Mickael ANGEL

Comment intégrer Sudoers à un serveur FreeIPA

Sudo est une application permettant d'obtenir les privilèges d'administrateur (ou root) sur les systèmes d'exploitation Linux et Unix. Généralement installé par défaut sur les distributions Linux, Sudo permet également de configurer l'autorisation des utilisateurs via le fichier `/etc/sudoers`, autorisant ainsi des utilisateurs non privilégiés à exécuter certaines commandes en tant qu'administrateur.

Il existe plusieurs façons de configurer sudo. Vous pouvez l'installer sur un ordinateur ou un serveur local, ou centraliser son déploiement via un logiciel tiers. Dans cet exemple, nous utiliserons le serveur FreeIPA pour configurer une distribution sudo centralisée.

FreeIPA intègre de nombreux outils facilitant les tâches d'administration, notamment l'intégration de sudo. Vous pouvez configurer un accès sudo complet pour les utilisateurs FreeIPA et limiter l'utilisation de sudo à des commandes spécifiques, grâce aux règles HBAC (Host-based Access Control) et au groupe de commandes sudo.

Ce tutoriel vous apprendra à intégrer Sudoers et FreeIPA à travers deux scénarios. Vous découvrirez également l'utilisation de base de l'utilitaire de commande « ipa » pour la gestion des utilisateurs, des groupes d'utilisateurs, des groupes d'hôtes, des règles Sudo, des règles HBAC et des groupes de commandes Sudo. Enfin, vous apprendrez à configurer et à intégrer le service SSSD avec Sudo et FreeIPA sur les machines clientes.

Prérequis

Pour suivre ce tutoriel, le serveur FreeIPA doit être installé et entièrement configuré. Vous trouverez des tutoriels pour Rocky Linux.

Vous devez également disposer d'un utilisateur FreeIPA et d'une machine cliente configurée comme client FreeIPA. De plus, un utilisateur non root disposant des privilèges d'administrateur (sudo/root) est requis à la fois sur le serveur FreeIPA et sur le client.

Cet exemple utilise deux serveurs Rocky Linux présentant les caractéristiques suivantes :

```
Nom d'hôte           IPAdresse       Utilisécomme
------------------------------------------------------
ipa.mickaelangel.lan 192.168.5.20    FreeIPA
Serveur client.mickaelangel.lan 192.168.5.75
                     FreeIPAClient
```

Une fois ces prérequis remplis, commencez l'intégration de Sudoers avec FreeIPA.

Permet aux utilisateurs de FreeIPA d'exécuter pleinement la commande sudo

Vous apprendrez tout d'abord à intégrer Sudoers au serveur FreeIPA en créant une règle Sudo spécifique autorisant les utilisateurs à exécuter la commande « sudo ». Dans cet exemple, vous configurerez un utilisateur FreeIPA existant, nommé « rocky », pour qu'il puisse exécuter la commande « sudo » sur chaque machine cliente et obtenir les privilèges root.

Voici les étapes à suivre pour intégrer Sudoers à FreeIPA Server :• Vérifier

l'utilisateur et les connexions

- Activez la fonctionnalité Sudo sur le service SSSD (sur la machine cliente).
- Configuration des règles sudo
- Vérifiez l'intégration de

Sudoers. Commençons

maintenant.

Vérifier l'utilisateur et la connexion FreeIPA

Dans cette section, vous vérifierez et vous assurerez que l'utilisateur FreeIPA 'rocky' est disponible sur le serveur FreeIPA et que cet utilisateur peut se connecter à la machine cliente 'client.mickaelangel.lan'.

Cet exemple utilise un utilisateur FreeIPA nommé « rocky ». Saisissez la commande « ipa » suivante pour vérifier que l'utilisateur « rocky » est disponible.sur le serveur FreeIPA.

```
utilisateur ipa trouver rocky
```

Une sortie '*1 utilisateur a trouvé une correspondance*' confirme que l'utilisateur '*rocheux*' is available on the FreeIPA server.

Mickael ANGEL

```
[root@ipa ~]#
[root@ipa ~]# ipa user-find rocky
--------------
1 user matched
--------------
  User login: rocky
  First name: Rocky
  Last name: Linux
  Home directory: /home/rocky
  Login shell: /bin/bash
  Principal name: rocky@[redacted].LAN
  Principal alias: rocky@[redacted].LAN
  Email address: rocky@[redacted].lan
  UID: 487800003
  GID: 487800003
  Account disabled: False
----------------------------
Number of entries returned 1
----------------------------
[root@ipa ~]#
```

Next, enter the following command to log in to the FreeIPA client machine using the user 'rocky'. This will ensure that the user can connect to FreeIPA client machines. In this example, the server called 'client.mickaelangel.lan' is used as the FreeIPA client machine.

```
sshrocky@client.mickaelangel.lan
```

Input the password for the user 'rocky'. After logging in to the client machine, enter the following command to identify the current user.

```
whoami
id
```

```
[root@ipa ~]#
[root@ipa ~]# ssh rocky@client.[redacted].lan
(rocky@client.[redacted].lan) Password:
Last login: [redacted] from 192.168.5.25
[rocky@client ~]$
[rocky@client ~]$ id
uid=487800003(rocky) gid=487800003(rocky) groups=487800003(rocky),487800004(development)
[rocky@client ~]$
[rocky@client ~]$ whoami
rocky
[rocky@client ~]$
```

Enter the following '*sudo su*' command to gain root access or privileges. After typing your password, you should get an error such as '*rocky is not in the sudoers file. This incident will be reported.*'

```
sudo su
```

Enable Sudo Feature on SSSD Service

Before setting up Sudoers with FreeIPA, you must enable the feature 'with-sudo' features on the SSSD service on the client machine. In this section, you will enable the SSSD feature 'sudo' via the 'authselect' utility. So be sure to run these commands on the client machine 'client.mickaelangel.lan'.

