scriting

Astier Guillaume

08/01/2026

Disques Durs en **Read-Only** à la Fin du Démarrage

Lire une recommendation CIS

Découpage du guide CIS

CIS initial setup

pem-cyber-linux.sh en service

Modification de la Bannière pour Indiquer le Niveau de Sécurisation

CIS services

CIS_Network

Mettre à jour la bannière en fonction du script de sécurité

Disques Durs en **Read-Only** à la Fin du Démarrage

Introduction

Dans le cadre de la sécurité informatique, il existe des mécanismes qui permettent de sécuriser l'intégrité des systèmes d'exploitation en limitant les modifications sur certains composants du système, notamment le disque dur. Un des mécanismes utilisés est de mettre en place un mode **read-only** (lecture seule) pour le disque dur ou une partition spécifique à la fin du processus de démarrage du système. Cette stratégie est couramment utilisée pour protéger les systèmes contre les attaques potentielles, les malwares et les manipulations non autorisées des fichiers système.

1. Qu'est-ce que le mode "Read-Only" ?

Un disque dur ou une partition **read-only** signifie que les données présentes sur ce disque ne peuvent pas être modifiées, ajoutées ou supprimées pendant le fonctionnement du système. Cela permet de préserver l'intégrité des fichiers importants, surtout ceux qui sont nécessaires au démarrage du système d'exploitation.

Le mode **read-only** peut être appliqué à différents niveaux, en fonction des besoins de sécurité :

- **Partition système** : Par exemple, une partition contenant le noyau du système d'exploitation.

1. Qu'est-ce que le mode "Read-Only" ?

Un disque dur ou une partition **read-only** signifie que les données présentes sur ce disque ne peuvent pas être modifiées, ajoutées ou supprimées pendant le fonctionnement du système. Cela permet de préserver l'intégrité des fichiers importants, surtout ceux qui sont nécessaires au démarrage du système d'exploitation.

Le mode **read-only** peut être appliqué à différents niveaux, en fonction des besoins de sécurité :

- **Partition système** : Par exemple, une partition contenant le noyau du système d'exploitation.
- **Disque entier** : Une protection complète de tous les fichiers du disque.

2. Pourquoi utiliser le mode “Read-Only” à la fin du démarrage ?

Le but principal d’un disque dur en mode **read-only** à la fin du démarrage est de rendre le système plus résistant aux attaques et manipulations malveillantes. Voici quelques raisons pour lesquelles cette technique est mise en œuvre :

- **Protection contre les malwares** : Beaucoup de malwares nécessitent des privilèges d'écriture sur le disque pour se propager et effectuer des modifications malveillantes. En verrouillant le disque en lecture seule après le démarrage, on empêche l'installation de ces malwares.

- **Protection contre les malwares** : Beaucoup de malwares nécessitent des privilèges d'écriture sur le disque pour se propager et effectuer des modifications malveillantes. En verrouillant le disque en lecture seule après le démarrage, on empêche l'installation de ces malwares.
- **Prévention contre les attaques Rootkit** : Les rootkits visent à masquer leur présence en modifiant des fichiers systèmes ou en injectant du code dans des processus légitimes. Un disque en **read-only** empêche toute modification de ces fichiers, rendant l'infection plus difficile.

- **Protection contre les malwares** : Beaucoup de malwares nécessitent des privilèges d'écriture sur le disque pour se propager et effectuer des modifications malveillantes. En verrouillant le disque en lecture seule après le démarrage, on empêche l'installation de ces malwares.
- **Prévention contre les attaques Rootkit** : Les rootkits visent à masquer leur présence en modifiant des fichiers systèmes ou en injectant du code dans des processus légitimes. Un disque en **read-only** empêche toute modification de ces fichiers, rendant l'infection plus difficile.
- **Maintien de l'intégrité du système** : Lors d'attaques physiques ou logiques sur les systèmes (ex : accès physique non autorisé), le mode **read-only** garantit que les fichiers essentiels ne sont pas altérés, ce qui permet de maintenir un environnement stable.

- **Protection contre les malwares** : Beaucoup de malwares nécessitent des privilèges d'écriture sur le disque pour se propager et effectuer des modifications malveillantes. En verrouillant le disque en lecture seule après le démarrage, on empêche l'installation de ces malwares.
- **Prévention contre les attaques Rootkit** : Les rootkits visent à masquer leur présence en modifiant des fichiers systèmes ou en injectant du code dans des processus légitimes. Un disque en **read-only** empêche toute modification de ces fichiers, rendant l'infection plus difficile.
- **Maintien de l'intégrité du système** : Lors d'attaques physiques ou logiques sur les systèmes (ex : accès physique non autorisé), le mode **read-only** garantit que les fichiers essentiels ne sont pas altérés, ce qui permet de maintenir un environnement stable.
- **Sécurisation des partitions sensibles** : Certaines partitions peuvent contenir des données ou des configurations sensibles qui ne doivent en aucun cas être modifiées. En les mettant en **read-only**, on les protège de toute altération.

