Configuration de SSH

Introduction

Le présent compte rendu porte sur les travaux réalisés dans le cadre du Bloc 2 - SISR, axé sur la connexion sécurisée à distance.

Les travaux ont été axés sur l'utilisation du protocole SSH pour établir une connexion sécurisée entre un client et un serveur distant.

Les objectifs des travaux étaient de comprendre les notions de sécurité liées à la mise en place d'un service de connexion à distance SSH, de mettre en place l'authentification SSH par paire de clés RSA, de déployer une clé RSA ou DSA sur un serveur, de configurer le serveur SSH pour qu'il accepte et utilise l'authentification par clé RSA, et de tester le fonctionnement de l'authentification par clés en se connectant en SSH sur le serveur.

Le compte rendu présentera les résultats des travaux, les solutions proposées pour minimiser les attaques sur le service SSH, ainsi que les problèmes rencontrés et les solutions pour y remédier.

Présentation

Dans le cadre de ces travaux, plusieurs solutions ont été proposées pour minimiser les attaques sur le service SSH.

Tout d'abord, l'utilisation de clés d'authentification à la place des mots de passe a été recommandée pour renforcer la sécurité de la connexion SSH.

Ensuite, il a été suggéré de limiter le nombre de tentatives de connexion pour éviter les attaques par force brute. Enfin, la mise en place d'un pare-feu pour restreindre l'accès au service SSH aux adresses IP autorisées a été recommandée.

Au cours des travaux, un problème courant rencontré lors de la connexion en SSH a été l'erreur “REMOTE HOST IDENTIFICATION HAS CHANGED”. Pour résoudre ce problème, il a été recommandé de vérifier l'authenticité de la nouvelle clé en contactant l'administrateur système du serveur distant, de mettre à jour la clé dans le fichier “known_hosts” en utilisant la commande “ssh-keygen -f” pour supprimer l'ancienne clé et ajouter la nouvelle clé, et de se connecter en SSH une fois la clé mise à jour.

TRAVAUX - SSH

- **Que signifie -p666 dans la commande de connexion ssh ?**

Cela signifie que l'on spécifie que l'on souhaite se connecter au service ssh sur le port 666. Dans le cas où le port par défaut (22) a été changé dans la configuration du serveur, ceci est nécessaire.

- **Que signifie le message encadré “The authenticity of host … continue connecting (yes/no)?” qui apparaît lors de la première connection à un serveur ssh ?**

Ce la signifie que le client ssh ne connait pas l'empreinte de la clé publique du serveur. Il souhaite donc savoir s'il doit abandonner la connexion (doute sur l'authenticité du serveur) ou s'il doit sauvegarder l'empreinte dans le fichier ~/.ssh/known_hosts. Cette empreinte est un hash (MD5 puis SHA256 depuis 2015) de la clé publique du serveur.

- **Quel problème de sécurité cela pose t-il ?**

Un intercepteur (Man in the Middle) qui intercèpte votre première connexion à un serveur ssh pourrait se faire passer pour lui sans que le serveur ni vous ne soyez au courant. .Une solution pourrait être par exemple de demander de vive voix à l'administrateur du serveur l'empreinte de la clé publique du serveur pour vérifier qu'elle est la même que celle affichée dans le message de première connexion.

- **Qu'est-ce qu'une attaque par brute force ?**

La méthode de recherche par force brute ou recherche exhaustive est une méthode de résolution de problème dans les domaines de la cryptologie, de l'informatique et de la théorie des jeux. Cette méthode tient justement son nom du fait qu'elle soit basée sur l'utilisation et l'essai de toutes les solutions possibles. Dans le cas de notre serveur ssh, une attaque par force brute consisterait à essayer tous les mots de passe possibles (soit par dictionnaire, soit par combinaison) pour un utilisateur donné, typiquement root. Donnez trois solutions pour minimiser ce type d'attaque sur le service ssh. On peut désactiver l'authentification par mot de passe, ne pas permettre de se connecter en tant que root, changer le port par défaut, fixer un nombre maximum de tentatives d'authentification, limiter le nombre de connexions par minute avec iptables... etc.

