Protection des données & Gestion des droits d'accès.

seance le 22 Mars 2022

BDOE656

Ferdinand TCHADJI,
enseignant-chercheur

SE Oracle Solaris et Sécurité des données

Services de sécurité (SE Oracle Solaris)

Pour préserver la sécurité du système d'exploitation Oracle Solaris (SE Oracle Solaris), le logiciel propose les fonctionnalités suivantes :

- Sécurité du système : la possibilité d'empêcher les intrusions, de protéger les ressources de la machine et les périphériques contre les utilisations inappropriées et de protéger les fichiers contre les modifications malveillantes ou involontaires par des utilisateurs ou des intrus.
- Services cryptographiques : capacité de brouiller les données afin que seul l'expéditeur et le destinataire désigné puissent lire le contenu, et de gérer les fournisseurs cryptographiques et les objets de clé publique
- Services d'authentification : capacité d'identifier un utilisateur de manière sécurisée, ce qui nécessite le nom de l'utilisateur et une forme quelconque de preuve, en général un mot de passe
- Authentification avec le chiffrement : capacité de s'assurer que les parties authentifiées peuvent communiquer sans interception, modification ou usurpation d'identité
- Service d'Audit : capacité d'identifier l'origine des modifications de sécurité apportées au système, y compris l'accès aux fichiers, les appels système associés à la sécurité, et les échecs d'authentification

Sécurité du système

Des contrôles d'accès permettent de limiter l'accès aux ressources du système à des utilisateurs autorisés.

Les fonctions du SE Oracle Solaris pour la sécurité du système et le contrôle de l'accès sont les suivantes :

- ✓**Outils d'administration de connexion** : commandes de surveillance et de contrôle de la capacité d'un utilisateur à se connecter.
- ✓**Hardware access** : commandes de limitation de l'accès à la PROM et de restriction des utilisateurs autorisés à démarrer le système.
- ✓**Resource access** : outils et stratégies permettant d'optimiser l'utilisation appropriée des ressources de la machine tout en minimisant l'utilisation inappropriée de ces ressources.
- ✓**Contrôle d'accès basé sur les rôles (RBAC)** : architecture permettant de créer des comptes utilisateur restreints spécifiques, qui sont autorisés à effectuer des tâches d'administration.

Sécurité du système

- ✓ **Privilèges** : droits discrets sur les processus pour effectuer des opérations. Ces droits sur les processus sont appliqués dans le noyau.
- ✓ **Gestion des périphériques** : la **stratégie** relative aux périphériques protège en outre les périphériques qui sont déjà protégés par des autorisations UNIX. L'**allocation** des périphériques contrôle l'accès aux périphériques, tels qu'un microphone ou une unité de CD-ROM. Lors de l'annulation de l'allocation, les scripts de nettoyage de périphérique peuvent ensuite effacer toutes les données du périphérique.
- ✓ **Outil de rapport d'audit de base (BART)** : instantané, appelé **manifeste**, des attributs des fichiers d'un système. En comparant les manifestes sur l'ensemble des systèmes ou sur un système dans le temps, les modifications apportées aux fichiers peuvent être surveillées pour réduire les risques de sécurité.
- ✓ **Autorisations de fichier** : attributs d'un fichier ou d'un répertoire. Les autorisations limitent les utilisateurs et les groupes qui sont autorisés à lire, écrire ou exécuter un fichier, ou effectuer une recherche dans un répertoire.
- ✓ **Scripts d'amélioration de la sécurité** : à l'aide des scripts, vous pouvez modifier de nombreux fichiers et paramètres système pour réduire les risques de sécurité.

Services d'authentification

- ***L'authentification est un mécanisme qui identifie un utilisateur ou un service en fonction de critères prédéfinis.***
- Les services d'authentification vont de simples paires nom-mot de passe à des systèmes de stimulation-réponse, tels que les cartes à jeton et la biométrie.
- Les mécanismes d'authentification forte reposent sur l'indication par un user d'un secret et sur un objet qui peut être vérifié.
- Un nom d'utilisateur est un exemple d'information que la personne connaît. Une carte à puce ou une empreinte digitale, par exemple, peut être vérifiée.