3. Mise en place d'un disque dur en mode **Read-Only**

Voici les étapes générales pour configurer un disque dur ou une partition en mode **read-only** à la fin du démarrage :

a. **Utilisation de Linux avec SELinux ou AppArmor**

Sur un système Linux, la configuration de la partition en mode **read-only** peut être réalisée en utilisant des outils comme SELinux (Security-Enhanced Linux) ou AppArmor, qui permettent de définir des politiques de sécurité avancées pour contrôler l'accès au système de fichiers.

1. **Édition du fichier fstab** :

3. Mise en place d'un disque dur en mode **Read-Only**

Voici les étapes générales pour configurer un disque dur ou une partition en mode **read-only** à la fin du démarrage :

a. **Utilisation de Linux avec SELinux ou AppArmor**

Sur un système Linux, la configuration de la partition en mode **read-only** peut être réalisée en utilisant des outils comme SELinux (Security-Enhanced Linux) ou AppArmor, qui permettent de définir des politiques de sécurité avancées pour contrôler l'accès au système de fichiers.

1. **Édition du fichier fstab** :
   - Le fichier **/etc/fstab** définit les points de montage des partitions. Pour mettre une partition en **read-only**, il suffit de définir l'option de montage en `ro` (read-only).

```
UUID=<UUID-du-disque> /  ext4  defaults,ro  0  1
```

3. Mise en place d'un disque dur en mode **Read-Only**

Voici les étapes générales pour configurer un disque dur ou une partition en mode **read-only** à la fin du démarrage :

a. **Utilisation de Linux avec SELinux ou AppArmor**

Sur un système Linux, la configuration de la partition en mode **read-only** peut être réalisée en utilisant des outils comme SELinux (Security-Enhanced Linux) ou AppArmor, qui permettent de définir des politiques de sécurité avancées pour contrôler l'accès au système de fichiers.

1. **Édition du fichier fstab** :
   - Le fichier **/etc/fstab** définit les points de montage des partitions. Pour mettre une partition en **read-only**, il suffit de définir l'option de montage en `ro` (read-only).

```
UUID=<UUID-du-disque> /  ext4  defaults,ro  0  1
```

2. **Activation de SELinux/AppArmor** :

3. Mise en place d'un disque dur en mode **Read-Only**

Voici les étapes générales pour configurer un disque dur ou une partition en mode **read-only** à la fin du démarrage :

a. **Utilisation de Linux avec SELinux ou AppArmor**

Sur un système Linux, la configuration de la partition en mode **read-only** peut être réalisée en utilisant des outils comme SELinux (Security-Enhanced Linux) ou AppArmor, qui permettent de définir des politiques de sécurité avancées pour contrôler l'accès au système de fichiers.

1. **Édition du fichier fstab** :
   - Le fichier **/etc/fstab** définit les points de montage des partitions. Pour mettre une partition en **read-only**, il suffit de définir l'option de montage en `ro` (read-only).

```
UUID=<UUID-du-disque> /  ext4  defaults,ro  0  1
```

2. **Activation de SELinux/AppArmor** :
   - Ces outils permettent de définir des règles strictes pour l'accès aux fichiers et peuvent être utilisés pour interdire toute modification sur les partitions en mode **read-only**.

Lire une recommendation CIS

Lire une recommendation CIS

1.1.1.1 Ensure cramfs kernel module is not available (Automated)

Profile Applicability:

- Level 1 - Server

Description:

The cramfs filesystem type is a compressed read-only Linux filesystem embedded in small footprint systems. A cramfs image can be used without having to first decompress the image.

Rationale:

Removing support for unneeded filesystem types reduces the local attack surface of the system. If this filesystem type is not needed, disable it.

Lire une recommendation CIS

1.1.1.1 Ensure cramfs kernel module is not available (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The cramfs filesystem type is a compressed read-only Linux filesystem embedded in small footprint systems. A cramfs image can be used without having to first decompress the image.

Rationale:

Removing support for unneeded filesystem types reduces the local attack surface of the system. If this filesystem type is not needed, disable it.

