Red Hat Enterprise Linux 9

Gestion des certificats dans IdM

Délivrance de certificats, configuration de l'authentification par certificat et contrôle de la validité des certificats

Last Updated: 2024-02-22

Red Hat Enterprise Linux 9 Gestion des certificats dans IdM

Délivrance de certificats, configuration de l'authentification par certificat et contrôle de la validité des certificats

Notice légale

Résumé

Les administrateurs utilisent des certificats X.509 pour authentifier les utilisateurs, les hôtes et les services, et pour permettre la signature numérique et le cryptage. Dans Red Hat Identity Management (IdM), vous pouvez gérer les certificats en utilisant l'autorité de certification (AC) intégrée ou externe. Vous pouvez demander et renouveler des certificats en utilisant le service certmonger, l'outil certutil ou les Ansible Playbooks. Pour remplacer les certificats du serveur web et du serveur LDAP des serveurs IdM, vous devez effectuer des actions manuelles. Les administrateurs peuvent créer des sous-CA légers pour émettre des certificats dans un but précis,

par exemple des certificats d'utilisateur pour une passerelle VPN. L'administrateur peut ensuite invalider tous les certificats pour ce service en révoquant le certificat du sous-CA lorsque cette passerelle VPN n'est plus nécessaire.

Table des matières

RENDRE L'OPEN SOURCE PLUS INCLUSIF

Red Hat s'engage à remplacer les termes problématiques dans son code, sa documentation et ses propriétés Web. Nous commençons par ces quatre termes : master, slave, blacklist et whitelist. En raison de l'ampleur de cette entreprise, ces changements seront mis en œuvre progressivement au cours de plusieurs versions à venir. Pour plus de détails, voir le message de notre directeur technique Chris Wright.

FOURNIR UN RETOUR D'INFORMATION SUR LA DOCUMENTATION DE RED HAT

Nous apprécions vos commentaires sur notre documentation. Faites-nous savoir comment nous pouvons l'améliorer.

Soumettre des commentaires sur des passages spécifiques

1. Consultez la documentation au format **Multi-page HTML** et assurez-vous que le bouton **Feedback** apparaît dans le coin supérieur droit après le chargement complet de la page.
2. Utilisez votre curseur pour mettre en évidence la partie du texte que vous souhaitez commenter.
3. Cliquez sur le bouton **Add Feedback** qui apparaît près du texte en surbrillance.
4. Ajoutez vos commentaires et cliquez sur **Submit**.

Soumettre des commentaires via Bugzilla (compte requis)

1. Connectez-vous au site Web de Bugzilla.
2. Sélectionnez la version correcte dans le menu **Version**.
3. Saisissez un titre descriptif dans le champ **Summary**.
4. Saisissez votre suggestion d'amélioration dans le champ **Description**. Incluez des liens vers les parties pertinentes de la documentation.
5. Cliquez sur **Submit Bug**.

CHAPITRE 1. LES CERTIFICATS DE CLÉ PUBLIQUE DANS LA GESTION DE L'IDENTITÉ

Ce chapitre décrit les certificats de clé publique X.509, qui sont utilisés pour authentifier les utilisateurs, les hôtes et les services dans le cadre de la gestion des identités (IdM). Outre l'authentification, les certificats X.509 permettent également la signature numérique et le cryptage afin de garantir la confidentialité, l'intégrité et la non-répudiation.

Un certificat contient les informations suivantes :

- Le sujet que le certificat authentifie.
- L'émetteur, c'est-à-dire l'autorité de certification qui a signé le certificat.
- Les dates de début et de fin de validité du certificat.
- Les utilisations valides du certificat.
- La clé publique du sujet.

Un message crypté par la clé publique ne peut être décrypté que par la clé privée correspondante. Alors qu'un certificat et la clé publique qu'il contient peuvent être rendus publics, l'utilisateur, l'hôte ou le service doit garder sa clé privée secrète.

1.1. AUTORITÉS DE CERTIFICATION DANS L'IDM

Les autorités de certification fonctionnent selon une hiérarchie de confiance. Dans un environnement IdM doté d'une autorité de certification (AC) interne, tous les hôtes, utilisateurs et services IdM font confiance aux certificats signés par l'AC. Outre cette autorité de certification racine, l'IdM prend en charge des sousautorités de certification auxquelles l'autorité de certification racine a accordé la possibilité de signer des certificats à leur tour. Souvent, les certificats que ces sous-AC peuvent signer sont des certificats d'un type spécifique, par exemple des certificats VPN. Enfin, IdM prend en charge l'utilisation d'AC externes. Le tableau ci-dessous présente les spécificités de l'utilisation des différents types d'AC dans l'IdM.

Tableau 1.1. Comparaison de l'utilisation d'autorités de certification intégrées et externes dans la gestion de l'identité

Nom de l'AC	Description	Utilisation	Liens utiles
ipa CA	Une AC intégrée basée sur le projet Dogtag en amont	Les autorités de certification intégrées peuvent créer, révoquer et émettre des certificats pour les utilisateurs, les hôtes et les services.	Utilisation de l'autorité de certification ipa pour demander un nouveau certificat d'utilisateur et l'exporter vers le client
Sous-activités de l'IdM	Une AC intégrée subordonnée à l'AC **ipa**	Les sous-CA IdM sont des AC auxquelles l'AC **ipa** a accordé la capacité de signer des certificats. Il s'agit souvent de certificats d'un type particulier, par exemple des certificats VPN.	Restreindre une application à ne faire confiance qu'à un sous-ensemble de certificats

Nom de l'AC	Description	Utilisation	Liens utiles
AC externes	Une AC externe est une AC autre que l'AC IdM intégrée ou ses sous-AAC.	Les outils IdM permettent d'ajouter ou de supprimer des certificats émis par ces autorités de certification à des utilisateurs, des services ou des hôtes.	Gestion des certificats signés en externe pour les utilisateurs, les hôtes et les services IdM

Du point de vue du certificat, il n'y a pas de différence entre la signature par une autorité de certification IdM auto-signée et la signature externe.

Le rôle de l'AC comprend les objectifs suivants :

- Il délivre des certificats numériques.
- En signant un certificat, il certifie que le sujet nommé dans le certificat possède une clé publique. Le sujet peut être un utilisateur, un hôte ou un service.
- Il peut révoquer des certificats et fournit un état de révocation via les listes de révocation de certificats (CRL) et le protocole d'état des certificats en ligne (OCSP).

Ressources supplémentaires

- Voir Planification des services de l'AC.

1.2. COMPARAISON DES CERTIFICATS ET DE KERBEROS

Les certificats remplissent une fonction similaire à celle des tickets Kerberos. Kerberos est un protocole d'authentification de réseau informatique qui fonctionne sur la base de tickets pour permettre aux nœuds communiquant sur un réseau non sécurisé de prouver leur identité les uns aux autres de manière sécurisée. Le tableau suivant présente une comparaison entre Kerberos et les certificats X.509 :

Tableau 1.2. Comparaison des certificats et de Kerberos

Characteristic	Kerberos	X.509
Authentication	Oui	Oui
Privacy	En option	Oui
Integrity	En option	Oui
Type of cryptography involved	Symétrique	Asymétrique
Default validity	Court (1 jour)	Longue (2 ans)

Par défaut, Kerberos dans la gestion de l'identité ne garantit que l'identité des parties qui communiquent.

1.3. AVANTAGES ET INCONVÉNIENTS DE L'UTILISATION DE CERTIFICATS POUR L'AUTHENTIFICATION DES UTILISATEURS DANS L'IDM

Les avantages de l'utilisation de certificats pour l'authentification des utilisateurs dans le cadre de l'IdM sont notamment les suivants :

- Un code PIN qui protège la clé privée d'une carte à puce est généralement moins complexe et plus facile à mémoriser qu'un mot de passe classique.
- Selon l'appareil, une clé privée stockée sur une carte à puce ne peut pas être exportée. Cela offre une sécurité supplémentaire.
- Les cartes à puce peuvent rendre la déconnexion automatique : l'IdM peut être configuré pour déconnecter les utilisateurs lorsqu'ils retirent leur carte à puce du lecteur.
- Le vol de la clé privée nécessite un accès physique à la carte à puce, ce qui rend les cartes à puce sûres contre les attaques de piratage.
- L'authentification par carte à puce est un exemple d'authentification à deux facteurs : elle nécessite à la fois quelque chose que vous possédez (la carte) et quelque chose que vous connaissez (le code PIN).
- Les cartes à puce sont plus souples que les mots de passe car elles fournissent des clés qui peuvent être utilisées à d'autres fins, comme le cryptage du courrier électronique.
- L'utilisation de cartes à puce sur des machines partagées qui sont des clients IdM ne pose généralement pas de problèmes de configuration supplémentaires aux administrateurs système. En fait, l'authentification par carte à puce est un choix idéal pour les machines partagées.

Les inconvénients de l'utilisation de certificats pour l'authentification des utilisateurs dans le cadre de l'IdM sont notamment les suivants :

- Les utilisateurs peuvent perdre ou oublier leur carte à puce ou leur certificat et se retrouver bloqués.
- Une erreur de saisie du code PIN à plusieurs reprises peut entraîner le blocage de la carte.
- Il y a généralement une étape intermédiaire entre la demande et l'autorisation par une sorte de responsable de la sécurité ou d'approbateur. Dans IdM, le responsable de la sécurité ou l'administrateur doit exécuter la commande **ipa cert-request**.
- Les cartes à puce et les lecteurs ont tendance à être spécifiques à un vendeur et à un pilote : bien qu'un grand nombre de lecteurs puissent être utilisés pour différentes cartes, une carte à puce d'un vendeur spécifique peut ne pas fonctionner dans le lecteur d'un autre vendeur ou dans un type de lecteur pour lequel elle n'a pas été conçue.
- Les certificats et les cartes à puce ont une courbe d'apprentissage abrupte pour les administrateurs.

CHAPITRE 2. GESTION DES CERTIFICATS POUR LES UTILISATEURS, LES HÔTES ET LES SERVICES À L'AIDE DE L'AUTORITÉ DE CERTIFICATION IDM INTÉGRÉE

Ce chapitre décrit comment gérer les certificats dans la gestion des identités (IdM) à l'aide de l'autorité de certification intégrée, de l'autorité de certification **ipa** et de ses sous-autorités de certification.

Ce chapitre contient les sections suivantes :

- Demande de nouveaux certificats pour un utilisateur, un hôte ou un service à l'aide de l'interface Web IdM.
- Demande de nouveaux certificats pour un utilisateur, un hôte ou un service auprès de l'autorité de certification IdM à l'aide de la CLI IdM :
  - Demande de nouveaux certificats pour un utilisateur, un hôte ou un service auprès de l'autorité de certification IdM à l'aide de certutil
    - Pour un exemple spécifique de demande d'un nouveau certificat d'utilisateur auprès de l'autorité de certification IdM à l'aide de l'utilitaire **certutil** et de son exportation vers un client IdM, voir Demande d'un nouveau certificat d'utilisateur et exportation vers le client.
  - Demande de nouveaux certificats pour un utilisateur, un hôte ou un service auprès de l'autorité de certification IdM à l'aide d'openssl

Vous pouvez également demander de nouveaux certificats pour un service à l'autorité de certification IdM à l'aide de l'utilitaire **certmonger**. Pour plus d'informations, voir Demande de nouveaux certificats pour un service auprès de l'autorité de certification IdM à l'aide de certmonger.

Conditions préalables

- Votre déploiement IdM contient une autorité de certification intégrée :
  - Pour plus d'informations sur la planification des services de l'autorité de certification dans IdM, voir Planification des services de l'autorité de certification .
  - Pour plus d'informations sur l'installation d'un serveur IdM avec DNS intégré et AC intégrée en tant qu'autorité de certification racine, voir Installation d'un serveur IdM : Avec DNS intégré, avec une autorité de certification intégrée comme autorité de certification racine
  - Pour plus d'informations sur l'installation d'un serveur IdM avec DNS intégré et une autorité de certification externe en tant qu'autorité de certification racine, voir Installation d'un serveur IdM : Avec DNS intégré, avec une autorité de certification externe comme autorité de certification racine
  - Pour plus d'informations sur l'installation d'un serveur IdM sans DNS intégré et avec une autorité de certification intégrée comme autorité de certification racine, voir Installation d'un serveur IdM : Sans DNS intégré, avec une autorité de certification intégrée comme autorité de certification racine.
  - [Facultatif] Votre déploiement IdM prend en charge les utilisateurs qui s'authentifient à l'aide d'un certificat :
    - Pour savoir comment configurer votre déploiement IdM pour prendre en charge l'authentification des utilisateurs à l'aide d'un certificat stocké dans le système de

fichiers du client IdM, voir Configuration de l'authentification à l'aide d'un certificat stocké sur le bureau d'un client IdM.

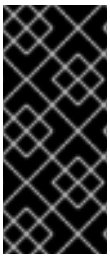
- Pour savoir comment configurer votre déploiement IdM pour prendre en charge l'authentification des utilisateurs à l'aide d'un certificat stocké sur une carte à puce insérée dans un client IdM, voir Configuration de la gestion des identités pour l'authentification par carte à puce.
- Pour savoir comment configurer votre déploiement IdM pour prendre en charge l'authentification des utilisateurs à l'aide de cartes à puce émises par un système de certificats Active Directory, voir Configuration des certificats émis par ADCS pour l'authentification par carte à puce dans IdM.

2.1. DEMANDE DE NOUVEAUX CERTIFICATS POUR UN UTILISATEUR, UN HÔTE OU UN SERVICE À L'AIDE DE L'INTERFACE WEB IDM

Cette section décrit comment utiliser l'interface Web de gestion des identités (IdM) pour demander un nouveau certificat pour n'importe quelle entité IdM auprès des autorités de certification (AC) IdM intégrées : l'AC **ipa** ou l'une de ses sous-AAC.

Les entités de l'IdM comprennent

- Utilisateurs
- Hosts
- Services



IMPORTANT

Les services sont généralement exécutés sur des nœuds de service dédiés sur lesquels les clés privées sont stockées. La copie de la clé privée d'un service sur le serveur IdM n'est pas considérée comme sûre. Par conséquent, lorsque vous demandez un certificat pour un service, créez la demande de signature de certificat (CSR) sur le nœud de service.

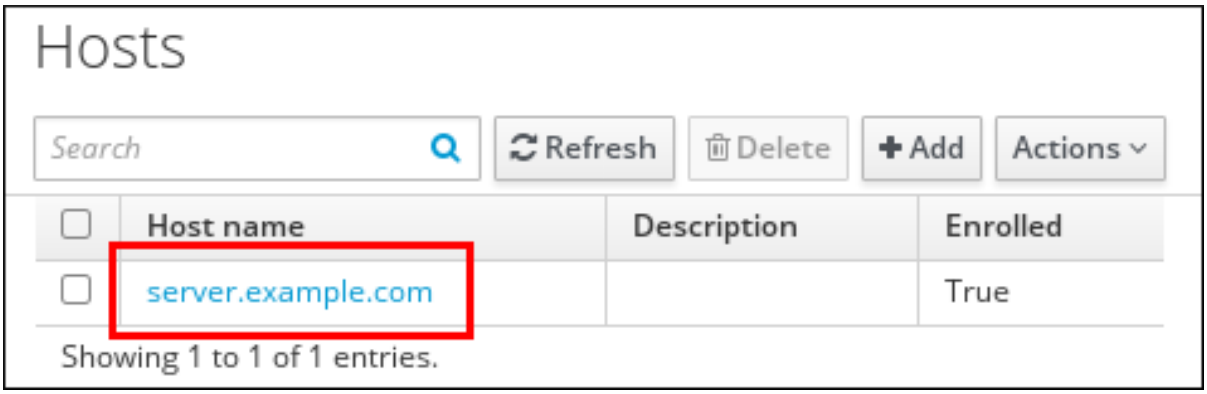
Conditions préalables

- Votre déploiement IdM contient une autorité de certification intégrée.
- Vous êtes connecté à l'interface Web IdM en tant qu'administrateur IdM.

Procédure

1. Sous l'onglet **Identity**, sélectionnez le sous-onglet **Users**, **Hosts** ou **Services**.
2. Cliquez sur le nom de l'utilisateur, de l'hôte ou du service pour ouvrir sa page de configuration.

Figure 2.1. Liste des hôtes



3. Cliquez sur **Actions → Nouveau certificat**.
4. Facultatif : Sélectionnez l'autorité de certification émettrice et l'ID de profil.
5. Suivez les instructions d'utilisation de l'utilitaire de ligne de commande (CLI) **certutil** à l'écran.
6. Cliquez sur **Issue**.

2.2. DEMANDE DE NOUVEAUX CERTIFICATS POUR UN UTILISATEUR, UN HÔTE OU UN SERVICE AUPRÈS DE L'AUTORITÉ DE CERTIFICATION IDM À L'AIDE DE CERTUTIL

Vous pouvez utiliser l'utilitaire **certutil** pour demander un certificat pour un utilisateur, un hôte ou un service de gestion d'identité (IdM) dans des situations IdM standard. Pour s'assurer qu'un alias Kerberos d'hôte ou de service peut utiliser un certificat, utilisez plutôt l'utilitaire openssl pour demander un certificat.

Cette section décrit comment demander un certificat pour un utilisateur, un hôte ou un service IdM à **ipa**, l'autorité de certification (AC) IdM, en utilisant **certutil**.

IMPORTANT

Les services sont généralement exécutés sur des nœuds de service dédiés sur lesquels les clés privées sont stockées. La copie de la clé privée d'un service sur le serveur IdM n'est pas considérée comme sûre. Par conséquent, lorsque vous demandez un certificat pour un service, créez la demande de signature de certificat (CSR) sur le nœud de service.

Conditions préalables

- Votre déploiement IdM contient une autorité de certification intégrée.
- Vous êtes connecté à l'interface de ligne de commande (CLI) d'IdM en tant qu'administrateur d'IdM.

Procédure

1. Créez un répertoire temporaire pour la base de données des certificats :

```
# mkdir ~/certdb/
```

2. Créez une nouvelle base de données temporaire de certificats, par exemple :

```
# certutil -N -d ~/certdb/
```

3. Créez la CSR et redirigez la sortie vers un fichier. Par exemple, pour créer une CSR pour un certificat de 4096 bits et définir l'objet à *CN=server.example.com,O=EXAMPLE.COM*:

```
# certutil -R -d ~/certdb/ -a -g 4096 -s \N-CN=server.example.com,O=EXAMPLE.COM\N-
server.example.com > certificate_request.csr
```

4. Soumettre le fichier de demande de certificat à l'autorité de certification fonctionnant sur le serveur IdM. Indiquer le principal Kerberos à associer au certificat nouvellement émis :

```
# ipa cert-request certificate_request.csr --principal=host/server.example.com
```

La commande **ipa cert-request** dans IdM utilise les valeurs par défaut suivantes :

- Le profil de certificat **caIPAserviceCert**
  Pour sélectionner un profil personnalisé, utilisez l'option **--profile-id**.
- L'autorité de certification racine de l'IdM intégrée, **ipa**
  Pour sélectionner un sous-CA, utilisez l'option **--ca**.

Ressources supplémentaires

- Voir le résultat de la commande **ipa cert-request --help**.
- Voir Création et gestion des profils de certificats dans la gestion des identités .

2.3. DEMANDE DE NOUVEAUX CERTIFICATS POUR UN UTILISATEUR, UN HÔTE OU UN SERVICE AUPRÈS DE L'AUTORITÉ DE CERTIFICATION IDM À L'AIDE D'OPENSSL

Vous pouvez utiliser l'utilitaire **openssl** pour demander un certificat pour un hôte ou un service de gestion d'identité (IdM) si vous voulez vous assurer que l'alias Kerberos de l'hôte ou du service peut utiliser le certificat. Dans les situations standard, il est préférable de demander un nouveau certificat à l'aide de l'utilitaire certutil.

Cette section décrit comment demander un certificat pour un hôte ou un service IdM à **ipa**, l'autorité de certification IdM, en utilisant **openssl**.

IMPORTANT

Les services sont généralement exécutés sur des nœuds de service dédiés sur lesquels les clés privées sont stockées. La copie de la clé privée d'un service sur le serveur IdM n'est pas considérée comme sûre. Par conséquent, lorsque vous demandez un certificat pour un service, créez la demande de signature de certificat (CSR) sur le nœud de service.

Conditions préalables

- Votre déploiement IdM contient une autorité de certification intégrée.

- Vous êtes connecté à l'interface de ligne de commande (CLI) d'IdM en tant qu'administrateur d'IdM.

Procédure

1. Créez un ou plusieurs alias pour votre principal Kerberos *test/server.example.com*. Par exemple, *test1/server.example.com* et *test2/server.example.com*.

2. Dans le CSR, ajoutez un subjectAltName pour dnsName (*server.example.com*) et otherName (*test2/server.example.com*). Pour ce faire, configurez le fichier **openssl.conf** pour qu'il contienne la ligne suivante spécifiant l'UPN otherName et subjectAltName :

```
otherName=1.3.6.1.4.1.311.20.2.3;UTF8:test2/server.example.com@EXAMPLE.COM
DNS.1 = server.example.com
```

3. Créez une demande de certificat à l'aide de **openssl**:

```
openssl req -new -newkey rsa :2048 -keyout test2service.key -sha256 -nodes -out
certificate_request.csr -config openssl.conf
```

4. Soumettre le fichier de demande de certificat à l'autorité de certification fonctionnant sur le serveur IdM. Indiquer le principal Kerberos à associer au certificat nouvellement émis :

```
# ipa cert-request certificate_request.csr --principal=host/server.example.com
```

La commande **ipa cert-request** dans IdM utilise les valeurs par défaut suivantes :

- Le profil de certificat **caIPAserviceCert**
  Pour sélectionner un profil personnalisé, utilisez l'option **--profile-id**.

- L'autorité de certification racine de l'IdM intégrée, **ipa**
  Pour sélectionner un sous-CA, utilisez l'option **--ca**.

Ressources supplémentaires

- Voir le résultat de la commande **ipa cert-request --help**.

- Voir Création et gestion des profils de certificats dans la gestion des identités .

2.4. RESSOURCES SUPPLÉMENTAIRES

- Voir Révoquer des certificats avec les AC IdM intégrées .

- Voir Restauration des certificats avec les autorités de certification IdM intégrées .

- Voir Restreindre une application à ne faire confiance qu'à un sous-ensemble de certificats .

CHAPITRE 3. GESTION DES CERTIFICATS SIGNÉS EN EXTERNE POUR LES UTILISATEURS, LES HÔTES ET LES SERVICES IDM

Ce chapitre explique comment utiliser l'interface de ligne de commande (CLI) de Identity Management (IdM) et l'interface Web IdM pour ajouter ou supprimer des certificats d'utilisateur, d'hôte ou de service émis par une autorité de certification (AC) externe.

3.1. AJOUT D'UN CERTIFICAT DÉLIVRÉ PAR UNE AUTORITÉ DE CERTIFICATION EXTERNE À UN UTILISATEUR, UN HÔTE OU UN SERVICE IDM À L'AIDE DE LA CLI IDM

En tant qu'administrateur de la gestion des identités (IdM), vous pouvez ajouter un certificat signé en externe au compte d'un utilisateur, d'un hôte ou d'un service IdM à l'aide de la CLI de la gestion des identités (IdM).

Conditions préalables

- Vous avez obtenu le ticket d'attribution de ticket d'un utilisateur administratif.

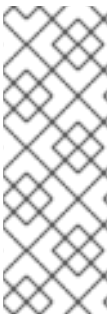
Procédure

- Pour ajouter un certificat à un utilisateur IdM, entrez :

```
$ ipa user-add-cert user --certificate=MIQTPrajQAwg...
```

  La commande vous demande de spécifier les informations suivantes :

  - Le nom de l'utilisateur
  - Le certificat DER encodé en Base64



NOTE

Au lieu de copier et de coller le contenu du certificat dans la ligne de commande, vous pouvez convertir le certificat au format DER, puis le réencoder en Base64. Par exemple, pour ajouter le certificat **user_cert.pem** à **user**, entrez :

```
$ ipa user-add-cert user --certificate="$(openssl x509 -outform der -in
user_cert.pem | base64 -w 0)"
```

Vous pouvez lancer la commande **ipa user-add-cert** de manière interactive en l'exécutant sans ajouter d'options.

Pour ajouter un certificat à un hôte IdM, entrez :

- **ipa host-add-cert**

Pour ajouter un certificat à un service IdM, entrez :

- **ipa service-add-cert**

Ressources supplémentaires

- Gestion des certificats pour les utilisateurs, les hôtes et les services à l'aide de l'autorité de certification IdM intégrée

3.2. AJOUT D'UN CERTIFICAT DÉLIVRÉ PAR UNE AUTORITÉ DE CERTIFICATION EXTERNE À UN UTILISATEUR, UN HÔTE OU UN SERVICE IDM À L'AIDE DE L'INTERFACE WEB IDM

En tant qu'administrateur de la gestion des identités (IdM), vous pouvez ajouter un certificat signé en externe au compte d'un utilisateur, d'un hôte ou d'un service IdM à l'aide de l'interface Web de la gestion des identités (IdM).

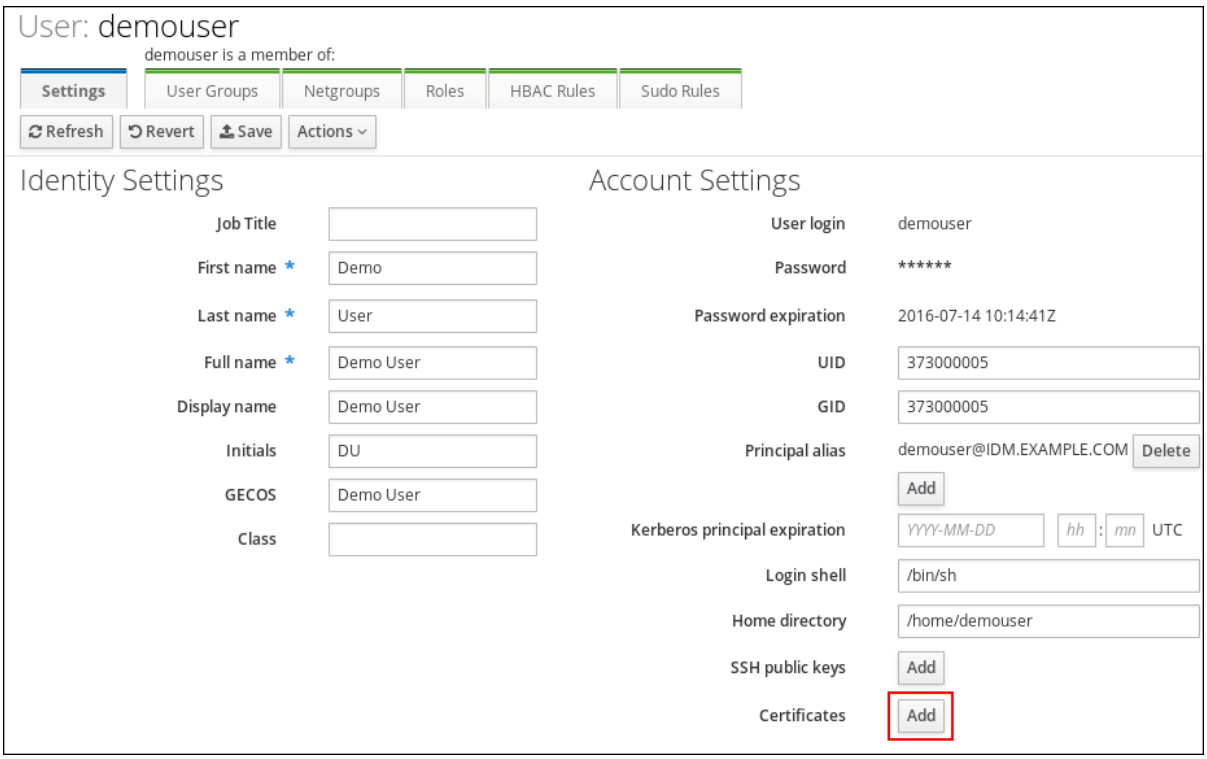
Conditions préalables

- Vous êtes connecté à l'interface Web de gestion des identités (IdM) en tant qu'utilisateur administratif.

Procédure

1. Ouvrez l'onglet **Identity** et sélectionnez le sous-onglet **Users**, **Hosts** ou **Services**.
2. Cliquez sur le nom de l'utilisateur, de l'hôte ou du service pour ouvrir sa page de configuration.
3. Cliquez sur **Ajouter à** côté de l'entrée **Certificates**.

Figure 3.1. Ajouter un certificat à un compte d'utilisateur



4. Collez le certificat au format Base64 ou PEM dans le champ de texte, puis cliquez sur **Ajouter**.
5. Cliquez sur **Sauvegarder** pour enregistrer les modifications.

3.3. SUPPRESSION D'UN CERTIFICAT ÉMIS PAR UNE AUTORITÉ DE CERTIFICATION EXTERNE À PARTIR D'UN COMPTE D'UTILISATEUR, D'HÔTE OU DE SERVICE IDM À L'AIDE DE LA CLI IDM

En tant qu'administrateur de la gestion des identités (IdM), vous pouvez supprimer un certificat signé en externe du compte d'un utilisateur, d'un hôte ou d'un service IdM à l'aide de la CLI de la gestion des identités (IdM).

Conditions préalables

- Vous avez obtenu le ticket d'attribution de ticket d'un utilisateur administratif.

Procédure

- Pour supprimer un certificat d'un utilisateur IdM, entrez :

```
$ ipa user-remove-cert user --certificate=MIQTPrajQAwg...
```

La commande vous demande de spécifier les informations suivantes :

  - Le nom de l'utilisateur
  - Le certificat DER encodé en Base64

NOTE

Au lieu de copier et de coller le contenu du certificat dans la ligne de commande, vous pouvez convertir le certificat au format DER, puis le réencoder en Base64. Par exemple, pour supprimer le certificat **user_cert.pem** de **user**, entrez :

```
$ ipa user-remove-cert user --certificate="$(openssl x509 -outform der -in
user_cert.pem | base64 -w 0)"
```

Vous pouvez lancer la commande **ipa user-remove-cert** de manière interactive en l'exécutant sans ajouter d'options.

Pour supprimer un certificat d'un hôte IdM, entrez :

- **ipa host-remove-cert**

Pour supprimer un certificat d'un service IdM, entrez :

- **ipa service-remove-cert**

Ressources supplémentaires

- Gestion des certificats pour les utilisateurs, les hôtes et les services à l'aide de l'autorité de certification IdM intégrée

3.4. SUPPRESSION D'UN CERTIFICAT ÉMIS PAR UNE AUTORITÉ DE CERTIFICATION EXTERNE D'UN COMPTE D'UTILISATEUR, D'HÔTE OU DE SERVICE IDM À L'AIDE DE L'INTERFACE WEB IDM

En tant qu'administrateur de la gestion des identités (IdM), vous pouvez supprimer un certificat signé en externe du compte d'un utilisateur, d'un hôte ou d'un service IdM à l'aide de l'interface Web de la gestion des identités (IdM).