Les fonctionnalités d'Oracle Solaris pour l'authentification sont les suivantes :

- ✓ **Appel de procédure à distance sécurisé** : mécanisme d'authentification qui utilise le protocole Diffie-Hellman pour protéger les montages NFS et un service de nommage, tel que NIS ou NIS+.
- ✓ **Module d'authentification enfichable (PAM)** : une structure qui permet à diverses technologies d'authentification d'être connectées à un service d'entrée système sans recompiler le service. Certains services d'entrée système comprennent login et ftp.
- ✓ **SASL (Simple Authentication and Security Layer)** : structure qui fournit des services d'authentification et de sécurité aux protocoles de réseau. Reportez-vous au
- ✓ **Secure Shell** : protocole de connexion et de transmission à distance qui chiffre les communications sur un réseau non sécurisé. Reportez-vous au
- ✓ **Service Kerberos** : architecture client-serveur qui fournit le chiffrement avec authentification.
- ✓ **Carte à puce Solaris** Une carte en plastique dotée d'un microprocesseur et d'une mémoire qui permettent à un lecteur de carte d'accéder aux systèmes.

Service Audit : SE Oracle

L'audit consiste à collecter des données sur l'utilisation des ressources système.

Les données d'audit fournissent un enregistrement des événements ayant trait à la sécurité.

Ces données peuvent ensuite être utilisées pour déterminer la responsabilité quant aux actions survenant sur un hôte.

Un audit réussi commence par deux fonctions de sécurité : ***identification et authentification.***

➔À chaque connexion, une fois qu'un user fournit un nom et un mot de passe, ***un ID de session d'audit unique est généré et associé au processus de l'utilisateur.***

➔L'ID de session d'audit est hérité par tous les processus démarrés au cours de la session de connexion.

➔Même si un user change d'identité au cours d'une session, toutes ses actions sont suivies avec le même ID de session d'audit.

Le service d'audit effectue les opérations suivantes :

✓*Surveillance des événements liés à la sécurité survenant sur l'hôte*

✓*Enregistrement des événements dans une piste d'audit à l'échelle du réseau*

✓*Détection des utilisations inappropriées et des activités non autorisées*

✓*Examen des modèles d'accès et des historiques d'accès des individus et des objets*

✓*Identification des tentatives de contournement des mécanismes de protection*

✓*Détection de l'utilisation étendue d'un privilège survenant lorsqu'un utilisateur change d'identité*

Audit: son Fonctionnement

L'audit génère des enregistrements d'audit lorsque des événements donnés se produisent.
Le plus souvent, les événements générant des enregistrements d'audit sont les suivants :

- ✓ *Démarrage et arrêt du système*
- ✓ *Connexion et déconnexion*
- ✓ *Création ou destruction de processus et création ou destruction de threads*
- ✓ *Ouverture, fermeture, création, destruction, ou modification du nom d'objets*
- ✓ *Utilisation des capacités de privilège ou du contrôle d'accès basé sur les rôles (RBAC)*
- ✓ *Actions d'identification et d'authentification*
- ✓ *Modification d'autorisations par un processus ou un utilisateur*
- ✓ *Actions d'administration, telles que l'installation d'un package*
- ✓ *Applications spécifiques à un site*

Les enregistrements d'audit sont générés à partir de trois sources :

1. *Par une application*
2. *À la suite d'un événement d'audit asynchrone*
3. *À la suite d'un appel système de processus*

Une fois les informations de l'événement recueillies, elles sont formatées en un enregistrement d'audit. L'enregistrement est ensuite écrit dans les fichiers d'audit. Les enregistrements d'audit complets sont stockés au ***format binaire***.

Dans la version Solaris10, les enregistrements d'audit peuvent également être consignés par l'utilitaire *syslog*.

structure cryptographique Oracle Solaris

Commandes au niveau de l'utilisateur dans la structure cryptographique Oracle Solaris

La structure cryptographique Oracle Solaris fournit des commandes au niveau de l'utilisateur pour vérifier l'intégrité des fichiers et les chiffrer/déchiffrer.