Audit:

Run the following script to verify the cramfs module is disabled:

IF the module is available in the running kernel:

- An entry including /bin/true or /bin/false exists in a file within the /etc/modprobe.d/ directory

IF available in ANY installed kernel:

IF the kernel module is not available on the system, or pre-compiled into the kernel: - No additional configuration is necessary

Audit:

Run the following script to verify the cramfs module is disabled:

IF the module is available in the running kernel:

- An entry including /bin/true or /bin/false exists in a file within the /etc/modprobe.d/ directory
- The module is deny listed in a file within the /etc/modprobe.d/ directory

IF available in ANY installed kernel:

IF the kernel module is not available on the system, or pre-compiled into the kernel: - No additional configuration is necessary

Audit:

Run the following script to verify the cramfs module is disabled:

IF the module is available in the running kernel:

- An entry including /bin/true or /bin/false exists in a file within the /etc/modprobe.d/ directory
- The module is deny listed in a file within the /etc/modprobe.d/ directory
- The module is not loaded in the kernel

IF available in ANY installed kernel:

IF the kernel module is not available on the system, or pre-compiled into the kernel: - No additional configuration is necessary

Audit:

Run the following script to verify the cramfs module is disabled:

IF the module is available in the running kernel:

- An entry including /bin/true or /bin/false exists in a file within the /etc/modprobe.d/ directory
- The module is deny listed in a file within the /etc/modprobe.d/ directory
- The module is not loaded in the kernel

IF available in ANY installed kernel:

- The module is deny listed in a file within the /etc/modprobe.d/ directory

IF the kernel module is not available on the system, or pre-compiled into the kernel: - No additional configuration is necessary

```
#!/usr/bin/env bash

{
   l_output="" l_output2="" l_output3="" l_dl="" # Unset output variables
   l_mname="cramfs" # set module name
   l_mtype="fs" # set module type
   l_searchloc="/lib/modprobe.d/*.conf /usr/local/lib/modprobe.d/*.conf /run/modprobe.d/*.conf
/etc/modprobe.d/*.conf"
   l_mpath="/lib/modules/**/kernel/$l_mtype"
   l_mpname="$(tr '-' '_' <<< "$l_mname")"
   l_mndir="$(tr '-' '/' <<< "$l_mname")"

   module_loadable_chk()
   {
      # Check if the module is currently loadable
      l_loadable="$(modprobe -n -v "$l_mname")"
      [ "$(wc -l <<< "$l_loadable")" -gt "1" ] && l_loadable="$(grep -P --
"(^\h*install|\b$l_mname)\b" <<< "$l_loadable")"
      if grep -Pq -- '^\h*install \/bin\/(true|false)' <<< "$l_loadable"; then
         l_output="$l_output\n - module: \"$l_mname\" is not loadable: \"$l_loadable\""
      else
         l_output2="$l_output2\n - module: \"$l_mname\" is loadable: \"$l_loadable\""
      fi
   }
   module_loaded_chk()
   {
      # Check if the module is currently loaded
      if ! lsmod | grep "$l_mname" > /dev/null 2>&1; then
         l_output="$l_output\n - module: \"$l_mname\" is not loaded"
      else
         l_output2="$l_output2\n - module: \"$l_mname\" is loaded"
      fi
   }
   module_deny_chk()
   {
      # Check if the module is deny listed
      l_dl="y"
      if modprobe --showconfig | grep -Pq -- '^\h*blacklist\h+'"$l_mpname"'\b'; then
         l_output="$l_output\n - module: \"$l_mname\" is deny listed in: \"$(grep -Pls --
"^\h*blacklist\h+$l_mname\b" $l_searchloc)\""
      else
         l_output2="$l_output2\n - module: \"$l_mname\" is not deny listed"
      fi
   }
   # Check if the module exists on the system
   for l_mdir in $l_mpath; do
      if [ -d "$l_mdir/$l_mndir" ] && [ -n "$(ls -A $l_mdir/$l_mndir)" ]; then
         l_output3="$l_output3\n  - \"$l_mdir\""
         [ "$l_dl" != "y" ] && module_deny_chk
         if [ "$l_mdir" = "/lib/modules/$(uname -r)/kernel/$l_mtype" ]; then
            module_loadable_chk
            module_loaded_chk
         fi
      else
         l_output="$l_output\n - module: \"$l_mname\" doesn't exist in \"$l_mdir\""
```

Remediation:

Run the following script to disable the cramfs module:

IF the module is available in the running kernel:

- Create a file ending in .conf with install cramfs /bin/false in the /etc/modprobe.d/ directory

IF available in ANY installed kernel:

IF the kernel module is not available on the system or pre-compiled into the kernel:

Remediation:

Run the following script to disable the cramfs module:

IF the module is available in the running kernel:

- Create a file ending in .conf with install cramfs /bin/false in the /etc/modprobe.d/ directory
- Create a file ending in .conf with blacklist cramfs in the /etc/modprobe.d/ directory

IF available in ANY installed kernel:

IF the kernel module is not available on the system or pre-compiled into the kernel:

Remediation:

Run the following script to disable the cramfs module:

IF the module is available in the running kernel:

- Create a file ending in .conf with install cramfs /bin/false in the /etc/modprobe.d/ directory
- Create a file ending in .conf with blacklist cramfs in the /etc/modprobe.d/ directory
- Unload cramfs from the kernel

IF available in ANY installed kernel:

IF the kernel module is not available on the system or pre-compiled into the kernel:

Remediation:

Run the following script to disable the cramfs module:

IF the module is available in the running kernel:

- Create a file ending in .conf with install cramfs /bin/false in the /etc/modprobe.d/ directory
- Create a file ending in .conf with blacklist cramfs in the /etc/modprobe.d/ directory
- Unload cramfs from the kernel

IF available in ANY installed kernel:

- Create a file ending in .conf with blacklist cramfs in the /etc/modprobe.d/ directory

IF the kernel module is not available on the system or pre-compiled into the kernel:

Remediation:

Run the following script to disable the cramfs module:

IF the module is available in the running kernel:

- Create a file ending in .conf with install cramfs /bin/false in the /etc/modprobe.d/ directory
- Create a file ending in .conf with blacklist cramfs in the /etc/modprobe.d/ directory
- Unload cramfs from the kernel

IF available in ANY installed kernel:

- Create a file ending in .conf with blacklist cramfs in the /etc/modprobe.d/ directory

IF the kernel module is not available on the system or pre-compiled into the kernel:

- No remediation is necessary

Découpage du guide CIS

Découpage du guide CIS

Voici comment se présente le découpage du guide CIS dans ses “grandes lignes”

Initial Setup

Module Kernel

Désactivation des modules non nécessaire pour le kernel

Partitions

Vérifier que les partitions sont correctement configurées

Update

Vérification de la gestion des mise à jours

Grub

S'assurer que le grub est sécurisé par mot de passe

Gestion de la RAM

Vérifier que la configuration ne permette pas d'attaque en mémoire

Selinux

Verifier que selinux est installé et configuré

Banner

Verifier que la banière est bien configuré (local et réseau)

Desktop

Verifier que gnome est configuré (banière, gestion du login ...)

Service

NTP

Vérifier que chrony (client/server de temps) est configuré

Service réseau

Vérifier que la présence de certains services réseau est sécurisé et/ou désactivé (ftp, samba ...)

Client réseau

Vérifier que la présence de certains client réseau est sécurisé et/ou désactivé (ftp, ldap ...)

Network

Devices

Verification de la présence légitime de certaines interfaces (IPV6, wifi ...)

Kernel modules

Supprimer les modules kernels indésirables (dccp, rds ...)

Network kernel parameters

Verifier que certaines fonctionnalités du kernel d'un point de vue réseau soient désactivé

Firewall

Vérifier qu'un seul firewall soit bien présent et configuré

Access, Authentication and Authorization

Cron / at

Vérifier que le service de planification Cron soit présent et sécurisé contrairement à at

ssh

Vérifier que ssh est configuré correctement

sudo

Vérifier que la configuration de sudo soit faite de manière “intelligente”

PAM

Vérifier que la configuration de PAM (authentification) soit faite correctement.

User account

Vérifier les droits et la gestion des utilisateurs client de l’instance

Logging and Auditing

Logging

Vérifier la présence et le bon fonctionnement dans un environnement sécurisé d'un service de gestion de logs

Auditd

Auditd permet le logging des activités de l'instance. Il doit être installé et configuré

Aide

Aide est un outils permettant de vérifier l'intégrité d'un système GNU/Linux. Il doit être installé et configuré

System Maintenance

File system permission

Vérifier que certains fichiers aient les bons droit UNIX

Local User and Group

Vérifier la cohérence des fichiers permettant la gestion des utilisateurs ainsi que la cohérence des UID/GID

CIS initial setup

Désactiver les modules

- Ensure cramfs kernel module is not available

Désactiver les modules

- Ensure cramfs kernel module is not available
- Ensure freevxfs kernel module is not available

Désactiver les modules

- Ensure cramfs kernel module is not available
- Ensure freevxfs kernel module is not available
- Ensure hfs kernel module is not available

Désactiver les modules

- Ensure cramfs kernel module is not available
- Ensure freevxfs kernel module is not available
- Ensure hfs kernel module is not available
- Ensure hfsplus kernel module is not available

Désactiver les modules

- Ensure cramfs kernel module is not available
- Ensure freevxfs kernel module is not available
- Ensure hfs kernel module is not available
- Ensure hfsplus kernel module is not available
- Ensure jffs2 kernel module is not available

Désactiver les modules

- Ensure cramfs kernel module is not available
- Ensure freevxfs kernel module is not available
- Ensure hfs kernel module is not available
- Ensure hfsplus kernel module is not available
- Ensure jffs2 kernel module is not available
- Ensure squashfs kernel module is not available

Désactiver les modules

- Ensure cramfs kernel module is not available
- Ensure freevxfs kernel module is not available
- Ensure hfs kernel module is not available
- Ensure hfsplus kernel module is not available
- Ensure jffs2 kernel module is not available
- Ensure squashfs kernel module is not available
- Ensure udf kernel module is not available