Conditions préalables

- Vous êtes connecté à l'interface Web de gestion des identités (IdM) en tant qu'utilisateur administratif.

Procédure

1. Ouvrez l'onglet **Identity** et sélectionnez le sous-onglet **Users**, **Hosts** ou **Services**.
2. Cliquez sur le nom de l'utilisateur, de l'hôte ou du service pour ouvrir sa page de configuration.
3. Cliquez sur les **Actions** en regard du certificat à supprimer, puis sélectionnez **Supprimer**.
4. Cliquez sur **Sauvegarder** pour enregistrer les modifications.

3.5. RESSOURCES SUPPLÉMENTAIRES

- Assurer la présence d'un certificat signé en externe dans une entrée de service IdM à l'aide d'un playbook Ansible

CHAPITRE 4. CONVERTIR LES FORMATS DE CERTIFICATS POUR QU'ILS FONCTIONNENT AVEC L'IDM

Cette histoire d'utilisateur décrit comment s'assurer qu'en tant qu'administrateur du système IdM, vous utilisez le format correct d'un certificat avec des commandes IdM spécifiques. Ceci est utile, par exemple, dans les situations suivantes :

- Vous êtes en train de charger un certificat externe dans un profil d'utilisateur. Pour plus d'informations, voir Conversion d'un certificat externe à charger dans un compte utilisateur IdM .
- Vous utilisez un certificat d'autorité de certification externe lorsque vous configurez le serveur IdM pour l'authentification par carte à puce ou lorsque vous configurez le client IdM pour l'authentification par carte à puce afin que les utilisateurs puissent s'authentifier auprès d'IdM à l'aide de cartes à puce dotées de certificats émis par l'autorité de certification externe.
- Vous exportez un certificat d'une base de données NSS dans un format PKCS #12 qui comprend à la fois le certificat et la clé privée. Pour plus de détails, voir Exportation d'un certificat et d'une clé privée d'une base de données NSS vers un fichier PKCS #12.

4.1. FORMATS ET ENCODAGES DES CERTIFICATS DANS L'IDM

L'authentification par certificat, y compris l'authentification par carte à puce dans l'IdM, se fait en comparant le certificat présenté par l'utilisateur avec le certificat ou les données du certificat qui sont stockés dans le profil IdM de l'utilisateur.

Configuration du système

Ce qui est stocké dans le profil IdM est uniquement le certificat, et non la clé privée correspondante. Lors de l'authentification, l'utilisateur doit également montrer qu'il est en possession de la clé privée correspondante. Pour ce faire, l'utilisateur présente soit un fichier PKCS #12 contenant à la fois le certificat et la clé privée, soit deux fichiers : l'un contenant le certificat et l'autre la clé privée.

Par conséquent, les processus tels que le chargement d'un certificat dans un profil d'utilisateur n'acceptent que les fichiers de certificat qui ne contiennent pas la clé privée.

De même, lorsqu'un administrateur système vous fournit un certificat d'une autorité de certification externe, il ne fournit que les données publiques : le certificat sans la clé privée. L'utilitaire **ipa-advise** permettant de configurer le serveur IdM ou le client IdM pour l'authentification par carte à puce s'attend à ce que le fichier d'entrée contienne le certificat de l'autorité de certification externe, mais pas la clé privée.

Encodage des certificats

Il existe deux encodages de certificats courants : Privacy-enhanced Electronic Mail (**PEM**) et Distinguished Encoding Rules (**DER**). Le format **base64** est presque identique au format **PEM**, mais il ne contient pas l'en-tête et le pied de page **-----BEGIN CERTIFICATE-----/-----END CERTIFICATE-----**.

Un certificat qui a été encodé à l'aide de **DER** est un fichier de certificat numérique X509 binaire. En tant que fichier binaire, le certificat n'est pas lisible par l'homme. Les fichiers **DER** utilisent parfois l'extension de nom de fichier **.der**, mais les fichiers portant les extensions **.crt** et **.cer** contiennent aussi parfois des certificats **DER**. Les fichiers **DER** contenant des clés peuvent être nommés **.key**.

Un certificat qui a été encodé à l'aide de **PEM** Base64 est un fichier lisible par l'homme. Le fichier contient des données blindées ASCII (Base64) préfixées par une ligne "-----BEGIN ...". Les fichiers **PEM** utilisent parfois l'extension de nom de fichier **.pem**, mais les fichiers avec les extensions de nom de

fichier **.crt** et **.cer** contiennent aussi parfois des certificats **PEM**. Les fichiers **PEM** contenant des clés peuvent être nommés **.key**.

Les différentes commandes **ipa** ont des limitations différentes en ce qui concerne les types de certificats qu'elles acceptent. Par exemple, la commande **ipa user-add-cert** n'accepte que les certificats encodés au format **base64**, tandis que **ipa-server-certinstall** accepte les certificats **PEM, DER, PKCS #7, PKCS #8** et **PKCS #12**.

Tableau 4.1. Encodage des certificats

Format d'encodage	Lisible par l'homme	Extensions de nom de fichier courantes	Exemples de commandes IdM acceptant le format d'encodage
PEM/base64	Oui	.pem, .crt, .cer	ipa user-add-cert, ipa-server-certinstall, ..
DER	Non	.der, .crt, .cer	ipa-server-certinstall, ..

Les commandes et formats de certificats dans l'IdM répertorient d'autres commandes **ipa** avec les formats de certificats acceptés par les commandes.

Authentification de l'utilisateur

Lorsqu'il utilise l'interface web pour accéder à l'IdM, l'utilisateur prouve qu'il est en possession de la clé privée correspondant au certificat, les deux étant stockés dans la base de données du navigateur.

Lorsqu'il utilise le CLI pour accéder à IdM, l'utilisateur prouve qu'il est en possession de la clé privée correspondant au certificat par l'une des méthodes suivantes :

- L'utilisateur ajoute, comme valeur du paramètre **X509_user_identity** de la commande **kinit -X**, le chemin d'accès au module de carte à puce connecté à la carte à puce qui contient à la fois le certificat et la clé :

```
$ kinit -X X509_user_identity='PKCS11:opensc-pkcs11.so' idm_user
```

- L'utilisateur ajoute deux fichiers comme valeurs du paramètre **X509_user_identity** de la commande **kinit -X**, l'un contenant le certificat et l'autre la clé privée :

```
$ kinit -X X509_user_identity='FILE:`/path/to/cert.pem,/path/to/cert.key`' idm_user
```

Commandes de certificats utiles

Pour afficher les données du certificat, telles que le sujet et l'émetteur :

```
$ openssl x509 -noout -text -in ca.pem
```

Comparer en quoi deux certificats diffèrent :

```
$ diff cert1.crt cert2.crt
```

Pour comparer les lignes sur lesquelles deux certificats diffèrent, les résultats étant affichés sur deux colonnes :

```
$ diff cert1.crt cert2.crt -y
```

4.2. CONVERSION D'UN CERTIFICAT EXTERNE POUR LE CHARGER DANS UN COMPTE D'UTILISATEUR IDM

Cette section explique comment s'assurer qu'un certificat externe est correctement encodé et formaté avant de l'ajouter à une entrée utilisateur.

4.2.1. Conditions préalables

- Si votre certificat a été émis par une autorité de certification Active Directory et utilise le codage **PEM**, assurez-vous que le fichier **PEM** a été converti au format **UNIX**. Pour convertir un fichier, utilisez l'utilitaire **dos2unix** fourni par le paquetage éponyme.

4.2.2. Conversion d'un certificat externe dans le CLI IdM et chargement dans un compte utilisateur IdM

Le site **IdM CLI** n'accepte qu'un certificat **PEM** dont la première et la dernière ligne (-----BEGIN CERTIFICATE----- et -----END CERTIFICATE-----) ont été supprimées.

Suivez cette procédure pour convertir un certificat externe au format **PEM** et l'ajouter à un compte utilisateur IdM à l'aide de la CLI IdM.

Procédure

1. Convertir le certificat au format **PEM**:
   - Si votre certificat est au format **DER**:

     ```
     $ openssl x509 -in cert.crt -inform der -outform pem -out cert.pem
     ```

   - Si votre fichier est au format **PKCS #12**, dont les extensions courantes sont **.pfx** et **.p12**, et qu'il contient un certificat, une clé privée et éventuellement d'autres données, extrayez le certificat à l'aide de l'utilitaire **openssl pkcs12**. Lorsque vous y êtes invité, saisissez le mot de passe protégeant la clé privée stockée dans le fichier :

     ```
     $ openssl pkcs12 -in cert_and_key.p12 -clcerts -nokeys -out cert.pem
     Enter Import Password:
     ```

2. Obtenir les informations d'identification de l'administrateur :

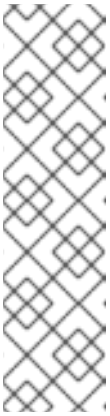
   ```
   $ kinit admin
   ```

3. Ajoutez le certificat au compte d'utilisateur à l'aide du site **IdM CLI** en suivant l'une des méthodes suivantes :
   - Supprimez la première et la dernière ligne (-----BEGIN CERTIFICATE----- et -----END CERTIFICATE-----) du fichier **PEM** à l'aide de l'utilitaire **sed** avant d'ajouter la chaîne à la commande **ipa user-add-cert**:

     ```
     $ ipa user-add-cert some_user --certificate="$(sed -e '/BEGIN
     CERTIFICATE/d;/END CERTIFICATE/d' cert.pem)"
     ```

- Copiez et collez le contenu du fichier de certificat sans les première et dernière lignes (-----BEGIN CERTIFICATE----- et -----END CERTIFICATE-----) dans la commande **ipa user-add-cert**:

```
$ ipa user-add-cert some_user --certificate=MIIDlzCCAn
gAwIBAgIBATANBgkqhki...
```



NOTE

Vous ne pouvez pas transmettre directement à la commande **ipa user-add-cert** un fichier **PEM** contenant le certificat, sans supprimer au préalable la première et la dernière ligne (-----BEGIN CERTIFICATE----- et -----END CERTIFICATE-----) :

```
$ ipa user-add-cert some_user --cert=some_user_cert.pem
```

Cette commande entraîne le message d'erreur "ipa : ERROR : Base64 decoding failed : Incorrect padding".

4. Optionnellement, vérifier si le certificat a été accepté par le système :

```
[idm_user@r8server]$ ipa user-show some_user
```

4.2.3. Conversion d'un certificat externe dans l'interface web IdM pour le charger dans un compte utilisateur IdM

Suivez cette procédure pour convertir un certificat externe au format **PEM** et l'ajouter à un compte utilisateur IdM dans l'interface web IdM.

Procédure

1. A l'aide de **CLI**, convertissez le certificat au format **PEM**:

   - Si votre certificat est au format **DER**:

```
$ openssl x509 -in cert.crt -inform der -outform pem -out cert.pem
```

   - Si votre fichier est au format **PKCS #12**, dont les extensions courantes sont **.pfx** et **.p12**, et qu'il contient un certificat, une clé privée et éventuellement d'autres données, extrayez le certificat à l'aide de l'utilitaire **openssl pkcs12**. Lorsque vous y êtes invité, saisissez le mot de passe protégeant la clé privée stockée dans le fichier :

```
$ openssl pkcs12 -in cert_and_key.p12 -clcerts -nokeys -out cert.pem
Enter Import Password:
```

2. Ouvrez le certificat dans un éditeur et copiez-en le contenu. Vous pouvez inclure les lignes d'en-tête et de pied de page "-----BEGIN CERTIFICATE-----" et "-----END CERTIFICATE-----", mais ce n'est pas nécessaire, car les formats **PEM** et **base64** sont tous deux acceptés par l'interface utilisateur Web de l'IdM.

3. Dans l'interface web IdM, connectez-vous en tant que responsable de la sécurité.

4. Allez à **Identity → Users → some_user**.

5. Cliquez sur **Add** à côté de **Certificates**.
6. Collez le contenu du certificat au format PEM dans la fenêtre qui s'ouvre.
7. Cliquez sur **Add**.

Si le certificat a été accepté par le système, il figure parmi les **Certificates** dans le profil de l'utilisateur.

4.3. PRÉPARATION DU CHARGEMENT D'UN CERTIFICAT DANS LE NAVIGATEUR

Avant d'importer un certificat d'utilisateur dans le navigateur, assurez-vous que le certificat et la clé privée correspondante sont au format **PKCS #12**. Deux situations courantes nécessitent des préparatifs supplémentaires :

- Le certificat se trouve dans une base de données NSS. Pour savoir comment procéder dans ce cas, voir Exportation d'un certificat et d'une clé privée d'une base de données NSS vers un fichier PKCS #12.
- Le certificat et la clé privée se trouvent dans deux fichiers **PEM** distincts. Pour savoir comment procéder dans cette situation, voir Combiner les fichiers PEM de certificat et de clé privée en un fichier PKCS #12.

Ensuite, pour importer le certificat de l'autorité de certification au format **PEM** et le certificat de l'utilisateur au format **PKCS #12** dans le navigateur, suivez les procédures de Configuration d'un navigateur pour activer l'authentification par certificat et Authentification à l'interface utilisateur Web de gestion d'identité avec un certificat en tant qu'utilisateur de gestion d'identité.

4.3.1. Exportation d'un certificat et d'une clé privée d'une base de données NSS dans un fichier PKCS #12

Procédure

1. Utilisez la commande **pk12util** pour exporter le certificat de la base de données NSS au format **PKCS12**. Par exemple, pour exporter le certificat avec le pseudonyme **some_user** de la base de données NSS stockée dans le répertoire **~/certdb** vers le fichier **~/some_user.p12**:

```
$ pk12util -d ~/certdb -o ~/some_user.p12 -n some_user
Enter Password or Pin for "NSS Certificate DB":
Enter password for PKCS12 file:
Re-enter password:
pk12util: PKCS12 EXPORT SUCCESSFUL
```

2. Définissez les autorisations appropriées pour le fichier **.p12**:

```
# chmod 600 ~/some_user.p12
```

Comme le fichier **PKCS #12** contient également la clé privée, il doit être protégé pour éviter que d'autres utilisateurs ne l'utilisent. Sinon, ils pourraient usurper l'identité de l'utilisateur.

4.3.2. Combinaison des fichiers PEM de certificats et de clés privées en un fichier PKCS #12

Cette section décrit comment combiner un certificat et la clé correspondante stockés dans des fichiers **PEM** distincts dans un fichier **PKCS #12**.

Procédure

- Pour combiner un certificat stocké dans **certfile.cer** et une clé stockée dans **certfile.key** dans un fichier **certfile.p12** qui contient à la fois le certificat et la clé :

```
$ openssl pkcs12 -export -in certfile.cer -inkey certfile.key -out certfile.p12
```

4.4. COMMANDES ET FORMATS LIÉS AUX CERTIFICATS DANS L'IDM

Le tableau des commandes et formats de certificats IdM présente les commandes liées aux certificats dans IdM avec les formats acceptables.

Tableau 4.2. Commandes et formats des certificats IdM

Commandement	Formats acceptables	Notes
ipa user-add-cert some_user --certificate	certificat PEM base64	
ipa-server-certinstall	Certificat PEM et DER ; chaîne de certificats PKCS#7 ; PKCS#8 et clé privée brute ; certificat et clé privée PKCS#12	
ipa-cacert-manage install	DER ; PEM ; PKCS#7	
ipa-cacert-manage renew --external-cert-file	Certificat PEM et DER ; chaîne de certificats PKCS#7	
ipa-ca-install --external-cert-file	Certificat PEM et DER ; chaîne de certificats PKCS#7	
ipa cert-show <cert serial> --certificate-out /path/to/file.pem	N/A	Crée le fichier **file.pem** codé en PEM avec le certificat portant le numéro de série **<cert_serial>**.
ipa cert-show <cert serial> --certificate-out /path/to/file.pem	N/A	Crée le fichier **file.pem** codé en PEM avec le certificat portant le numéro de série **<cert_serial>**. Si l'option **--chain** est utilisée, le fichier PEM contient le certificat, y compris la chaîne de certificats.
ipa cert-request --certificate-out=FILE /path/to/req.csr	N/A	Crée le fichier **req.csr** au format PEM avec le nouveau certificat.

Commandement	Formats acceptables	Notes
ipa cert-request --certificate-out=FILE /path/to/req.csr	N/A	Crée le fichier **req.csr** au format PEM avec le nouveau certificat. Si l'option **--chain** est utilisée, le fichier PEM contient le certificat, y compris la chaîne de certificats.

CHAPITRE 5. CRÉATION ET GESTION DE PROFILS DE CERTIFICATS DANS LA GESTION DES IDENTITÉS

Les profils de certificats sont utilisés par l'autorité de certification (AC) lors de la signature des certificats pour déterminer si une demande de signature de certificat (CSR) est acceptable et, le cas échéant, quelles sont les caractéristiques et les extensions présentes sur le certificat. Un profil de certificat est associé à l'émission d'un type particulier de certificat. En combinant les profils de certificats et les listes de contrôle d'accès (ACL) de l'autorité de certification, vous pouvez définir et contrôler l'accès aux profils de certificats personnalisés.

Les procédures décrivant la création de profils de certificats utilisent les certificats S/MIME à titre d'exemple. Certains programmes de messagerie électronique prennent en charge le courrier électronique signé et crypté numériquement à l'aide du protocole Secure Multipurpose Internet Mail Extension (S/MIME). L'utilisation de S/MIME pour signer ou crypter des messages électroniques exige que l'expéditeur du message dispose d'un certificat S/MIME.

5.1. QU'EST-CE QU'UN PROFIL DE CERTIFICAT ?

Vous pouvez utiliser les profils de certificat pour déterminer le contenu des certificats, ainsi que les contraintes liées à l'émission des certificats, telles que les suivantes :

- Algorithme de signature à utiliser pour chiffrer la demande de signature du certificat.
- La validité par défaut du certificat.
- Les motifs de révocation qui peuvent être utilisés pour révoquer un certificat.
- Si le nom commun du mandant est copié dans le champ du nom alternatif du sujet.
- Les caractéristiques et les extensions qui doivent figurer sur le certificat.

Un profil de certificat unique est associé à l'émission d'un type particulier de certificat. Vous pouvez définir différents profils de certificats pour les utilisateurs, les services et les hôtes dans l'IdM. L'IdM inclut par défaut les profils de certificats suivants :

- **caIPAserviceCert**
- **IECUserRoles**
- **KDCs_PKINIT_Certs** (utilisé en interne)

En outre, vous pouvez créer et importer des profils personnalisés, qui vous permettent d'émettre des certificats à des fins spécifiques. Par exemple, vous pouvez limiter l'utilisation d'un profil particulier à un seul utilisateur ou à un seul groupe, empêchant ainsi les autres utilisateurs et groupes d'utiliser ce profil pour émettre un certificat à des fins d'authentification. Pour créer des profils de certificats personnalisés, utilisez la commande **ipa certprofile**.

Ressources supplémentaires

- Voir la commande **ipa help certprofile**.

5.2. CRÉATION D'UN PROFIL DE CERTIFICAT

Cette procédure décrit comment créer un profil de certificat via la ligne de commande en créant un fichier de configuration de profil pour demander des certificats S/MIME.

Procédure

1. Créer un profil personnalisé en copiant un profil par défaut existant :

```
$ ipa certprofile-show --out smime.cfg caIPAserviceCert
-----------------------------------------------
Profile configuration stored in file 'smime.cfg'
-----------------------------------------------
  Profile ID: caIPAserviceCert
  Profile description: Standard profile for network services
  Store issued certificates: TRUE
```

2. Ouvrez le fichier de configuration du profil nouvellement créé dans un éditeur de texte.

```
$ vi  smime.cfg
```

3. Remplacez **Profile ID** par un nom qui reflète l'utilisation du profil, par exemple **smime**.

NOTE

Lorsque vous importez un profil nouvellement créé, le champ **profileId**, s'il est présent, doit correspondre à l'identifiant spécifié dans la ligne de commande.

4. Mettez à jour la configuration de l'extension Extended Key Usage. La configuration par défaut de l'extension Extended Key Usage concerne l'authentification du serveur et du client TLS. Par exemple, pour S/MIME, l'Extended Key Usage doit être configuré pour la protection du courrier électronique :

```
policyset.serverCertSet.7.default.params.exKeyUsageOIDs=1.3.6.1.5.5.7.3.4
```

5. Importer le nouveau profil :

```
$ ipa certprofile-import smime --file smime.cfg \
  --desc "S/MIME certificates" --store TRUE

-----------------------
Imported profile "smime"
-----------------------
```

```
  Profile ID: smime
  Profile description: S/MIME certificates
  Store issued certificates: TRUE
```

Verification steps

- Vérifiez que le nouveau profil de certificat a été importé :

```
$ ipa certprofile-find

-----------------
4 profiles matched
-----------------
  Profile ID: caIPAserviceCert
  Profile description: Standard profile for network services
  Store issued certificates: TRUE

  Profile ID: IECUserRoles
  Profile description: User profile that includes IECUserRoles extension from request
  Store issued certificates: TRUE

  Profile ID: KDCs_PKINIT_Certs
  Profile description: Profile for PKINIT support by KDCs
  Store issued certificates: TRUE

  Profile ID: smime
  Profile description: S/MIME certificates
  Store issued certificates: TRUE
----------------------------
Number of entries returned 4
----------------------------
```

Ressources supplémentaires

- Voir **ipa help certprofile**.
- Voir RFC 5280, section 4.2.1.12 .

5.3. QU'EST-CE QU'UNE LISTE DE CONTRÔLE D'ACCÈS CA ?

Les règles de la liste de contrôle d'accès de l'autorité de certification (CA ACL) définissent quels profils peuvent être utilisés pour délivrer des certificats à quels mandants. Vous pouvez utiliser les listes de contrôle d'accès de l'autorité de certification à cette fin, par exemple :

- Déterminer quel utilisateur, hôte ou service peut recevoir un certificat avec un profil particulier
- Déterminer quelle autorité de certification de l'IdM ou quelle sous autorité de certification est autorisée à délivrer le certificat

Par exemple, à l'aide d'ACL, vous pouvez limiter l'utilisation d'un profil destiné aux employés travaillant dans un bureau situé à Londres aux seuls utilisateurs membres du groupe d'utilisateurs IdM lié au bureau de Londres.

L'utilitaire **ipa caacl** pour la gestion des règles CA ACL permet aux utilisateurs privilégiés d'ajouter, d'afficher, de modifier ou de supprimer une CA ACL spécifiée.

Ressources supplémentaires

- Voir **ipa help caacl**.

5.4. DÉFINITION D'UNE CA ACL POUR CONTRÔLER L'ACCÈS AUX PROFILS DE CERTIFICATS

Cette procédure décrit comment utiliser l'utilitaire **caacl** pour définir une règle de liste de contrôle d'accès (ACL) à l'autorité de certification afin d'autoriser les utilisateurs d'un groupe à accéder à un profil de certificat personnalisé. Dans ce cas, la procédure décrit comment créer un groupe d'utilisateurs S/MIME et une liste de contrôle d'accès à l'autorité de certification pour permettre aux utilisateurs de ce groupe d'accéder au modèle de certificat **smime**.

Conditions préalables

- Assurez-vous d'avoir obtenu les informations d'identification de l'administrateur IdM.

Procédure

1. Créez un nouveau groupe pour les utilisateurs du profil de certificat :

```
$ ipa group-add smime_users_group
--------------------------------
Added group "smime users group"
--------------------------------
  Group name: smime_users_group
  GID: 75400001
```

2. Créez un nouvel utilisateur à ajouter au groupe **smime_user_group**:

```
$ ipa user-add smime_user
First name: smime
Last name: user
----------------------
Added user "smime_user"
----------------------
  User login: smime_user
  First name: smime
  Last name: user
  Full name: smime user
  Display name: smime user
  Initials: TU
  Home directory: /home/smime_user
  GECOS: smime user
  Login shell: /bin/sh
  Principal name: smime_user@IDM.EXAMPLE.COM
  Principal alias: smime_user@IDM.EXAMPLE.COM
  Email address: smime_user@idm.example.com
  UID: 1505000004
  GID: 1505000004
  Password: False
  Member of groups: ipausers
  Kerberos keys available: False
```

3. Ajouter le **smime_user** au groupe **smime_users_group**:

```
$ ipa group-add-member smime_users_group --users=smime_user
  Group name: smime_users_group
  GID: 1505000003
  Member users: smime_user
-------------------------
Number of members added 1
-------------------------
```

4. Créez l'ACL CA pour permettre aux utilisateurs du groupe d'accéder au profil de certificat :

```
$ ipa caacl-add smime_acl
-------------------------
Added CA ACL "smime_acl"
-------------------------
  ACL name: smime_acl
  Enabled: TRUE
```

5. Ajoutez le groupe d'utilisateurs à la CAL :

```
$ ipa caacl-add-user smime_acl --group smime_users_group
  ACL name: smime_acl
  Enabled: TRUE
  User Groups: smime_users_group
-------------------------
Number of members added 1
-------------------------
```

6. Ajoutez le profil de certificat à la liste de contrôle de l'autorité de certification :

```
$ ipa caacl-add-profile smime_acl --certprofile smime
  ACL name: smime_acl
  Enabled: TRUE
  Profiles: smime
  User Groups: smime_users_group
-------------------------
Number of members added 1
-------------------------
```

Verification steps

- Affichez les détails de l'ACL CA que vous avez créée :

```
$ ipa caacl-show smime_acl
  ACL name: smime_acl
  Enabled: TRUE
  Profiles: smime
  User Groups: smime_users_group
...
```

Ressources supplémentaires

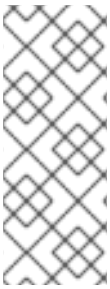
- Voir la page de manuel **ipa**.
- Voir **ipa help caacl**.

5.5. UTILISATION DES PROFILS DE CERTIFICATS ET DES LISTES DE CONTRÔLE DE L'AUTORITÉ DE CERTIFICATION POUR ÉMETTRE DES CERTIFICATS

Vous pouvez demander des certificats à l'aide d'un profil de certificat lorsque les listes de contrôle d'accès (CA ACL) de l'autorité de certification l'autorisent. Cette procédure décrit comment demander un certificat S/MIME pour un utilisateur utilisant un profil de certificat personnalisé auquel l'accès a été accordé via une liste de contrôle d'accès de l'autorité de certification.

Conditions préalables

- Votre profil de certificat a été créé.
- Une liste de contrôle CA a été créée pour permettre à l'utilisateur d'utiliser le profil de certificat requis pour demander un certificat.



NOTE

Vous pouvez contourner la vérification de la CAL si l'utilisateur qui exécute la commande **cert-request**:

- Est l'utilisateur de **admin**.
- A l'autorisation de **Request Certificate ignoring CA ACLs**.

Procédure

1. Générer une demande de certificat pour l'utilisateur. Par exemple, en utilisant OpenSSL :

```
$ openssl req -new -newkey rsa:2048 -days 365 -nodes -keyout private.key -out cert.csr -subj '/CN=smime_user'
```

2. Demander un nouveau certificat pour l'utilisateur à l'autorité de certification IdM :

```
$ ipa cert-request cert.csr --principal=smime_user --profile-id=smime
```

L'option --ca *sub-CA_name* peut être ajoutée à la commande pour demander le certificat à une autorité de certification secondaire plutôt qu'à l'autorité de certification racine.

Verification steps

- Vérifiez que le certificat nouvellement émis est attribué à l'utilisateur :

```
$ ipa user-show user
  User login: user
  ...
  Certificate: MIICfzCCAWcCAQA...
  ...
```

Ressources supplémentaires

- Voir la page de manuel **ipa(a)**.
- Voir la commande **ipa help user-show**.
- Voir la commande **ipa help cert-request**.
- Voir la page de manuel **openssl(lssl)**.

5.6. MODIFICATION D'UN PROFIL DE CERTIFICAT

Cette procédure décrit comment modifier les profils de certificats directement via la ligne de commande à l'aide de la commande **ipa certprofile-mod**.

Procédure

1. Déterminez l'identifiant du profil de certificat que vous modifiez. Pour afficher tous les modèles de certificats actuellement stockés dans l'IdM :

```
# ipa certprofile-find

------------------
4 profiles matched
------------------
  Profile ID: caIPAserviceCert
  Profile description: Standard profile for network services
  Store issued certificates: TRUE

  Profile ID: IECUserRoles
  ...

  Profile ID: smime
  Profile description: S/MIME certificates
  Store issued certificates: TRUE
----------------------------
Number of entries returned
----------------------------
```

2. Modifiez la description du profil de certificat. Par exemple, si vous avez créé un modèle de certificat personnalisé pour les certificats S/MIME en utilisant un modèle existant, modifiez la description en fonction de la nouvelle utilisation :

```
# ipa certprofile-mod smime --desc "New certificate profile description"
-----------------------------------
Modified Certificate Profile "smime"
-----------------------------------
   Profile ID: smime
   Profile description: New certificate profile description
   Store issued certificates: TRUE
```

3. Ouvrez votre fichier de profil de certificat client dans un éditeur de texte et modifiez-le en fonction de vos besoins :

```
# vi smime.cfg
```

-

Pour plus de détails sur les options qui peuvent être configurées dans le fichier de configuration du profil de certificat, voir Paramètres de configuration du profil de certificat .

4. Mettre à jour le fichier de configuration du profil de certificat existant :

```
# ipa certprofile-mod _profile_ID_ --file=smime.cfg
```

Verification steps

- Vérifiez que le profil de certificat a été mis à jour :

```
$ ipa certprofile-show smime
  Profile ID: smime
  Profile description: New certificate profile description
  Store issued certificates: TRUE
```

Ressources supplémentaires

- Voir la page de manuel **ipa(a)**.
- Voir **ipa help certprofile-mod**.