Enter the following command 'authselect' to enable 'sudo' on the SSSD service. You must enable the 'sudo' feature on SSSD so the FreeIPA users can execute the 'sudo' command on the client machine.

```
sudo authselect enable-feature with-sudo
```

Next, restart the SSSD service via the following systemctl command utility. Then, verify the SSSD service to ensure thatthe service is running.

```
sudo systemctl restart sssd
sudo systemctl status sssd
```

An output 'active (running)' confirms that the SSSD service is running on the client machine.

```
[root@client ~]#
[root@client ~]# sudo authselect enable-feature with-sudo
Make sure that SSSD service is configured and enabled. See SSSD documentation for more information.

[root@client ~]# sudo systemctl restart sssd
[root@client ~]#
[root@client ~]# sudo systemctl status sssd
● sssd.service - System Security Services Daemon
     Loaded: loaded (/usr/lib/systemd/system/sssd.service; enabled; vendor preset: enabled)
     Active: active (running) since                          5s ago
   Main PID: 5259 (sssd)
      Tasks: 7 (limit: 4636)
     Memory: 52.5M
        CPU: 304ms
     CGroup: /system.slice/sssd.service
             ├─5259 /usr/sbin/sssd -i --logger=files
             ├─5260 /usr/libexec/sssd/sssd_be --domain [redacted].lan --uid 0 --gid 0 --logger=files
             ├─5261 /usr/libexec/sssd/sssd_nss --uid 0 --gid 0 --logger=files
             ├─5262 /usr/libexec/sssd/sssd_pam --uid 0 --gid 0 --logger=files
             ├─5263 /usr/libexec/sssd/sssd_ssh --uid 0 --gid 0 --logger=files
             ├─5264 /usr/libexec/sssd/sssd_sudo --uid 0 --gid 0 --logger=files
             └─5265 /usr/libexec/sssd/sssd_pac --uid 0 --gid 0 --logger=files
```

Lastly, enter the following command to show the list of enabled features on the current authentication source. On the FreeIPA client machine, SSSD is enabled by default as an authentication source.

```
authselect current
```

Under the 'Enabled features' section, you should see the parameter 'with-sudo'. This confirms that the 'sudo' feature is enabled on the SSSD authentication profile.

```
[root@client ~]#
[root@client ~]# authselect current
Profile ID: sssd
Enabled features:
- with-mkhomedir
- with-sudo
[root@client ~]#
[root@client ~]#
```

Setting up Sudo Rule

Sudo is an application that allows you to execute the command as the root user or you can also get the root user with it. By default, Freeipa supports Sudo and provides a command-line utility for integrating Sudo with the FreeIPA server.

In this section, you will integrate Sudo with FreeIPA by creating a Sudo Rule. You will first set up the user group, thenyou will set up the Sudo Rule that allows any users within the specific group to access and execute Sudo.

On the FreeIPA server, enter the following command to create a new group called 'sysadmin' on the FreeIPA server. Then, verify the details of the 'sysadmin' group.

```
ipa group-add --desc='Sysadmin Team' sysadmin
ipa group-find sysadmin
```

An output 'Added group "sysadmin"' confirms that the new group is created on the FreeIPA server. After theLorsque la commande « ipa group-find » est exécutée, vous devriez obtenir un résultat tel que « 1 groupe correspond », ce qui signifie que le groupe « sysadmin » est ajouté et disponible.

```
[root@ipa ~]#
[root@ipa ~]# ipa group-add --desc='Sysadmin Team' sysadmin
----------------------
Added group "sysadmin"
----------------------
  Group name: sysadmin
  Description: Sysadmin Team
  GID: 487800005
[root@ipa ~]#
[root@ipa ~]# ipa group-find sysadmin
---------------
1 group matched
---------------
  Group name: sysadmin
  Description: Sysadmin Team
  GID: 487800005
----------------------------
Number of entries returned 1
----------------------------
[root@ipa ~]#
```

Ensuite, saisissez la commande suivante pour créer une nouvelle règle Sudo appelée « *sysadmin_sudo*« ipa sudorule-add » est un utilitaire permettant d'ajouter la règle sudo au serveur FreeIPA. Vous créez également une nouvelle règle sudo avec certains paramètres.*--hostcat=all --runasusercat=all --runasgroupcat=all --cmdcat=all*'qui permet à cette règle d'exécuter sudo sur chaque machine cliente, et tout utilisateur ou groupe qui fera partie du «*sysadmin_sudo*' règle.

```
ipa sudorule-ajouter sysadmin_sudo \
    --hostcat=all --runasusercat=all --runasgroupcat=all --cmdcat=all
```

Une sortie '*Règle Sudo ajoutée : « sysadmin_sudo »*' confirme que la nouvelle règle sudo a été créée.

```
[root@ipa ~]#
[root@ipa ~]# ipa sudorule-add sysadmin_sudo \
    --hostcat=all --runasusercat=all --runasgroupcat=all --cmdcat=all
-------------------------------
Added Sudo Rule "sysadmin_sudo"
-------------------------------
  Rule name: sysadmin_sudo
  Enabled: True
  Host category: all
  Command category: all
  RunAs User category: all
  RunAs Group category: all
[root@ipa ~]#
[root@ipa ~]#
```

Saisissez maintenant la commande suivante pour ajouter le groupe « sysadmin » à la règle sudo « sysadmin_sudo ». Cela permettra à tous les utilisateurs FreeIPA appartenant au groupe « sysadmin » d'exécuter la commande sudo sur chaque machine cliente FreeIPA.

```
ipa sudorule-add-user sysadmin_sudo --group sysadmin
```

```
[root@ipa ~]#
[root@ipa ~]# ipa sudorule-add-user sysadmin_sudo --group sysadmin
  Rule name: sysadmin_sudo
  Enabled: True
  Host category: all
  Command category: all
  RunAs User category: all
  RunAs Group category: all
  User Groups: sysadmin
------------------------
Number of members added 1
------------------------
[root@ipa ~]#
```

