctionnement dynamique

6.1 UC.1 : Récupération des certificats SSL/TLS

UC.1.2 : Récupération des certificats SSL/TLS	
Composants mis en jeu	Appli RC SSL/TLS
Intervenants	Utilisateur, Machine D
Processus de mise en œuvre	

Pour toutes les adresses IP ayant un port 443 ouvert :

1. Établissement de la connexion TCP entre Appli RC SSL/TLS et Machine D
2. Échange des clefs/certificats entre Appli RC SSL/TLS et Machine D
3. Appli RC SSL/TLS stocke la clef/le certificat de Machine D dans un répertoire `keys/certs` sous forme de fichier dont le nom est l'adresse IP de Machine D (si la clef n'est pas déjà présente)
4. Fermeture de la connexion



6.2 UC.2 : Factorisation des moduli des certificats

UC.2 : Factorisation des moduli des certificats	
Composants mis en jeu	Appli F
Intervenants	Utilisateur
Processus de mise en œuvre	

1. Récupérer la liste des clefs
2. Construire l'arbre des produits en stockant chaque niveau dans un fichier
3. Construire l'arbre des restes
4. Stocker les facteurs communs dans un fichier

Entrées : Tableau des moduli des clefs publiques : T
Sorties : Hauteur de l'arbre, produits des moduli des clefs publiques
Données : Tableaux v, tmp ; Entier i, level

$v \leftarrow T$;
$level \leftarrow 0$;
tant que $|v| > 1$ **faire**
  $tmp \leftarrow \emptyset$;
  pour chaque $i \in \{0, .., |v|/2\}$ **faire**
    $tmp[i] \leftarrow v[i \times 2] \times v[i \times 2 + 1]$;
  fin
  $storeProductLevel(v, level)$;
  $v \leftarrow tmp$;
  $level \leftarrow level + 1$;
fin
retourner *level*

Algorithme 1 : Construction de l'arbre des produits

Entrées : Hauteur de l'arbre des produits : level
Sorties : PGCDs des moduli des clefs publiques
Données : Tableaux P, v, w ; Entier i
$P \leftarrow getProductLevel(level)$;
tant que $level > 0$ **faire**
  $v \leftarrow getProductLevel(level - 1)$;
  pour chaque $i \in \{0, .., |v|\}$ **faire**
    $v[i] \leftarrow P[i/2] \pmod{v[i]^2}$;
  fin
  $level \leftarrow 1$;
  $storeRemainderLevel(v, level)$;
  $v \leftarrow tmp$;
  $level \leftarrow level + 1$;
fin
$w \leftarrow \emptyset$;
pour chaque $i \in \{0, .., |v|\}$ **faire**
  $w[i] \leftarrow P[i/2] \pmod{v[i]^2}$;
  $w[i] \leftarrow w[i]/v[i]$;
  $w[i] \leftarrow pgcd(w[i], v[i])$;
fin
retourner w

Algorithme 2 : Construction de l'arbre des restes

6.3 UC.3 : Présentation des résultats

UC.3 : Présentation des résultats	
Composants mis en jeu	Serveur web
Intervenants	Utilisateur
Processus de mise en œuvre	

1. Développement partie clefs publiques récupérées (tri par critères : taille, issuer si connu etc.)
2. Développement partie facteurs communs avec graphiques HighCharts
3. Documentation avec doxygen de l'audit d'OpenSSL

6.4 UC.4 : Audit d'OpenSSL

UC.4 : Audit d'OpenSSL	
Composants mis en jeu	code source d'OpenSSL
Intervenants	
Processus de mise en œuvre	

1. Sélectionner les fonctions à auditer
2. Développer les codes de tests sur ces fonctions
3. Générer une documentation doxygen associée à l'audit

6.5 UC.5 : Évaluation du niveau de sécurité du navigateur client

UC.5 : Évaluation du niveau de sécurité du navigateur client	
Composants mis en jeu	Serveur web
Intervenants	Utilisateur
Processus de mise en œuvre	

1. Lister la liste des protocoles supportés par le navigateur de l'utilisateur
2. Lister la liste des systèmes de chiffrement supportés par le navigateur de l'utilisateur
3. Lister les détails du protocole utilisé pour la connexion HTTPS
4. Établir un score selon les résultats précédents