5.7. PARAMÈTRES DE CONFIGURATION DU PROFIL DE CERTIFICAT

Les paramètres de configuration du profil de certificat sont stockés dans un fichier *profile_name*.cfg dans le répertoire du profil de l'autorité de certification, **/var/lib/pki/pki-tomcat/ca/profiles/ca**. Tous les paramètres d'un profil - valeurs par défaut, entrées, sorties et contraintes - sont configurés dans un seul ensemble de règles. Un ensemble de règles pour un profil de certificat porte le nom suivant **policyset.*policyName.policyNumber*.** Par exemple, pour l'ensemble de règles **serverCertSet**:

```
policyset.list=serverCertSet
policyset.serverCertSet.list=1,2,3,4,5,6,7,8
policyset.serverCertSet.1.constraint.class_id=subjectNameConstraintImpl
policyset.serverCertSet.1.constraint.name=Subject Name Constraint
policyset.serverCertSet.1.constraint.params.pattern=CN=[^,]+,.+
policyset.serverCertSet.1.constraint.params.accept=true
policyset.serverCertSet.1.default.class_id=subjectNameDefaultImpl
policyset.serverCertSet.1.default.name=Subject Name Default
policyset.serverCertSet.1.default.params.name=CN=$request.req_subject_name.cn$, OU=pki-ipa,
O=IPA
policyset.serverCertSet.2.constraint.class_id=validityConstraintImpl
policyset.serverCertSet.2.constraint.name=Validity Constraint
policyset.serverCertSet.2.constraint.params.range=740
policyset.serverCertSet.2.constraint.params.notBeforeCheck=false
policyset.serverCertSet.2.constraint.params.notAfterCheck=false
policyset.serverCertSet.2.default.class_id=validityDefaultImpl
policyset.serverCertSet.2.default.name=Validity Default
policyset.serverCertSet.2.default.params.range=731
policyset.serverCertSet.2.default.params.startTime=0
```

Chaque ensemble de règles contient une liste de règles configurées pour le profil de certificat par numéro d'identification de la règle dans l'ordre dans lequel elles doivent être évaluées. Le serveur évalue chaque ensemble de règles pour chaque demande qu'il reçoit. Lorsqu'une seule demande de

certificat est reçue, un seul ensemble est évalué et tous les autres ensembles du profil sont ignorés. Lorsque des paires de clés doubles sont émises, le premier ensemble de règles est évalué pour la première demande de certificat, et le deuxième ensemble est évalué pour la deuxième demande de certificat. Il n'est pas nécessaire de disposer de plus d'un ensemble de règles lors de l'émission de certificats individuels ou de plus de deux ensembles lors de l'émission de paires de clés.

Tableau 5.1. Paramètres du fichier de configuration du profil de certificat

Paramètres	Description
desc	Une description en texte libre du profil de certificat, qui est affichée sur la page des entités finales. Par exemple, **desc=This certificate profile is for enrolling server certificates with agent authentication**.
permettre	Active le profil pour qu'il soit accessible via la page des entités finales. Par exemple, **enable=true**.
auth.instance_id	Définit le plug-in du gestionnaire d'authentification à utiliser pour authentifier la demande de certificat. Dans le cas d'une inscription automatique, l'autorité de certification délivre un certificat immédiatement si l'authentification est réussie. Si l'authentification échoue ou si aucun plug-in d'authentification n'est spécifié, la demande est mise en file d'attente pour être approuvée manuellement par un agent. Par exemple, **auth.instance_id=AgentCertAuth**.
authz.acl	Spécifie la contrainte d'autorisation. Ce paramètre est principalement utilisé pour définir la liste de contrôle d'accès (ACL) de l'évaluation du groupe. Par exemple, le paramètre **caCMCUserCert** exige que le signataire de la requête CMC appartienne au groupe des agents du gestionnaire de certificats : **authz.acl=group="Certificate Manager Agents** Dans le cadre du renouvellement des certificats d'utilisateur basé sur un annuaire, cette option est utilisée pour s'assurer que le demandeur initial et l'utilisateur actuellement authentifié sont les mêmes. Une entité doit s'authentifier (se lier ou, essentiellement, se connecter au système) avant que l'autorisation puisse être évaluée.
nom	Le nom du profil de certificat. Par exemple, **name=Agent-Authenticated Server Certificate Enrollment**. Ce nom est affiché sur la page d'inscription ou de renouvellement de l'utilisateur final.

Paramètres	Description
liste.d'entrée	Liste les entrées autorisées pour le profil de certificat par nom. Par exemple, **input.list=i1,i2**.
input.input_id.class_id	Indique le nom de la classe java pour l'entrée par ID d'entrée (le nom de l'entrée répertoriée dans input.list). Par exemple, **input.i1.class_id=certReqInputImpl**.
liste.de.sortie	Liste les formats de sortie possibles pour le profil de certificat par nom. Par exemple, **output.list=o1**.
output.output_id.class_id	Spécifie le nom de la classe Java pour le format de sortie indiqué dans output.list. Par exemple, **output.o1.class_id=certOutputImpl**.
policyset.list	Liste les règles de profil de certificat configurées. Pour les certificats doubles, un ensemble de règles s'applique à la clé de signature et l'autre à la clé de chiffrement. Les certificats simples n'utilisent qu'un seul ensemble de règles de profil de certificat. Par exemple, **policyset.list=serverCertSet**.
policyset.policyset_id.list	Répertorie les politiques de l'ensemble de politiques configuré pour le profil de certificat par numéro d'identification de politique dans l'ordre dans lequel elles doivent être évaluées. Par exemple, **policyset.serverCertSet.list=1,2,3,4,5,6,7,8**.
policyset.policyset_id.policy_number.constraint.class_id	Indique le nom de la classe java du plug-in de contrainte défini pour la valeur par défaut configurée dans la règle de profil. Par exemple, policyset.serverCertSet.1.constraint.class_id=subjectNameConstraintImpl.
policyset.policyset_id.policy_number.constraint.name	Indique le nom de la contrainte défini par l'utilisateur. Par exemple, policyset.serverCertSet.1.constraint.name=Contrainte de nom de sujet.
policyset.policyset_id.policy_number.constraint.params.attribute	Spécifie une valeur pour un attribut autorisé pour la contrainte. Les attributs possibles varient en fonction du type de contrainte. Par exemple, policyset.serverCertSet.1.constraint.params.pattern=CN=.*.

Paramètres	Description
policyset.policyset_id.policy_number.default.class_id	Indique le nom de la classe Java pour l'ensemble de valeurs par défaut dans la règle de profil. Par exemple, policyset.serverCertSet.1.default.class_id=userSubjectNameDefaultImpl
policyset.policyset_id.policy_number.default.name	Indique le nom défini par l'utilisateur pour la valeur par défaut. Par exemple, policyset.serverCertSet.1.default.name=Nom du sujet Défaut
policyset.policyset_id.policy_number.default.params.attribute	Spécifie une valeur pour un attribut autorisé pour la valeur par défaut. Les attributs possibles varient en fonction du type de défaut. Par exemple, policyset.serverCertSet.1.default.params.name=CN=(Name)$request.requestor_name$.

CHAPITRE 6. GESTION DE LA VALIDITÉ DES CERTIFICATS DANS IDM

Dans le cadre de la gestion des identités (IdM), vous pouvez gérer la validité des certificats existants et des certificats que vous souhaitez émettre à l'avenir, mais les méthodes sont différentes.

6.1. GESTION DE LA VALIDITÉ D'UN CERTIFICAT EXISTANT QUI A ÉTÉ DÉLIVRÉ PAR L'AC IDM

Dans IdM, les méthodes suivantes permettent d'afficher la date d'expiration d'un certificat :

- Visualisation de la date d'expiration dans l'IdM WebUI .
- Visualisation de la date d'expiration dans le CLI .

Vous pouvez gérer la validité d'un certificat existant qui a été délivré par l'autorité de certification IdM de la manière suivante :

- Renouveler un certificat en demandant un nouveau certificat à l'aide de la demande de signature de certificat (CSR) d'origine ou d'une nouvelle CSR générée à partir de la clé privée. Vous pouvez demander un nouveau certificat à l'aide des utilitaires suivants :

  marchand de cerises

  Vous pouvez utiliser **certmonger** pour demander un certificat de service. Avant l'expiration du certificat, **certmonger** renouvellera automatiquement le certificat, garantissant ainsi la validité continue du certificat de service. Pour plus de détails, voir Obtention d'un certificat IdM pour un service à l'aide de certmonger;

  certutil

  Vous pouvez utiliser **certutil** pour renouveler les certificats d'utilisateur, d'hôte et de service. Pour plus d'informations sur la demande d'un certificat d'utilisateur, voir Demande d'un nouveau certificat d'utilisateur et exportation vers le client;

  openssl

  Vous pouvez utiliser **openssl** pour renouveler les certificats d'utilisateur, d'hôte et de service.

- Révoquer un certificat. Pour plus de détails, voir :
  - Révoquer des certificats avec les autorités de certification IdM intégrées à l'aide de l'interface Web IdM;
  - Révoquer des certificats avec les autorités de certification IdM intégrées en utilisant IdM CLI;
- Rétablir un certificat s'il a été temporairement révoqué. Pour plus de détails, voir :
  - Restauration des certificats avec les AC IdM intégrées à l'aide de l'IdM WebUI ;
  - Restauration des certificats avec les autorités de certification IdM intégrées à l'aide de IdM CLI.

6.2. GESTION DE LA VALIDITÉ DES FUTURS CERTIFICATS ÉMIS PAR L'AC IDM

Pour gérer la validité des futurs certificats émis par l'AC IdM, modifiez, importez ou créez un profil de certificat. Pour plus d'informations, voir Création et gestion des profils de certificats dans la gestion des identités.

6.3. VISUALISATION DE LA DATE D'EXPIRATION D'UN CERTIFICAT DANS L'IDM WEBUI

Vous pouvez utiliser l'interface Web IdM pour afficher la date d'expiration de tous les certificats qui ont été émis par l'autorité de certification IdM.

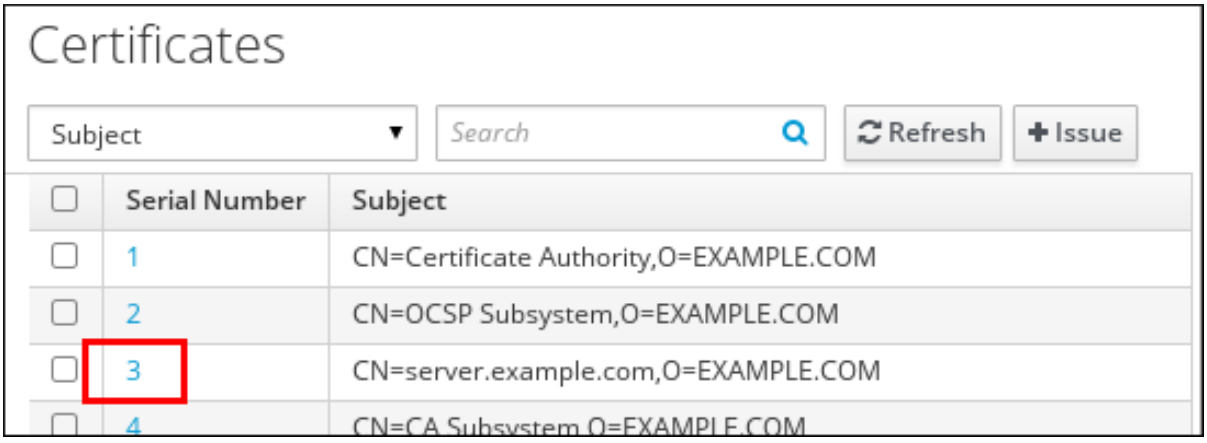
Conditions préalables

- Assurez-vous d'avoir obtenu les informations d'identification de l'administrateur.

Procédure

1. Dans le menu **Authentication**, cliquez sur **Certificates** > **Certificates**.
2. Cliquez sur le numéro de série du certificat pour ouvrir la page d'information sur le certificat.

   Figure 6.1. Liste des certificats



3. Dans la page d'information sur le certificat, recherchez les informations sur **Expires On**.

6.4. AFFICHAGE DE LA DATE D'EXPIRATION D'UN CERTIFICAT DANS LE CLI

Vous pouvez utiliser l'interface de ligne de commande (CLI) pour afficher la date d'expiration d'un certificat.

Procédure

- Utilisez l'utilitaire **openssl** pour ouvrir le fichier dans un format lisible par l'homme :

```
$ openssl x509 -noout -text -in ca.pem
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: O = IDM.EXAMPLE.COM, CN = Certificate Authority
```

```
Validity
    Not Before: Oct 30 19:39:14 2017 GMT
    Not After : Oct 30 19:39:14 2037 GMT
```

6.5. RÉVOQUER DES CERTIFICATS AVEC LES AC IDM INTÉGRÉES

6.5.1. Raisons de la révocation du certificat

Un certificat révoqué est invalide et ne peut être utilisé pour l'authentification. Toutes les révocations sont permanentes, sauf pour la raison 6 : **Certificate Hold**.

Le motif de révocation par défaut est 0 : **unspecified**.

Tableau 6.1. Motifs de révocation

ID	Raison	Explication
0	Non spécifié	
1	Clé compromise	La clé qui a émis le certificat n'est plus fiable. Causes possibles : perte du jeton, accès incorrect au fichier.
2	CA compromis	L'autorité de certification qui a émis le certificat n'est plus fiable.
3	Affiliation modifiée	Causes possibles : * Une personne a quitté l'entreprise ou a été affectée à un autre service. * Un hôte ou un service est en cours de retrait.
4	Remplacé	Un nouveau certificat a remplacé le certificat actuel.
5	Cessation d'activité	L'hôte ou le service est en cours de déclassement.
6	Maintien du certificat	Le certificat est temporairement révoqué. Vous pourrez restaurer le certificat ultérieurement.
8	Supprimer de la LCR	Le certificat n'est pas inclus dans la liste de révocation des certificats (CRL).
9	Retrait du privilège	L'utilisateur, l'hôte ou le service n'est plus autorisé à utiliser le certificat.
10	Compromis de l'autorité d'attribut (AA)	Le certificat AA n'est plus fiable.

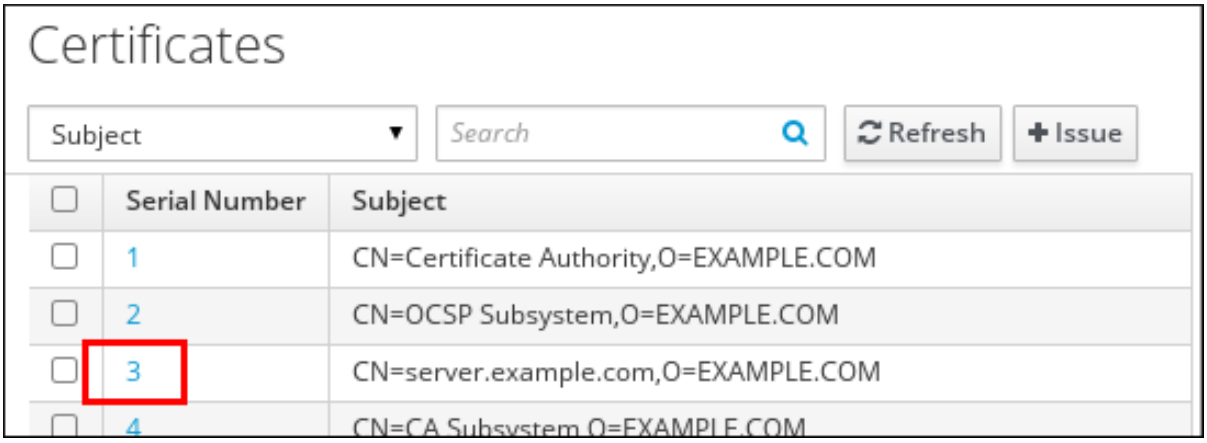
6.5.2. Révoquer des certificats avec les AC IdM intégrées à l'aide de l'interface Web IdM

Si vous savez que vous avez perdu la clé privée de votre certificat, vous devez révoquer le certificat pour empêcher son utilisation abusive. Suivez cette procédure pour utiliser l'interface Web IdM afin de révoquer un certificat émis par l'autorité de certification IdM.

Procédure

1. Cliquez sur **Authentication** > **Certificates** > **Certificates**.
2. Cliquez sur le numéro de série du certificat pour ouvrir la page d'information sur le certificat.

   Figure 6.2. Liste des certificats



3. Dans la page d'information sur le certificat, cliquez sur **Actions → Révoquer le certificat**
4. Sélectionnez le motif de révocation et cliquez sur **Révoquer**. Pour plus d'informations, reportez-vous à la section Motifs de révocation des certificats.

6.5.3. Révoquer des certificats avec les AC IdM intégrées à l'aide de la CLI IdM

Si vous savez que vous avez perdu la clé privée de votre certificat, vous devez révoquer le certificat pour empêcher son utilisation abusive. Suivez cette procédure pour utiliser la CLI IdM afin de révoquer un certificat émis par l'autorité de certification IdM.

Procédure

- Utilisez la commande **ipa cert-revoke** et spécifiez :
  - le numéro de série du certificat
  - le numéro d'identification du motif de révocation ; voir Motifs de révocation de certificats pour plus de détails

Par exemple, pour révoquer le certificat portant le numéro de série **1032** pour la raison 1 : **Key Compromised**, entrez :

```
ipa cert-revoke 1032 --revocation-reason=1
```

Pour plus de détails sur la demande d'un nouveau certificat, voir la documentation suivante :

- Demander un nouveau certificat d'utilisateur et l'exporter vers le client
- Obtention d'un certificat IdM pour un service à l'aide de certmonger .

6.6. RESTAURATION DES CERTIFICATS AVEC LES AC IDM INTÉGRÉES

Si vous avez révoqué un certificat pour la raison 6 : **Certificate Hold**, vous pouvez le restaurer à nouveau si la clé privée du certificat n'a pas été compromise. Pour restaurer un certificat, utilisez l'une des procédures suivantes :

- Restaurer les certificats avec les AC IdM intégrées à l'aide de l'interface Web IdM ;
- Restaurer les certificats avec les AC IdM intégrées à l'aide de la CLI IdM .

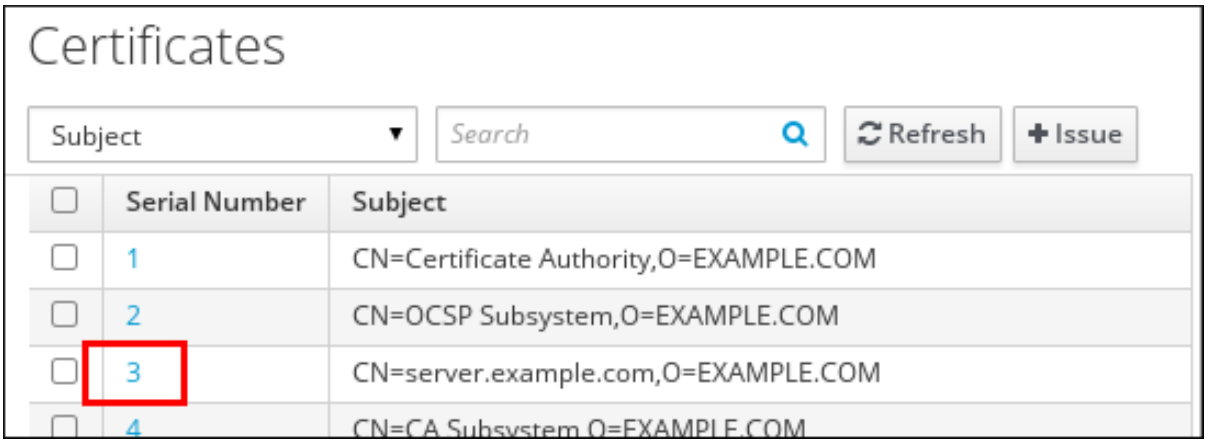
6.6.1. Restauration des certificats avec les AC IdM intégrées à l'aide de l'interface Web IdM

Suivez cette procédure pour utiliser l'interface Web IdM afin de restaurer un certificat IdM qui a été révoqué pour la raison suivante : **Certificate Hold**.

Procédure

1. Dans le menu **Authentication**, cliquez sur **Certificates** > **Certificates**.
2. Cliquez sur le numéro de série du certificat pour ouvrir la page d'information sur le certificat.

   Figure 6.3. Liste des certificats



3. Dans la page d'information sur le certificat, cliquez sur **Actions → Restaurer le certificat**

6.6.2. Restauration des certificats avec les AC IdM intégrées à l'aide de la CLI IdM

Suivez cette procédure pour utiliser la CLI IdM afin de restaurer un certificat IdM qui a été révoqué pour la raison 6 : **Certificate Hold**.

Procédure

- Utilisez la commande **ipa cert-remove-hold** et indiquez le numéro de série du certificat. Par exemple :

```
ipa cert-remove-hold 1032
```

CHAPITRE 7. CONFIGURATION DE LA GESTION DES IDENTITÉS POUR L'AUTHENTIFICATION PAR CARTE À PUCE

La gestion des identités (IdM) prend en charge l'authentification par carte à puce avec :

- Certificats d'utilisateur délivrés par l'autorité de certification IdM
- Certificats d'utilisateur délivrés par une autorité de certification externe

Cette histoire d'utilisateur montre comment configurer l'authentification par carte à puce dans IdM pour les deux types de certificats. Dans l'histoire de l'utilisateur, le certificat CA **rootca.pem** est le fichier contenant le certificat d'une autorité de certification externe de confiance.

Pour plus d'informations sur l'authentification par carte à puce dans IdM, voir Comprendre l'authentification par carte à puce.

L'histoire de l'utilisateur contient les modules suivants :

- Configuration du serveur IdM pour l'authentification par carte à puce
- Configuration du client IdM pour l'authentification par carte à puce
- Ajout d'un certificat à une entrée utilisateur dans l'interface Web IdM
- Ajout d'un certificat à une entrée utilisateur dans la CLI IdM
- Installation d'outils de gestion et d'utilisation des cartes à puce
- Stocker un certificat sur une carte à puce
- Connexion à l'IdM avec des cartes à puce
- Configuration de l'accès au GDM à l'aide de l'authentification par carte à puce
- Configuration de l'accès su à l'aide de l'authentification par carte à puce

7.1. CONFIGURATION DU SERVEUR IDM POUR L'AUTHENTIFICATION PAR CARTE À PUCE

Si vous souhaitez activer l'authentification par carte à puce pour les utilisateurs dont les certificats ont été émis par l'autorité de certification (AC) du domaine <EXAMPLE.ORG> auquel votre AC de gestion des identités (IdM) fait confiance, vous devez obtenir les certificats suivants afin de pouvoir les ajouter lors de l'exécution du script **ipa-advise** qui configure le serveur IdM :

- Le certificat de l'autorité de certification racine qui a délivré le certificat pour l'autorité de certification <EXAMPLE.ORG> directement ou par l'intermédiaire d'une ou de plusieurs de ses autorités de certification secondaires. Vous pouvez télécharger la chaîne de certificats à partir d'une page web dont le certificat a été délivré par l'autorité. Pour plus de détails, voir les étapes 1 à 4a de la section Configuration d'un navigateur pour activer l'authentification par certificat .
- Le certificat de l'autorité de certification IdM. Vous pouvez obtenir le certificat de l'autorité de certification à partir du fichier **/etc/ipa/ca.crt** du serveur IdM sur lequel tourne une instance de l'autorité de certification IdM.

- Les certificats de toutes les autorités de certification intermédiaires, c'est-à-dire intermédiaires entre l'autorité de certification <EXAMPLE.ORG> et l'autorité de certification IdM.

Pour configurer un serveur IdM pour l'authentification par carte à puce :

1. Obtenir les fichiers contenant les certificats de l'autorité de certification au format PEM.
2. Exécutez le script intégré **ipa-advise**.
3. Recharger la configuration du système.

Conditions préalables

- Vous avez un accès root au serveur IdM.
- Vous disposez du certificat de l'autorité de certification racine et de tous les certificats des autorités de certification intermédiaires.

Procédure

1. Créez un répertoire dans lequel vous effectuerez la configuration :

```
[root@server]# mkdir ~/SmartCard/
```

2. Naviguez jusqu'au répertoire :

```
[root@server]# cd ~/SmartCard/
```

3. Obtenez les certificats d'autorité de certification pertinents stockés dans des fichiers au format PEM. Si votre certificat d'autorité de certification est stocké dans un fichier d'un format différent, tel que DER, convertissez-le au format PEM. Le certificat de l'autorité de certification IdM est au format PEM et se trouve dans le fichier **/etc/ipa/ca.crt**.
Convertit un fichier DER en fichier PEM :

```
# openssl x509 -in <filename>.der -inform DER -out <filename>.pem -outform PEM
```

4. Pour plus de commodité, copiez les certificats dans le répertoire dans lequel vous souhaitez effectuer la configuration :

```
[root@server SmartCard]# cp /tmp/rootca.pem ~/SmartCard/
[root@server SmartCard]# cp /tmp/subca.pem ~/SmartCard/
[root@server SmartCard]# cp /tmp/issuingca.pem ~/SmartCard/
```

5. En option, si vous utilisez des certificats d'autorités de certification externes, utilisez l'utilitaire **openssl x509** pour visualiser le contenu des fichiers au format **PEM** et vérifier que les valeurs **Issuer** et **Subject** sont correctes :

```
[root@server SmartCard]# openssl x509 -noout -text -in rootca.pem | more
```

6. Générer un script de configuration avec l'utilitaire intégré **ipa-advise**, en utilisant les privilèges de l'administrateur :

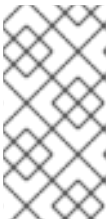
```
[root@server SmartCard]# kinit admin
[root@server SmartCard]# ipa-advise config-server-for-smart-card-auth > config-server-for-smart-card-auth.sh
```

Le script **config-server-for-smart-card-auth.sh** effectue les actions suivantes :

- Il configure le serveur HTTP Apache de l'IdM.
- Il active la cryptographie à clé publique pour l'authentification initiale dans Kerberos (PKINIT) sur le centre de distribution de clés (KDC).
- Il configure l'interface Web IdM pour qu'elle accepte les demandes d'autorisation de carte à puce.

7. Exécutez le script en ajoutant les fichiers PEM contenant les certificats de l'autorité de certification racine et de l'autorité de certification secondaire en tant qu'arguments :

```
[root@server SmartCard]# chmod +x config-server-for-smart-card-auth.sh
[root@server SmartCard]# ./config-server-for-smart-card-auth.sh rootca.pem subca.pem issuingca.pem
Ticket cache:KEYRING:persistent:0:0
Default principal: admin@IDM.EXAMPLE.COM
[...]
Systemwide CA database updated.
The ipa-certupdate command was successful
```



NOTE

Assurez-vous que vous ajoutez le certificat de l'autorité de certification racine en tant qu'argument avant tout certificat d'autorité de certification secondaire et que les certificats de l'autorité de certification ou de l'autorité de certification secondaire n'ont pas expiré.

8. En option, si l'autorité de certification qui a émis le certificat utilisateur ne fournit pas de répondeur OCSP (Online Certificate Status Protocol), il peut être nécessaire de désactiver la vérification OCSP pour l'authentification à l'IdM Web UI :

   a. Définissez le paramètre **SSLOCSPEnable** à **off** dans le fichier **/etc/httpd/conf.d/ssl.conf**:

   ```
   SSLOCSPEnable off
   ```

   b. Redémarrez le démon Apache (httpd) pour que les modifications prennent effet immédiatement :

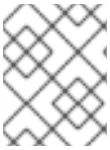
   ```
   [root@server SmartCard]# systemctl restart httpd
   ```

AVERTISSEMENT

Ne désactivez pas le contrôle OCSP si vous n'utilisez que des certificats d'utilisateur émis par l'autorité de certification IdM. Les répondeurs OCSP font partie de l'IdM.

Pour savoir comment maintenir la vérification OCSP activée, tout en empêchant un certificat d'utilisateur d'être rejeté par le serveur IdM s'il ne contient pas les informations relatives à l'emplacement où l'autorité de certification qui a délivré le certificat d'utilisateur écoute les demandes de service OCSP, voir la directive **SSLOCSPDefaultResponder** dans les options de configuration de Apache mod_ssl.

Le serveur est maintenant configuré pour l'authentification par carte à puce.



NOTE

Pour activer l'authentification par carte à puce dans l'ensemble de la topologie, exécutez la procédure sur chaque serveur IdM.

7.2. UTILISER ANSIBLE POUR CONFIGURER LE SERVEUR IDM POUR L'AUTHENTIFICATION PAR CARTE À PUCE

Vous pouvez utiliser Ansible pour activer l'authentification par carte à puce pour les utilisateurs dont les certificats ont été émis par l'autorité de certification (AC) du domaine <EXAMPLE.ORG> auquel votre AC de gestion des identités (IdM) fait confiance. Pour ce faire, vous devez obtenir les certificats suivants afin de pouvoir les utiliser lors de l'exécution d'un playbook Ansible avec le script de rôle **ipasmartcard_server ansible-freeipa** :

- Le certificat de l'autorité de certification racine qui a délivré le certificat pour l'autorité de certification <EXAMPLE.ORG> directement ou par l'intermédiaire d'une ou de plusieurs de ses autorités de certification secondaires. Vous pouvez télécharger la chaîne de certificats à partir d'une page web dont le certificat a été émis par l'autorité. Pour plus d'informations, voir l'étape 4 de la section Configuration d'un navigateur pour activer l'authentification par certificat .
- Le certificat de l'autorité de certification IdM. Vous pouvez obtenir le certificat de l'autorité de certification à partir du fichier **/etc/ipa/ca.crt** sur n'importe quel serveur de l'autorité de certification IdM.
- Les certificats de toutes les AC intermédiaires entre l'AC <EXAMPLE.ORG> et l'AC IdM.

Conditions préalables

- Vous avez un accès **root** au serveur IdM.
- Vous connaissez le mot de passe de l'IdM **admin**.
- Vous disposez du certificat de l'autorité de certification racine, du certificat de l'autorité de certification IdM et de tous les certificats des autorités de certification intermédiaires.

- Vous avez configuré votre nœud de contrôle Ansible pour qu'il réponde aux exigences suivantes :
  - Vous utilisez la version 2.8 ou ultérieure d'Ansible.
  - Vous avez installé le paquetage **ansible-freeipa** sur le contrôleur Ansible.
  - L'exemple suppose que dans le répertoire ***~/MyPlaybooks/*** vous avez créé un fichier d'inventaire Ansible avec le nom de domaine complet (FQDN) du serveur IdM.
  - L'exemple suppose que le coffre-fort **secret.yml** Ansible stocke votre **ipaadmin_password**.

Procédure

1. Si vos certificats d'autorité de certification sont stockés dans des fichiers d'un format différent, tel que **DER**, convertissez-les au format **PEM**:

```
# openssl x509 -in <filename>.der -inform DER -out <filename>.pem -outform PEM
```

   Le certificat de l'autorité de certification IdM est au format **PEM** et se trouve dans le fichier **/etc/ipa/ca.crt**.