Désactiver les modules

- Ensure cramfs kernel module is not available
- Ensure freevxfs kernel module is not available
- Ensure hfs kernel module is not available
- Ensure hfsplus kernel module is not available
- Ensure jffs2 kernel module is not available
- Ensure squashfs kernel module is not available
- Ensure udf kernel module is not available
- Ensure usb-storage kernel module is not available

function_Apply

Créer une fonction permettant automatiquement de générer un fichier dans /etc/modprobe.d/

function_Check

Créer une fonction permettant automatiquement de vérifier un fichier dans /etc/modprobe.d/

Filesystem Partition

- Configure /tmp

function_Apply

Créer une fonction permettant de modifier le fichier /etc/fstab afin de créer des partition en ramfs

function_Check

Créer une fonction permettant de vérifier la présence d'une chaine de caractère spécifique dans un fichier et du retour "non vide" lors de la requête

Filesystem Partition

- Configure /tmp
- Configure /dev/shm

function_Apply

Créer une fonction permettant de modifier le fichier /etc/fstab afin de créer des partition en ramfs

function_Check

Créer une fonction permettant de vérifier la présence d'une chaine de caractère spécifique dans un fichier et du retour "non vide" lors de la requête

Banner

- Ensure message of the day is configured properly

function_Apply

... à vous de jouer

function_Check

... à vous de jouer

Banner

- Ensure message of the day is configured properly
- Ensure local login warning banner is configured properly

function_Apply

... à vous de jouer

function_Check

... à vous de jouer

Banner

- Ensure message of the day is configured properly
- Ensure local login warning banner is configured properly
- Ensure remote login warning banner is configured properly

function_Apply

... à vous de jouer

function_Check

... à vous de jouer

Banner

- Ensure message of the day is configured properly
- Ensure local login warning banner is configured properly
- Ensure remote login warning banner is configured properly
- Ensure access to /etc/motd is configured

function_Apply

... à vous de jouer

function_Check

... à vous de jouer

Banner

- Ensure message of the day is configured properly
- Ensure local login warning banner is configured properly
- Ensure remote login warning banner is configured properly
- Ensure access to /etc/motd is configured
- Ensure access to /etc/issue is configured

function_Apply

... à vous de jouer

function_Check

... à vous de jouer

Banner

- Ensure message of the day is configured properly
- Ensure local login warning banner is configured properly
- Ensure remote login warning banner is configured properly
- Ensure access to /etc/motd is configured
- Ensure access to /etc/issue is configured
- Ensure access to /etc/issue.net is configured

function_Apply

... à vous de jouer

function_Check

... à vous de jouer

Instruction supplémentaire

Il n'est pas nécessaire de créer systématiquement une fonction pour chaque chapitre. Il est soit :

- possible de mutualiser (restons générique)

Prenez le temps de comprendre les indications du guide CIS.

Si vous ne comprenez pas vous ne serez pas comment appliquer et/ou tester l'application d'une règle.

Instruction supplémentaire

Il n'est pas nécessaire de créer systématiquement une fonction pour chaque chapitre. Il est soit :

- possible de mutualiser (restons générique)
- utiliser directement dans les run.sh des command UNIX "classique"

Prenez le temps de comprendre les indications du guide CIS.

Si vous ne comprenez pas vous ne serez pas comment appliquer et/ou tester l'application d'une règle.

pem-cyber-linux.sh en service

Introduction

Dans cet exercice, nous allons créer un service systemd sur un système CentOS 7 afin d'exécuter le script `pem-cyber-linux.sh` à la fin du démarrage du système. Ce script est utilisé pour vérifier que les règles de cybersécurité du guide CIS (Center for Internet Security) sont bien appliquées sur le serveur. Le script prend les arguments `-c -l 1 -t server`.

Étape 1 : Créer un fichier de service systemd

Systemd est le système d'init utilisé dans CentOS 7 pour gérer les services et les processus du système. Nous allons créer un fichier de service pour exécuter le script au démarrage.

1.1 Créer le fichier de service systemd

1. Ouvrez un terminal et connectez-vous en tant que root ou utilisez `sudo` pour les commandes suivantes.

```
sudo vim /etc/systemd/system/pem-cyber-linux.service
```

1.1 Créer le fichier de service systemd

1. Ouvrez un terminal et connectez-vous en tant que root ou utilisez `sudo` pour les commandes suivantes.
2. Créez un fichier de service systemd dans le répertoire `/etc/systemd/system/` :

```
sudo vim /etc/systemd/system/pem-cyber-linux.service
```

1.2 Contenu du fichier de service

Ajoutez le contenu suivant dans le fichier :

```
[Unit]
Description=Vérification des règles CIS Cybersecurity
After=network.target

[Service]
ExecStart=/bin/bash /usr/local/bin/pem-cyber-linux.sh -c -l 1 -t server
Restart=on-failure
User=root
Environment=PATH=/usr/local/bin:/usr/bin:/bin

[Install]
WantedBy=multi-user.target
```

** Explication des options : **

- [Unit] : Cette section définit des métadonnées et des dépendances pour le service. After=network.target assure que le service démarre après la mise en réseau.
- Service : Cette section définit la commande à exécuter pour démarrer le service. Nous utilisons /bin/bash pour exécuter le script pem-cyber-linux.sh avec les arguments -c -l 1 -t server. Restart=on-failure indique que le service redémarrera en cas d'échec. Le service est exécuté en tant qu'utilisateur root car il peut nécessiter des privilèges élevés pour effectuer des vérifications.

