Guide OVH Debian-CIS

Les scripts de sécurité modulaires pour Debian 10/11/12 basés sur les recommandations de CIS (Center for Internet Security) sont utilisés chez OVHcloud pour renforcer l'infrastructure conforme à PCI-DSS. Voici un aperçu de leur utilisation et de leur configuration :

Fonctionnement des Scripts

Ces scripts vérifient et appliquent des recommandations de sécurité sur Debian. Par exemple, pour auditer tout le système, vous pouvez utiliser la commande :

```
# bin/hardening.sh --audit-all

[...]
hardening [INFO] Treating /opt/cis-hardening/bin/hardening/6.2.19_check_duplicate_groupname.sh
6.2.19_check_duplicate_gr [INFO] Working on 6.2.19_check_duplicate_groupname
6.2.19_check_duplicate_gr [INFO] Checking Configuration
6.2.19_check_duplicate_gr [INFO] Performing audit
6.2.19_check_duplicate_gr [ OK ] No duplicate GIDs
6.2.19_check_duplicate_gr [ OK ] Check Passed
[...]
################### SUMMARY ###################
      Total Available Checks : 232
         Total Runned Checks : 166
         Total Passed Checks : [ 142/166 ]
         Total Failed Checks : [  24/166 ]
   Enabled Checks Percentage : 71.00 %
       Conformity Percentage : 85.00 %
```

Installation Rapide

```
git clone https://github.com/ovh/debian-cis.git && cd debian-cis
cp debian/default /etc/default/cis-hardening
sed -i "s#CIS_LIB_DIR=.*#CIS_LIB_DIR='$(pwd)'/lib#" /etc/default/cis-hardening
sed -i "s#CIS_CHECKS_DIR=.*#CIS_CHECKS_DIR='$(pwd)'/bin/hardening#" /etc/default/cis-hardening
sed -i "s#CIS_CONF_DIR=.*#CIS_CONF_DIR='$(pwd)'/etc#" /etc/default/cis-hardening
sed -i "s#CIS_TMP_DIR=.*#CIS_TMP_DIR='$(pwd)'/tmp#" /etc/default/cis-hardening
bin/hardening.sh --create-config-files-only
```

Utilisation et Configuration

Les scripts de durcissement se trouvent dans bin/hardening. Chaque script a un fichier de configuration correspondant dans etc/conf.d/[script_name].cfg. Vous pouvez activer ou désactiver chaque script individuellement.

```
# Configuration for script of same name
status=disabled
# Put here your exceptions concerning admin accounts shells separated by spaces
EXCEPTIONS=""
```

Les valeurs de status :

disabled : Le script ne s'exécutera pas.
audit : Le script vérifiera sans appliquer de modifications.
enabled : Le script vérifiera et appliquera les modifications nécessaires.

Commandes Principales

--audit : Vérifie le système avec tous les scripts en mode audit.
--apply : Vérifie et applique les modifications avec les scripts activés.

Options Avancées

--audit-all : Exécute tous les scripts d'audit, même ceux désactivés.
--audit-all-enable-passed : Active automatiquement les scripts qui passent en mode audit.
--sudo : Permet l'exécution en tant qu'utilisateur normal avec escalade sudo pour certains fichiers.
--batch : Simplifie la sortie de l'audit en une seule ligne par script.
--only < CHECK *NUMBER> : Exécute uniquement les vérifications sélectionnées.*
--set-hardening-level : Applique tous les contrôles jusqu'au niveau sélectionné.
--allow-service < SERVICE> : Permet de spécifier des services autorisés.
--set-log-level < LEVEL> : Définit le niveau de journalisation (info, warning, error, ok, debug).
--create-config-files-only: Crée uniquement les fichiers de configuration.
--allow-unsupported-distribution : Autorise l'exécution sur des distributions non supportées.

Score 100% : Ajout d'un script bash

Pour une implémentation dans la CI gitlab, il faut un score de 100%. Pour cela, faut **disabled** les modules en Check Failed de la benchmark avec l'ajout de ce script :

```
#!/bin/bash

# Vérifie si le fichier list_CIS.txt existe
if [ ! -f list_CIS.txt ]; then
    echo "Le fichier list_CIS.txt est introuvable."
    exit 1
fi

# Boucle sur chaque ligne du fichier list_CIS.txt
while IFS= read -r filename; do
    # Vérifie si le fichier etc/conf.d/$filename existe
    if [ -f "etc/conf.d/$filename" ]; then
```

```
        # Utilise sed pour remplacer status=audit par status=disabled dans le fichier
        sed -i 's/status=audit/status=disabled/g' "etc/conf.d/$filename"
        echo "Modifié: etc/conf.d/$filename"
    else
        echo "Le fichier etc/conf.d/$filename est introuvable."
    fi
done < list_CIS.txt
```