2. En option, utilisez l'utilitaire **openssl x509** pour visualiser le contenu des fichiers au format **PEM** et vérifier que les valeurs **Issuer** et **Subject** sont correctes :

```
# openssl x509 -noout -text -in root-ca.pem | more
```

3. Naviguez jusqu'à votre répertoire ***~/MyPlaybooks/*** répertoire :

```
$ cd ~/MyPlaybooks/
```

4. Créez un sous-répertoire dédié aux certificats d'autorité de certification :

```
$ mkdir SmartCard/
```

5. Pour plus de commodité, copiez tous les certificats requis dans le répertoire **~/MyPlaybooks/SmartCard/**:

```
# cp /tmp/root-ca.pem ~/MyPlaybooks/SmartCard/
# cp /tmp/intermediate-ca.pem ~/MyPlaybooks/SmartCard/
# cp /etc/ipa/ca.crt ~/MyPlaybooks/SmartCard/ipa-ca.crt
```

6. Dans votre fichier d'inventaire Ansible, spécifiez ce qui suit :
   - Les serveurs IdM que vous souhaitez configurer pour l'authentification par carte à puce.
   - Le mot de passe de l'administrateur de l'IdM.
   - Les chemins d'accès aux certificats des autorités de certification dans l'ordre suivant :
     - Le fichier du certificat de l'autorité de certification racine
     - Les fichiers des certificats de l'autorité de certification intermédiaire

- Le fichier du certificat de l'autorité de certification IdM

Le fichier peut se présenter comme suit :

```
[ipaserver]
ipaserver.idm.example.com

[ipareplicas]
ipareplica1.idm.example.com
ipareplica2.idm.example.com

[ipacluster:children]
ipaserver
ipareplicas

[ipacluster:vars]
ipaadmin_password=SomeADMINpassword
ipasmartcard_server_ca_certs=/home/<user_name>/MyPlaybooks/SmartCard/root-ca.pem,/home/<user_name>/MyPlaybooks/SmartCard/intermediate-ca.pem,/home/<user_name>/MyPlaybooks/SmartCard/ipa-ca.crt
```

7. Créez un playbook **install-smartcard-server.yml** avec le contenu suivant :

```
---
- name: Playbook to set up smart card authentication for an IdM server
  hosts: ipaserver
  become: true

  roles:
  - role: ipasmartcard_server
    state: present
```

8. Enregistrer le fichier.

9. Exécutez le playbook Ansible. Spécifiez le fichier du livre de jeu, le fichier contenant le mot de passe protégeant le fichier **secret.yml** et le fichier d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory install-smartcard-server.yml
```

Le rôle **ipasmartcard_server** Ansible effectue les actions suivantes :

- Il configure le serveur HTTP Apache de l'IdM.
- Il active la cryptographie à clé publique pour l'authentification initiale dans Kerberos (PKINIT) sur le centre de distribution de clés (KDC).
- Il configure l'interface Web IdM pour qu'elle accepte les demandes d'autorisation de carte à puce.

10. En option, si l'autorité de certification qui a émis le certificat utilisateur ne fournit pas de répondeur OCSP (Online Certificate Status Protocol), il peut être nécessaire de désactiver la vérification OCSP pour l'authentification à l'IdM Web UI :

a. Se connecter au serveur IdM en tant que **root**:

```
ssh root@ipaserver.idm.example.com
```

b. Définissez le paramètre **SSLOCSPEnable** à **off** dans le fichier **/etc/httpd/conf.d/ssl.conf**:

```
SSLOCSPEnable off
```

c. Redémarrez le démon Apache (httpd) pour que les modifications prennent effet immédiatement :

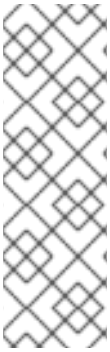
```
# systemctl restart httpd
```

AVERTISSEMENT

Ne désactivez pas le contrôle OCSP si vous n'utilisez que des certificats d'utilisateur émis par l'autorité de certification IdM. Les répondeurs OCSP font partie de l'IdM.

Pour savoir comment maintenir la vérification OCSP activée, tout en empêchant un certificat d'utilisateur d'être rejeté par le serveur IdM s'il ne contient pas les informations relatives à l'emplacement où l'autorité de certification qui a délivré le certificat d'utilisateur écoute les demandes de service OCSP, voir la directive **SSLOCSPDefaultResponder** dans les options de configuration de Apache mod_ssl.

Le serveur figurant dans le fichier d'inventaire est maintenant configuré pour l'authentification par carte à puce.



NOTE

Pour activer l'authentification par carte à puce dans l'ensemble de la topologie, définissez la variable **hosts** dans le playbook Ansible à **ipacluster**:

```
---
- name: Playbook to setup smartcard for IPA server and replicas
  hosts: ipacluster
[...]
```

Ressources supplémentaires

- Exemples de playbooks utilisant le rôle **ipasmartcard_server** dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/**

7.3. CONFIGURATION DU CLIENT IDM POUR L'AUTHENTIFICATION PAR CARTE À PUCE

Cette section décrit comment configurer les clients IdM pour l'authentification par carte à puce. La procédure doit être exécutée sur chaque système IdM, client ou serveur, auquel vous souhaitez vous connecter en utilisant une carte à puce pour l'authentification. Par exemple, pour activer une connexion

ssh de l'hôte A à l'hôte B, le script doit être exécuté sur l'hôte B.

En tant qu'administrateur, exécutez cette procédure pour activer l'authentification par carte à puce à l'aide de

- Le protocole **ssh**
  Pour plus d'informations, voir Configuration de l'accès SSH à l'aide de l'authentification par carte à puce.
- Le login de la console
- Le gestionnaire d'affichage Gnome (GDM)
- La commande **su**

Cette procédure n'est pas nécessaire pour s'authentifier auprès de l'interface Web IdM.
L'authentification à l'interface Web IdM implique deux hôtes, dont aucun ne doit être un client IdM :

- La machine sur laquelle le navigateur s'exécute. La machine peut être en dehors du domaine IdM.
- Le serveur IdM sur lequel **httpd** est exécuté.

La procédure suivante suppose que vous configurez l'authentification par carte à puce sur un client IdM et non sur un serveur IdM. C'est pourquoi vous avez besoin de deux ordinateurs : un serveur IdM pour générer le script de configuration et le client IdM sur lequel le script sera exécuté.

Conditions préalables

- Votre serveur IdM a été configuré pour l'authentification par carte à puce, comme décrit dans la section Configuration du serveur IdM pour l'authentification par carte à puce .
- Vous disposez d'un accès root au serveur IdM et au client IdM.
- Vous disposez du certificat de l'autorité de certification racine et de tous les certificats des autorités de certification intermédiaires.
- Vous avez installé le client IdM avec l'option **--mkhomedir** pour vous assurer que les utilisateurs distants peuvent se connecter avec succès. Si vous ne créez pas de répertoire personnel, l'emplacement de connexion par défaut est la racine de la structure de répertoires, /.

Procédure

1. Sur un serveur IdM, générez un script de configuration avec **ipa-advise** en utilisant les privilèges de l'administrateur :

```
[root@server SmartCard]# kinit admin
[root@server SmartCard]# ipa-advise config-client-for-smart-card-auth > config-client-for-smart-card-auth.sh
```

   Le script **config-client-for-smart-card-auth.sh** effectue les actions suivantes :

   - Il configure le démon de la carte à puce.
   - Il définit la réserve de confiance du système.

- Il configure le System Security Services Daemon (SSSD) pour permettre aux utilisateurs de s'authentifier soit avec leur nom d'utilisateur et leur mot de passe, soit avec leur carte à puce. Pour plus de détails sur les options du profil SSSD pour l'authentification par carte à puce, voir Options d'authentification par carte à puce dans RHEL .

2. A partir du serveur IdM, copiez le script dans un répertoire de votre choix sur la machine du client IdM :

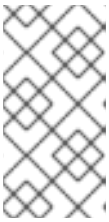
```
[root@server SmartCard]# scp config-client-for-smart-card-auth.sh
root@client.idm.example.com:/root/SmartCard/
Password:
config-client-for-smart-card-auth.sh        100%   2419       3.5MB/s   00:00
```

3. À partir du serveur IdM, copiez les fichiers de certificats d'autorité de certification au format PEM, pour plus de commodité, dans le même répertoire de la machine du client IdM que celui utilisé à l'étape précédente :

```
[root@server SmartCard]# scp {rootca.pem,subca.pem,issuingca.pem}
root@client.idm.example.com:/root/SmartCard/
Password:
rootca.pem                    100%   1237     9.6KB/s   00:00
subca.pem                     100%   2514    19.6KB/s   00:00
issuingca.pem                 100%   2514    19.6KB/s   00:00
```

4. Sur l'ordinateur client, exécutez le script en ajoutant les fichiers PEM contenant les certificats d'autorité de certification en tant qu'arguments :

```
[root@client SmartCard]# kinit admin
[root@client SmartCard]# chmod +x config-client-for-smart-card-auth.sh
[root@client SmartCard]# ./config-client-for-smart-card-auth.sh rootca.pem subca.pem
issuingca.pem
Ticket cache:KEYRING:persistent:0:0
Default principal: admin@IDM.EXAMPLE.COM
[...]
Systemwide CA database updated.
The ipa-certupdate command was successful
```



NOTE

Assurez-vous que vous ajoutez le certificat de l'autorité de certification racine en tant qu'argument avant tout certificat d'autorité de certification secondaire et que les certificats de l'autorité de certification ou de l'autorité de certification secondaire n'ont pas expiré.

Le client est maintenant configuré pour l'authentification par carte à puce.

7.4. UTILISER ANSIBLE POUR CONFIGURER LES CLIENTS IDM POUR L'AUTHENTIFICATION PAR CARTE À PUCE

Cette section explique comment utiliser le module **ansible-freeipa ipasmartcard_client** pour configurer des clients Identity Management (IdM) spécifiques afin de permettre aux utilisateurs IdM de s'authentifier à l'aide d'une carte à puce. Exécutez cette procédure pour activer l'authentification par carte à puce pour les utilisateurs IdM qui utilisent l'un des éléments suivants pour accéder à IdM :

- Le protocole **ssh**
  Pour plus d'informations, voir Configuration de l'accès SSH à l'aide de l'authentification par carte à puce.
- Le login de la console
- Le gestionnaire d'affichage Gnome (GDM)
- La commande **su**

NOTE

Cette procédure n'est pas nécessaire pour s'authentifier auprès de l'interface Web IdM. L'authentification à l'interface Web IdM implique deux hôtes, dont aucun ne doit être un client IdM :

- La machine sur laquelle le navigateur s'exécute. La machine peut être en dehors du domaine IdM.
- Le serveur IdM sur lequel **httpd** est exécuté.

Conditions préalables

- Votre serveur IdM a été configuré pour l'authentification par carte à puce, comme décrit dans la section Utilisation d'Ansible pour configurer le serveur IdM pour l'authentification par carte à puce.
- Vous disposez d'un accès root au serveur IdM et au client IdM.
- Vous disposez du certificat de l'autorité de certification racine, du certificat de l'autorité de certification IdM et de tous les certificats des autorités de certification intermédiaires.
- Vous avez configuré votre nœud de contrôle Ansible pour qu'il réponde aux exigences suivantes :
  - Vous utilisez la version 2.8 ou ultérieure d'Ansible.
  - Vous avez installé le paquetage **ansible-freeipa** sur le contrôleur Ansible.
  - L'exemple suppose que dans le répertoire ***~/MyPlaybooks/*** vous avez créé un fichier d'inventaire Ansible avec le nom de domaine complet (FQDN) du serveur IdM.
  - L'exemple suppose que le coffre-fort **secret.yml** Ansible stocke votre **ipaadmin_password**.

Procédure

1. Si vos certificats d'autorité de certification sont stockés dans des fichiers d'un format différent, tel que **DER**, convertissez-les au format **PEM**:

   ```
   # openssl x509 -in <filename>.der -inform DER -out <filename>.pem -outform PEM
   ```

   Le certificat de l'autorité de certification IdM est au format **PEM** et se trouve dans le fichier **/etc/ipa/ca.crt**.

2. En option, utilisez l'utilitaire **openssl x509** pour visualiser le contenu des fichiers au format **PEM** et vérifier que les valeurs **Issuer** et **Subject** sont correctes :

```
# openssl x509 -noout -text -in root-ca.pem | more
```

3. Sur votre nœud de contrôle Ansible, naviguez jusqu'à votre répertoire ***~/MyPlaybooks/*** répertoire :

```
$ cd ~/MyPlaybooks/
```

4. Créez un sous-répertoire dédié aux certificats d'autorité de certification :

```
$ mkdir SmartCard/
```

5. Pour plus de commodité, copiez tous les certificats requis dans le répertoire **~/MyPlaybooks/SmartCard/**, par exemple :

```
# cp /tmp/root-ca.pem ~/MyPlaybooks/SmartCard/
# cp /tmp/intermediate-ca.pem ~/MyPlaybooks/SmartCard/
# cp /etc/ipa/ca.crt ~/MyPlaybooks/SmartCard/ipa-ca.crt
```

6. Dans votre fichier d'inventaire Ansible, spécifiez ce qui suit :

   - Les clients IdM que vous souhaitez configurer pour l'authentification par carte à puce.
   - Le mot de passe de l'administrateur de l'IdM.
   - Les chemins d'accès aux certificats des autorités de certification dans l'ordre suivant :
     - Le fichier du certificat de l'autorité de certification racine
     - Les fichiers des certificats de l'autorité de certification intermédiaire
     - Le fichier du certificat de l'autorité de certification IdM

   Le fichier peut se présenter comme suit :

```
[ipaclients]
ipaclient1.example.com
ipaclient2.example.com

[ipaclients:vars]
ipaadmin_password=SomeADMINpassword
ipasmartcard_client_ca_certs=/home/<user_name>/MyPlaybooks/SmartCard/root-ca.pem,/home/<user_name>/MyPlaybooks/SmartCard/intermediate-ca.pem,/home/<user_name>/MyPlaybooks/SmartCard/ipa-ca.crt
```

7. Créez un playbook **install-smartcard-clients.yml** avec le contenu suivant :

```
---
- name: Playbook to set up smart card authentication for an IdM client
  hosts: ipaclients
  become: true
```

```
roles:
- role: ipasmartcard_client
  state: present
```

8. Enregistrer le fichier.

9. Exécutez le playbook Ansible. Spécifiez le playbook et les fichiers d'inventaire :

```
$ ansible-playbook --vault-password-file=password_file -v -i inventory install-smartcard-clients.yml
```

Le rôle **ipasmartcard_client** Ansible effectue les actions suivantes :

- Il configure le démon de la carte à puce.
- Il définit la réserve de confiance du système.
- Il configure le System Security Services Daemon (SSSD) pour permettre aux utilisateurs de s'authentifier soit avec leur nom d'utilisateur et leur mot de passe, soit avec leur carte à puce. Pour plus de détails sur les options de profil SSSD pour l'authentification par carte à puce, voir Options d'authentification par carte à puce dans RHEL .

Les clients répertoriés dans la section **ipaclients** du fichier d'inventaire sont maintenant configurés pour l'authentification par carte à puce.

NOTE

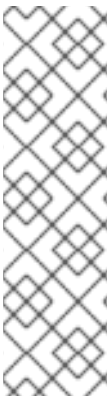
Si vous avez installé les clients IdM avec l'option **--mkhomedir**, les utilisateurs distants pourront se connecter à leur répertoire personnel. Sinon, l'emplacement de connexion par défaut est la racine de la structure de répertoires, /.

Ressources supplémentaires

- Exemples de playbooks utilisant le rôle **ipasmartcard_server** dans le répertoire **/usr/share/doc/ansible-freeipa/playbooks/**

7.5. AJOUT D'UN CERTIFICAT À UNE ENTRÉE UTILISATEUR DANS L'INTERFACE WEB IDM

Cette procédure décrit comment ajouter un certificat externe à une entrée utilisateur dans l'interface Web IdM.



NOTE

Au lieu de télécharger le certificat complet, il est également possible de télécharger des données de mappage de certificat vers une entrée d'utilisateur dans IdM. Les entrées utilisateur contenant des certificats complets ou des données de mappage de certificats peuvent être utilisées conjointement avec les règles de mappage de certificats correspondantes pour faciliter la configuration de l'authentification par carte à puce pour les administrateurs de système. Pour plus de détails, voir

Règles de mappage des certificats pour la configuration de l'authentification sur les cartes à puce.

NOTE

Si le certificat de l'utilisateur a été délivré par l'autorité de certification IdM, le certificat est déjà stocké dans l'entrée de l'utilisateur et vous pouvez ignorer cette section.

Conditions préalables

- Vous disposez du certificat que vous souhaitez ajouter à l'entrée de l'utilisateur.

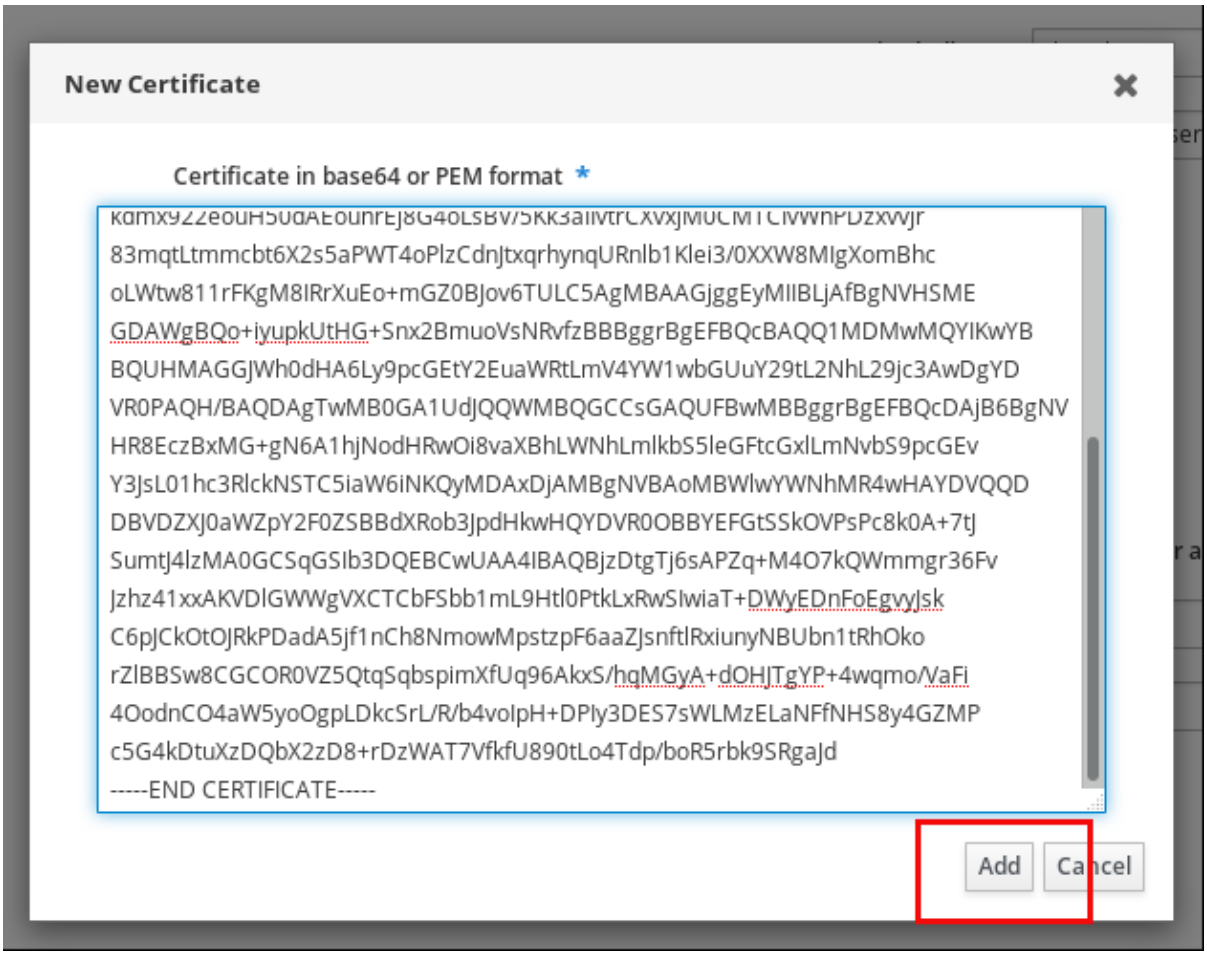
Procédure

1. Connectez-vous à l'interface Web IdM en tant qu'administrateur si vous souhaitez ajouter un certificat à un autre utilisateur. Pour ajouter un certificat à votre propre profil, vous n'avez pas besoin des informations d'identification de l'administrateur.
2. Naviguez vers **Users → Active users → sc_user**.
3. Recherchez l'option **Certificate** et cliquez sur **Add**.
4. Sur le site **Command-Line Interface**, affichez le certificat au format **PEM** à l'aide de l'utilitaire **cat** ou d'un éditeur de texte :

```
[user@client SmartCard]$ cat testuser.crt
```

5. Copiez et collez le certificat de l'interface de gestion dans la fenêtre qui s'est ouverte dans l'interface Web.
6. Cliquez sur **Add**.

Figure 7.1. Ajout d'un nouveau certificat dans l'interface Web IdM



L'entrée **sc_user** contient maintenant un certificat externe.

7.6. AJOUT D'UN CERTIFICAT À UNE ENTRÉE UTILISATEUR DANS LA CLI IDM

Cette procédure décrit comment ajouter un certificat externe à une entrée utilisateur dans IdM CLI.

NOTE

Au lieu de télécharger le certificat complet, il est également possible de télécharger des données de mappage de certificat vers une entrée d'utilisateur dans IdM. Les entrées utilisateur contenant des certificats complets ou des données de mappage de certificats peuvent être utilisées conjointement avec les règles de mappage de certificats correspondantes pour faciliter la configuration de l'authentification par carte à puce pour les administrateurs système. Pour plus de détails, voir Règles de mappage de certificats pour la configuration de l'authentification par carte à puce.

NOTE

Si le certificat de l'utilisateur a été délivré par l'autorité de certification IdM, le certificat est déjà stocké dans l'entrée de l'utilisateur et vous pouvez ignorer cette section.

Conditions préalables

- Vous disposez du certificat que vous souhaitez ajouter à l'entrée de l'utilisateur.

Procédure

1. Connectez-vous au CLI IdM en tant qu'administrateur si vous souhaitez ajouter un certificat à un autre utilisateur :

```
[user@client SmartCard]$ kinit admin
```

   Pour ajouter un certificat à votre propre profil, vous n'avez pas besoin des informations d'identification de l'administrateur :

```
[user@client SmartCard]$ kinit sc_user
```

2. Créez une variable d'environnement contenant le certificat dont l'en-tête et le pied de page ont été supprimés et concaténés en une seule ligne, ce qui correspond au format attendu par la commande **ipa user-add-cert**:

```
[user@client SmartCard]$ export CERT=`openssl x509 -outform der -in testuser.crt | base64 -w0 -`
```

   Notez que le certificat dans le fichier **testuser.crt** doit être au format **PEM**.

3. Ajoutez le certificat au profil de l'utilisateur sc_user à l'aide de la commande **ipa user-add-cert**:

```
[user@client SmartCard]$ ipa user-add-cert sc_user --certificate=$CERT
```

L'entrée **sc_user** contient maintenant un certificat externe.

7.7. INSTALLATION D'OUTILS DE GESTION ET D'UTILISATION DES CARTES À PUCE

Pour configurer votre carte à puce, vous avez besoin d'outils qui peuvent générer des certificats et les stocker sur une carte à puce.

Vous devez :

- Installez le paquet **gnutls-utils**, qui vous aide à gérer les certificats.
- Installez le paquetage **opensc**, qui fournit un ensemble de bibliothèques et d'utilitaires pour travailler avec des cartes à puce.
- Démarrez le service **pcscd**, qui communique avec le lecteur de cartes à puce.

Procédure

1. Installez les paquets **opensc** et **gnutls-utils**:

```
# dnf -y install opensc gnutls-utils
```

2. Démarrez le service **pcscd**.

```
# systemctl start pcscd
```

Vérifiez que le service **pcscd** est opérationnel.

7.8. PRÉPARATION DE VOTRE CARTE À PUCE ET TÉLÉCHARGEMENT DE VOS CERTIFICATS ET CLÉS SUR VOTRE CARTE À PUCE

Cette section décrit la configuration de la carte à puce avec l'outil **pkcs15-init**, qui vous aide à configurer :

- Effacer votre carte à puce
- Définition de nouveaux codes PIN et de clés de déblocage de code PIN (PUK) en option
- Création d'un nouvel emplacement sur la carte à puce
- Stockage du certificat, de la clé privée et de la clé publique dans la fente
- Si nécessaire, verrouiller les paramètres de la carte à puce, car certaines cartes à puce nécessitent ce type de finalisation

NOTE

L'outil **pkcs15-init** peut ne pas fonctionner avec toutes les cartes à puce. Vous devez utiliser les outils qui fonctionnent avec la carte à puce que vous utilisez.

Conditions préalables

- Le paquet **opensc**, qui comprend l'outil **pkcs15-init**, est installé.
  Pour plus de détails, voir Installation des outils de gestion et d'utilisation des cartes à puce .
- La carte est insérée dans le lecteur et connectée à l'ordinateur.
- Vous disposez de la clé privée, de la clé publique et du certificat à stocker sur la carte à puce. Dans cette procédure, **testuser.key**, **testuserpublic.key**, et **testuser.crt** sont les noms utilisés pour la clé privée, la clé publique et le certificat.
- Vous disposez du code PIN de l'utilisateur de votre carte à puce actuelle et du code PIN de l'agent de sécurité (SO-PIN).

Procédure

1. Effacez votre carte à puce et authentifiez-vous avec votre code PIN :

```
$ pkcs15-init --erase-card --use-default-transport-keys
Using reader with a card: Reader name
PIN [Security Officer PIN] required.
Please enter PIN [Security Officer PIN]:
```

   La carte a été effacée.

2. Initialisez votre carte à puce, définissez votre code PIN et PUK d'utilisateur, ainsi que le code PIN et PUK de votre responsable de la sécurité :

```
$ pkcs15-init --create-pkcs15 --use-default-transport-keys \
    --pin 963214 --puk 321478 --so-pin 65498714 --so-puk 784123
Using reader with a card: Reader name
```

   L'outil **pcks15-init** crée un nouvel emplacement sur la carte à puce.

3. Définir l'étiquette et l'ID d'authentification pour l'emplacement :

```
$ pkcs15-init --store-pin --label testuser \
   --auth-id 01 --so-pin 65498714 --pin 963214 --puk 321478
Using reader with a card: Reader name
```

L'étiquette est définie sur une valeur lisible par l'homme, dans ce cas, **testuser**. L'adresse **auth-id** doit être composée de deux valeurs hexadécimales ; dans ce cas, elle est fixée à **01**.

4. Stockez et étiquetez la clé privée dans le nouvel emplacement de la carte à puce :

```
$ pkcs15-init --store-private-key testuser.key --label testuser_key \
   --auth-id 01 --id 01 --pin 963214
Using reader with a card: Reader name
```

NOTE

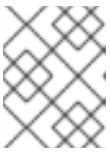
La valeur que vous indiquez pour **--id** doit être la même lorsque vous stockez votre clé privée et votre certificat à l'étape suivante. Il est recommandé de spécifier votre propre valeur pour **--id**, sinon l'outil calculera une valeur plus complexe.

5. Stockez et étiquetez le certificat dans le nouvel emplacement de la carte à puce :

```
$ pkcs15-init --store-certificate testuser.crt --label testuser_crt \
   --auth-id 01 --id 01 --format pem --pin 963214
Using reader with a card: Reader name
```

6. (Facultatif) Stockez et étiquetez la clé publique dans le nouvel emplacement de la carte à puce :

```
$ pkcs15-init --store-public-key testuserpublic.key
   --label testuserpublic_key --auth-id 01 --id 01 --pin 963214
Using reader with a card: Reader name
```



NOTE

Si la clé publique correspond à une clé privée ou à un certificat, indiquez le même ID que celui de la clé privée ou du certificat.

7. (Facultatif) Certaines cartes à puce exigent que vous finalisiez la carte en verrouillant les paramètres :