Vérifiez les détails de la règle sudo « sysadmin_sudo » à l'aide de la commande suivante. Dans la section « Groupes d'utilisateurs », le groupe « sysadmin » devrait être ajouté et disponible pour « sysadmin_sudo ».

```
ipa sudorule-add-option --sudooption='!authenticate' Mickael_sudo
```

```
[root@ipa ~]#
[root@ipa ~]# ipa sudorule-show sysadmin_sudo
  Rule name: sysadmin_sudo
  Enabled: True
  Host category: all
  Command category: all
  RunAs User category: all
  RunAs Group category: all
  User Groups: sysadmin
[root@ipa ~]#
```

Enfin, saisissez la commande suivante pour ajouter l'utilisateur FreeIPA « rocky » au groupe « sysadmin ». Vérifiez ensuite les détails du groupe « sysadmin » pour vous assurer que votre utilisateur y a bien été ajouté.

```
ipa group-add-member --user=rocky sysadmin
ipa group-show sysadmin
```

Dans les détails du groupe « sysadmin », vous devriez voir l'utilisateur « rocky » ajouté et ce groupe fait également partie de la règle sudo « sysadmin_sudo ».

```
[root@ipa ~]#
[root@ipa ~]# ipa group-add-member --user=rocky sysadmin
  Group name: sysadmin
  Description: Sysadmin Team
  GID: 487800005
  Member users: rocky
  Member of Sudo rule: sysadmin_sudo
-------------------------
Number of members added 1
-------------------------
[root@ipa ~]#
[root@ipa ~]# ipa group-show sysadmin
  Group name: sysadmin
  Description: Sysadmin Team
  GID: 487800005
  Member users: rocky
  Member of Sudo rule: sysadmin_sudo
[root@ipa ~]#
[root@ipa ~]#
```

À ce stade, vous avez configuré et autorisé l'utilisateur « rocky » via le groupe « sysadmin » et la règle Sudo « sysadmin_sudo » à exécuter Sudo sur toutes les machines clientes. L'étape suivante consiste à vérifier l'intégration de Sudo avec le serveur FreeIPA.

Vérifier l'intégration de Sudoers avec FreeIPA

Dans cette section, vous vérifierez que l'intégration de Sudo avec FreeIPA fonctionne correctement. Pour ce faire, connectez-vous à la machine cliente en tant qu'utilisateur « rocky », puis exécutez la commande « sudo » afin d'obtenir les privilèges d'administrateur.

Saisissez la commande suivante pour vous connecter à la machine cliente FreeIPA « client.mickaelangel.lan » avec l'utilisateur « rocky ». Lorsque vous y êtes invité, saisissez le mot de passe de l'utilisateur « rocky ».

```
sshrocky@client.mickaelangel.lan
```

Une fois connecté à la machine cliente, exécutez la commande suivante pour identifier votre utilisateur actuel. Vous devriez constater que vous êtes connecté en tant qu'utilisateur « rocky ».

```
je
suis
qui
```

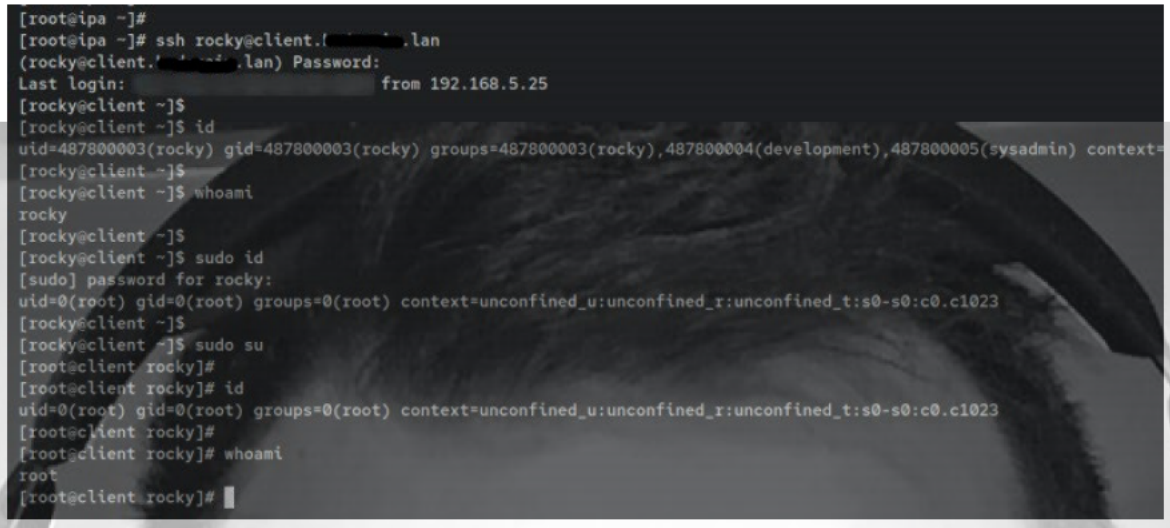
Saisissez maintenant la commande « sudo » ci-dessous pour vérifier les privilèges root « sudo » de l'utilisateur FreeIPA « rocky ». Lorsque vous y êtes invité, saisissez le mot de passe de l'utilisateur « rocky ».

```
sudo id
sudo su
```

Une fois l'opération réussie, votre invite de commande devrait afficher « root@hostname... ». Cela confirme que vous disposez désormais des privilèges root et que vous êtes connecté en tant qu'utilisateur root.