Vous êtes aministrateur sur un serveur web, auquel vous pouvez vous connecter par ssh. Un technicien de votre hébergeur renverse son café sur votre serveur qui tombe en panne. L'hébergeur vous met à disposition immédiatement un second serveur web qui possède exactement la même configuration IP que votre serveur habituel.

- **Vous essayez de vous connecter en ssh, un message d'erreur s'affiche... Pourquoi ?**

Bien que le nouveau serveur ait la même configuration IP, il possède une clé publique différente. L'empreinte de la clé publique du serveur qui est stockée dans le fichier ~/.ssh/known_hosts du client n'est donc pas la même que celle du serveur de secours. Le client interprète donc ça comme une potentielle attaque de Man in the Middle et donc rejette la connexion.

- **Comment résoudre ce problème ?**

ssh-keygen -R [hostname] permet de supprimer l'ancienne empreinte de clé qui était stockée dans ~/.ssh/known_hosts. Supprimer ou éditer le fichier ~/.ssh/known_hosts afin d'enlever l'ancienne empreinte du serveur. Désactiver la vérification de l'empreinte de la clé de l'hôte en ajoutant StrictHostKeyChecking no dans le fichier ~/.ssh/config.

Les logiciels avec interface graphique pour la prise de contrôle à distance présentent des avantages et des inconvénients

Avantages

1. **Interaction visuelle :** Les logiciels avec interface graphique permettent d'avoir un accès visuel au bureau de l'ordinateur distant, facilitant l'interaction avec les applications et les fichiers comme si vous étiez physiquement devant l'ordinateur distant.

2. **Convivialité :** L'interface graphique rend la prise de contrôle à distance plus conviviale pour les utilisateurs qui ne sont pas familiers avec les lignes de commande ou les interfaces textuelles.
3. **Polyvalence :** Certains logiciels avec interface graphique offrent des fonctionnalités avancées telles que le transfert de fichiers, le partage d'écran et la collaboration en temps réel.

Inconvénients

1- **Utilisation des ressources** :Les logiciels avec interface graphique peuvent consommer plus de ressources système que les solutions en ligne de commande, ce qui peut être un inconvénient sur des connexions à faible bande passante ou des ordinateurs distants avec des capacités limitées.

2- **Dépendance à l'environnement graphique** :Certains logiciels avec interface graphique peuvent nécessiter un environnement graphique spécifique sur l'ordinateur distant, ce qui peut limiter leur utilisation dans des scénarios où seul un accès en ligne de commande est disponible.

3- **Sécurité** :Les interfaces graphiques peuvent introduire des vulnérabilités supplémentaires en raison de la complexité des interactions utilisateur, nécessitant une attention particulière à la sécurité lors de la configuration et de l'utilisation de ces logiciels.

Il est important de prendre en compte ces avantages et inconvénients lors du choix d'un logiciel de prise de contrôle à distance, en fonction des besoins spécifiques de l'utilisateur et des contraintes techniques de l'environnement d'utilisation.

Sécurité lors de la connexion à distance

1. **Utilisez une connexion sécurisée :** Utilisez un protocole de connexion sécurisé tel que SSH (Secure Shell) pour établir une connexion cryptée entre l'ordinateur local et l'ordinateur distant. Évitez d'utiliser des protocoles non sécurisés tels que Telnet ou FTP.

2. **Utilisez des mots de passe forts :** Utilisez des mots de passe forts et complexes pour les comptes d'utilisateur sur l'ordinateur distant et pour la connexion à distance. **Évitez d'utiliser des mots de passe courants ou faciles à deviner**.

3. **Limitez l'accès :** Limitez l'accès à l'ordinateur distant en n'autorisant que les utilisateurs autorisés à se connecter. Utilisez des pare-feux pour bloquer les connexions non autorisées.