** Explication des options : **

- [Unit] : Cette section définit des métadonnées et des dépendances pour le service. After=network.target assure que le service démarre après la mise en réseau.
- Service : Cette section définit la commande à exécuter pour démarrer le service. Nous utilisons /bin/bash pour exécuter le script pem-cyber-linux.sh avec les arguments -c -l 1 -t server. Restart=on-failure indique que le service redémarrera en cas d'échec. Le service est exécuté en tant qu'utilisateur root car il peut nécessiter des privilèges élevés pour effectuer des vérifications.
- [Install] : Cette section indique quand le service doit être démarré. multi-user.target est un niveau de fonctionnement qui représente un système multi-utilisateurs standard (ce qui correspond à un démarrage normal de CentOS 7).

Étape 2 : Recharger systemd et activer le service

Rechargez la configuration de systemd pour prendre en compte le nouveau service :

```
sudo systemctl daemon-reload
```

Activez le service pour qu'il se lance automatiquement à chaque démarrage du système :

```
sudo systemctl enable pem-cyber-linux.service
```

Vérifiez que le service est bien activé :

```
sudo systemctl is-enabled pem-cyber-linux.service
```

Cela devrait renvoyer enabled, confirmant que le service sera lancé au démarrage.

Étape 3 : Démarrer et tester le service

```
Démarrez immédiatement le service pour tester qu'il fonctionne correctement :
```

```
sudo systemctl start pem-cyber-linux.service
```

Vérifiez l'état du service pour vous assurer qu'il fonctionne comme prévu :

```
sudo systemctl status pem-cyber-linux.service
```

Vous devriez voir quelque chose comme :

```
- pem-cyber-linux.service - Vérification des règles CIS Cybersecurity
   Loaded: loaded (/etc/systemd/system/pem-cyber-linux.service; enabled; vendor
      preset: disabled)
   Active: active (exited) since Sat 2025-02-10 10:00:00 UTC; 10s ago
```

Pour consulter les logs du service et voir les sorties du script, utilisez :

```
sudo journalctl -u pem-cyber-linux.service
```

Étape 4 : Redémarrer le système pour tester le démarrage automatique

```
Redémarrez le système pour vérifier que le service est exécuté à la fin du
    démarrage :
```

```
sudo reboot
```

Après le redémarrage, vérifiez que le service a bien été lancé automatiquement :

```
    sudo systemctl status pem-cyber-linux.service
```

```
Vous devriez voir que le service est actif et qu'il a été exécuté avec succès.
```

Conclusion

Nous avons maintenant un service systemd sur CentOS 7 qui exécute le script pem-cyber-linux.sh à la fin du démarrage, avec les arguments nécessaires pour vérifier l'application des règles cybersécurité du guide CIS. Cette méthode permet de s'assurer que les configurations de sécurité sont vérifiées à chaque démarrage du système, ce qui renforce la sécurité de votre serveur.

Modification de la Bannière pour Indiquer le Niveau de Sécurisation

Modification de la Bannière pour Indiquer le Niveau de Sécurisation

Dans un environnement sécurisé, il est souvent utile de personnaliser la bannière de connexion qui apparaît au moment où un utilisateur se connecte au système. Cette bannière peut fournir des informations importantes, comme un message de bienvenue ou un avertissement sur le niveau de sécurisation du système. Dans ce chapitre, nous allons modifier la bannière de connexion pour afficher le niveau de sécurisation appliqué au serveur.

Objectifs

- Personnaliser la bannière de connexion pour afficher le niveau de sécurisation du système.

Objectifs

- Personnaliser la bannière de connexion pour afficher le niveau de sécurisation du système.
- Utiliser un script pour mettre à jour automatiquement cette bannière en fonction des résultats du script `pem-cyber-linux.sh`.

Personnalisation de la Bannière

La bannière de connexion sur CentOS 7 peut être configurée en modifiant le fichier **/etc/issue** ou **/etc/issue.net**. Ces fichiers sont utilisés pour afficher des messages avant la demande de connexion.