Identifiez votre utilisateur actuel à l'aide de la commande ci-dessous. Vous devriez constater que vous êtes actuellement connecté en tant qu'utilisateur « root ».


```
je
suis
qui
```



Grâce à cela, vous avez intégré avec succès Sudoers au serveur FreeIPA. Tout utilisateur du groupe « sysadmin » peut désormais exécuter la commande « sudo » et obtenir les privilèges root sur chaque machine cliente.

Permet aux utilisateurs de FreeIPA d'exécuter des commandes sudo spécifiques.

Dans ce scénario, vous allez créer un nouvel utilisateur FreeIPA qui pourra se connecter à la machine cliente et exécuter Sudo pour des commandes spécifiques.

Dans cet exemple, vous allez configurer un nouvel utilisateur appelé « max » qui pourra se connecter à la machine cliente « client.mickaelangel.lan » via SSH et exécuter Sudo, à l'exception de certaines commandes de gestion de la pile LEMP.

Pour ce faire, vous devrez suivre les étapes

suivantes :• Création d'un utilisateur et d'un

groupe FreeIPA
- Création d'un groupe hôte
- Création d'une règle HBAC (contrôle d'accès basé sur l'hôte)• Création d'une règle Sudo
- Création d'un groupe de commandes sudo
- Vérifiez l'intégration de Sudo avec

FreeIPA. Commençons maintenant.

Création d'un utilisateur et d'un groupe FreeIPA

Vous devrez tout d'abord créer et définir un utilisateur et un groupe spécifiques. Dans cet exemple, vous créerez un nouvel utilisateur nommé « max » et le groupe d'utilisateurs « systemadmin » sur le serveur FreeIPA. Vous vérifierez ensuite que le nouvel utilisateur « max » peut accéder à la machine cliente « client.mickaelangel.lan » et s'y connecter.

Créez un nouvel utilisateur FreeIPA nommé « max » en saisissant la commande « ipa user-add » ci-dessous. Saisissez le mot de passe de votre nouvel utilisateur lorsque vous y êtes invité, puis répétez l'opération.

```
ipa user-add max --first=Max --last=Rods --password
```

```
[root@ipa ~]#
[root@ipa ~]# ipa user-add max --first=Max --last=Rods --password
Password:
Enter Password again to verify:
----------------
Added user "max"
----------------
  User login: max
  First name: Max
  Last name: Rods
  Full name: Max Rods
  Display name: Max Rods
  Initials: MR
  Home directory: /home/max
  GECOS: Max Rods
  Login shell: /bin/bash
  Principal name: max@[redacted].LAN
  Principal alias: max@[redacted].LAN
  User password expiration: 20230225133119Z
  Email address: max@[redacted].lan
  UID: 487800008
  GID: 487800008
  Password: True
  Member of groups: ipausers
  Kerberos keys available: True
[root@ipa ~]#
```

Vérifiez maintenant les détails de l'utilisateur « max » à l'aide de la commande suivante. Vous devriez voir la configuration détaillée par défaut del'utilisateur 'max'.

```
recherche d'utilisateurs ipa max
```

```
[root@ipa ~]#
[root@ipa ~]# ipa user-find max
--------------
1 user matched
--------------
  User login: max
  First name: Max
  Last name: Rods
  Home directory: /home/max
  Login shell: /bin/bash
  Principal name: max@[redacted].LAN
  Principal alias: max@[redacted].LAN
  Email address: max@[redacted].lan
  UID: 487800008
  GID: 487800008
  Account disabled: False
----------------------------
Number of entries returned 1
----------------------------
[root@ipa ~]#
```

Ensuite, saisissez la commande suivante pour ajouter un nouveau groupe « systemadmin ». Puis, ajoutez l'utilisateur « max » au groupe « systemadmin ».

```
ipa group-add --desc='Équipe d'administration
système' systemadmin ipa group-add-member --user=max
svstemadmin
```

Vérifiez la configuration détaillée du groupe « systemadmin » à l'aide de la commande suivante. L'utilisateur « max » devrait y être ajouté et disponible.

```
ipa group-show systemadmin
```

```
[root@ipa ~]#
[root@ipa ~]# ipa group-add --desc='System Admin Team' systemadmin
------------------------
Added group "systemadmin"
------------------------
  Group name: systemadmin
  Description: System Admin Team
  GID: 487800009
[root@ipa ~]#
```

```
[root@ipa ~]#
[root@ipa ~]# ipa group-add-member --user=max systemadmin
  Group name: systemadmin
  Description: System Admin Team
  GID: 487800009
  Member users: max
-------------------------
Number of members added 1
-------------------------
[root@ipa ~]# ipa group-show systemadmin
  Group name: systemadmin
  Description: System Admin Team
  GID: 487800009
  Member users: max
[root@ipa ~]#
```

Pour permettre au nouvel utilisateur de se connecter à la machine cliente, saisissez la commande ssh suivante. Dans cet exemple,L'utilisateur 'max' se connectera à la machine cliente 'client.mickaelangel.lan'.

```
sshmax@client.mickaelangel.lan
```

Saisissez votre mot de passe lorsque vous y êtes invité. Il vous sera ensuite demandé de modifier le mot de passe par défaut. Saisissez donc votre mot de passe actuel, puis le nouveau mot de passe, et répétez l'opération.

```
[root@ipa ~]#
[root@ipa ~]# ssh max@client.[illegible].lan
(max@client.[illegible].lan) Password:
(max@client.[illegible].lan) Password expired. Change your password now.
Current Password:
(max@client.[illegible].lan) New password:
(max@client.[illegible].lan) Retype new password:
[max@client ~]$
[max@client ~]$ id
uid=487800008(max) gid=487800008(max) groups=487800008(max),487800009(systemadmin) c
[max@client ~]$
[max@client ~]$ whoami
max
[max@client ~]$
```

Après vous être connecté, identifiez votre utilisateur actuel à l'aide de la commande suivante.

```
je
suis
qui
```

Vous devriez voir que vous êtes connecté en tant qu'utilisateur 'max' au 'client.mickaelangel.lan'.

