



Plan d'assurance qualité : P.A.Q

Sommaire

- 1 Identification des acteurs du projet AdminSSH
- 2 Description du projet
 - 2.1 Introduction.
 - 2.2 AdminSSH est une solution d'identification et de sécurisation de l'administration
 - 2.3 Division du travail
- 3 Livrables
- 4 Communication
- 5 Gestion des ressources
- 6 Contraintes
- 7 Procédures

Identification des acteurs du projet AdminSSH

- **Le client** : Lalitte Eric 01.43.90.21.12
- **chef de projet** : Chasles Philippe
- **L'équipe de développement** :
 - Grandsart Emeric
 - Le Cann Vincent
 - Goldery Yohan
 - Chasles Philippe
- **commanditaire** : Lalitte Eric 01.43.90.21.12

Description du projet

Introduction.

Admin SSH est un projet qui a 2 objectifs principaux.

- **Le premier est de sécuriser l'accès et la copie vers des serveurs de types différents.**
Ceci est mis en oeuvre à l'aide de SSH.
- **Le deuxième objectif est d'avoir une trace des accès et des copies à ces serveurs.**
Pour ce faire nous allons mettre un serveur "tampon" par lequel tout les accès et les copies devront passer. En effet ce serveur que nous appellerons bastion sera le seul à avoir directement accès à ces serveurs. Tout les admins devront donc passer par ce serveur (qui au passage gardera une trace des accès et des copies) avant que celui-ci ne redirige leur requête vers le serveur qui les intéresse.

Non seulement les accès seront sécurisés, et ensuite a partir d'un site web nous pourrons voir qui à accéder à quel serveur à quel instant.

AdminSSH est une solution d'identification et de sécurisation de l'administration

- **Un bastion** : qui correspond à notre serveur central

Les administrateurs se connectent à ce serveur de manière transparente. Celui-ci "log" cet accès et l'accès au serveur qui intéresse l'administrateur. Lorsque l'administrateur se connecte au bastion, ce dernier créé automatiquement une nouvelle connexion vers le serveur désiré. Cela doit être transparent pour l'administrateur.

- **Un serveur Key :**

Il va générer les clés ssh qui seront ensuite placées sur le bastion, sur les machines des administrateurs et sur les serveurs.

- **Une gestion sécurisée de transferts de fichier.**

Le transfert de fichier doit être aussi transparent qu'une connexion normale à un serveur. Celle-ci doit laisser une trace sur le bastion.

- **Une solution d'administration à distance sécurisée.**

Nous allons développer un site web qui va permettre à un administrateur central de gérer les clés et les accès des administrateurs. Il pourra décider qui a accès à un serveur en particulier. Il pourra aussi à partir de site web observer les connexions et les copie de fichier et ainsi avoir un historique des accès sur les serveurs.

Division du travail

Notre travail sera divisé en 3 grandes parties

- **Installation et configuration des serveurs.**

Bastion : installation de base pour l'instant : nous avons besoin que de ssh

Mise en place d'un environnement chrooter pour chaque nouvel utilisateur créer

Mise en place d'un profil modifier qui permet d'avoir un lancement de script au démarrage du Shell utilisateur

Mise en place des connexions vers les serveurs de destination

Mise en place des scripts de connexion utilisateur (pour se diriger vers un serveur de destination), et d'une création d'un fichier de log

Mise en place d'une tache automatisant le rapatriement du fichier de log vers le serveur web

Serveur key (Marge): installation de base + apache 2 /PHP/MySQL

Mise en place du serveur apache2

Mise en place de PHP et MySQL

Mise en place d'un script de création de paires de clé RSA

Mise en place d'un déclenchement de commande et script à distance

Gestion des utilisateurs dans une base de données

Mise en place de la page web

Mise en place de la page Admin et Affichage des logs

Serveur de test (Homer) : une debian de base avec ssh

Blocage de toutes les entrées ssh sauf celle du bastion et du serveur web

Bonne configuration de ssh

- **Développement des scripts Perl**

Nous utiliserons le perl plutôt que le bash. En effet le perl est commun à tous les systèmes d'exploitations que nous utiliserons (même Windows). Cela évite le problème de compatibilité.

- **Développement de l'interface web**

Livrables

Voici la liste des éléments qui seront livrés à la fin du projet :

- **Sur un CD :**
 - Document d'avant projet
 - Plan d'assurance qualité (PAQ)
 - Planning
 - Un manuel d'utilisation de la solution.
 - Codes sources des scripts
 - Fichiers de configuration des différents serveurs installés.
- **Sur la machine :**
 - Serveur Web : Apache2/PHP5
 - Serveur générateur de clés OpenSSH

Communication

- **Réunions :**
 - Les réunions entre l'équipe de développement et l'enseignant référant auront lieu tous lundi matin à 10h.
 - L'équipe de développement se réunira pour des réunions d'avancement du projet tous les lundis et vendredis pendant la matinée afin de faire le point sur le travail effectué, ainsi que les prochaines tâches à accomplir.
 - Toutes les réunions auront lieu à l'IN'TECH Info à Ivry
- **Autres**
 - Contact par mail pour de l'aide technique au chef de projet client, et pour de l'aide fonctionnelle au commanditaire.

Gestion des ressources

- **Equipe de développement :**

- PC portable équipé de Windows XP professionnel
- Installation d'OpenSSH-server sur les 3 machines serveurs pour un travail a distance possible. Configuration des machines :

Serveur sous Open BSD

*Mémoire vive 768 Mo (2 x 256)
CPU Intel Pentium 4 1.70GHz (cache 256kb)
Carte mère Compaq
2 machines de ce type: V7E003 et V7E015*

Serveur sous debian

*Mémoire vive 512 Mo
CPU Intel Pentium 3 1Ghz
1 machine de ce type: V7G010*

Contraintes

Ce projet concerne en premier lieu la sécurisation des accès d'administration, la contrainte principale est donc de pouvoir offrir une solution ayant un fort niveau de sécurité, tout en restant exploitable facilement pour les administrateurs, et compatible avec tout système d'exploitation.

Il faut aussi que la solution puisse s'intégrer à tout type de système d'information composé d'éléments hétérogènes.

Pour cela :

- Les serveurs seront sous Linux, les distributions seront : OpenBsd, Debian.
- Les pages web seront validés W3C pour une compatibilité avec tous les navigateurs.
- Les logiciels utilisés seront gratuits pour éviter les problèmes liés aux licences.

Le temps de réalisation de ce projet est d'environ 12 semaines, le projet devra donc impérativement être fonctionnel pour le client à la fin de ces 12 semaines.

Procédures

- *Sauvegardes*

Les documents sont sauvegardés sur un serveur externe box2.hostyle.fr:

- Les principaux fichiers de configuration des services que nous utiliserons.
- Un fichier qui contient les adresses IP, les ports et noms des serveurs, ainsi que les comptes root et mots de passe.
- Les scripts que nous développerons.
- Les documents de chacun. (Nous avons tous un répertoire personnel, accessible à tous.
- Les comptes rendus de réunion.

- *Gestion des documents*

Les différents documents seront nommés de la manière suivante :

TypeDeDoc-JJ_MM_AAAA

Exemple pour un compte rendu : CR_26032008.doc

- *Planning*

Il sera mise à jour hebdomadairement lors de chaque réunion.
Planning en ligne sur Itipédia