```
$ pkcs15-init -F
```

À ce stade, votre carte à puce comprend le certificat, la clé privée et la clé publique dans l'emplacement nouvellement créé. Vous avez également créé votre code PIN et PUK d'utilisateur ainsi que le code PIN et PUK de l'agent de sécurité.

7.9. CONNEXION À L'IDM AVEC DES CARTES À PUCE

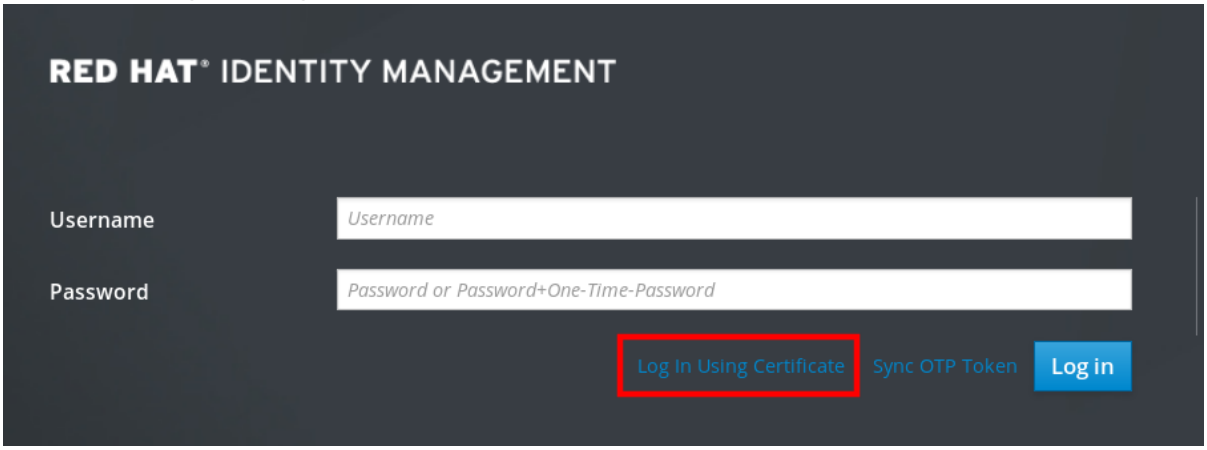
Cette section décrit l'utilisation des cartes à puce pour se connecter à l'interface Web IdM.

Conditions préalables

- Le navigateur web est configuré pour utiliser l'authentification par carte à puce.
- Le serveur IdM est configuré pour l'authentification par carte à puce.
- Le certificat installé sur votre carte à puce est soit émis par le serveur IdM, soit ajouté à l'entrée de l'utilisateur dans IdM.
- Vous connaissez le code PIN requis pour déverrouiller la carte à puce.
- La carte à puce a été insérée dans le lecteur.

Procédure

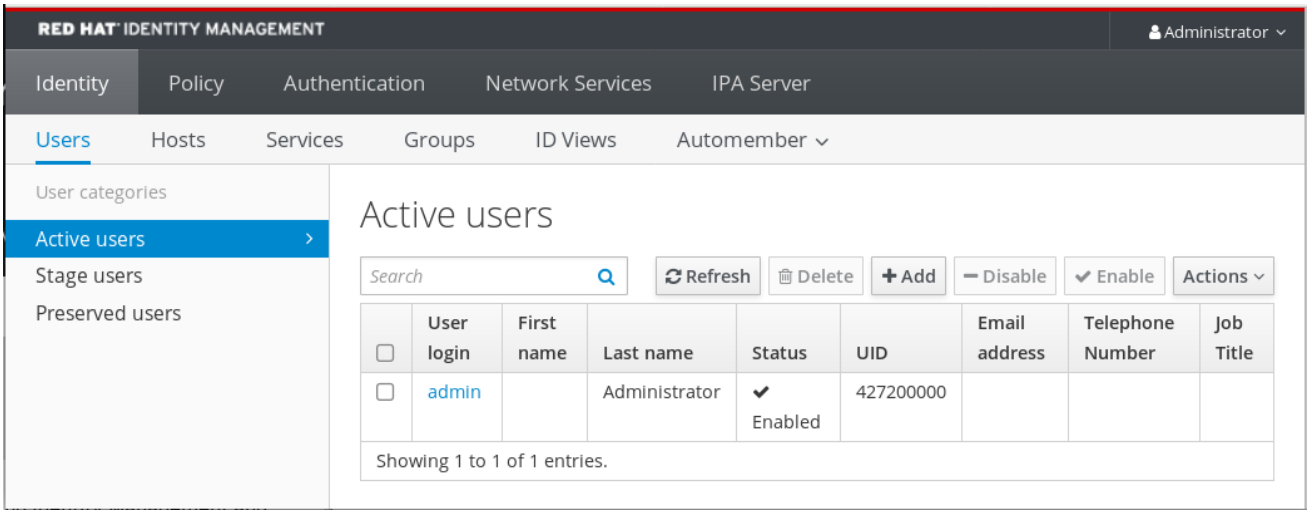
1. Ouvrez l'interface Web IdM dans le navigateur.
2. Cliquez sur **Log In Using Certificate**



3. Si la boîte de dialogue **Password Required** s'ouvre, ajoutez le code PIN pour déverrouiller la carte à puce et cliquez sur le bouton **OK**.
   La boîte de dialogue **User Identification Request** s'ouvre.

   Si la carte à puce contient plus d'un certificat, sélectionnez le certificat que vous souhaitez utiliser pour l'authentification dans la liste déroulante située sous **Choose a certificate to present as identification**.
4. Cliquez sur le bouton **OK**.

Vous êtes maintenant connecté avec succès à l'interface Web IdM.



7.10. SE CONNECTER À GDM EN UTILISANT L'AUTHENTIFICATION PAR CARTE À PUCE SUR UN CLIENT IDM

Le Gnome Desktop Manager (GDM) nécessite une authentification. Vous pouvez utiliser votre mot de passe, mais vous pouvez également utiliser une carte à puce pour l'authentification.

Cette section décrit l'authentification par carte à puce pour accéder à GDM.

Conditions préalables

- Le système a été configuré pour l'authentification par carte à puce. Pour plus de détails, voir Configuration du client IdM pour l'authentification par carte à puce.
- La carte à puce contient votre certificat et votre clé privée.
- Le compte d'utilisateur est membre du domaine IdM.
- Le certificat de la carte à puce correspond à l'entrée de l'utilisateur :
  - Affecter le certificat à une entrée utilisateur particulière. Pour plus de détails, voir Ajouter un certificat à une entrée utilisateur dans l'interface Web Id M ou Ajouter un certificat à une entrée utilisateur dans l'interface CLI IdM.
  - Les données de mappage de certificats appliquées au compte. Pour plus de détails, voir Règles de mappage de certificats pour la configuration de l'authentification par carte à puce.

Procédure

1. Insérez la carte à puce dans le lecteur.
2. Saisissez le code PIN de la carte à puce.
3. Cliquez sur **Sign In**.

Vous êtes connecté avec succès au système RHEL et vous disposez d'un TGT fourni par le serveur IdM.

Verification steps

- Dans la fenêtre **Terminal**, entrez **klist** et vérifiez le résultat :

```
$ klist
Ticket cache: KEYRING:persistent:1358900015:krb_cache_TObtNMd
Default principal: example.user@REDHAT.COM

Valid starting       Expires              Service principal
04/20/2020 13:58:24  04/20/2020 23:58:24  krbtgt/EXAMPLE.COM@EXAMPLE.COM
 renew until 04/27/2020 08:58:15
```

7.11. UTILISATION DE L'AUTHENTIFICATION PAR CARTE À PUCE AVEC LA COMMANDE SU

Le passage à un autre utilisateur nécessite une authentification. Vous pouvez utiliser un mot de passe ou un certificat. Cette section décrit l'utilisation de votre carte à puce avec la commande **su**. Cela signifie qu'après avoir entré la commande **su**, vous êtes invité à saisir le code PIN de la carte à puce.

Conditions préalables

- Votre serveur et votre client IdM ont été configurés pour l'authentification par carte à puce.
  - Voir Configuration du serveur IdM pour l'authentification par carte à puce
  - Voir Configuration du client IdM pour l'authentification par carte à puce
- La carte à puce contient votre certificat et votre clé privée. Voir Stocker un certificat sur une carte à puce
- La carte est insérée dans le lecteur et connectée à l'ordinateur.

Procédure

- Dans une fenêtre de terminal, changez d'utilisateur à l'aide de la commande **su**:

  ```
  $ su - example.user
  PIN for smart_card
  ```

  Si la configuration est correcte, vous êtes invité à saisir le code PIN de la carte à puce.

CHAPITRE 8. CONFIGURATION DES CERTIFICATS ÉMIS PAR ADCS POUR L'AUTHENTIFICATION PAR CARTE À PUCE DANS IDM

Ce scénario décrit la situation suivante :

- Votre déploiement est basé sur une confiance inter-forêts entre Identity Management (IdM) et Active Directory (AD).
- Vous souhaitez autoriser l'authentification par carte à puce pour les utilisateurs dont les comptes sont stockés dans AD.
- Les certificats sont créés et stockés dans Active Directory Certificate Services (ADCS).

Pour une vue d'ensemble de l'authentification par carte à puce, voir Comprendre l'authentification par carte à puce.

La configuration s'effectue selon les étapes suivantes :

- Copie des certificats d'autorité de certification et d'utilisateur d'Active Directory vers le serveur et le client IdM
- Configuration du serveur IdM et des clients pour l'authentification par carte à puce à l'aide de certificats ADCS
- Conversion d'un fichier PFX (PKCS#12) pour pouvoir stocker le certificat et la clé privée dans la carte à puce
- Configuration des délais d'attente dans le fichier sssd.conf
- Création de règles de mappage de certificats pour l'authentification par carte à puce

Conditions préalables

- La gestion des identités (IdM) et la confiance dans Active Directory (AD) sont installées
  Pour plus de détails, voir Installer la confiance entre IdM et AD .
- Active Directory Certificate Services (ADCS) est installé et les certificats pour les utilisateurs sont générés

8.1. PARAMÈTRES DU SERVEUR WINDOWS REQUIS POUR LA CONFIGURATION DE LA CONFIANCE ET L'UTILISATION DU CERTIFICAT

Cette section résume ce qui doit être configuré sur Windows Server :

- Active Directory Certificate Services (ADCS) est installé
- L'autorité de certification est créée
- [Facultatif] Si vous utilisez l'inscription Web de l'autorité de certification, les services d'information Internet (IIS) doivent être configurés

Exporter le certificat :

- La clé doit avoir **2048** bits ou plus
- Inclure une clé privée
- Vous aurez besoin d'un certificat au format suivant : Échange d'informations personnelles -**PKCS #12(.PFX)**
    - Activer la confidentialité des certificats

8.2. COPIER DES CERTIFICATS À PARTIR D'ACTIVE DIRECTORY À L'AIDE DE SFTP

Pour pouvoir utiliser l'authentification par carte à puce, vous devez copier les fichiers de certificats suivants :

- Un certificat d'autorité de certification racine au format **CER**: **adcs-winserver-ca.cer** sur votre serveur IdM.
- Un certificat d'utilisateur avec une clé privée au format **PFX**: **aduser1.pfx** sur un client IdM.

NOTE

Cette procédure suppose que l'accès SSH est autorisé. Si SSH n'est pas disponible, l'utilisateur doit copier le fichier du serveur AD vers le serveur IdM et le client.

Procédure

1. Connectez-vous à partir de **the IdM server** et copiez le certificat racine de **adcs-winserver-ca.cer** sur le serveur IdM :

```
root@idmserver ~]# sftp Administrator@winserver.ad.example.com
Administrator@winserver.ad.example.com's password:
Connected to Administrator@winserver.ad.example.com.
sftp> cd <Path to certificates>
sftp> ls
adcs-winserver-ca.cer    aduser1.pfx
sftp>
sftp> get adcs-winserver-ca.cer
Fetching <Path to certificates>/adcs-winserver-ca.cer to adcs-winserver-ca.cer
<Path to certificates>/adcs-winserver-ca.cer                100%  1254    15KB/s 00:00
sftp quit
```

2. Connectez-vous à partir de **the IdM client** et copiez le certificat d'utilisateur de **aduser1.pfx** sur le client :

```
[root@client1 ~]# sftp Administrator@winserver.ad.example.com
Administrator@winserver.ad.example.com's password:
Connected to Administrator@winserver.ad.example.com.
sftp> cd /<Path to certificates>
sftp> get aduser1.pfx
Fetching <Path to certificates>/aduser1.pfx to aduser1.pfx
<Path to certificates>/aduser1.pfx                 100%  1254    15KB/s 00:00
sftp quit
```

Le certificat de l'autorité de certification est stocké dans le serveur IdM et les certificats des utilisateurs sont stockés sur la machine du client.

8.3. CONFIGURATION DU SERVEUR IDM ET DES CLIENTS POUR L'AUTHENTIFICATION PAR CARTE À PUCE À L'AIDE DE CERTIFICATS ADCS

Vous devez configurer le serveur IdM (Identity Management) et les clients pour pouvoir utiliser l'authentification par carte à puce dans l'environnement IdM. IdM inclut les scripts **ipa-advise** qui effectuent tous les changements nécessaires :

- installer les paquets nécessaires
- il configure le serveur et les clients IdM
- copier les certificats de l'autorité de certification dans les emplacements prévus

Vous pouvez exécuter **ipa-advise** sur votre serveur IdM.

Cette procédure décrit

- Sur un serveur IdM : Préparation du script **ipa-advise** pour configurer votre serveur IdM pour l'authentification par carte à puce.
- Sur un serveur IdM : Préparation du script **ipa-advise** pour configurer votre client IdM pour l'authentification par carte à puce.
- Sur un serveur IdM : Appliquer le script du serveur **ipa-advise** sur le serveur IdM en utilisant le certificat AD.
- Déplacement du script client vers la machine client IdM.
- Sur un client IdM : Appliquer le script du client **ipa-advise** sur le client IdM en utilisant le certificat AD.

Conditions préalables

- Le certificat a été copié sur le serveur IdM.
- Obtenir le ticket Kerberos.
- Connectez-vous en tant qu'utilisateur disposant de droits d'administration.

Procédure

1. Sur le serveur IdM, utilisez le script **ipa-advise** pour configurer un client :

```
[root@idmserver ~]# ipa-advise config-client-for-smart-card-auth > sc_client.sh
```

2. Sur le serveur IdM, utilisez le script **ipa-advise** pour configurer un serveur :

```
[root@idmserver ~]# ipa-advise config-server-for-smart-card-auth > sc_server.sh
```

3. Sur le serveur IdM, exécuter le script :

```
[root@idmserver ~]# sh -x sc_server.sh adcs-winserver-ca.cer
```

- Il configure le serveur HTTP Apache de l'IdM.
- Il active la cryptographie à clé publique pour l'authentification initiale dans Kerberos (PKINIT) sur le centre de distribution de clés (KDC).
- Il configure l'interface Web IdM pour qu'elle accepte les demandes d'autorisation de carte à puce.

4. Copiez le script **sc_client.sh** sur le système client :

```
[root@idmserver ~]# scp sc_client.sh root@client1.idm.example.com:/root
Password:
sc_client.sh                  100%  2857   1.6MB/s   00:00
```

5. Copiez le certificat Windows sur le système client :

```
[root@idmserver ~]# scp adcs-winserver-ca.cer root@client1.idm.example.com:/root
Password:
adcs-winserver-ca.cer                 100%  1254   952.0KB/s   00:00
```

6. Sur le système client, exécutez le script client :

```
[root@idmclient1 ~]# sh -x sc_client.sh adcs-winserver-ca.cer
```

Le certificat de l'autorité de certification est installé dans le bon format sur le serveur IdM et les systèmes clients, et l'étape suivante consiste à copier les certificats des utilisateurs sur la carte à puce elle-même.

8.4. CONVERSION DU FICHIER PFX

Avant d'enregistrer le fichier PFX (PKCS#12) dans la carte à puce, vous devez

- convertir le fichier au format PEM
- extraire la clé privée et le certificat dans deux fichiers différents

Conditions préalables

- Le fichier PFX est copié sur la machine du client IdM.

Procédure

1. Sur le client IdM, dans le format PEM :

```
[root@idmclient1 ~]# openssl pkcs12 -in aduser1.pfx -out aduser1_cert_only.pem -clcerts -
nodes
Enter Import Password:
```

2. Extraire la clé dans un fichier séparé :

```
[root@idmclient1 ~]# openssl pkcs12 -in adduser1.pfx -nocerts -out adduser1.pem >
aduser1.key
```

3. Extraire le certificat public dans un fichier séparé :

```
[root@idmclient1 ~]# openssl pkcs12 -in adduser1.pfx -clcerts -nokeys -out
aduser1_cert_only.pem > aduser1.crt
```

À ce stade, vous pouvez enregistrer les adresses **aduser1.key** et **aduser1.crt** dans la carte à puce.

8.5. INSTALLATION D'OUTILS DE GESTION ET D'UTILISATION DES CARTES À PUCE

Pour configurer votre carte à puce, vous avez besoin d'outils qui peuvent générer des certificats et les stocker sur une carte à puce.

Vous devez :

- Installez le paquet **gnutls-utils**, qui vous aide à gérer les certificats.
- Installez le paquetage **opensc**, qui fournit un ensemble de bibliothèques et d'utilitaires pour travailler avec des cartes à puce.
- Démarrez le service **pcscd**, qui communique avec le lecteur de cartes à puce.

Procédure

1. Installez les paquets **opensc** et **gnutls-utils**:

```
# dnf -y install opensc gnutls-utils
```

2. Démarrez le service **pcscd**.

```
# systemctl start pcscd
```

Vérifiez que le service **pcscd** est opérationnel.

8.6. PRÉPARATION DE VOTRE CARTE À PUCE ET TÉLÉCHARGEMENT DE VOS CERTIFICATS ET CLÉS SUR VOTRE CARTE À PUCE

Cette section décrit la configuration de la carte à puce avec l'outil **pkcs15-init**, qui vous aide à configurer :

- Effacer votre carte à puce
- Définition de nouveaux codes PIN et de clés de déblocage de code PIN (PUK) en option
- Création d'un nouvel emplacement sur la carte à puce
- Stockage du certificat, de la clé privée et de la clé publique dans la fente
- Si nécessaire, verrouiller les paramètres de la carte à puce, car certaines cartes à puce nécessitent ce type de finalisation

NOTE

L'outil **pkcs15-init** peut ne pas fonctionner avec toutes les cartes à puce. Vous devez utiliser les outils qui fonctionnent avec la carte à puce que vous utilisez.

Conditions préalables

- Le paquet **opensc**, qui comprend l'outil **pkcs15-init**, est installé.
  Pour plus de détails, voir Installation des outils de gestion et d'utilisation des cartes à puce .
- La carte est insérée dans le lecteur et connectée à l'ordinateur.
- Vous disposez de la clé privée, de la clé publique et du certificat à stocker sur la carte à puce. Dans cette procédure, **testuser.key**, **testuserpublic.key**, et **testuser.crt** sont les noms utilisés pour la clé privée, la clé publique et le certificat.
- Vous disposez du code PIN de l'utilisateur de votre carte à puce actuelle et du code PIN de l'agent de sécurité (SO-PIN).

Procédure

1. Effacez votre carte à puce et authentifiez-vous avec votre code PIN :

```
$ pkcs15-init --erase-card --use-default-transport-keys
Using reader with a card: Reader name
PIN [Security Officer PIN] required.
Please enter PIN [Security Officer PIN]:
```

   La carte a été effacée.

2. Initialisez votre carte à puce, définissez votre code PIN et PUK d'utilisateur, ainsi que le code PIN et PUK de votre responsable de la sécurité :

```
$ pkcs15-init --create-pkcs15 --use-default-transport-keys \
    --pin 963214 --puk 321478 --so-pin 65498714 --so-puk 784123
Using reader with a card: Reader name
```

   L'outil **pcks15-init** crée un nouvel emplacement sur la carte à puce.

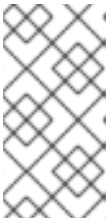
3. Définir l'étiquette et l'ID d'authentification pour l'emplacement :

```
$ pkcs15-init --store-pin --label testuser \
    --auth-id 01 --so-pin 65498714 --pin 963214 --puk 321478
Using reader with a card: Reader name
```

   L'étiquette est définie sur une valeur lisible par l'homme, dans ce cas, **testuser**. L'adresse **auth-id** doit être composée de deux valeurs hexadécimales ; dans ce cas, elle est fixée à **01**.

4. Stockez et étiquetez la clé privée dans le nouvel emplacement de la carte à puce :

```
$ pkcs15-init --store-private-key testuser.key --label testuser_key \
    --auth-id 01 --id 01 --pin 963214
Using reader with a card: Reader name
```



NOTE

La valeur que vous indiquez pour **--id** doit être la même lorsque vous stockez votre clé privée et votre certificat à l'étape suivante. Il est recommandé de spécifier votre propre valeur pour **--id**, sinon l'outil calculera une valeur plus complexe.

5. Stockez et étiquetez le certificat dans le nouvel emplacement de la carte à puce :

```
$ pkcs15-init --store-certificate testuser.crt --label testuser_crt \
    --auth-id 01 --id 01 --format pem --pin 963214
Using reader with a card: Reader name
```

6. (Facultatif) Stockez et étiquetez la clé publique dans le nouvel emplacement de la carte à puce :

```
$ pkcs15-init --store-public-key testuserpublic.key
    --label testuserpublic_key --auth-id 01 --id 01 --pin 963214
Using reader with a card: Reader name
```

NOTE

Si la clé publique correspond à une clé privée ou à un certificat, indiquez le même ID que celui de la clé privée ou du certificat.

7. (Facultatif) Certaines cartes à puce exigent que vous finalisiez la carte en verrouillant les paramètres :

```
$ pkcs15-init -F
```

À ce stade, votre carte à puce comprend le certificat, la clé privée et la clé publique dans l'emplacement nouvellement créé. Vous avez également créé votre code PIN et PUK d'utilisateur ainsi que le code PIN et PUK de l'agent de sécurité.

8.7. CONFIGURATION DES DÉLAIS D'ATTENTE DANS SSSD.CONF

L'authentification à l'aide d'un certificat de carte à puce peut prendre plus de temps que les délais par défaut utilisés par SSSD. L'expiration du délai peut être causée par :

- lecteur lent
- un transfert d'un dispositif physique vers un environnement virtuel
- trop de certificats stockés sur la carte à puce
- réponse lente du répondeur OCSP (Online Certificate Status Protocol) si OCSP est utilisé pour vérifier les certificats

Dans ce cas, vous pouvez prolonger les délais suivants dans le fichier **sssd.conf**, par exemple jusqu'à 60 secondes :

- **p11_child_timeout**
- **krb5_auth_timeout**

Conditions préalables

- Vous devez être connecté en tant que root.

Procédure

1. Ouvrez le fichier **sssd.conf**:

```
[root@idmclient1 ~]# vim /etc/sssd/sssd.conf
```

2. Modifier la valeur de **p11_child_timeout**:

```
[pam]
p11_child_timeout = 60
```

3. Modifier la valeur de **krb5_auth_timeout**:

```
[domain/IDM.EXAMPLE.COM]
krb5_auth_timeout = 60
```

4. Sauvegarder les paramètres.

Maintenant, l'interaction avec la carte à puce est autorisée pendant 1 minute (60 secondes) avant que l'authentification n'échoue avec un délai d'attente.

8.8. CRÉATION DE RÈGLES DE MAPPAGE DE CERTIFICATS POUR L'AUTHENTIFICATION PAR CARTE À PUCE

Si vous souhaitez utiliser un seul certificat pour un utilisateur qui possède des comptes dans AD (Active Directory) et dans IdM (Identity Management), vous pouvez créer une règle de mappage des certificats sur le serveur IdM.

Après avoir créé une telle règle, l'utilisateur peut s'authentifier avec sa carte à puce dans les deux domaines.

Pour plus d'informations sur les règles de mappage des certificats, voir Règles de mappage des certificats pour la configuration de l'authentification sur les cartes à puce.

CHAPITRE 9. CONFIGURATION DES RÈGLES DE MAPPAGE DES CERTIFICATS DANS LA GESTION DES IDENTITÉS

9.1. RÈGLES DE MAPPAGE DES CERTIFICATS POUR LA CONFIGURATION DE L'AUTHENTIFICATION SUR LES CARTES À PUCE

Les règles de mappage de certificats sont un moyen pratique de permettre aux utilisateurs de s'authentifier à l'aide de certificats dans des scénarios où l'administrateur de la gestion des identités (IdM) n'a pas accès aux certificats de certains utilisateurs. Ce manque d'accès est généralement dû au fait que les certificats ont été délivrés par une autorité de certification externe. Un cas d'utilisation particulier est représenté par les certificats émis par le système de certification d'un Active Directory (AD) avec lequel le domaine IdM entretient une relation de confiance.

Les règles de mappage des certificats sont également pratiques si l'environnement IdM est vaste et que de nombreux utilisateurs utilisent des cartes à puce. Dans ce cas, l'ajout de certificats complets peut s'avérer compliqué. Le sujet et l'émetteur sont prévisibles dans la plupart des scénarios et donc plus faciles à ajouter à l'avance que le certificat complet. En tant qu'administrateur système, vous pouvez créer une règle de mappage de certificats et ajouter des données de mappage de certificats à une entrée utilisateur avant même qu'un certificat ne soit délivré à un utilisateur particulier. Une fois le certificat émis, l'utilisateur peut se connecter à l'aide du certificat, même si le certificat complet n'a pas encore été téléchargé dans l'entrée utilisateur.

En outre, comme les certificats doivent être renouvelés à intervalles réguliers, les règles de mappage des certificats réduisent la charge administrative. Lorsque le certificat d'un utilisateur est renouvelé, l'administrateur ne doit pas mettre à jour l'entrée de l'utilisateur. Par exemple, si le mappage est basé sur les valeurs **Subject** et **Issuer**, et si le nouveau certificat a le même sujet et le même émetteur que l'ancien, le mappage s'applique toujours. En revanche, si le certificat complet est utilisé, l'administrateur doit télécharger le nouveau certificat dans l'entrée utilisateur pour remplacer l'ancien.

Pour configurer le mappage des certificats :

1. Un administrateur doit charger les données de mappage du certificat (généralement l'émetteur et le sujet) ou le certificat complet dans un compte utilisateur.

2. Un administrateur doit créer une règle de mappage de certificats pour permettre à un utilisateur de se connecter avec succès à l'IdM

    a. dont le compte contient une entrée de données de mappage de certificats

    b. dont la saisie des données de mappage du certificat correspond aux informations figurant sur le certificat

    Pour plus d'informations sur les différents composants d'une règle de correspondance et sur la manière de les obtenir et de les utiliser, voir Composants d'une règle de correspondance des identités dans IdM et Obtention de l'émetteur d'un certificat en vue de son utilisation dans une règle de correspondance.

Ensuite, lorsque l'utilisateur final présente le certificat, stocké soit dans le système de fichiers, soit sur une carte à puce, l'authentification est réussie.

9.1.1. Règles de mappage des certificats pour les trusts avec les domaines Active Directory

Cette section décrit les différents cas d'utilisation du mappage de certificats qui sont possibles si un déploiement IdM est en relation de confiance avec un domaine Active Directory (AD).

Les règles de mappage des certificats sont un moyen pratique d'autoriser l'accès aux ressources IdM pour les utilisateurs qui possèdent des certificats de carte à puce émis par le système de certification AD de confiance. Selon la configuration d'AD, les scénarios suivants sont possibles :

- Si le certificat est émis par AD mais que l'utilisateur et le certificat sont stockés dans IdM, le mappage et l'ensemble du traitement de la demande d'authentification ont lieu du côté d'IdM. Pour plus de détails sur la configuration de ce scénario, voir Configuration du mappage des certificats pour les utilisateurs stockés dans IdM

- Si l'utilisateur est enregistré dans AD, le traitement de la demande d'authentification a lieu dans AD. Il existe trois sous-cas différents :

  - L'entrée utilisateur AD contient l'intégralité du certificat. Pour plus de détails sur la configuration de l'IdM dans ce scénario, voir Configuration du mappage de certificats pour les utilisateurs dont l'entrée utilisateur AD contient le certificat entier.

  - AD est configuré pour associer des certificats d'utilisateur à des comptes d'utilisateur. Dans ce cas, l'entrée utilisateur AD ne contient pas l'intégralité du certificat, mais un attribut appelé **altSecurityIdentities**. Pour plus d'informations sur la configuration de l'IdM dans ce scénario, voir Configuration du mappage de certificats si AD est configuré pour mapper des certificats d'utilisateur sur des comptes d'utilisateur.

  - L'entrée de l'utilisateur AD ne contient ni le certificat complet ni les données de mappage. Dans ce cas, la seule solution consiste à utiliser la commande **ipa idoverrideuser-add** pour ajouter le certificat complet à l'ID override de l'utilisateur AD dans IdM. Pour plus de détails, voir Configuration du mappage de certificats si l'entrée de l'utilisateur AD ne contient pas de certificat ou de données de mappage.

.

9.1.2. Composants d'une règle de mappage d'identité dans IdM

Cette section décrit les composants d'un site *identity mapping rule* dans IdM et la manière de les configurer. Chaque composant a une valeur par défaut que vous pouvez remplacer. Vous pouvez définir les composants dans l'interface Web ou dans l'interface de ligne de commande. Dans la CLI, la règle de mappage d'identité est créée à l'aide de la commande **ipa certmaprule-add**.

Règle de cartographie

Le composant règle de mappage associe (ou *maps*) un certificat à un ou plusieurs comptes d'utilisateurs. La règle définit un filtre de recherche LDAP qui associe un certificat au compte d'utilisateur voulu.
Les certificats délivrés par différentes autorités de certification (AC) peuvent avoir des propriétés différentes et être utilisés dans des domaines différents. C'est pourquoi l'IdM n'applique pas les règles de mappage de manière inconditionnelle, mais uniquement aux certificats appropriés. Les certificats appropriés sont définis à l'aide de *matching rules*.

Notez que si vous laissez l'option "mapping rule" vide, les certificats sont recherchés dans l'attribut **userCertificate** sous la forme d'un fichier binaire encodé en DER.

Définissez la règle de mappage dans le CLI en utilisant l'option **--maprule**.

Règle de correspondance

Le composant règle de correspondance sélectionne un certificat auquel vous souhaitez appliquer la règle de correspondance. La règle de correspondance par défaut fait correspondre les certificats avec l'utilisation **digitalSignature key** et **clientAuth extended key**.
Définissez la règle de correspondance dans le CLI en utilisant l'option **--matchrule**.

Liste des domaines

La liste des domaines spécifie les domaines d'identité dans lesquels vous souhaitez que l'IdM recherche les utilisateurs lors du traitement des règles de mappage d'identité. Si l'option n'est pas spécifiée, l'IdM recherche les utilisateurs uniquement dans le domaine local auquel appartient le client IdM.
Définissez le domaine dans le CLI en utilisant l'option **--domain**.

Priorité

Lorsque plusieurs règles s'appliquent à un certificat, la règle ayant la priorité la plus élevée est prioritaire. Toutes les autres règles sont ignorées.

- Plus la valeur numérique est faible, plus la priorité de la règle de mise en correspondance des identités est élevée. Par exemple, une règle de priorité 1 est plus prioritaire qu'une règle de priorité 2.
- Si une règle n'a pas de valeur de priorité définie, elle a la priorité la plus basse.

Définissez la priorité de la règle de mappage dans le CLI à l'aide de l'option **--priority**.

Exemple de règle de mappage de certificats

Définir, à l'aide de la CLI, une règle de mappage de certificats appelée **simple_rule** qui autorise l'authentification d'un certificat émis par **Smart Card CA** de l'organisation **EXAMPLE.ORG** tant que le **Subject** de ce certificat correspond à une entrée **certmapdata** dans un compte d'utilisateur de l'IdM :

```
# ipa certmaprule-add simple_rule --matchrule '<ISSUER>CN=Smart Card CA,O=EXAMPLE.ORG' --maprule '(ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>{subject_dn!nss_x500})'
```

9.1.3. Obtenir l'émetteur d'un certificat pour l'utiliser dans une règle de correspondance

Cette procédure décrit comment obtenir les informations relatives à l'émetteur d'un certificat afin de pouvoir les copier et les coller dans la règle de correspondance d'une règle de mappage de certificats. Pour obtenir le format de l'émetteur requis par une règle de correspondance, utilisez l'utilitaire **openssl x509**.

Conditions préalables

- Vous disposez du certificat d'utilisateur au format **.pem** ou **.crt**

Procédure

1. Obtenez les informations sur l'utilisateur à partir du certificat. Utilisez l'utilitaire d'affichage et de signature de certificats **openssl x509** avec :
   - l'option **-noout** pour empêcher la sortie d'une version encodée de la requête

- l'option **-issuer** pour éditer le nom de l'émetteur
- l'option **-in** pour spécifier le nom du fichier d'entrée à partir duquel le certificat doit être lu
- l'option **-nameopt** avec la valeur **RFC2253** pour afficher la sortie avec le nom distinctif relatif (RDN) le plus spécifique en premier
  Si le fichier d'entrée contient un certificat de gestion d'identité, la sortie de la commande montre que l'émetteur est défini à l'aide des informations de **Organisation**:

```
# openssl x509 -noout -issuer -in idm_user.crt -nameopt RFC2253
issuer=CN=Certificate Authority,O=REALM.EXAMPLE.COM
```

  Si le fichier d'entrée contient un certificat Active Directory, la sortie de la commande montre que l'émetteur est défini à l'aide des informations de **Domain Component**:

```
# openssl x509 -noout -issuer -in ad_user.crt -nameopt RFC2253
issuer=CN=AD-WIN2012R2-CA,DC=AD,DC=EXAMPLE,DC=COM
```

2. Optionnellement, pour créer une nouvelle règle de mappage dans le CLI basée sur une règle de correspondance qui spécifie que l'émetteur du certificat doit être le **AD-WIN2012R2-CA** extrait du domaine **ad.example.com** et que le sujet du certificat doit correspondre à l'entrée **certmapdata** dans un compte d'utilisateur dans l'IdM :