Création d'un groupe hôte

Après avoir créé l'utilisateur et le groupe, vous allez maintenant créer et configurer le groupe hôte sur FreeIPA. Vous créerez un nouveau groupe hôte appelé « appserver », et l'un des membres de ce groupe sera « client.mickaelangel.lan ».

Vérifiez la liste des hôtes disponibles sur FreeIPA à l'aide de la commande suivante. Elle affichera la liste des machines disponibles sur le serveur FreeIPA. Dans cet exemple, deux hôtes, « ipa.mickaelangel.lan » et « client.mickaelangel.lan », sont disponibles sur le serveur FreeIPA.

```
ipa host-find
```

```
[root@ipa ~]#
[root@ipa ~]# ipa host-find
----------
2 hosts matched
----------
  Host name: client.[redacted].lan
  Platform: x86_64
  Operating system: 5.14.0-162.12.1.el9_1.0.2.x86_64
  Principal name: host/client.[redacted].lan@[redacted].LAN
  Principal alias: host/client.[redacted].lan@[redacted].LAN
  SSH public key fingerprint: SHA256:MDjwndHGk2ninB51p6191iG6zJix+ivwiKIIqnmCP+E (ssh-ed25519), SH
                              SHA256:HBICPqvhnleV41/rewgeW8MwhO/cXPD07w1e0OzrK88 (ssh-rsa)

  Host name: ipa.[redacted].lan
  Principal name: host/ipa.[redacted].lan@[redacted].LAN
  Principal alias: host/ipa.[redacted].lan@[redacted].LAN
  SSH public key fingerprint: SHA256:1uWwWhUHEYa4o6A9V40nmnQvke3r9p3fGolBlvakeME (ssh-ed25519), SH
                              SHA256:8+dKtl7sGPZOx/COWkektxKZSZflM4Bb00jyHVesGko (ssh-rsa)
----------
Number of entries returned 2
----------
[root@ipa ~]#
```

Créez maintenant un nouveau groupe d'hôtes appelé «*serveur d'applications*Ensuite, ajoutez l'hôte.*client.mickaelangel.lan*' au groupe hôte '*serveur d'applications*'.

```
ipa hostgroup-add appserver
ipa hostgroup-add-member appserver --hosts=client.mickaelangel.lan
```

```
[root@ipa ~]#
[root@ipa ~]# ipa hostgroup-add appserver
----------------------------
Added hostgroup "appserver"
----------------------------
  Host-group: appserver
[root@ipa ~]#
```

```
[root@ipa ~]#
[root@ipa ~]# ipa hostgroup-add-member appserver --hosts=client.[redacted].lan
  Host-group: appserver
  Member hosts: client.[redacted].lan
-------------------------
Number of members added 1
-------------------------
[root@ipa ~]#
[root@ipa ~]# ipa hostgroup-show appserver
  Host-group: appserver
  Member hosts: client.[redacted].lan
[root@ipa ~]#
```

Vérifiez le groupe hôte détaillé '*serveur d'applications*' en utilisant la commande ci-dessous. Vous devriez voir l'hôte '*client.mickaelangel.lan*est ajouté et disponible sur le*serveur d'applications*' groupe hôte.

```
ipa hostgroup-show appserver
```

Création d'une règle HBAC (Host-Based Access Control)

Une fois le groupe d'hôtes « appserver » créé, vous pouvez configurer la règle HBAC (Host-based Access Control) sur le serveur FreeIPA. Cette règle vous permet de définir des politiques de restriction d'accès aux hôtes et services en fonction de l'utilisateur qui tente de se connecter, de ses groupes d'hôtes, de l'hôte auquel il tente d'accéder (ou de ses groupes d'hôtes) et, éventuellement, du service utilisé.

Dans cette section, vous désactiverez la règle HBAC par défaut et configurerez une règle HBAC personnalisée appelée « operation_admin » qui autorisera l'utilisateur « max » ou le groupe « systemadmin » à administrer tous les hôtes du groupe d'hôtes « appserver » via SSH. Cet utilisateur ou ce groupe pourra également exécuter les commandes « sudo » et « sudo -l ».

Saisissez la commande suivante pour désactiver la règle HBAC par défaut « allow_all ». Vérifiez ensuite les détails de cette règle.

```
ipa hbacrule-disable
autoriser_tout ipa hbacrule-
show autoriser tout
```

Une sortie '*La règle HBAC désactivée « allow_all » est confirmée.*La règle est désactivée. De plus, sur le '*Activé*Dans cette section, vous devriez voir la valeur changée en '*FALSE*'.

```
[root@ipa ~]#
[root@ipa ~]# ipa hbacrule-disable allow_all
-----------------------------
Disabled HBAC rule "allow_all"
-----------------------------
[root@ipa ~]#
[root@ipa ~]# ipa hbacrule-show allow_all
  Rule name: allow_all
  User category: all
  Host category: all
  Service category: all
  Description: Allow all users to access any host from any host
  Enabled: False
[root@ipa ~]#
```

Avec la règle HBAC par défaut « allow_all » désactivée, aucun utilisateur ne pourra se connecter à la machine cliente.