Et la liste des modules à `disabled` que le script va récupérer. Il faut nommer `list_CIS.txt` :

```
1.1.2_tmp_partition.cfg
1.1.3_tmp_nodev.cfg
1.1.4_tmp_nosuid.cfg
1.1.5_tmp_noexec.cfg
1.1.6.1_var_nodev.cfg
1.1.6.2_var_nosuid.cfg
1.1.6_var_partition.cfg
1.1.7_var_tmp_partition.cfg
1.1.8_var_tmp_nodev.cfg
1.1.9_var_tmp_nosuid.cfg
1.1.10_var_tmp_noexec.cfg
1.1.11.1_var_log_noexec.cfg
1.1.11.2_var_log_nosuid.cfg
1.1.11.3_var_log_nodev.cfg
1.1.11_var_log_partition.cfg
1.1.12.1_var_log_audit_noexec.cfg
1.1.12.2_var_log_audit_nosuid.cfg
1.1.12.3_var_log_audit_nodev.cfg
1.1.12_var_log_audit_partition.cfg
1.1.13_home_partition.cfg
1.1.14.1_home_nosuid.cfg
1.1.14_home_nodev.cfg
1.1.17_run_shm_noexec.cfg
1.3.3_logfile_sudo.cfg
1.4.1_install_tripwire.cfg
1.4.2_tripwire_cron.cfg
1.5.1_bootloader_ownership.cfg
1.5.2_bootloader_password.cfg
1.7.1.1_install_apparmor.cfg
1.7.1.2_enable_apparmor.cfg
1.7.1.3_enforce_or_complain_apparmor.cfg
1.7.1.4_enforcing_apparmor.cfg
1.9_install_updates.cfg
2.2.1.3_configure_chrony.cfg
2.2.1.4_configure_ntp.cfg
2.2.3_disable_avahi_server.cfg
2.2.4_disable_print_server.cfg
2.2.10_disable_http_server.cfg
2.2.14_disable_snmp_server.cfg
2.3.4_disable_telnet_client.cfg
2.3.5_disable_ldap_client.cfg
3.5.4.1.1_net_fw_default_policy_drop.cfg
4.1.1.1_install_auditd.cfg
4.1.1.2_enable_auditd.cfg
4.1.1.3_audit_bootloader.cfg
4.1.1.4_audit_backlog_limit.cfg
4.1.2.1_audit_log_storage.cfg
```

```
4.1.2.2_halt_when_audit_log_full.cfg
4.1.2.3_keep_all_audit_logs.cfg
4.1.3_record_date_time_edit.cfg
4.1.4_record_user_group_edit.cfg
4.1.5_record_network_edit.cfg
4.1.6_record_mac_edit.cfg
4.1.7_record_login_logout.cfg
4.1.8_record_session_init.cfg
4.1.9_record_dac_edit.cfg
4.1.10_record_failed_access_file.cfg
4.1.11_record_privileged_commands.cfg
4.1.12_record_successful_mount.cfg
4.1.13_record_file_deletions.cfg
4.1.14_record_sudoers_edit.cfg
4.1.15_record_sudo_usage.cfg
4.1.16_record_kernel_modules.cfg
4.1.17_freeze_auditd_conf.cfg
4.2.1.1_install_syslog-ng.cfg
4.2.1.2_enable_syslog-ng.cfg
4.2.1.4_syslog_ng_logfiles_perm.cfg
4.2.1.5_syslog-ng_remote_host.cfg
4.2.1.6_remote_syslog-ng_acl.cfg
4.2.2.3_journald_write_persistent.cfg
4.2.3_logs_permissions.cfg
4.4_logrotate_permissions.cfg
5.1.2_crontab_perm_ownership.cfg
5.1.3_cron_hourly_perm_ownership.cfg
5.1.4_cron_daily_perm_ownership.cfg
5.1.5_cron_weekly_perm_ownership.cfg
5.1.6_cron_monthly_perm_ownership.cfg
5.1.7_cron_d_perm_ownership.cfg
5.2.13_sshd_ciphers.cfg
5.2.14_ssh_cry_mac.cfg
5.2.15_ssh_cry_kex.cfg
5.2.16_sshd_idle_timeout.cfg
5.2.18_sshd_limit_access.cfg
5.2.21_disable_ssh_allow_tcp_forwarding.cfg
5.3.1_enable_pwquality.cfg
5.3.2_enable_lockout_failed_password.cfg
5.3.3_limit_password_reuse.cfg
5.4.1.1_set_password_exp_days.cfg
5.4.1.2_set_password_min_days_change.cfg
5.4.2_disable_system_accounts.cfg
5.4.4_default_umask.cfg
5.6_restrict_su.cfg
6.1.10_find_world_writable_file.cfg
6.1.11_find_unowned_files.cfg
6.1.12_find_ungrouped_files.cfg
6.1.13_find_suid_files.cfg
6.1.14_find_sgid_files.cfg
6.2.3_users_homedir_exist.cfg
6.2.8_check_user_dir_perm.cfg
6.2.9_users_homedir_ownership.cfg
99.1.1.23_disable_usb_devices.cfg
99.1.3_acc_sudoers_no_all.cfg
99.3.3.1_install_tcp_wrapper.cfg
99.3.3.3_hosts_deny.cfg
99.5.2.1_ssh_auth_pubk_only.cfg
```

```
99.5.2.2_ssh_cry_rekey.cfg
99.5.2.3_ssh_disable_features.cfg
99.5.2.4_ssh_keys_from.cfg
99.5.2.6_ssh_sys_accept_env.cfg
99.5.4.5.2_acc_shadow_sha512.cfg
```