4. **Mettez à jour les logiciels :** Assurez vous que les logiciels utilisés pour la connexion à distance sont à jour avec les derniers correctifs de sécurité pour éviter les vulnérabilités connues.

5. **Utilisez l'authentification à deux facteurs :** Utilisez l'authentification à deux facteurs pour renforcer la sécurité de la connexion à distance. Cela peut inclure l'utilisation d'un code de vérification envoyé par SMS ou l'utilisation d'un jeton de sécurité.

6. **Surveillez les connexions :** Surveillez les connexions à distance pour détecter les activités suspectes ou les tentatives d'accès non autorisées.

7. **Chiffrez les données :** Utilisez le chiffrement pour protéger les données sensibles qui sont transférées entre l'ordinateur local et l'ordinateur distant.

En suivant ces mesures de sécurité, vous pouvez réduire les risques de violation de la sécurité lors de la connexion à distance à un ordinateur distant.

Voici une liste des principales commandes et leur utilisation

1. **ssh :** Cette commande est utilisée pour se connecter à un serveur distant via SSH. Par exemple, `ssh root@10.31.96.1` se connectera à l'ordinateur distant avec le nom d'utilisateur “user” et le nom d'hôte “hostname”.

2. **ssh-keygen :** Cette commande est utilisée pour générer une paire de clés SSH (publique et privée) pour une utilisation avec SSH. Par exemple, `ssh-keygen -t rsa` générera une paire de clés RSA.

3. **ssh-copy-id :** Cette commande est utilisée pour copier la clé publique SSH sur un serveur distant pour permettre une connexion sans mot de passe. Par exemple, `ssh-copy-id root@10.31.96.1` copiera la clé publique de l'utilisateur sur l'ordinateur distant avec le nom d'utilisateur “root” et le nom d'hôte “10.31.96.1”.

4. **ssh-add :** Cette commande est utilisée pour ajouter une clé privée SSH à l'agent SSH pour permettre une connexion sans mot de passe pendant une période spécifiée. Par exemple, `ssh-add -t 1h` ajoutera la clé privée à l'agent SSH pour une durée d'une heure.

5. **ssh-agent :** Cette commande est utilisée pour démarrer l'agent SSH, qui stocke les clés privées pour une utilisation avec SSH. Par exemple, `eval $(ssh-agent)` démarrera l'agent SSH.

6. **sshfs :** Cette commande est utilisée pour monter un système de fichiers distant via SSH. Par exemple, `sshfs user@hostname:/remote/directory /local/mount/point` montera le répertoire distant “/remote/directory” sur le point de montage local “/local/mount/point”.

7. **scp :** Cette commande est utilisée pour copier des fichiers entre des ordinateurs locaux et distants via SSH. Par exemple, `scp file.txt root@10.31.96.1:/remote/directory` copiera le fichier “file.txt” sur l'ordinateur distant avec le nom d'utilisateur “user” et le nom d'hôte “root” dans le répertoire “/remote/directory”.

Ces commandes sont utilisées pour configurer et utiliser SSH de manière sécurisée et efficace. Il est important de comprendre leur utilisation et leur syntaxe pour utiliser SSH de manière appropriée.

CONCLUSION

En conclusion, les travaux réalisés dans le cadre du Bloc 2 - SISR ont permis de mieux comprendre les enjeux de sécurité liés à la mise en place d'un service de connexion à distance SSH, ainsi que les solutions pour minimiser les risques d'attaques. Les travaux ont également permis de mettre en pratique les connaissances acquises en déployant une clé RSA ou DSA sur un serveur, en configurant le serveur SSH pour qu'il accepte et utilise l'authentification par clé RSA, et en testant le fonctionnement de l'authentification par clés en se connectant en SSH sur le serveur.

Fichier externe

comment_sauvegarde_ca_configuration.pdf
la_haute_disponibilite_de_serveurs_heartbeat.pdf
installation.pdf
authentification_ssh_et_chiffrement_asymetriques.pdf