Modifier le fichier `/etc/issue` et /etc/issue.net

Ouvrez le fichier **/etc/issue** dans un éditeur de texte avec des privilèges administratifs :

```
sudo vim  /etc/issue /etc/issue.net
```

Vous pouvez personnaliser ce message pour refléter l'état actuel du serveur ou du système, par exemple :

```
Niveau de sécurisation : Basé sur CIS Level 1 (Serveur)
```

Mettre à jour la bannière en fonction du script de sécurité

Il est possible de rendre cette bannière dynamique, c'est-à-dire qu'elle reflète le niveau de sécurisation actuel du système en fonction des résultats du script pem-cyber-linux.sh. Pour cela, nous allons modifier le script pour qu'il mette à jour la bannière de manière automatique.

Modifier le script pem-cyber-linux.sh

Vous pouvez ajouter une partie au script pem-cyber-linux.sh pour qu'il mette à jour la bannière en fonction des résultats de l'analyse des règles CIS.

Ouvrez le script pem-cyber-linux.sh dans un éditeur de texte :

```
sudo vim  /usr/local/bin/pem-cyber-linux.sh
```

À la fin du script, après avoir effectué les vérifications de sécurité, ajoutez un bloc de code qui met à jour la bannière de sécurité en fonction du niveau de conformité des règles CIS. Par exemple :

```
# Vérification du niveau de conformité
if [ "$conformite" == "true" ]; then
    echo "Niveau de sécurisation : Conforme aux règles CIS" > /etc/issue
    echo "Niveau de sécurisation : Conforme aux règles CIS" > /etc/issue.net
else
    echo "Niveau de sécurisation : Non conforme aux règles CIS" > /etc/issue
    echo "Niveau de sécurisation : Non conforme aux règles CIS" > /etc/issue.net
fi
```

Ce code met à jour les fichiers /etc/issue et /etc/issue.net avec un message en fonction de l'état de conformité du système aux règles CIS.

Vous devrez définir la variable $conformite dans votre script en fonction des résultats des vérifications effectuées.

Pour tester la bannière locale, ouvrez un terminal et vérifiez la bannière avec la commande suivante :

```
cat /etc/issue
```

Pour tester la bannière SSH, déconnectez-vous et reconnectez-vous en SSH, puis vérifiez la bannière qui s'affiche avant l'authentification.

```
ssh localhost
```

Conclusion

En personnalisant la bannière de connexion de votre serveur CentOS 7, vous pouvez afficher un message informatif sur le niveau de sécurisation du système. Cela peut être très utile dans un environnement de production pour rappeler aux utilisateurs et aux administrateurs que des contrôles de sécurité rigoureux sont appliqués sur le système. La mise à jour automatique de cette bannière via le script pem-cyber-linux.sh garantit que l'état de conformité aux règles CIS est toujours reflété de manière dynamique.

CIS services

CIS services

Bien que l'application de mises à jour et de correctifs système permette de corriger les vulnérabilités connues, l'un des meilleurs moyens de protéger le système contre les vulnérabilités non encore signalées est de désactiver tous les services qui ne sont pas nécessaires au fonctionnement normal du système. Cela empêche l'exploitation des vulnérabilités découvertes ultérieurement. Si un service n'est pas activé, il ne peut pas être exploité. Les actions décrites dans cette section du document fournissent des conseils sur certains services qui peuvent être désactivés en toute sécurité et dans quelles circonstances, réduisant ainsi considérablement le nombre de menaces potentielles pour le système résultant. De plus, certains services qui doivent rester activés mais avec une configuration sécurisée sont couverts ainsi que les clients de services non sécurisés.

Services et binaires à activer et/ou supprimer

- Vérifier que l'instance a le paquet chrony

Services et binaires à activer et/ou supprimer

- Vérifier que l'instance a le paquet chrony
- Vérifier que le service chrony est lancé et que sa configuration est fonctionelle

Services et binaires à activer et/ou supprimer

- Vérifier que l'instance a le paquet chrony
- Vérifier que le service chrony est lancé et que sa configuration est fonctionelle
- Vérifier que chrony n'est pas lancé en tant que root

Services et binaires à activer et/ou supprimer

- Vérifier que l'instance a le paquet chrony
- Vérifier que le service chrony est lancé et que sa configuration est fonctionelle
- Vérifier que chrony n'est pas lancé en tant que root
- Vérifier que les service autofs, samba, cups telnet et ftp ne sont pas activé

Services et binaires à activer et/ou supprimer

- Vérifier que l'instance a le paquet chrony
- Vérifier que le service chrony est lancé et que sa configuration est fonctionelle
- Vérifier que chrony n'est pas lancé en tant que root
- Vérifier que les service autofs, samba, cups telnet et ftp ne sont pas activé
- Vérifier que le server X n'est pas utilisé