Ensuite, créez une nouvelle règle HBAC personnalisée appelée «*administrateur d'opérations*' en utilisant ce qui suit '*ipa hbacrule-add*' commande.

```
ipa hbacrule-ajouter opération_admin
```

```
[root@ipa ~]#
[root@ipa ~]# ipa hbacrule-add operation_admin
---------------------------------
Added HBAC rule "operation_admin"
---------------------------------
  Rule name: operation_admin
  Enabled: True
[root@ipa ~]#
```

Ajoutez maintenant le groupe hôte '*serveur d'applications*' via la commande '*ipa hbacrule-ajouter-hôte*'et le groupe d'utilisateurs '*administrateur système*' via la commande'*ipa hbacrule-add-user*'à la règle HBAC «*administrateur d'opérations*'.

```
ipa hbacrule-add-host operation_admin --hostgroup appserver
ipa hbacrule-add-user operation_admin --group systemadmin
```

```
[root@ipa ~]#
[root@ipa ~]# ipa hbacrule-add-host operation_admin --hostgroup appserver
  Rule name: operation_admin
  Enabled: True
  Host Groups: appserver
-------------------------
Number of members added 1
-------------------------
[root@ipa ~]#
[root@ipa ~]# ipa hbacrule-add-user operation_admin --group systemadmin
  Rule name: operation_admin
  Enabled: True
  User Groups: systemadmin
  Host Groups: appserver
-------------------------
Number of members added 1
-------------------------
[root@ipa ~]#
```

Ensuite, ajoutez les services « sshd », « sudo » et « su » à la règle HBAC « operation_admin ». Tous les hôtes ou groupes d'hôtes inclus dans cette règle seront concernés et autorisés à accéder à ces commandes ou à les exécuter.

```
ipa hbacrule-add-service operation_admin --hbacsvcs=sshd
ipa hbacrule-add-service opération_admin --hbacsvcs=sudo --hbacsvcs=su-l
```

```
[root@ipa ~]#
[root@ipa ~]# ipa hbacrule-add-service operation_admin --hbacsvcs=sshd
  Rule name: operation_admin
  Enabled: True
  User Groups: systemadmin
  Host Groups: appserver
  HBAC Services: sshd
-------------------------
Number of members added 1
-------------------------
[root@ipa ~]#
[root@ipa ~]# ipa hbacrule-add-service operation_admin --hbacsvcs=sudo --hbacsvcs=su-l
  Rule name: operation_admin
  Enabled: True
  User Groups: systemadmin
  Host Groups: appserver
  HBAC Services: sshd, sudo, su-l
-------------------------
Number of members added 2
-------------------------
[root@ipa ~]#
```

Vérifiez les détails de la règle HBAC.*administrateur d'opérations*« Utilisez la commande suivante. Vous devriez voir trois services. »*sshd*', '*sudo*', et '*su-l*' ajouté et disponible sur le '*administrateur d'opérations*'.

```
ipa hbacrule-show operation_admin
```

```
[root@ipa ~]#
[root@ipa ~]# ipa hbacrule-show operation_admin
  Rule name: operation_admin
  Enabled: True
  User Groups: systemadmin
  Host Groups: appserver
  HBAC Services: sshd, sudo, su-l
[root@ipa ~]#
```

À ce stade, seuls les utilisateurs ou groupes associés à la règle « operation_admin » pourront se connecter à la machine cliente. L'utilisateur « rocky » ne peut pas se connecter à la machine cliente « client.mickaelangel.lan », contrairement à l'utilisateur « max », qui peut s'y connecter. En effet, « max » fait partie du groupe « systemadmin », auquel est appliquée la règle HBAC « operation_admin ».

La connexion en tant qu'utilisateur « rocky » sera définie par la machine cliente.

```
sshrocky@client.mickaelangel.lan
```

La connexion en tant qu'utilisateur « max » est autorisée car il fait partie du groupe « systemadmin ».

```
sshmax@client.mickaelangel.lan
```

```
[root@ipa ~]#
[root@ipa ~]# ssh rocky@client.[redacted].lan
(rocky@client.[redacted].lan) Password:
Connection closed by UNKNOWN port 65535
[root@ipa ~]#
[root@ipa ~]# ssh max@client.[redacted].lan
(max@client.[redacted].lan) Password:
Last login: Tue Feb 28 03:52:36 2023 from 192.168.5.25
[max@client ~]$
[max@client ~]$ id
uid=487800008(max) gid=487800008(max) groups=487800008(max),487800009(systemadmin) context=
[max@client ~]$
[max@client ~]$ whoami
max
[max@client ~]$
```

Création d'une règle Sudo

Dans cette section, vous allez créer et configurer une nouvelle règle sudo sur FreeIPA. Ensuite, vous attribuerez cette règle sudo à...groupe spécifique « systemadmin » et le groupe hôte « appserver ».

Saisissez la commande suivante pour ajouter une nouvelle règle Sudo appelée « *systemadmin_sudo*Dans cet exemple, vous créez une nouvelle règle sudo qui ne s'appliquera qu'à l'utilisateur et au groupe avec le paramètre *--runasusercat=all*

--runasgroupcat=all'.

Mickael ANGEL

Ainsi, toutes les commandes ne peuvent pas être exécutées avec sudo ou en tant que root, et cette règle peut être appliquée à des clients ou hôtes spécifiques.


```
ipa sudorule-add systemadmin_sudo \
    --runasusercat=all --runasgroupcat=all
```

```
[root@ipa ~]#
[root@ipa ~]# ipa sudorule-add systemadmin_sudo \
    --runasusercat=all --runasgroupcat=all
---------------------------------
Added Sudo Rule "systemadmin_sudo"
---------------------------------
  Rule name: systemadmin_sudo
  Enabled: True
  RunAs User category: all
  RunAs Group category: all
[root@ipa ~]#
[root@ipa ~]#
```