```
# ipa certmaprule-add simple_rule --matchrule '<ISSUER>CN=AD-WIN2012R2-
CA,DC=AD,DC=EXAMPLE,DC=COM' --maprule '(ipacertmapdata=X509:<I>
{issuer_dn!nss_x500}<S>{subject_dn!nss_x500})'
```

9.1.4. Ressources supplémentaires

- Voir la page de manuel **sss-certmap(5)**.

9.2. CONFIGURATION DU MAPPAGE DES CERTIFICATS POUR LES UTILISATEURS STOCKÉS DANS IDM

Cette histoire d'utilisateur décrit les étapes qu'un administrateur système doit suivre pour activer le mappage de certificats dans IdM si l'utilisateur pour lequel l'authentification par certificat est configurée est stocké dans IdM. Cette section décrit les points suivants :

- Comment configurer une règle de mappage de certificats pour que les utilisateurs de l'IdM dont les certificats correspondent aux conditions spécifiées dans la règle de mappage et dans leurs entrées de données de mappage de certificats puissent s'authentifier auprès de l'IdM.
- Comment ajouter des données de mappage de certificats à une entrée utilisateur IdM afin que l'utilisateur puisse s'authentifier à l'aide de plusieurs certificats, à condition qu'ils contiennent tous les valeurs spécifiées dans l'entrée de données de mappage de certificats.

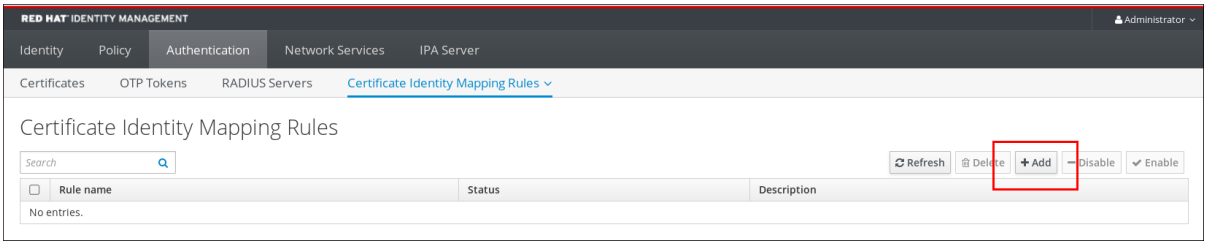
Conditions préalables

- L'utilisateur dispose d'un compte dans l'IdM.
- L'administrateur dispose soit du certificat complet, soit des données de mappage du certificat à ajouter à l'entrée de l'utilisateur.

9.2.1. Ajout d'une règle de mappage de certificats dans l'interface web IdM

1. Connectez-vous à l'interface web IdM en tant qu'administrateur.
2. Naviguez vers **Authentication → Certificate Identity Mapping Rules → Certificate Identity Mapping Rules**.
3. Cliquez sur **Add**.

Figure 9.1. Ajout d'une nouvelle règle de mappage de certificats dans l'interface web IdM



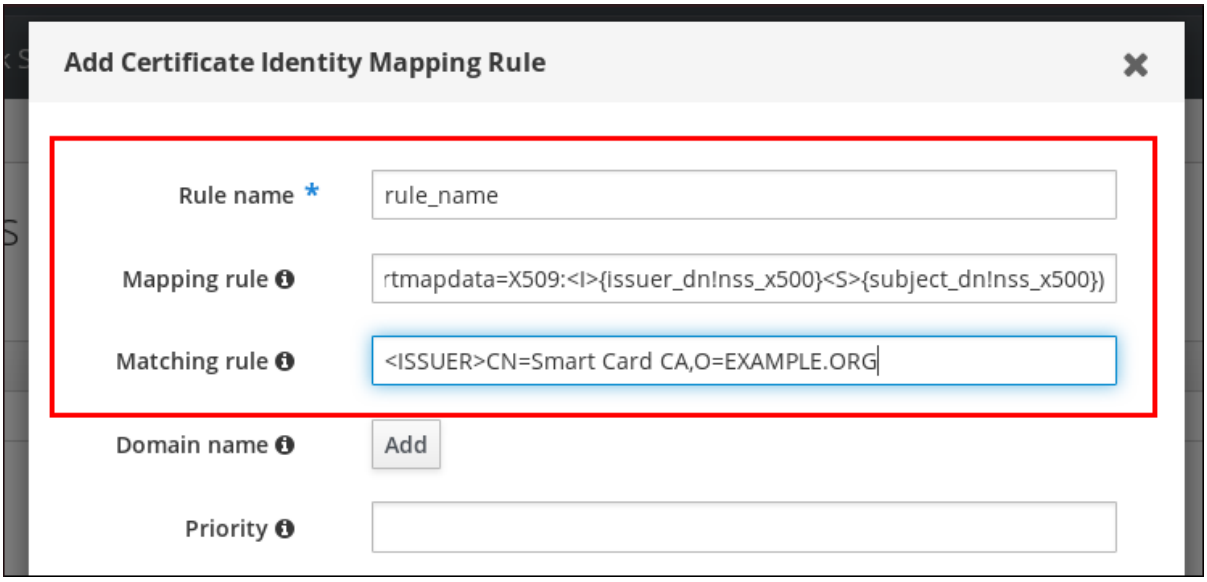
4. Saisissez le nom de la règle.
5. Saisir la règle de mise en correspondance. Par exemple, pour que l'IdM recherche les entrées **Issuer** et **Subject** dans tout certificat qui lui est présenté et base sa décision d'authentifier ou non sur les informations trouvées dans ces deux entrées du certificat présenté :

```
(ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>{subject_dn!nss_x500})
```

6. Saisir la règle de correspondance. Par exemple, pour n'autoriser que les certificats émis par **Smart Card CA** de l'organisation **EXAMPLE.ORG** à authentifier les utilisateurs de l'IdM :

```
<ISSUER>CN=Smart Card CA,O=EXAMPLE.ORG
```

Figure 9.2. Saisir les détails d'une règle de mappage de certificats dans l'interface web IdM



7. Cliquez sur **Add** en bas de la boîte de dialogue pour ajouter la règle et fermer la boîte.
8. Le System Security Services Daemon (SSSD) relit périodiquement les règles de mappage des certificats. Pour forcer le chargement immédiat de la règle nouvellement créée, redémarrez SSSD :

```
# systemctl restart sssd
```

Vous disposez à présent d'une règle de mappage de certificats qui compare le type de données spécifié dans la règle de mappage qu'elle trouve sur un certificat de carte à puce avec les données de mappage de certificats dans vos entrées d'utilisateur IdM. Lorsqu'elle trouve une correspondance, elle authentifie l'utilisateur correspondant.

9.2.2. Ajout d'une règle de mappage de certificats dans la CLI IdM

1. Obtenir les informations d'identification de l'administrateur :

```
# kinit admin
```

2. Saisir la règle de mise en correspondance et la règle de correspondance sur laquelle la règle de mise en correspondance est basée. Par exemple, pour que l'IdM recherche les entrées **Issuer** et **Subject** dans tout certificat présenté et fonde sa décision d'authentification ou non sur les informations trouvées dans ces deux entrées du certificat présenté, en ne reconnaissant que les certificats émis par l'organisation **Smart Card CA** de l'organisation **EXAMPLE.ORG**:

```
# ipa certmaprule-add rule_name --matchrule '<ISSUER>CN=Smart Card CA,O=EXAMPLE.ORG' --maprule '(ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>{subject_dn!nss_x500})'
----------------------------------------------------
Added Certificate Identity Mapping Rule "rule_name"
----------------------------------------------------
  Rule name: rule_name
  Mapping rule: (ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>{subject_dn!nss_x500})
  Matching rule: <ISSUER>CN=Smart Card CA,O=EXAMPLE.ORG
  Enabled: TRUE
```

3. Le System Security Services Daemon (SSSD) relit périodiquement les règles de mappage des certificats. Pour forcer le chargement immédiat de la règle nouvellement créée, redémarrez SSSD :

```
# systemctl restart sssd
```

Vous disposez à présent d'une règle de mappage de certificats qui compare le type de données spécifié dans la règle de mappage qu'elle trouve sur un certificat de carte à puce avec les données de mappage de certificats dans vos entrées d'utilisateur IdM. Lorsqu'elle trouve une correspondance, elle authentifie l'utilisateur correspondant.

9.2.3. Ajout de données de mappage de certificats à une entrée utilisateur dans l'interface web IdM

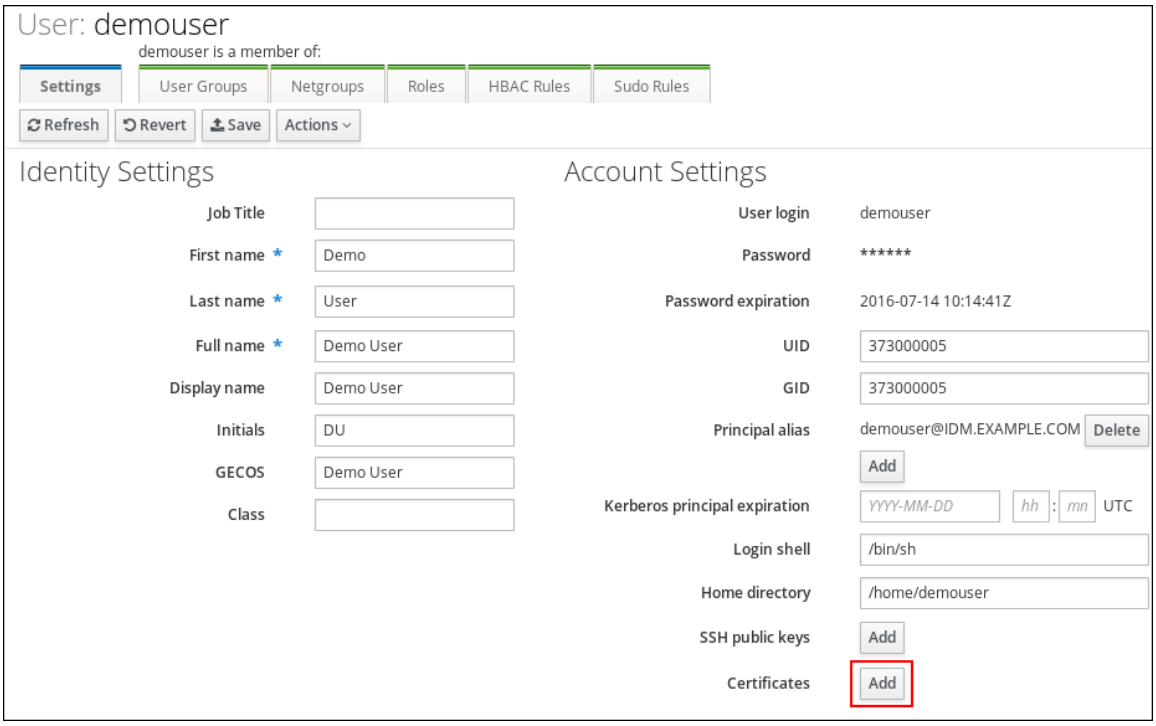
1. Se connecter à l'interface web IdM en tant qu'administrateur.
2. Naviguez vers **Users → Active users → idm_user**.
3. Recherchez l'option **Certificate mapping data** et cliquez sur **Add**.
4. Si vous disposez du certificat de **idm_user**:
   a. Dans l'interface ligne de commande, affichez le certificat à l'aide de l'utilitaire **cat** ou d'un éditeur de texte :

```
[root@server ~]# cat idm_user_certificate.pem
-----BEGIN CERTIFICATE-----
MIIFFTCCA/2gAwIBAgIBEjANBgkqhkiG9w0BAQsFADA6MRgwFgYDVQQKDA9JRE0u
RVhBTVBMRS5DT00xHjAcBgNVBAMMFUNlcnRpZmljYXRlIEF1dGhvcml0eTAeFw0x
ODA5MDIxODE1MzlaFw0yMDA5MDIxODE1MzlaMCwxGDAWBgNVBAoMD0lETS5FWE
FN
[...output truncated...]
```

b. Copier le certificat.

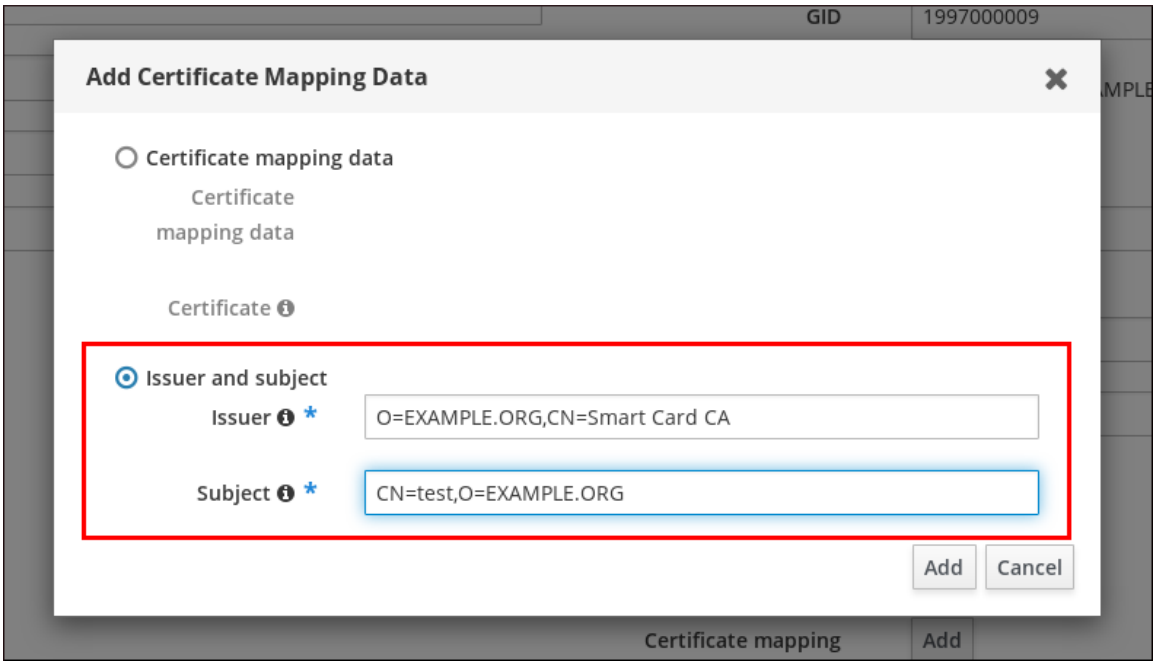
c. Dans l'interface web IdM, cliquez sur **Add** à côté de **Certificate** et collez le certificat dans la fenêtre qui s'ouvre.

Figure 9.3. Ajout des données de mappage du certificat d'un utilisateur : certificat



Alternativement, si vous ne disposez pas du certificat de **idm_user** mais connaissez le **Issuer** et le **Subject** du certificat, cochez le bouton radio de **Issuer and subject** et introduisez les valeurs dans les deux cases respectives.

Figure 9.4. Ajout des données de mappage du certificat d'un utilisateur : émetteur et sujet



5. Cliquez sur **Add**.

6. Si vous avez accès à l'ensemble du certificat au format **.pem**, vérifiez que l'utilisateur et le certificat sont liés :

   a. Utilisez l'utilitaire **sss_cache** pour invalider l'enregistrement de **idm_user** dans le cache SSSD et forcer le rechargement des informations de **idm_user**:

   ```
   # sss_cache -u idm_user
   ```

   b. Exécutez la commande **ipa certmap-match** avec le nom du fichier contenant le certificat de l'utilisateur IdM :

   ```
   # ipa certmap-match idm_user_cert.pem
   ----------
   1 user matched
   ----------
    Domain: IDM.EXAMPLE.COM
    User logins: idm_user
   ----------------------------
   Number of entries returned 1
   ----------------------------
   ```

   Le résultat confirme que des données de mappage de certificats ont été ajoutées à **idm_user** et qu'une règle de mappage correspondante existe. Cela signifie que vous pouvez utiliser n'importe quel certificat correspondant aux données de mappage de certificats définies pour vous authentifier en tant que **idm_user**.

9.2.4. Ajout de données de mappage de certificats à une entrée utilisateur dans la CLI IdM

1. Obtenir les informations d'identification de l'administrateur :

```
# kinit admin
```

2. Si vous disposez du certificat de **idm_user**, ajoutez le certificat au compte d'utilisateur à l'aide de la commande **ipa user-add-cert**:

```
# CERT=`cat idm_user_cert.pem | tail -n +2| head -n -1 | tr -d '\r\n'\`
# ipa user-add-certmapdata idm_user --certificate $CERT
```

Alternativement, si vous n'avez pas le certificat de **idm_user** à votre disposition mais que vous connaissez le **Issuer** et le **Subject** du certificat de idm_user :

```
# ipa user-add-certmapdata idm_user --subject "O=EXAMPLE.ORG,CN=test" --issuer
"CN=Smart Card CA,O=EXAMPLE.ORG"
----------------------------------------
Added certificate mappings to user "idm_user"
----------------------------------------
  User login: idm_user
  Certificate mapping data: X509:<I>O=EXAMPLE.ORG,CN=Smart Card
CA<S>CN=test,O=EXAMPLE.ORG
```

3. Si vous avez accès à l'ensemble du certificat au format **.pem**, vérifiez que l'utilisateur et le certificat sont liés :

   a. Utilisez l'utilitaire **sss_cache** pour invalider l'enregistrement de **idm_user** dans le cache SSSD et forcer le rechargement des informations de **idm_user**:

   ```
   # sss_cache -u idm_user
   ```

   b. Exécutez la commande **ipa certmap-match** avec le nom du fichier contenant le certificat de l'utilisateur IdM :

   ```
   # ipa certmap-match idm_user_cert.pem
   --------------
   1 user matched
   --------------
    Domain: IDM.EXAMPLE.COM
    User logins: idm_user
   ----------------------------
   Number of entries returned 1
   ----------------------------
   ```

   Le résultat confirme que des données de mappage de certificats ont été ajoutées à **idm_user** et qu'une règle de mappage correspondante existe. Cela signifie que vous pouvez utiliser n'importe quel certificat correspondant aux données de mappage de certificats définies pour vous authentifier en tant que **idm_user**.

9.3. CONFIGURATION DU MAPPAGE DES CERTIFICATS POUR LES UTILISATEURS DONT L'ENTRÉE AD CONTIENT L'INTÉGRALITÉ DU CERTIFICAT

Cette histoire d'utilisateur décrit les étapes nécessaires pour activer le mappage de certificats dans IdM si le déploiement d'IdM est en confiance avec Active Directory (AD), l'utilisateur est stocké dans AD et l'entrée de l'utilisateur dans AD contient le certificat entier.

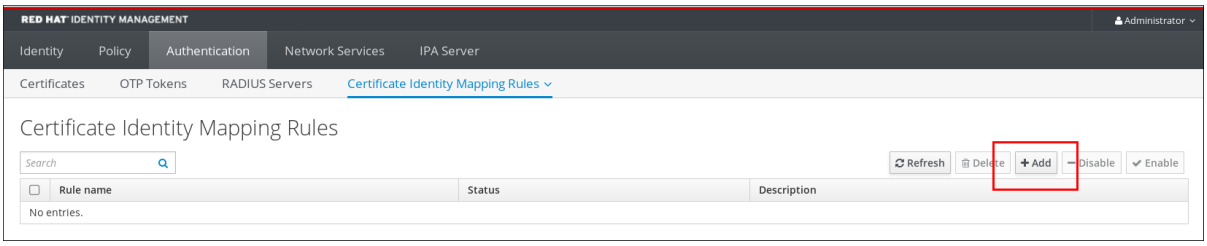
Conditions préalables

- L'utilisateur n'a pas de compte dans IdM.
- L'utilisateur dispose d'un compte dans AD qui contient un certificat.
- L'administrateur IdM a accès aux données sur lesquelles la règle de mappage des certificats IdM peut être basée.

9.3.1. Ajout d'une règle de mappage de certificats dans l'interface web IdM

1. Se connecter à l'interface web IdM en tant qu'administrateur.
2. Naviguez vers **Authentication → Certificate Identity Mapping Rules → Certificate Identity Mapping Rules**.
3. Cliquez sur **Add**.

Figure 9.5. Ajout d'une nouvelle règle de mappage de certificats dans l'interface web IdM



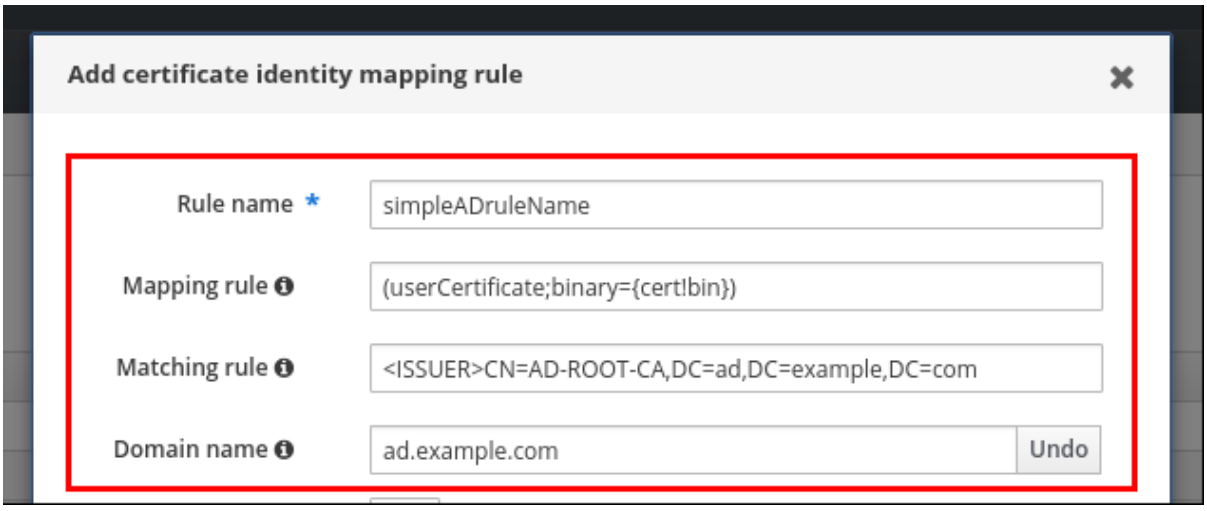
4. Saisissez le nom de la règle.
5. Saisissez la règle de mappage. Pour que l'ensemble du certificat présenté à IdM pour l'authentification soit comparé à ce qui est disponible dans AD :

```
(userCertificate;binary={cert!bin})
```

6. Saisissez la règle de correspondance. Par exemple, pour autoriser uniquement les certificats émis par **AD-ROOT-CA** du domaine **AD.EXAMPLE.COM** à s'authentifier :

```
<ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
```

Figure 9.6. Règle de mappage des certificats pour un utilisateur dont le certificat est stocké dans AD



7. Cliquez sur **Add**.

8. Le System Security Services Daemon (SSSD) relit périodiquement les règles de mappage des certificats. Pour forcer le chargement immédiat de la règle nouvellement créée, redémarrez SSSD dans le CLI: :

```
# systemctl restart sssd
```

9.3.2. Ajout d'une règle de mappage de certificats dans la CLI IdM

1. Obtenir les informations d'identification de l'administrateur :

```
# kinit admin
```

2. Saisissez la règle de mappage et la règle de correspondance sur laquelle la règle de mappage est basée. Comparer l'ensemble du certificat présenté pour l'authentification à ce qui est disponible dans AD, en autorisant uniquement les certificats émis par **AD-ROOT-CA** du domaine **AD.EXAMPLE.COM** pour l'authentification :

```
# ipa certmaprule-add simpleADrule --matchrule '<ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com' --maprule '(userCertificate;binary={cert!bin})' --domain ad.example.com
----------------------------------------------------
Added Certificate Identity Mapping Rule "simpleADrule"
----------------------------------------------------
  Rule name: simpleADrule
  Mapping rule: (userCertificate;binary={cert!bin})
  Matching rule: <ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
  Domain name: ad.example.com
  Enabled: TRUE
```

3. Le System Security Services Daemon (SSSD) relit périodiquement les règles de mappage des certificats. Pour forcer le chargement immédiat de la règle nouvellement créée, redémarrez SSSD :

```
# systemctl restart sssd
```

9.4. CONFIGURATION DU MAPPAGE DES CERTIFICATS SI AD EST CONFIGURÉ POUR MAPPER LES CERTIFICATS D'UTILISATEUR AUX COMPTES D'UTILISATEUR

Cette histoire d'utilisateur décrit les étapes nécessaires pour activer le mappage de certificats dans IdM si le déploiement d'IdM est en confiance avec Active Directory (AD), l'utilisateur est stocké dans AD et l'entrée de l'utilisateur dans AD contient des données de mappage de certificats.

Conditions préalables

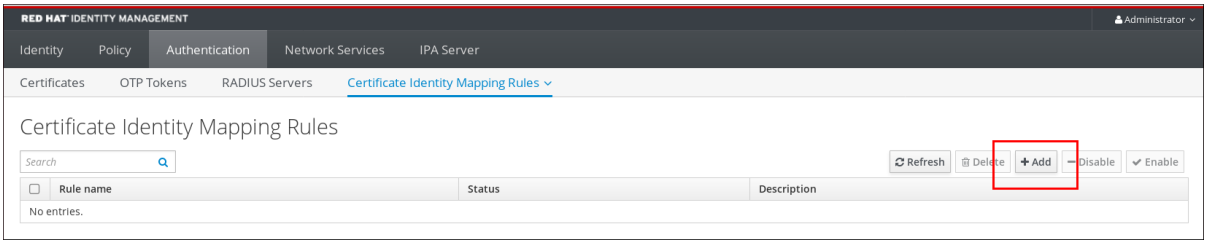
- L'utilisateur n'a pas de compte dans IdM.
- L'utilisateur possède un compte dans AD qui contient l'attribut **altSecurityIdentities**, l'équivalent AD de l'attribut IdM **certmapdata**.

- L'administrateur IdM a accès aux données sur lesquelles la règle de mappage des certificats IdM peut être basée.

9.4.1. Ajout d'une règle de mappage de certificats dans l'interface web IdM

1. Se connecter à l'interface web IdM en tant qu'administrateur.
2. Naviguez vers **Authentication → Certificate Identity Mapping Rules → Certificate Identity Mapping Rules**.
3. Cliquez sur **Add**.

Figure 9.7. Ajout d'une nouvelle règle de mappage de certificats dans l'interface web IdM



4. Saisissez le nom de la règle.
5. Saisissez la règle de mappage. Par exemple, pour que AD DC recherche les entrées **Issuer** et **Subject** dans tout certificat présenté et base sa décision d'authentification ou non sur les informations trouvées dans ces deux entrées du certificat présenté :

```
(altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}<S>{subject_dn!ad_x500})
```

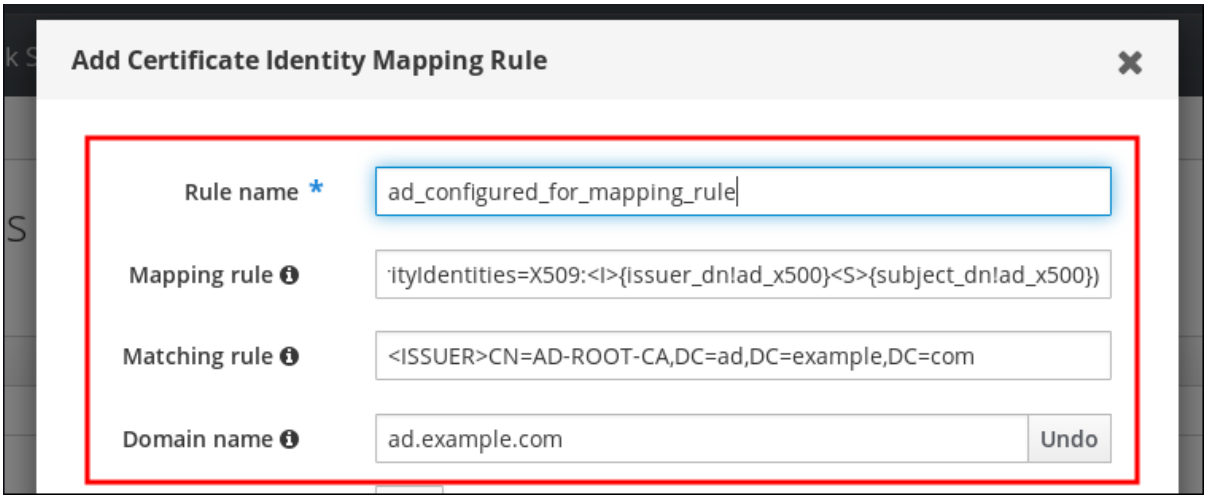
6. Saisissez la règle de correspondance. Par exemple, pour n'autoriser que les certificats émis par **AD-ROOT-CA** du domaine **AD.EXAMPLE.COM** à authentifier les utilisateurs de l'IdM :

```
<ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
```

7. Entrez le domaine :

```
ad.example.com
```

Figure 9.8. Règle de mappage des certificats si AD est configuré pour le mappage



8. Cliquez sur **Add**.

9. Le System Security Services Daemon (SSSD) relit périodiquement les règles de mappage des certificats. Pour forcer le chargement immédiat de la règle nouvellement créée, redémarrez SSSD dans le CLI: :

```
# systemctl restart sssd
```

9.4.2. Ajout d'une règle de mappage de certificats dans la CLI IdM

1. Obtenir les informations d'identification de l'administrateur :

```
# kinit admin
```

2. Saisissez la règle de mappage et la règle de correspondance sur laquelle la règle de mappage est basée. Par exemple, pour que AD recherche les entrées **Issuer** et **Subject** dans tout certificat présenté et n'autorise que les certificats émis par **AD-ROOT-CA** du domaine **AD.EXAMPLE.COM**:

```
# ipa certmaprule-add ad_configured_for_mapping_rule --matchrule
'<ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com' --maprule
'(altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}<S>{subject_dn!ad_x500})' --
domain=ad.example.com
----------------------------------------------------
Added Certificate Identity Mapping Rule "ad_configured_for_mapping_rule"
----------------------------------------------------
  Rule name: ad_configured_for_mapping_rule
  Mapping rule: (altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}<S>
{subject_dn!ad_x500})
  Matching rule: <ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
  Domain name: ad.example.com
  Enabled: TRUE
```

3. Le System Security Services Daemon (SSSD) relit périodiquement les règles de mappage des certificats. Pour forcer le chargement immédiat de la règle nouvellement créée, redémarrez SSSD :