Utilisation

1. Pour que le script fonctionne correctement, le fichier `list_CIS.txt` doit être placé dans le même répertoire que le script bash. Comme dans le répertoire `debian-cis/`
2. Enregistrez ce script dans un fichier, par exemple `update_status.sh`.
3. Rendez le script exécutable : `chmod +x update_status.sh`.
4. Exécutez le script : `./update_status.sh`.

Exit code 0 : Si le pourcentage est de 100%

Pour une implémentation dans la CI gitlab, il faut que le résultat en sortie du Benchmark réussi avec succès. Si c'est le cas, `echo $?` affiche 0.

Pour cela, il faut créer un script qui exécute le benchmark OVH CIS et renvoie un code de sortie de 0 si le score est de 100%, et de 1 sinon :

```
#!/bin/bash

# Exécuter le benchmark OVH CIS
result=$(bin/hardening.sh --audit)

# Extraire le pourcentage de conformité
conformity_percentage=$(echo "$result" | grep -oP 'Conformity Percentage\s+:\s+\K[0-9.]+')

# Vérifier si le pourcentage est de 100%
if [ "$conformity_percentage" == "100.00" ]; then
  echo "Le score est de 100%."
  exit 0
else
  echo "Le score n'est pas de 100%."
  exit 1
fi
```

Explication

1. **Exécuter le benchmark :** La commande `bin/hardening.sh --audit` lance le benchmark OVH CIS.
2. **Extraction du pourcentage de conformité :** La commande `grep` extrait la ligne contenant "Conformity Percentage" et isole le pourcentage.
3. **Vérification du score :** Le script compare le pourcentage extrait à "100.00". Si le pourcentage est de 100%, le script affiche un message de succès et se termine avec un code de sortie de 0. Sinon, il affiche un message d'échec et se termine avec un code de sortie de 1.

Utilisation

1. Placez le script ci-dessus dans un fichier, par exemple `check_benchmark.sh`.
2. Rendez le script exécutable : `chmod +x check_benchmark.sh`.
3. Exécutez le script : `./check_benchmark.sh`.

Personnalisation et Tests : Ajout de Scripts Personnalisés

```
cp src/skel bin/hardening/99.99_custom_script.sh
chmod +x bin/hardening/99.99_custom_script.sh
cp src/skel.cfg etc/conf.d/99.99_custom_script.cfg
```

Test des Scripts : Les tests fonctionnels peuvent être exécutés dans un Docker :

```
./tests/docker_build_and_run_tests.sh <target> [name of test script...]
```

Style de Codage: Utilisez Shellcheck et Shellfmt pour vérifier / maintenir le style des scripts :

```
./shellcheck/launch_shellcheck.sh [name of script...]
./shellfmt/launch_shellfmt.sh
```

Avertissement

Ce projet est un ensemble d'outils. Ils sont destinés à aider l'administrateur système à construire un environnement sécurisé. Bien que nous les utilisions chez OVHcloud pour renforcer notre infrastructure conforme au PCI-DSS, nous ne pouvons pas garantir qu'ils fonctionneront pour vous. Ils ne sécuriseront pas magiquement n'importe quel hôte.

Un mot sur la numérotation, la mise en œuvre et la durabilité de ce référentiel au fil du temps : Ce projet est né avec la distribution Debian 7 en 2016. Au fil du temps, le PDF du CIS Benchmark a évolué, changeant sa numérotation et supprimant des vérifications obsolètes.

Afin de maintenir la rétro-compatibilité avec la dernière version maintenue de Debian, la numérotation n'a pas été modifiée en même temps que le PDF, car les scripts de configuration sont nommés en fonction de celle-ci. **Changer la numérotation pourrait casser l'automatisation** pour les administrateurs l'utilisant depuis des années, et gérer ce problème sans rien casser nécessiterait une refonte importante.

Par conséquent, ne vous inquiétez pas pour la numérotation, les vérifications sont là, mais la numérotation peut différer d'un PDF à l'autre. Veuillez également noter que toutes les vérifications du PDF du CIS Benchmark ne sont peut-être pas implémentées dans cet ensemble de scripts. Nous avons choisi les plus pertinentes pour nous chez OVHcloud, n'hésitez pas à faire une Pull Request pour ajouter le script manquant que vous pourriez juger pertinent pour vous.