Saisissez maintenant la commande ci-dessous pour ajouter le groupe d'utilisateurs « systemdadmin » et le groupe d'hôtes « appserver » à la règle sudo « systemadmin_sudo ». Ainsi, la règle « systemadmin_sudo » ne s'appliquera qu'aux utilisateurs du groupe « systemadmin » et aux hôtes/clients du groupe « appserver ».

```
ipa sudorule-add-user systemadmin_sudo --group systemadmin
ipa sudorule-add-host systemadmin_sudo --hostgroup appserver
```

```
[root@ipa ~]#
[root@ipa ~]# ipa sudorule-add-user systemadmin_sudo --group systemadmin
  Rule name: systemadmin_sudo
  Enabled: True
  RunAs User category: all
  RunAs Group category: all
  User Groups: systemadmin
-------------------------
Number of members added 1
-------------------------
[root@ipa ~]#
[root@ipa ~]# ipa sudorule-add-host systemadmin_sudo --hostgroup appserver
  Rule name: systemadmin_sudo
  Enabled: True
  RunAs User category: all
  RunAs Group category: all
  User Groups: systemadmin
  Host Groups: appserver
-------------------------
Number of members added 1
-------------------------
[root@ipa ~]#
```

Création d'un groupe de commandes sudo

Une fois la règle sudo créée, vous devez configurer le groupe de commandes sudo dans FreeIPA. Vous pourrez ainsi associer plusieurs commandes à ce groupe, puis appliquer ce dernier à la règle sudo correspondante.

Dans cet exemple, vous définirez des commandes FreeIPA permettant de démarrer et de redémarrer la pile LEMP.Ensuite, vous créerez un nouveau groupe de commandes Sudo appelé « systemadmin_cmds » et y ajouterez vos commandes. Enfin, vous associerez le groupe de commandes Sudo « systemadmin_cmds » à la règle Sudo « systemadmin_sudo ».

Saisissez la commande suivante pour ajouter de nouvelles commandes au serveur FreeIPA. Dans cet exemple, vous ajouterez des commandes permettant de gérer la pile LEMP. Vous pourrez ainsi démarrer et redémarrer les services Nginx, MariaDB et PHP-FPM.

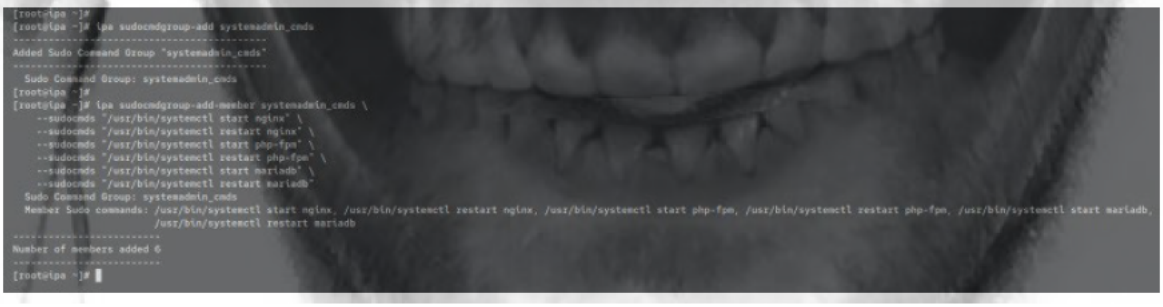
```
ipa sudocmd-add "/usr/bin/systemctl start nginx"
ipa sudocmd-add "/usr/bin/systemctl restart nginx"

ipa sudocmd-add "/usr/bin/systemctl start php-fpm"
ipa sudocmd-add "/usr/bin/systemctl restart php-fpm"

ipa sudocmd-add "/usr/bin/systemctl start mariadb"
ipa sudocmd-add "/usr/bin/systemctl restart mariadb"
```

```
[root@ipa ~]#
[root@ipa ~]# ipa sudocmd-add "/usr/bin/systemctl start nginx"
----------------------------------------------------
Added Sudo Command "/usr/bin/systemctl start nginx"
----------------------------------------------------
  Sudo Command: /usr/bin/systemctl start nginx
[root@ipa ~]#
[root@ipa ~]# ipa sudocmd-add "/usr/bin/systemctl restart nginx"
----------------------------------------------------
Added Sudo Command "/usr/bin/systemctl restart nginx"
----------------------------------------------------
  Sudo Command: /usr/bin/systemctl restart nginx
[root@ipa ~]#
[root@ipa ~]# ipa sudocmd-add "/usr/bin/systemctl start php-fpm"
----------------------------------------------------
Added Sudo Command "/usr/bin/systemctl start php-fpm"
----------------------------------------------------
  Sudo Command: /usr/bin/systemctl start php-fpm
[root@ipa ~]#
[root@ipa ~]# ipa sudocmd-add "/usr/bin/systemctl restart php-fpm"
----------------------------------------------------
Added Sudo Command "/usr/bin/systemctl restart php-fpm"
----------------------------------------------------
  Sudo Command: /usr/bin/systemctl restart php-fpm
[root@ipa ~]#
[root@ipa ~]# ipa sudocmd-add "/usr/bin/systemctl start mariadb"
----------------------------------------------------
Added Sudo Command "/usr/bin/systemctl start mariadb"
----------------------------------------------------
  Sudo Command: /usr/bin/systemctl start mariadb
[root@ipa ~]#
[root@ipa ~]# ipa sudocmd-add "/usr/bin/systemctl restart mariadb"
----------------------------------------------------
Added Sudo Command "/usr/bin/systemctl restart mariadb"
----------------------------------------------------
  Sudo Command: /usr/bin/systemctl restart mariadb
[root@ipa ~]#
```

Ensuite, créez un nouveau groupe de commandes sudo nommé « systemadmin_cmds ». Puis, ajoutez à ce groupe toutes les commandes permettant de gérer la pile LEMP.

```
ipa sudocmdgroup-add systemadmin_cmds
ipa sudocmdgroup-add-member systemadmin_cmds \
    --sudocmds "/usr/bin/systemctl start nginx" \
    --sudocmds "/usr/bin/systemctl restart nginx" \
    --sudocmds "/usr/bin/systemctl start php-fpm" \
    --sudocmds "/usr/bin/systemctl restart php-fpm" \
    --sudocmds "/usr/bin/systemctl start mariadb" \
    --sudocmds "/usr/bin/systemctl restart mariadb"
```

Une sortie '*Ajout du groupe de commandes sudo « systemdadmin_cmds »*' confirme que le nouveau groupe de commandes sudo a été créé. De plus, une sortie '*Nombre de membres ajoutés : 6*' confirme que vous avez ajouté 6 commandes au '*commandes systemdadmin*' groupe de commandes sudo.