Services et binaires à activer et/ou supprimer

- Vérifier que l'instance a le paquet chrony
- Vérifier que le service chrony est lancé et que sa configuration est fonctionelle
- Vérifier que chrony n'est pas lancé en tant que root
- Vérifier que les service autofs, samba, cups telnet et ftp ne sont pas activé
- Vérifier que le server X n'est pas utilisé
- Vérifier que les binaires ftp, ldap, nis et tftp ne sont pas présent sur le système

function_Apply

- Créer une fonction permettant de lancer la désactivation de services

function_Check

function_Apply

- Créer une fonction permettant de lancer la désactivation de services
- Créer une fonction permettant la suppression de rpms

function_Check

function_Apply

- Créer une fonction permettant de lancer la désactivation de services
- Créer une fonction permettant la suppression de rpms

function_Check

- Créer une fonction permettant de vérifier la présence ou l'absence d'un service

function_Apply

- Créer une fonction permettant de lancer la désactivation de services
- Créer une fonction permettant la suppression de rpms

function_Check

- Créer une fonction permettant de vérifier la présence ou l'absance d'un service
- Créer une fonction permettant la présence ou l'absance de rpms

CIS_Network

CIS_Network

Pour réduire la surface d'attaque d'un système, les périphériques inutilisés doivent être désactivés.

Gestion du réseau

- Vérifier que le protocole IPV6 est désactivé

Gestion du réseau

- Vérifier que le protocole IPV6 est désactivé
- Vérifier que les périphérique de type wifi sont désactivé

Gestion du réseau

- Vérifier que le protocole IPV6 est désactivé
- Vérifier que les périphérique de type wifi sont désactivé
- Vérifier que les périphérique de type bluetooth sont désactivé

Gestion du réseau

- Vérifier que le protocole IPV6 est désactivé
- Vérifier que les périphérique de type wifi sont désactivé
- Vérifier que les périphérique de type bluetooth sont désactivé
- Vérifier que l'ip forwarding est désactivé

Gestion du réseau

- Vérifier que le protocole IPV6 est désactivé
- Vérifier que les périphérique de type wifi sont désactivé
- Vérifier que les périphérique de type bluetooth sont désactivé
- Vérifier que l'ip forwarding est désactivé
- Vérifier que la redirection de paquet via l'icmp est désactivé

Gestion du réseau

- Vérifier que le protocole IPV6 est désactivé
- Vérifier que les périphérique de type wifi sont désactivé
- Vérifier que les périphérique de type bluetooth sont désactivé
- Vérifier que l'ip forwarding est désactivé
- Vérifier que la redirection de paquet via l'icmp est désactivé
- Vérifier la présence d'un firewawll et de son avtivation

Mettre à jour la bannière en fonction du script de sécurité

Mettre à jour la bannière en fonction du script de sécurité

Il est possible de rendre cette bannière dynamique, c'est-à-dire qu'elle reflète le niveau de sécurisation actuel du système en fonction des résultats du script pem-cyber-linux.sh. Pour cela, nous allons modifier le script pour qu'il mette à jour la bannière de manière automatique.

Modifier le script pem-cyber-linux.sh

Vous pouvez ajouter une partie au script pem-cyber-linux.sh pour qu'il mette à jour la bannière en fonction des résultats de l'analyse des règles CIS.

Ouvrez le script pem-cyber-linux.sh dans un éditeur de texte :

```
sudo vim  /usr/local/bin/pem-cyber-linux.sh
```

À la fin du script, après avoir effectué les vérifications de sécurité, ajoutez un bloc de code qui met à jour la bannière de sécurité en fonction du niveau de conformité des règles CIS. Par exemple :

```
# Vérification du niveau de conformité
if [ "$conformite" == "true" ]; then
    echo "Niveau de sécurisation : Conforme aux règles CIS" > /etc/issue
    echo "Niveau de sécurisation : Conforme aux règles CIS" > /etc/issue.net
else
    echo "Niveau de sécurisation : Non conforme aux règles CIS" > /etc/issue
    echo "Niveau de sécurisation : Non conforme aux règles CIS" > /etc/issue.net
fi
```

Ce code met à jour les fichiers /etc/issue et /etc/issue.net avec un message en fonction de l'état de conformité du système aux règles CIS.

Vous devrez définir la variable $conformite dans votre script en fonction des résultats des vérifications effectuées.

Pour tester la bannière locale, ouvrez un terminal et vérifiez la bannière avec la commande suivante :

```
cat /etc/issue
```

Pour tester la bannière SSH, déconnectez-vous et reconnectez-vous en SSH, puis vérifiez la bannière qui s'affiche avant l'authentification.

```
ssh localhost
```

Conclusion

En personnalisant la bannière de connexion de votre serveur CentOS 7, vous pouvez afficher un message informatif sur le niveau de sécurisation du système. Cela peut être très utile dans un environnement de production pour rappeler aux utilisateurs et aux administrateurs que des contrôles de sécurité rigoureux sont appliqués sur le système. La mise à jour automatique de cette bannière via le script pem-cyber-linux.sh garantit que l'état de conformité aux règles CIS est toujours reflété de manière dynamique.