```
# systemctl restart sssd
```

9.4.3. Vérification des données de mappage des certificats du côté AD

L'attribut **altSecurityIdentities** est l'équivalent dans Active Directory (AD) de l'attribut utilisateur **certmapdata** dans IdM. Lors de la configuration du mappage des certificats dans IdM, dans le scénario où un domaine AD de confiance est configuré pour mapper les certificats d'utilisateur aux comptes d'utilisateur, l'administrateur du système IdM doit vérifier que l'attribut **altSecurityIdentities** est correctement défini dans les entrées d'utilisateur dans AD.

Pour vérifier que AD contient les bonnes informations pour l'utilisateur stocké dans AD, utilisez la commande **ldapsearch**.

- Par exemple, entrez la commande ci-dessous pour vérifier auprès du serveur **adserver.ad.example.com** que les conditions suivantes s'appliquent :
  - L'attribut **altSecurityIdentities** est défini dans l'entrée utilisateur de **ad_user**.
  - La règle du match stipule que les conditions suivantes s'appliquent :

- Le certificat que **ad_user** utilise pour s'authentifier auprès d'AD a été émis par **AD-ROOT-CA** du domaine **ad.example.com**.
- Le sujet est **<S>DC=com,DC=example,DC=ad,CN=Users,CN=ad_user**:

```
$ ldapsearch -o ldif-wrap=no -LLL -h adserver.ad.example.com \
-p 389 -D cn=Administrator,cn=users,dc=ad,dc=example,dc=com \
-W -b cn=users,dc=ad,dc=example,dc=com "(cn=ad_user)" \
altSecurityIdentities
Enter LDAP Password:
dn: CN=ad_user,CN=Users,DC=ad,DC=example,DC=com
altSecurityIdentities: X509:<I>DC=com,DC=example,DC=ad,CN=AD-ROOT-
CA<S>DC=com,DC=example,DC=ad,CN=Users,CN=ad_user
```

9.5. CONFIGURATION DU MAPPAGE DES CERTIFICATS SI L'ENTRÉE DE L'UTILISATEUR AD NE CONTIENT PAS DE CERTIFICAT OU DE DONNÉES DE MAPPAGE

Cette histoire d'utilisateur décrit les étapes nécessaires pour activer le mappage de certificats dans IdM si le déploiement IdM est en confiance avec Active Directory (AD), l'utilisateur est stocké dans AD et l'entrée de l'utilisateur dans AD ne contient ni le certificat entier ni les données de mappage de certificats.

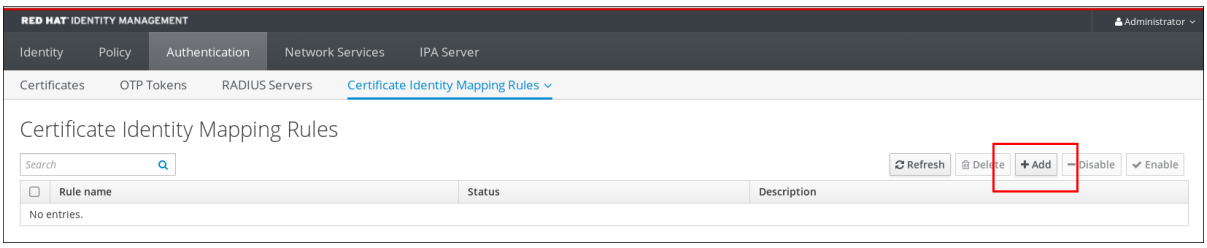
Conditions préalables

- L'utilisateur n'a pas de compte dans IdM.
- L'utilisateur a un compte dans AD qui ne contient ni le certificat complet ni l'attribut **altSecurityIdentities**, l'équivalent AD de l'attribut IdM **certmapdata**.
- L'administrateur IdM dispose de l'ensemble du certificat de l'utilisateur AD à ajouter au site **user ID override** de l'utilisateur AD dans IdM.

9.5.1. Ajout d'une règle de mappage de certificats dans l'interface web IdM

1. Se connecter à l'interface web IdM en tant qu'administrateur.
2. Naviguez vers **Authentication → Certificate Identity Mapping Rules → Certificate Identity Mapping Rules**.
3. Cliquez sur **Add**.

Figure 9.9. Ajout d'une nouvelle règle de mappage de certificats dans l'interface web IdM



4. Saisissez le nom de la règle.

5. Saisissez la règle de mappage. Pour que l'ensemble du certificat présenté à l'IdM pour l'authentification soit comparé au certificat stocké dans l'entrée de remplacement de l'ID utilisateur de l'entrée de l'utilisateur AD dans l'IdM :

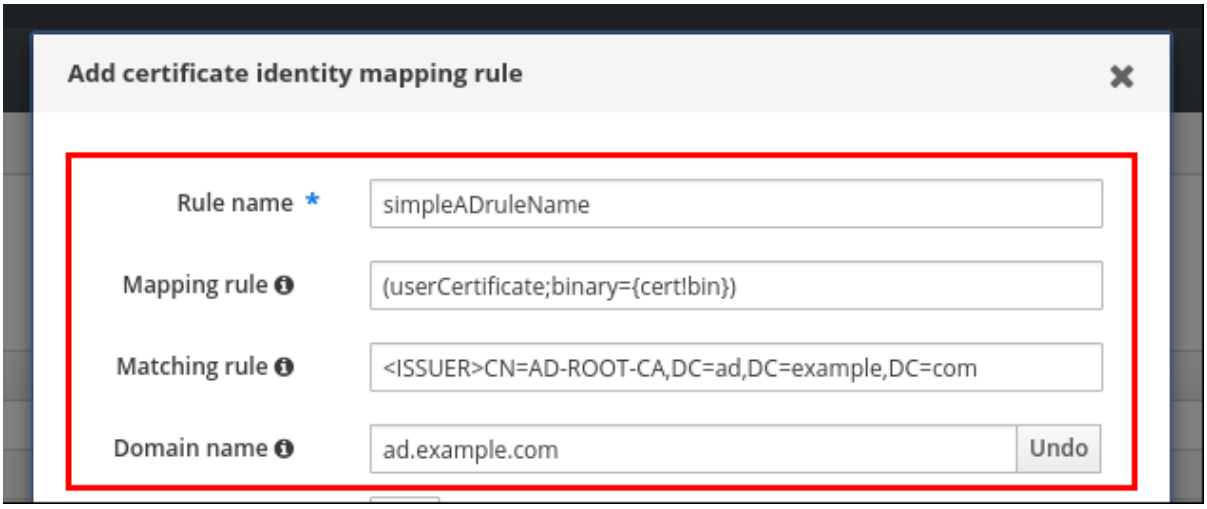
```
(userCertificate;binary={cert!bin})
```

6. Saisissez la règle de correspondance. Par exemple, pour autoriser uniquement les certificats émis par **AD-ROOT-CA** du domaine **AD.EXAMPLE.COM** à s'authentifier :

```
<ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
```

7. Saisissez le nom de domaine. Par exemple, pour rechercher des utilisateurs dans le domaine **ad.example.com**:

Figure 9.10. Règle de mappage des certificats pour un utilisateur sans certificat ni données de mappage stockées dans AD



8. Cliquez sur **Add**.

9. Le System Security Services Daemon (SSSD) relit périodiquement les règles de mappage des certificats. Pour forcer le chargement immédiat de la règle nouvellement créée, redémarrez SSSD dans l'interface CLI :

```
# systemctl restart sssd
```

9.5.2. Ajout d'une règle de mappage de certificats dans la CLI IdM

1. Obtenir les informations d'identification de l'administrateur :

```
# kinit admin
```

2. Saisissez la règle de mappage et la règle de correspondance sur laquelle la règle de mappage est basée. Pour que l'ensemble du certificat présenté pour l'authentification soit comparé au certificat stocké dans l'entrée de remplacement de l'ID utilisateur de l'entrée de l'utilisateur AD dans IdM, autorisant uniquement les certificats émis par **AD-ROOT-CA** du domaine **AD.EXAMPLE.COM** pour l'authentification :

```
# ipa certmaprule-add simpleADrule --matchrule '<ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com' --maprule '(userCertificate;binary={cert!bin})' --domain ad.example.com
```

```
---------------------------------------------------
Added Certificate Identity Mapping Rule "simpleADrule"
---------------------------------------------------
  Rule name: simpleADrule
  Mapping rule: (userCertificate;binary={cert!bin})
  Matching rule: <ISSUER>CN=AD-ROOT-CA,DC=ad,DC=example,DC=com
  Domain name: ad.example.com
  Enabled: TRUE
```

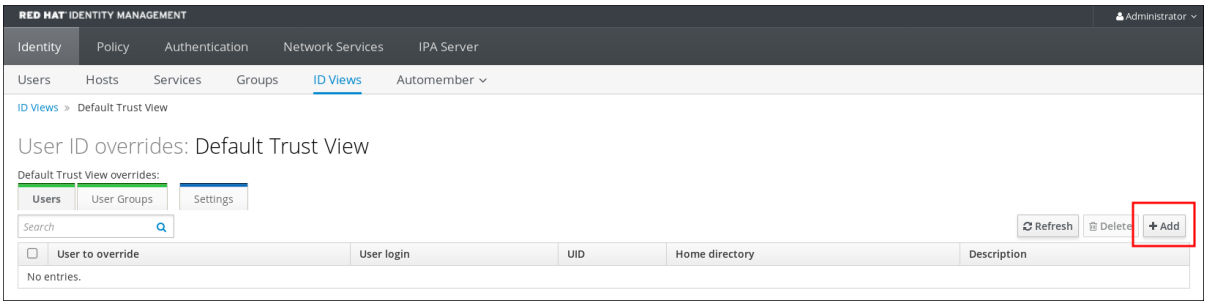
3. Le System Security Services Daemon (SSSD) relit périodiquement les règles de mappage des certificats. Pour forcer le chargement immédiat de la règle nouvellement créée, redémarrez SSSD :

```
# systemctl restart sssd
```

9.5.3. Ajout d'un certificat à l'ID override d'un utilisateur AD dans l'interface web IdM

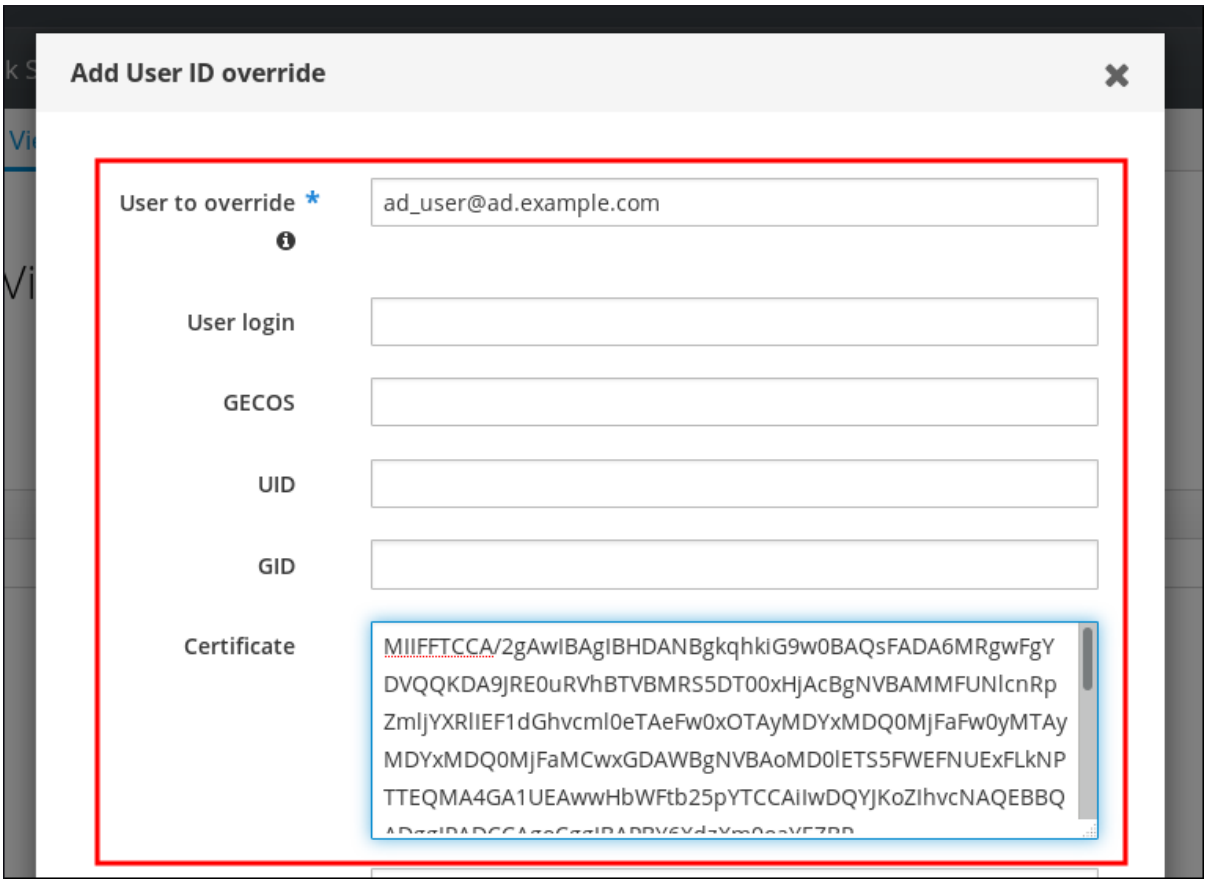
1. Naviguez vers **Identity → ID Views → Default Trust View**.
2. Cliquez sur **Add**.

Figure 9.11. Ajout d'un nouvel identifiant d'utilisateur dans l'interface web de l'IdM



3. Dans le champ **User to override**, entrez **ad_user@ad.example.com**.
4. Copiez et collez le certificat de **ad_user** dans le champ **Certificate**.

Figure 9.12. Configuration de l'annulation de l'ID utilisateur pour un utilisateur AD



5. Cliquez sur **Add**.

Verification steps

- Vérifiez que l'utilisateur et le certificat sont liés :
  - Utilisez l'utilitaire **sss_cache** pour invalider l'enregistrement de **ad_user@ad.example.com** dans le cache SSSD et forcer le rechargement des informations de **ad_user@ad.example.com**:

    ```
    # sss_cache -u ad_user@ad.example.com
    ```

  - Exécutez la commande **ipa certmap-match** avec le nom du fichier contenant le certificat de l'utilisateur AD :

    ```
    # ipa certmap-match ad_user_cert.pem
    ----------
    1 user matched
    ----------
     Domain: AD.EXAMPLE.COM
     User logins: ad_user@ad.example.com
    ----------------------------
    Number of entries returned 1
    ----------------------------
    ```

Le résultat confirme que des données de mappage de certificats ont été ajoutées à **ad_user@ad.example.com** et qu'une règle de mappage correspondante définie dans Ajout d'une règle de mappage de certificats si l'entrée de l'utilisateur AD ne contient pas de certificat ou de données de

mappage existe. Cela signifie que vous pouvez utiliser n'importe quel certificat correspondant aux données de mappage de certificats définies pour vous authentifier en tant que **ad_user@ad.example.com**.

9.5.4. Ajout d'un certificat à l'ID override d'un utilisateur AD dans la CLI IdM

Ressources supplémentaires

1. Obtenir les informations d'identification de l'administrateur :

```
# kinit admin
```

2. Stockez le blob du certificat dans une nouvelle variable appelée **CERT**:

```
# CERT=`cat ad_user_cert.pem | tail -n 2| head -n -1 | tr -d '\r\n'\`
```

3. Ajoutez le certificat de **ad_user@ad.example.com** au compte d'utilisateur à l'aide de la commande **ipa idoverrideuser-add-cert**:

```
# ipa idoverrideuser-add-cert ad_user@ad.example.com --certificate $CERT
```

Verification steps

- Vérifiez que l'utilisateur et le certificat sont liés :
  - Utilisez l'utilitaire **sss_cache** pour invalider l'enregistrement de **ad_user@ad.example.com** dans le cache SSSD et forcer le rechargement des informations de **ad_user@ad.example.com**:

```
# sss_cache -u ad_user@ad.example.com
```

  - Exécutez la commande **ipa certmap-match** avec le nom du fichier contenant le certificat de l'utilisateur AD :

```
# ipa certmap-match ad_user_cert.pem
--------------
1 user matched
--------------
 Domain: AD.EXAMPLE.COM
 User logins: ad_user@ad.example.com
----------------------------
Number of entries returned 1
----------------------------
```

Le résultat confirme que des données de mappage de certificats ont été ajoutées à **ad_user@ad.example.com** et qu'une règle de mappage correspondante définie dans Ajout d'une règle de mappage de certificats si l'entrée de l'utilisateur AD ne contient pas de certificat ou de données de mappage existe. Cela signifie que vous pouvez utiliser n'importe quel certificat correspondant aux données de mappage de certificats définies pour vous authentifier en tant que **ad_user@ad.example.com**.

9.6. COMBINAISON DE PLUSIEURS RÈGLES DE MAPPAGE D'IDENTITÉ EN UNE SEULE

Ressources supplémentaires

Pour combiner plusieurs règles de mappage d'identité en une seule règle combinée, utilisez le caractère | (ou) pour précéder les règles de mappage individuelles et séparez-les à l'aide de crochets **()**, par exemple :

Exemple de filtre de mappage de certificats 1

```
$ ipa certmaprule-add ad_cert_for_ipa_and_ad_users \ --maprule='(|(ipacertmapdata=X509:<I>
{issuer_dn!nss_x500}<S>{subject_dn!nss_x500})(altSecurityIdentities=X509:<I>
{issuer_dn!ad_x500}<S>{subject_dn!ad_x500}))' \ --matchrule='<ISSUER>CN=AD-ROOT-
CA,DC=ad,DC=example,DC=com' \ --domain=ad.example.com
```

Dans l'exemple ci-dessus, la définition du filtre dans l'option **--maprule** inclut ces critères :

- **ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>{subject_dn!nss_x500}** est un filtre qui relie le sujet et l'émetteur d'un certificat de carte à puce à la valeur de l'attribut **ipacertmapdata** dans un compte d'utilisateur IdM, comme décrit dans Ajouter une règle de mappage de certificat dans IdM
- **altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}<S>{subject_dn!ad_x500}** est un filtre qui relie le sujet et l'émetteur d'un certificat de carte à puce à la valeur de l'attribut **altSecurityIdentities** dans un compte d'utilisateur AD, comme décrit dans Ajouter une règle de mappage de certificats si le domaine AD de confiance est configuré pour mapper les certificats d'utilisateur
- L'ajout de l'option **--domain=ad.example.com** signifie que les utilisateurs associés à un certificat donné sont recherchés non seulement dans le domaine local **idm.example.com**, mais aussi dans le domaine **ad.example.com**

La définition du filtre dans l'option **--maprule** accepte l'opérateur logique | (ou), ce qui permet de spécifier plusieurs critères. Dans ce cas, la règle met en correspondance tous les comptes d'utilisateurs qui répondent à au moins un des critères.

Exemple de filtre de mappage de certificats 2

```
$ ipa certmaprule-add ipa_cert_for_ad_users \
  --maprule='(|(userCertificate;binary={cert!bin})(ipacertmapdata=X509:<I>
{issuer_dn!nss_x500}<S>{subject_dn!nss_x500})(altSecurityIdentities=X509:<I>
{issuer_dn!ad_x500}<S>{subject_dn!ad_x500}))' \
  --matchrule='<ISSUER>CN=Certificate Authority,O=REALM.EXAMPLE.COM' \
  --domain=idm.example.com --domain=ad.example.com
```

Dans l'exemple ci-dessus, la définition du filtre dans l'option **--maprule** inclut ces critères :

- **userCertificate;binary={cert!bin}** est un filtre qui renvoie les entrées d'utilisateurs contenant le certificat complet. Pour les utilisateurs AD, la création de ce type de filtre est décrite en détail dans Ajouter une règle de mappage de certificats si l'entrée de l'utilisateur AD ne contient pas de certificat ou de données de mappage.
- **ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>{subject_dn!nss_x500}** est un filtre qui relie le sujet et l'émetteur d'un certificat de carte à puce à la valeur de l'attribut **ipacertmapdata**

dans un compte d'utilisateur IdM, comme décrit dans Ajouter une règle de mappage de certificat dans IdM.

- **altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}<S>{subject_dn!ad_x500}** est un filtre qui relie le sujet et l'émetteur d'un certificat de carte à puce à la valeur de l'attribut **altSecurityIdentities** dans un compte d'utilisateur AD, comme décrit dans Ajouter une règle de mappage de certificats si le domaine AD de confiance est configuré pour mapper les certificats d'utilisateur.

La définition du filtre dans l'option **--maprule** accepte l'opérateur logique | (ou), ce qui permet de spécifier plusieurs critères. Dans ce cas, la règle met en correspondance tous les comptes d'utilisateurs qui répondent à au moins un des critères.

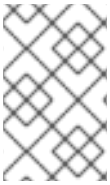
CHAPITRE 10. CONFIGURATION DE L'AUTHENTIFICATION AVEC UN CERTIFICAT STOCKÉ SUR LE BUREAU D'UN CLIENT IDM

En configurant la gestion des identités (IdM), les administrateurs du système IdM peuvent permettre aux utilisateurs de s'authentifier auprès de l'interface web IdM et de l'interface de ligne de commande (CLI) à l'aide d'un certificat qu'une autorité de certification (CA) a délivré aux utilisateurs.

Le navigateur web peut fonctionner sur un système qui ne fait pas partie du domaine IdM.

Cette histoire d'utilisateur fournit des instructions sur la manière de configurer et de tester efficacement la connexion à l'interface utilisateur Web et à la CLI de Identity Management avec un certificat stocké sur le bureau d'un client IdM. En suivant cette histoire d'utilisateur,

- vous pouvez ignorer la demande d'un nouveau certificat utilisateur et son exportation vers le client si l'utilisateur que vous souhaitez authentifier à l'aide d'un certificat dispose déjà d'un certificat ;
- vous pouvez ne pas vous assurer que le certificat et l'utilisateur sont liés si le certificat de l'utilisateur a été émis par l'autorité de certification IdM.



NOTE

Seuls les utilisateurs de la gestion des identités peuvent se connecter à l'interface web à l'aide d'un certificat. Les utilisateurs d'Active Directory peuvent se connecter avec leur nom d'utilisateur et leur mot de passe.

10.1. CONFIGURATION DU SERVEUR DE GESTION DES IDENTITÉS POUR L'AUTHENTIFICATION PAR CERTIFICAT DANS L'INTERFACE WEB

En tant qu'administrateur de la gestion des identités (IdM), vous pouvez autoriser les utilisateurs à utiliser des certificats pour s'authentifier dans votre environnement IdM.

Procédure

En tant qu'administrateur de la gestion des identités :

1. Sur un serveur de gestion des identités, obtenez les privilèges d'administrateur et créez un script shell pour configurer le serveur.
   a. Exécutez la commande **ipa-advise config-server-for-smart-card-auth** et enregistrez son résultat dans un fichier, par exemple **server_certificate_script.sh**:

   ```
   # kinit admin
   # ipa-advise config-server-for-smart-card-auth > server_certificate_script.sh
   ```

   b. Ajoutez des autorisations d'exécution au fichier à l'aide de l'utilitaire **chmod**:

   ```
   # chmod x server_certificate_script.sh
   ```

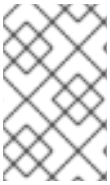
2. Sur tous les serveurs du domaine Identity Management, exécutez le script **server_certificate_script.sh**

a. avec le chemin du certificat de l'autorité de certification IdM, **/etc/ipa/ca.crt**, comme entrée si l'autorité de certification IdM est la seule autorité de certification qui a émis les certificats des utilisateurs pour lesquels vous voulez activer l'authentification par certificat :

```
# ./server_certificate_script.sh /etc/ipa/ca.crt
```

b. avec en entrée les chemins d'accès aux certificats d'autorité de certification concernés si différentes autorités de certification externes ont signé les certificats des utilisateurs pour lesquels vous souhaitez activer l'authentification par certificat :

```
# ./server_certificate_script.sh /tmp/ca1.pem /tmp/ca2.pem
```

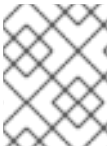


NOTE

N'oubliez pas d'exécuter le script sur chaque nouvelle réplique que vous ajouterez au système à l'avenir si vous souhaitez que l'authentification par certificat des utilisateurs soit activée dans l'ensemble de la topologie.

10.2. DEMANDER UN NOUVEAU CERTIFICAT D'UTILISATEUR ET L'EXPORTER VERS LE CLIENT

En tant qu'administrateur de la gestion des identités (IdM), vous pouvez créer des certificats pour les utilisateurs dans votre environnement IdM et les exporter vers les clients IdM sur lesquels vous souhaitez activer l'authentification par certificat pour les utilisateurs.



NOTE

Vous pouvez ignorer cette section si l'utilisateur que vous souhaitez authentifier à l'aide d'un certificat dispose déjà d'un certificat.

Procédure

1. Si vous le souhaitez, créez un nouveau répertoire, par exemple **~/certdb/**, et faites-en une base de données de certificats temporaire. Si on vous le demande, créez un mot de passe pour la base de données des certificats NSS afin de crypter les clés du certificat qui sera généré lors d'une étape ultérieure :

```
# mkdir ~/certdb/
# certutil -N -d ~/certdb/
Enter a password which will be used to encrypt your keys.
The password should be at least 8 characters long,
and should contain at least one non-alphabetic character.

Enter new password:
Re-enter password:
```

2. Créez la demande de signature de certificat (CSR) et redirigez la sortie vers un fichier. Par exemple, pour créer une CSR avec le nom **certificate_request.csr** pour un certificat bit **4096** pour l'utilisateur **idm_user** dans le domaine **IDM.EXAMPLE.COM**, en définissant le surnom des clés privées du certificat à **idm_user** pour faciliter la recherche, et en définissant le sujet à **CN=idm_user,O=IDM.EXAMPLE.COM**:

```
# certutil -R -d ~/certdb/ -a -g 4096 -n idm_user -s "CN=idm_user,O=IDM.EXAMPLE.COM"
> certificate_request.csr
```

3. À l'invite, saisissez le même mot de passe que celui que vous avez saisi lorsque vous avez utilisé **certutil** pour créer la base de données temporaire. Continuez ensuite à taper randlomly jusqu'à ce qu'on vous dise d'arrêter :

```
Enter Password or Pin for "NSS Certificate DB":

A random seed must be generated that will be used in the
creation of your key.  One of the easiest ways to create a
random seed is to use the timing of keystrokes on a keyboard.

To begin, type keys on the keyboard until this progress meter
is full.  DO NOT USE THE AUTOREPEAT FUNCTION ON YOUR KEYBOARD!