Enfin, saisissez la commande suivante pour ajouter le groupe de commandes sudo « systemadmin_cmds » à la règle sudo « systemadmin_sudo ». Ainsi, les utilisateurs relevant de la règle sudo « systemadmin_sudo » pourront exécuter les six commandes sudo disponibles dans le groupe « systemadmin_cmds ».

```
ipa sudorule-add-allow-command systemadmin_sudo --sudocmdgroups systemadmin_cmds
```

```
[root@ipa ~]#
[root@ipa ~]# ipa sudorule-add-allow-command systemadmin_sudo --sudocmdgroups systemadmin_cmds
  Rule name: systemadmin_sudo
  Enabled: True
  RunAs User category: all
  RunAs Group category: all
  User Groups: systemadmin
  Host Groups: appserver
  Sudo Allow Command Groups: systemadmin_cmds
-------------------------
Number of members added 1
-------------------------
[root@ipa ~]#
[root@ipa ~]#
```

Vérifier l'intégration de Sudoers avec FreeIPA

Pour vérifier l'implémentation de Sudo et son intégration avec FreeIPA dans ce scénario, vous vous connecterez àConnectez-vous à « client.mickaelangel.lan » avec le nouvel utilisateur « max ». Vous exécuterez ensuite certaines commandes permettant de gérer les services de la pile LEMP.

Connectez-vous au client.mickaelangel.lan avec l'utilisateur « max » en utilisant la commande ssh suivante. Saisissez le mot de passe lorsque vous y êtes invité.

```
sshmax@client.mickaelangel.lan
```

Après vous être connecté, saisissez la commande suivante pour identifier votre utilisateur actuel. Vous constaterez que vous êtes connecté en tant qu'utilisateur « max ».

```
je
suis
qui
```

```
[max@client ~]$
[max@client ~]$ su -l max
Password:
Last login:                              on pts/1
[max@client ~]$
[max@client ~]$ id
uid=487800008(max) gid=487800008(max) groups=487800008(max),487800009(systemadmin) conte
[max@client ~]$
[max@client ~]$ whoami
max
[max@client ~]$
```

Ensuite, saisissez la commande suivante pour redémarrer les services LEMP Stack. Lorsque vous y êtes invité, entrez le mot de passe de l'utilisateur « max ». L'opération devrait réussir car l'utilisateur « max » est autorisé à exécuter ces commandes avec les privilèges « sudo ».

```
sudo /usr/bin/systemctl restart nginx
sudo /usr/bin/systemctl restart php-fpm
sudo /usr/bin/systemctl restart mariadb
```

```
[max@client ~]$
[max@client ~]$ sudo /usr/bin/systemctl restart nginx
[sudo] password for max:
[max@client ~]$
[max@client ~]$ sudo /usr/bin/systemctl restart php-fpm
[max@client ~]$ sudo /usr/bin/systemctl restart mariadb
[max@client ~]$
[max@client ~]$
```

Maintenant, si vous essayez le «*systemctl stop*Si la commande échoue, l'opération devrait échouer et vous obtiendrez une erreur telle que « *Désolé, l'utilisateur max est désormais autorisé à exécuter des commandes en tant que root sur le client.mickaelangel.lan*'.

```
sudo /usr/bin/systemctl stop nginx
sudo /usr/bin/systemctl stop php-fpm
sudo /usr/bin/systemctl stop mariadb
```

```
[max@client ~]$
[max@client ~]$ sudo /usr/bin/systemctl stop nginx
Sorry, user max is not allowed to execute '/usr/bin/systemctl stop nginx' as root on client.[redacted].lan.
[max@client ~]$
[max@client ~]$ sudo /usr/bin/systemctl stop php-fpm
Sorry, user max is not allowed to execute '/usr/bin/systemctl stop php-fpm' as root on client.[redacted].lan.
[max@client ~]$
[max@client ~]$ sudo /usr/bin/systemctl stop mariadb
Sorry, user max is not allowed to execute '/usr/bin/systemctl stop mariadb' as root on client.[redacted].lan.
[max@client ~]$
```

Vous avez ainsi terminé la configuration de Sudoers avec le serveur FreeIPA pour autoriser l'exécution de commandes spécifiques.exécuté en tant que sudo ou en tant que root.

Conclusion

Ce tutoriel vous a appris à intégrer Sudoers au serveur FreeIPA. Vous avez également découvert quelques commandes de base de l'utilitaire « ipa » pour la gestion des utilisateurs, des groupes, des groupes d'hôtes, des règles sudo, des règles HBAC (contrôle d'accès basé sur l'hôte) et des autorisations sudo.Groupe de commandes. Vous avez appris l'intégration de base de Sudoers avec le serveur FreeIPA à travers deux scénarios différents : autoriser les utilisateurs à exécuter sudo pour toutes les commandes et tous les hôtes, et autoriser les utilisateurs à exécuter sudo uniquement avec des commandes spécifiques sur des hôtes/clients spécifiques.

De plus, vous avez également appris la configuration de base du service SSSD et son intégration avec Sudo via leServeur FreeIPA.

Grâce à cette implémentation de Sudo, vous pouvez mettre en œuvre ces deux scénarios sur votre serveur de déploiement. Pour en savoir plus, consultez la documentation officielle de FreeIPA et Sudoer.