- ✓ *digest* **(commande)** **:** calcule une synthèse de message (Ex.: md5.fr) d'un fichier ou plus. Une synthèse permet de vérifier l'intégrité d'un fichier. SHA1 et MD5 sont des exemples de fonctions digest.
- ✓ *mac* **(commande)** **:** calcule un code d'authentification des messages (MAC) des fichiers. Un code MAC associe des données à un message authentifié. Un MAC permet à un destinataire de vérifier que le message provient de l'expéditeur et qu'il n'a pas été altéré. Les mécanismes sha1_mac et md5_hmac peuvent calculer un MAC.
- ✓ *encrypt* **(commande)** **:** chiffre des fichiers avec un chiffrement symétrique. La commande *encrypt -l* répertorie les algorithmes disponibles. Les mécanismes répertoriés dans une bibliothèque au niveau de l'utilisateur sont disponibles pour la commande encrypt. La structure offre les mécanismes AES, DES, 3DES et ARCFOUR pour le chiffrement utilisateur.
- ✓ *decrypt* **(commande)** **:** déchiffre des fichiers qui ont été chiffrés avec la commande *encrypt*. La commande *decrypt* utilise les mêmes clés et même mécanisme que ceux utilisés pour chiffrer le fichier d'origine.

Calculez la synthèse du fichier et enregistrez la liste des synthèses.

Fournissez un algorithme avec la commande `digest`.

`% digest -v -a` *algorithm input-file* `>` *digest-listing*

`-v`	Affiche la sortie au format suivant : *algorithm* (*input-file*) = *digest*
`-a` *algorithm*	Algorithme à utiliser pour calculer une synthèse du fichier. Saisissez l'algorithme lorsqu'il s'affiche dans la sortie de l'Étape 1.
input-file	Fichier d'entrée pour la commande `digest`.
digest-listing	Fichier de sortie pour la commande `digest`.

Créez un MAC pour un fichier.

Fournissez une clé et utilisez un algorithme de clé symétrique avec la commande `mac`.

`% mac [-v] -a` *algorithm* `[-k` *keyfile* `| -K` *key-label* `[-T` *token*`]]` *input-file*

`-v`	Affiche la sortie au format suivant : *algorithm* (*input-file*) = *mac*
`-a` *algorithm*	Algorithme à utiliser pour calculer le code MAC. Saisissez l'algorithme lorsqu'il s'affiche dans la sortie de la commande `mac -l`.
`-k` *keyfile*	Fichier contenant une clé de longueur spécifiée par algorithme.
`-K` *key-label*	Est l'étiquette d'une clé dans le keystore PKCS #11.
`-T` *token*	Est le nom du jeton. Par défaut, le jeton est `Sun Software PKCS#11 softtoken`. Est utilisé uniquement lorsque l'option `-K` *key-label* est utilisée.
input-file	Fichier d'entrée pour le MAC.

Dans l'exemple suivant, la commande digest utilise le mécanisme MD5 pour calculer la synthèse pour une pièce jointe d'un e-mail.

% **digest -v -a md5 email.attach >> $HOME/digest.emails.05.07**

% **cat ~/digest.emails.05.07**

md5 (email.attach) = 85c0a53d1a5cc71ea34d9ee7b1b28b01

Lorsque l'option -v n'est pas utilisée, la synthèse est enregistrée sans informations complémentaires :

% **digest -a md5 email.attach >> $HOME/digest.emails.05.07**

% **cat ~/digest.emails.05.07**

85c0a53d1a5cc71ea34d9ee7b1b28b01

Calcul d'un MAC avec DES_MAC et une phrase de passe

- Dans l'exemple suivant, la pièce jointe d'e-mail est authentifiée avec le mécanisme DES_MAC et une clé dérivée d'une phrase de passe. La liste MAC est enregistrée dans un fichier. Si la phrase de passe est stockée dans un fichier, celui-ci doit être lisible uniquement par l'utilisateur.
- % **mac -v -a des_mac email.attach**
  Enter passphrase: *<Typepassphrase>*
  des_mac (email.attach) = dd27870a
  % **echo "des_mac (email.attach) = dd27870a" >> ~/desmac.daily.05.07**

Calcul d'un MAC avec MD5_HMAC et un fichier de clés

- Dans l'exemple suivant, la pièce jointe d'e-mail est authentifiée avec le mécanisme MD5_HMAC et une clé secrète. La liste MAC est enregistrée dans un fichier.
- % **mac -v -a md5_hmac -k $HOME/keyf/05.07.mack64 email.attach**
  md5_hmac (email.attach) = 02df6eb6c123ff25d78877eb1d55710c
- % **echo "md5_hmac (email.attach) = 02df6eb6c123ff25d78877eb1d55710c" \ >> ~/mac.daily.05.07**

Chiffrer et déchiffrer

Chiffrez un fichier.

Fournissez une clé et utilisez un algorithme de clé symétrique avec la commande `encrypt`.

```
% encrypt -a algorithm [-v] \
[-k keyfile | -K key-label [-T token]] [-i input-file] [-o output-file]
```

`-a` *algorithm*	Algorithme à utiliser pour chiffrer le fichier. Saisissez l'algorithme lorsqu'il s'affiche dans la sortie de la commande `encrypt -l`.
`-k` *keyfile*	Fichier contenant une clé de longueur spécifiée par algorithme. La longueur de la clé pour chaque algorithme est répertoriée, en bits, dans la sortie de la commande `encrypt -l`.
`-K` *key-label*	Etiquette d'une clé dans le keystore PKCS #11.
`-T` *token*	Nom du jeton. Par défaut, le jeton est `Sun Software PKCS#11 softtoken`. Est utilisé uniquement lorsque l'option `-K` *key-label* est utilisée.
`-i` *input-file*	Fichier d'entrée que vous voulez chiffrer. Ce fichier n'est pas modifié par la commande.
`-o` *output-file*	Fichier de sortie correspondant à la forme chiffrée du fichier d'entrée.

Création d'une clé AES pour le chiffrement de vos fichiers

- Dans l'exemple suivant, un utilisateur crée et stocke une clé AES dans keystore PKCS #11 existant pour l'utilisation lors du chiffrement et du déchiffrement. L'utilisateur peut vérifier que la clé existe et l'utiliser, mais il ne peut pas l'afficher

% **pktool genkey label=MyAESkeynumber1 keytype=aes keylen=256**

Enter PIN for Sun Software PKCS#11 softtoken :Type password

% **pktool list objtype=key**

Enter PIN for Sun Software PKCS#11 softtoken :<*Type password*> Found 1 key

Key #1 - Sun Software PKCS#11 softtoken: MyAESkeynumber1 (256)

Chiffrer et déchiffrer

Pour utiliser la clé pour chiffrer un fichier, l'utilisateur le récupère la clé par son étiquette.

% **encrypt -a aes -K MyAESkeynumber1 -i encryptthisfile -o encryptedthisfile**

- Pour déchiffrer le encryptedthisfile, l'utilisateur récupère la clé par son étiquette

% **decrypt -a aes -K MyAESkeynumber1 -i encryptedthisfile -o sameasencryptthisfile**

Chiffrement et déchiffrement avec 3DES et un fichier de clés

- Dans l'exemple suivant, un fichier est chiffré avec l'algorithme 3DES. L'algorithme 3DES requiert une clé de 192 bits, ou 24 octets.
- **% encrypt -a 3des -k ~/keyf/05.07.des24 \ -i ~/personal2.txt -o ~/enc/e.personal2.txt**
- Pour déchiffrer le fichier de sortie, l'utilisateur utilise la même clé et le même mécanisme de chiffrement que ceux utilisés pour le chiffrement.
- Commande?
- **% decrypt -a 3des -k ~/keyf/05.07.des24 \ -i ~/enc/e.personal2.txt -o ~/personal2.txt**