Continue typing until the progress meter is full:
```

4. Soumettez le fichier de demande de certificat au serveur. Indiquez le principal Kerberos à associer au certificat nouvellement émis, le fichier de sortie dans lequel stocker le certificat et, éventuellement, le profil du certificat. Par exemple, pour obtenir un certificat du profil **IECUserRoles**, un profil avec l'extension des rôles d'utilisateur ajoutés, pour le principal **idm_user**@**IDM.EXAMPLE.COM**, et l'enregistrer dans le fichier **~/idm_user.pem**:

```
# ipa cert-request certificate_request.csr --principal=idm_user@IDM.EXAMPLE.COM --
profile-id=IECUserRoles --certificate-out=~/idm_user.pem
```

5. Ajoutez le certificat à la base de données NSS. Utilisez l'option **-n** pour définir le même surnom que celui que vous avez utilisé lors de la création de la CSR, afin que le certificat corresponde à la clé privée dans la base de données NSS. L'option **-t** définit le niveau de confiance. Pour plus de détails, voir la page de manuel certutil(1). L'option **-i** spécifie le fichier de certificat d'entrée. Par exemple, pour ajouter à la base de données NSS un certificat avec le pseudonyme **idm_user** qui est stocké dans le fichier **~/idm_user.pem** de la base de données **~/certdb/**:

```
# certutil -A -d ~/certdb/ -n idm_user -t "P,," -i ~/idm_user.pem
```

6. Vérifiez que la clé dans la base de données NSS n'indique pas **(orphan)** comme surnom. Par exemple, pour vérifier que le certificat stocké dans la base de données **~/certdb/** n'est pas orphelin :

```
# certutil -K -d ~/certdb/
< 0> rsa      5ad14d41463b87a095b1896cf0068ccc467df395   NSS Certificate
DB:idm_user
```

7. Utilisez la commande **pk12util** pour exporter le certificat de la base de données NSS au format PKCS12. Par exemple, pour exporter le certificat avec le pseudonyme **idm_user** de la base de données NSS **/root/certdb** vers le fichier **~/idm_user.p12**:

```
# pk12util -d ~/certdb -o ~/idm_user.p12 -n idm_user
Enter Password or Pin for "NSS Certificate DB":
Enter password for PKCS12 file:
Re-enter password:
pk12util: PKCS12 EXPORT SUCCESSFUL
```

8. Transférez le certificat vers l'hôte sur lequel vous souhaitez activer l'authentification par certificat pour **idm_user**:

```
# scp ~/idm_user.p12 idm_user@client.idm.example.com:/home/idm_user/
```

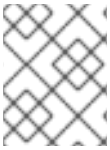
9. Sur l'hôte vers lequel le certificat a été transféré, rendez le répertoire dans lequel le fichier .pkcs12 est stocké inaccessible au groupe "other" pour des raisons de sécurité :

```
# chmod o-rwx /home/idm_user/
```

10. Pour des raisons de sécurité, supprimez la base de données NSS temporaire et le fichier .pkcs12 du serveur :

```
# rm ~/certdb/
# rm ~/idm_user.p12
```

10.3. S'ASSURER QUE LE CERTIFICAT ET L'UTILISATEUR SONT LIÉS



NOTE

Vous pouvez ignorer cette section si le certificat de l'utilisateur a été délivré par l'autorité de certification IdM.

Pour que l'authentification par certificat fonctionne, vous devez vous assurer que le certificat est lié à l'utilisateur qui l'utilisera pour s'authentifier auprès de la gestion des identités (IdM).

- Si le certificat est fourni par une autorité de certification qui ne fait pas partie de votre environnement de gestion des identités, reliez l'utilisateur et le certificat en suivant la procédure décrite dans la section Relier les comptes d'utilisateurs aux certificats .
- Si le certificat est fourni par l'autorité de certification de la gestion des identités, il est déjà automatiquement ajouté à l'entrée de l'utilisateur et vous n'avez pas besoin de lier le certificat au compte de l'utilisateur. Pour plus de détails sur la création d'un nouveau certificat dans IdM, voir Demander un nouveau certificat d'utilisateur et l'exporter vers le client .

10.4. CONFIGURATION D'UN NAVIGATEUR POUR ACTIVER L'AUTHENTIFICATION PAR CERTIFICAT

Pour pouvoir s'authentifier à l'aide d'un certificat lors de l'utilisation de l'interface WebUI pour se connecter à la gestion des identités (IdM), vous devez importer l'utilisateur et les certificats de l'autorité de certification (AC) concernés dans le navigateur Mozilla Firefox ou Google Chrome. L'hôte sur lequel le navigateur est exécuté ne doit pas nécessairement faire partie du domaine IdM.

IdM prend en charge les navigateurs suivants pour se connecter à l'interface WebUI :

- Mozilla Firefox 38 et versions ultérieures
- Google Chrome 46 et versions ultérieures

La procédure suivante montre comment configurer le navigateur Mozilla Firefox 57.0.1.

Conditions préalables

- Vous disposez du certificat d'utilisateur que vous souhaitez importer dans le navigateur au format PKCS#12.

Procédure

1. Ouvrez Firefox, puis naviguez vers **Preferences → Privacy & Security**.

   Figure 10.1. Section Vie privée et sécurité dans les préférences

2. Cliquez sur **Afficher les certificats**.

   Figure 10.2. Voir les certificats en protection de la vie privée et en sécurité

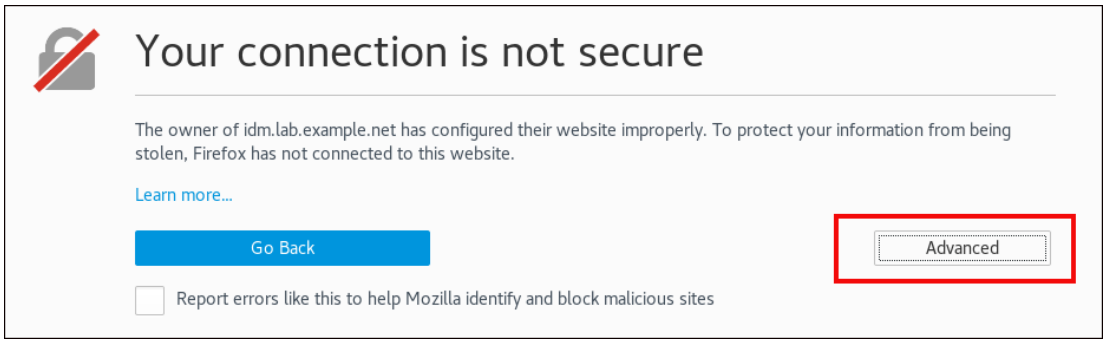
3. Dans l'onglet **Your Certificates**, cliquez sur **Importer**. Localisez et ouvrez le certificat de l'utilisateur au format PKCS12, puis cliquez sur **OK** et **OK**.

4. Assurez-vous que l'autorité de certification de la gestion d'identité est reconnue par Firefox comme une autorité de confiance :

   a. Enregistrer localement le certificat de l'autorité de certification IdM :

      - Naviguez vers l'interface web IdM en écrivant le nom de votre serveur IdM dans la barre d'adresse de Firefox. Cliquez sur **Advanced** sur la page d'avertissement de connexion non sécurisée.

Figure 10.3. Connexion non sécurisée

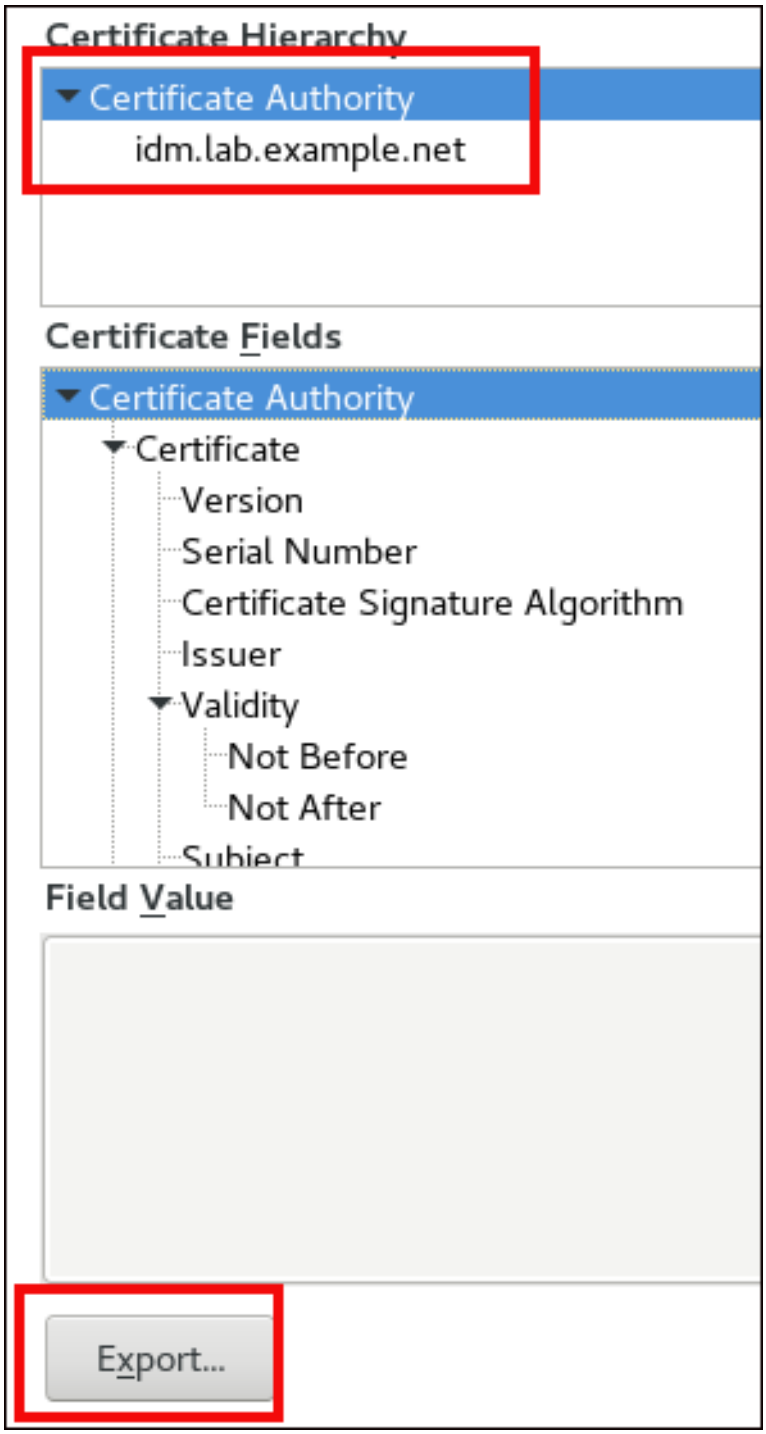


- **Add Exception**. Cliquez sur **View**.

Figure 10.4. Afficher les détails d'un certificat

- Dans l'onglet **Details**, mettez en évidence les champs **Certificate Authority**.

Figure 10.5. Exportation du certificat de l'autorité de certification



- Cliquez sur **Exporter**. Enregistrez le certificat d'autorité de certification, par exemple dans le fichier **CertificateAuthority.crt**, puis cliquez sur **Fermer** et sur **Annuler**.

b. Importer le certificat de l'autorité de certification IdM dans Firefox en tant que certificat d'autorité de certification de confiance :

- Ouvrez Firefox, allez dans Préférences et cliquez sur **Confidentialité & Sécurité**.

Figure 10.6. Section Vie privée et sécurité dans les préférences

- Cliquez sur **Afficher les certificats**.

Figure 10.7. Voir les certificats en protection de la vie privée et en sécurité

- Dans l'onglet **Authorities**, cliquez sur **Importer**. Localisez et ouvrez le certificat CA que vous avez enregistré à l'étape précédente dans le fichier **CertificateAuthority.crt**. Faites confiance au certificat pour identifier les sites web, puis cliquez sur **OK** et **OK**.

5. Passez à la section Authentification à l'interface Web de gestion des identités avec un certificat en tant qu'utilisateur de gestion des identités.

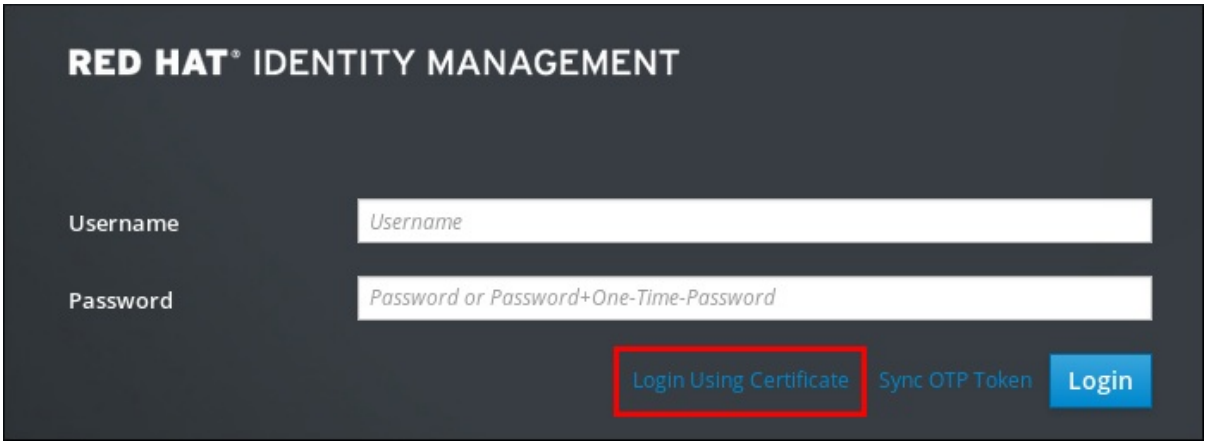
10.5. AUTHENTIFICATION À L'INTERFACE WEB DE GESTION DES IDENTITÉS AVEC UN CERTIFICAT EN TANT QU'UTILISATEUR DE GESTION DES IDENTITÉS

Cette procédure décrit l'authentification en tant qu'utilisateur de l'interface Web de gestion des identités (IdM) à l'aide d'un certificat stocké sur le bureau d'un client de gestion des identités.

Procédure

1. Dans le navigateur, accédez à l'interface utilisateur Web de gestion des identités à l'adresse suivante : **https://server.idm.example.com/ipa/ui**.

2. Cliquez sur **Connexion à l'aide d'un certificat**.
login Utiliser un certificat dans l'interface web de gestion des identités



3. Le certificat de l'utilisateur doit déjà être sélectionné. Décochez la case **Se souvenir de cette décision**, puis cliquez sur **OK**.

Vous êtes maintenant authentifié en tant qu'utilisateur correspondant au certificat.

Ressources supplémentaires

- Voir Configuration de la gestion des identités pour l'authentification par carte à puce .

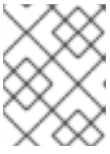
10.6. CONFIGURATION D'UN CLIENT IDM POUR PERMETTRE L'AUTHENTIFICATION À LA CLI À L'AIDE D'UN CERTIFICAT

Pour que l'authentification par certificat fonctionne pour un utilisateur IdM dans l'interface de ligne de commande (CLI) de votre client IdM, importez le certificat de l'utilisateur IdM et la clé privée dans le client IdM. Pour plus d'informations sur la création et le transfert du certificat d'utilisateur, voir Demander un nouveau certificat d'utilisateur et l'exporter vers le client .

Procédure

- Connectez-vous au client IdM et préparez le fichier .p12 contenant le certificat de l'utilisateur et la clé privée. Pour obtenir et mettre en cache le ticket d'octroi de ticket Kerberos (TGT), exécutez la commande **kinit** avec le principal de l'utilisateur, en utilisant l'option **-X** avec l'attribut **X509_username:/path/to/file.p12** pour spécifier où trouver les informations d'identité X509 de l'utilisateur. Par exemple, pour obtenir le TGT pour **idm_user** en utilisant les informations d'identité de l'utilisateur stockées dans le fichier **~/idm_user.p12**:

```
$ kinit -X X509_idm_user='PKCS12:~/idm_user.p12' idm_user
```



NOTE

La commande prend également en charge le format de fichier .pem : **kinit -X X509_username='FILE:/path/to/cert.pem,/path/to/key' user_principal**

CHAPITRE 11. UTILISATION DU SERVEUR DE RENOUVELLEMENT DE L'AUTORITÉ DE CERTIFICATION IDM

11.1. EXPLICATION DU SERVEUR DE RENOUVELLEMENT DE L'AUTORITÉ DE CERTIFICATION IDM

Dans un déploiement de gestion d'identité (IdM) qui utilise une autorité de certification (CA) intégrée, le serveur de renouvellement de la CA maintient et renouvelle les certificats du système IdM. Il garantit la robustesse des déploiements IdM.

Les certificats du système IdM comprennent

- **IdM CA** certificat
- **OCSP** certificat de signature
- **IdM CA subsystem** certificats
- **IdM CA audit signing** certificat
- **IdM renewal agent** (RA) certificat
- **KRA** certificats de transport et de stockage

Ce qui caractérise les certificats de système, c'est que leurs clés sont partagées par toutes les répliques de l'autorité de certification. En revanche, les certificats de service IdM (par exemple, les certificats **LDAP**, **HTTP** et **PKINIT**) ont des paires de clés et des noms de sujets différents sur les différents serveurs de l'AC IdM.

Dans la topologie IdM, par défaut, le premier serveur CA IdM est le serveur de renouvellement CA.

NOTE

Dans la documentation en amont, l'autorité de certification IdM est appelée **Dogtag**.

Le rôle du serveur de renouvellement de l'AC

Les certificats **IdM CA**, **IdM CA subsystem** et **IdM RA** sont essentiels pour le déploiement de l'IdM. Chaque certificat est stocké dans une base de données NSS dans le répertoire **/etc/pki/pki-tomcat/** et également en tant qu'entrée de base de données LDAP. Le certificat stocké dans LDAP doit correspondre au certificat stocké dans la base de données NSS. Si ce n'est pas le cas, l'authentification échoue entre le cadre IdM et l'autorité de certification IdM, ainsi qu'entre l'autorité de certification IdM et LDAP.

Toutes les répliques de l'AC IdM ont des demandes de suivi pour chaque certificat de système. Si un déploiement IdM avec AC intégrée ne contient pas de serveur de renouvellement d'AC, chaque serveur d'AC IdM demande le renouvellement des certificats de système de manière indépendante. Il en résulte que différentes répliques de l'autorité de certification disposent de différents certificats de système et que des échecs d'authentification se produisent.

La désignation d'une réplique de l'autorité de certification en tant que serveur de renouvellement permet de renouveler les certificats du système une seule fois, lorsque cela est nécessaire, et d'éviter ainsi les échecs d'authentification.

Le rôle du service certmonger sur les répliques de CA

Le service **certmonger** exécuté sur toutes les répliques de l'AC IdM utilise l'aide au renouvellement **dogtag-ipa-ca-renew-agent** pour assurer le suivi des certificats du système IdM. Le programme d'aide au renouvellement lit la configuration du serveur de renouvellement de l'autorité de certification. Sur chaque réplique d'autorité de certification qui n'est pas le serveur de renouvellement de l'autorité de certification, le programme d'aide au renouvellement récupère les derniers certificats de système à partir des entrées LDAP de **ca_renewal**. En raison de la non-détermination du moment exact où les tentatives de renouvellement de **certmonger** se produisent, l'assistant **dogtag-ipa-ca-renew-agent** tente parfois de mettre à jour un certificat système avant que le serveur de renouvellement de l'autorité de certification n'ait effectivement renouvelé le certificat. Dans ce cas, l'ancien certificat, qui va bientôt expirer, est renvoyé au service **certmonger** sur la réplique de l'autorité de certification. Le service **certmonger**, réalisant qu'il s'agit du même certificat que celui déjà stocké dans sa base de données, continue d'essayer de renouveler le certificat avec un certain délai entre chaque tentative jusqu'à ce qu'il puisse récupérer le certificat mis à jour auprès du serveur de renouvellement de l'autorité de certification.

Fonctionnement correct du serveur de renouvellement de l'autorité de certification IdM

Un déploiement IdM avec une autorité de certification intégrée est un déploiement IdM qui a été installé avec une autorité de certification IdM - ou dont le serveur d'autorité de certification IdM a été installé ultérieurement. Un déploiement IdM avec une autorité de certification intégrée doit toujours avoir exactement une réplique d'autorité de certification configurée comme serveur de renouvellement. Le serveur de renouvellement doit être en ligne et entièrement fonctionnel, et doit répliquer correctement avec les autres serveurs.

Si le serveur de renouvellement de l'AC actuel est supprimé à l'aide des commandes **ipa server-del**, **ipa-replica-manage del**, **ipa-csreplica-manage del** ou **ipa-server-install --uninstall**, une autre réplique de l'AC est automatiquement affectée en tant que serveur de renouvellement de l'AC. Cette politique garantit que la configuration du serveur de renouvellement reste valide.

Cette police ne couvre pas les situations suivantes :

- **Offline renewal server**
  Si le serveur de renouvellement est hors ligne pendant une longue période, il peut manquer une fenêtre de renouvellement. Dans ce cas, tous les serveurs d'autorité de certification non renouvelés continuent à réinstaller les certificats du système actuel jusqu'à ce que les certificats expirent. Dans ce cas, le déploiement de l'IdM est perturbé, car un seul certificat expiré peut entraîner des échecs de renouvellement pour d'autres certificats.

- **Replication problems**
  Si des problèmes de réplication existent entre le serveur de renouvellement et les autres répliques de l'autorité de certification, le renouvellement peut réussir, mais les autres répliques de l'autorité de certification peuvent ne pas être en mesure de récupérer les certificats mis à jour avant qu'ils n'expirent.

  Pour éviter cette situation, assurez-vous que vos accords de réplication fonctionnent correctement. Pour plus de détails, consultez les directives générales ou spécifiques de dépannage de la réplication dans le site RHEL 7 *Linux Domain Identity, Authentication, and Policy Guide*.

11.2. MODIFICATION ET RÉINITIALISATION DU SERVEUR DE RENOUVELLEMENT DE L'AUTORITÉ DE CERTIFICATION IDM

Lorsqu'un serveur de renouvellement d'autorité de certification (AC) est mis hors service, Identity Management (IdM) sélectionne automatiquement un nouveau serveur de renouvellement d'AC dans la liste des serveurs d'AC IdM. L'administrateur système ne peut pas influencer la sélection.

Pour pouvoir sélectionner le nouveau serveur de renouvellement de l'autorité de certification IdM, l'administrateur du système doit effectuer le remplacement manuellement. Choisissez le nouveau serveur de renouvellement de l'autorité de certification avant de commencer le processus de mise hors service du serveur de renouvellement actuel.

Si la configuration actuelle du serveur de renouvellement de l'autorité de certification n'est pas valide, réinitialisez le serveur de renouvellement de l'autorité de certification IdM.

Suivez cette procédure pour modifier ou réinitialiser le serveur de renouvellement de l'autorité de certification.

Conditions préalables

- Vous disposez des informations d'identification de l'administrateur IdM.

Procédure

1. Obtenir les informations d'identification de l'administrateur IdM :

```
~]$ kinit admin
Password for admin@IDM.EXAMPLE.COM:
```

2. Optionnellement, pour savoir quels serveurs IdM dans le déploiement ont le rôle d'autorité de certification nécessaire pour être éligibles à devenir le nouveau serveur de renouvellement de l'autorité de certification :

```
~]$ ipa server-role-find --role 'CA server'
----------------------
2 server roles matched
----------------------
  Server name: server.idm.example.com
  Role name: CA server
  Role status: enabled

  Server name: replica.idm.example.com
  Role name: CA server
  Role status: enabled
----------------------------
Number of entries returned 2
----------------------------
```

   Il y a deux serveurs d'autorité de certification dans le déploiement.

3. En option, pour savoir quel serveur CA est le serveur de renouvellement CA actuel, entrez :

```
~]$ ipa config-show | grep 'CA renewal'
  IPA CA renewal master: server.idm.example.com
```

   Le serveur de renouvellement actuel est **server.idm.example.com**.

4. Pour modifier la configuration du serveur de renouvellement, utilisez l'utilitaire **ipa config-mod** avec l'option **--ca-renewal-master-server**:

```
~]$ ipa config-mod --ca-renewal-master-server replica.idm.example.com | grep 'CA renewal'
  IPA CA renewal master: replica.idm.example.com
```

IMPORTANT

Vous pouvez également passer à un nouveau serveur de renouvellement de l'autorité de certification en utilisant :

- la commande **ipa-cacert-manage --renew**. Cette commande renouvelle le certificat de l'autorité de certification *and* et fait du serveur de l'autorité de certification sur lequel vous exécutez la commande le nouveau serveur de renouvellement de l'autorité de certification.
- la commande **ipa-cert-fix**. Cette commande rétablit le déploiement lorsque des certificats expirés sont à l'origine d'échecs. Elle fait également du serveur d'autorité de certification sur lequel vous exécutez la commande le nouveau serveur de renouvellement de l'autorité de certification.
  Pour plus de détails, voir Renouvellement des certificats système expirés lorsque l'IdM est hors ligne.

11.3. PASSAGE D'UNE AUTORITÉ DE CERTIFICATION EXTERNE À UNE AUTORITÉ DE CERTIFICATION AUTO-SIGNÉE DANS L'IDM

Suivez cette procédure pour passer d'un certificat signé en externe à un certificat auto-signé de l'autorité de certification (AC) de la gestion des identités (IdM). Avec une autorité de certification auto-signée, le renouvellement du certificat de l'autorité de certification est géré automatiquement : un administrateur système n'a pas besoin de soumettre une demande de signature de certificat (CSR) à une autorité externe.

Le passage d'une autorité de certification externe à une autorité de certification auto-signée ne remplace que le certificat de l'autorité de certification. Les certificats signés par l'ancienne autorité de certification restent valables et continuent d'être utilisés. Par exemple, la chaîne de certificats pour le certificat **LDAP** reste inchangée même après le passage à une autorité de certification auto-signée :

external_CA certificat > **IdM CA** certificat > **LDAP** certificat

Conditions préalables

- Vous avez un accès **root** au serveur de renouvellement de l'autorité de certification IdM et à tous les clients et serveurs IdM.

Procédure

1. Sur le serveur de renouvellement de l'autorité de certification IdM, renouveler le certificat de l'autorité de certification en tant que certificat auto-signé :

```
# ipa-cacert-manage renew --self-signed
Renewing CA certificate, please wait
CA certificate successfully renewed
```

```
The ipa-cacert-manage command was successful
```

2. **SSH** à tous les autres serveurs et clients IdM en tant que **root**. Par exemple :

```
# ssh root@idmclient01.idm.example.com
```

3. Sur le client IdM, mettre à jour les bases de données locales de certificats IdM avec les certificats du serveur :

```
[idmclient01 ~]# ipa-certupdate
Systemwide CA database updated.
Systemwide CA database updated.
The ipa-certupdate command was successful
```

4. Optionnellement, pour vérifier si la mise à jour a réussi et si le nouveau certificat d'autorité de certification a été ajouté au fichier **/etc/ipa/ca.crt**:

```
[idmclient01 ~]$ openssl crl2pkcs7 -nocrl -certfile /etc/ipa/ca.crt | openssl pkcs7 -
print_certs -text -noout
[...]
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 39 (0x27)
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: O=IDM.EXAMPLE.COM, CN=Certificate Authority
        Validity
            Not Before: Jul  1 16:32:45 2019 GMT
            Not After : Jul  1 16:32:45 2039 GMT
        Subject: O=IDM.EXAMPLE.COM, CN=Certificate Authority
[...]
```

La sortie montre que la mise à jour a réussi puisque le nouveau certificat d'autorité de certification est listé avec les anciens certificats d'autorité de certification.

11.4. RENOUVELLEMENT DU SERVEUR DE RENOUVELLEMENT DE L'AUTORITÉ DE CERTIFICATION IDM AVEC UN CERTIFICAT SIGNÉ EN EXTERNE

Cette section décrit comment renouveler le certificat de l'autorité de certification (AC) de Identity Management (IdM) en utilisant une AC externe pour signer la demande de signature de certificat (CSR). Dans cette configuration, le serveur de l'autorité de certification IdM est une sous-autorité de certification de l'autorité de certification externe. L'autorité de certification externe peut être un serveur de certificats Active Directory (AD CS), mais ce n'est pas obligatoire.

Si l'autorité de certification externe est AD CS, vous pouvez spécifier le modèle que vous souhaitez pour le certificat de l'autorité de certification IdM dans la RSC. Un modèle de certificat définit les politiques et les règles qu'une autorité de certification utilise lorsqu'elle reçoit une demande de certificat. Les modèles de certificat dans AD correspondent aux profils de certificat dans IdM.

Vous pouvez définir un modèle AD CS spécifique par son identifiant d'objet (OID). Les OID sont des valeurs numériques uniques émises par diverses autorités pour identifier de manière unique des éléments de données, des syntaxes et d'autres parties d'applications distribuées.

Vous pouvez également définir un modèle AD CS spécifique par son nom. Par exemple, le nom du profil par défaut utilisé dans une RSC soumise par une AC IdM à une CS AD est **subCA**.

Pour définir un profil en spécifiant son OID ou son nom dans la CSR, utilisez l'option **external-ca-profile**. Pour plus de détails, voir la page de manuel **ipa-cacert-manage**.

Outre l'utilisation d'un modèle de certificat prêt à l'emploi, vous pouvez également créer un modèle de certificat personnalisé dans l'AD CS et l'utiliser dans la CSR.

Conditions préalables

- Vous disposez d'un accès root au serveur de renouvellement de l'autorité de certification IdM.

Procédure

Suivez cette procédure pour renouveler le certificat de l'autorité de certification IdM avec une signature externe, que le certificat actuel de l'autorité de certification soit auto-signé ou signé de manière externe.

1. Créer une RSC à soumettre à l'autorité de certification externe :

   - Si l'autorité de certification externe est un CS AD, utilisez l'option **--external-ca-type=ms-cs**. Si vous souhaitez un modèle différent du modèle par défaut **subCA**, indiquez-le à l'aide de l'option **--external-ca-profile**:

     ```
     ~]# ipa-cacert-manage renew --external-ca --external-ca-type=ms-cs [--external-ca-profile=PROFILE]
     Exporting CA certificate signing request, please wait
     The next step is to get /var/lib/ipa/ca.csr signed by your CA and re-run ipa-cacert-manage as:
     ipa-cacert-manage renew --external-cert-file=/path/to/signed_certificate --external-cert-file=/path/to/external_ca_certificate
     The ipa-cacert-manage command was successful
     ```

   - Si l'autorité de certification externe n'est pas une autorité de certification AD :

     ```
     ~]# ipa-cacert-manage renew --external-ca
     Exporting CA certificate signing request, please wait
     The next step is to get /var/lib/ipa/ca.csr signed by your CA and re-run ipa-cacert-manage as:
     ipa-cacert-manage renew --external-cert-file=/path/to/signed_certificate --external-cert-file=/path/to/external_ca_certificate
     The ipa-cacert-manage command was successful
     ```

     La sortie montre qu'un CSR a été créé et qu'il est stocké dans le fichier **/var/lib/ipa/ca.csr**.

2. Soumettre le CSR situé dans **/var/lib/ipa/ca.csr** à l'autorité de certification externe. La procédure diffère selon le service utilisé comme autorité de certification externe.

3. Récupérer le certificat émis et la chaîne de certificats de l'autorité de certification émettrice dans un blob codé en base 64, qui est :

   - un fichier PEM si l'autorité de certification externe n'est pas une autorité de certification AD.

   - un certificat Base_64 si l'autorité de certification externe est un CS AD.

La procédure diffère d'un service de certification à l'autre. En général, un lien de téléchargement sur une page web ou dans l'e-mail de notification permet à l'administrateur de télécharger tous les certificats requis.

Si l'autorité de certification externe est un CS AD et que vous avez soumis la RSC avec un modèle connu via la fenêtre de gestion de l'autorité de certification de Microsoft Windows, le CS AD émet immédiatement le certificat et la boîte de dialogue Enregistrer le certificat apparaît dans l'interface web du CS AD, demandant où enregistrer le certificat émis.

4. Exécutez à nouveau la commande **ipa-cacert-manage renew**, en ajoutant tous les fichiers de certificats d'autorité de certification nécessaires pour fournir une chaîne de certificats complète. Spécifiez autant de fichiers que nécessaire, en utilisant plusieurs fois l'option **--external-cert-file**:

```
~]# ipa-cacert-manage renew --external-cert-file=/path/to/signed_certificate --external-cert-file=/path/to/external_ca_certificate_1 --external-cert-file=/path/to/external_ca_certificate_2
```

5. Sur tous les serveurs et clients IdM, mettre à jour les bases de données locales de certificats IdM avec les certificats du serveur :

```
[client ~]$ ipa-certupdate
Systemwide CA database updated.
Systemwide CA database updated.
The ipa-certupdate command was successful
```

6. Optionnellement, pour vérifier si la mise à jour a réussi et si le nouveau certificat d'autorité de certification a été ajouté au fichier **/etc/ipa/ca.crt**:

```
[client ~]$ openssl crl2pkcs7 -nocrl -certfile /etc/ipa/ca.crt | openssl pkcs7 -print_certs -text -noout
[...]
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 39 (0x27)
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: O=IDM.EXAMPLE.COM, CN=Certificate Authority
        Validity
            Not Before: Jul  1 16:32:45 2019 GMT
            Not After : Jul  1 16:32:45 2039 GMT
        Subject: O=IDM.EXAMPLE.COM, CN=Certificate Authority
[...]
```

La sortie montre que la mise à jour a réussi puisque le nouveau certificat d'autorité de certification est listé avec les anciens certificats d'autorité de certification.

CHAPITRE 12. RENOUVELLEMENT DES CERTIFICATS SYSTÈME EXPIRÉS LORSQUE L'IDM EST HORS LIGNE

Si un certificat système a expiré, la gestion des identités (IdM) ne démarre pas. IdM prend en charge le renouvellement des certificats système même dans cette situation en utilisant l'outil **ipa-cert-fix**.

- Assurez-vous que le service LDAP fonctionne en entrant la commande **ipactl start --ignore-service-failures** sur l'hôte.

12.1. RENOUVELLEMENT DES CERTIFICATS SYSTÈME EXPIRÉS SUR UN SERVEUR DE RENOUVELLEMENT DE L'AUTORITÉ DE CERTIFICATION

Cette section décrit comment appliquer l'outil **ipa-cert-fix** aux certificats IdM expirés.

IMPORTANT

Si vous exécutez l'outil **ipa-cert-fix** sur un hôte CA (Autorité de certification) qui n'est pas le serveur de renouvellement CA, et que l'utilitaire renouvelle les certificats partagés, cet hôte devient automatiquement le nouveau serveur de renouvellement CA dans le domaine. Il doit toujours y avoir un seul serveur de renouvellement de l'autorité de certification dans le domaine pour éviter les incohérences.

Conditions préalables

- Se connecter au serveur avec les droits d'administration

Procédure

1. Lancez l'outil **ipa-cert-fix** pour analyser le système et dresser la liste des certificats expirés qui doivent être renouvelés :

```
# ipa-cert-fix
...
The following certificates will be renewed:

Dogtag sslserver certificate:
  Subject: CN=ca1.example.com,O=EXAMPLE.COM 201905222205
  Serial:  13
  Expires: 2019-05-12 05:55:47
...
Enter "yes" to proceed:
```

2. Saisissez **yes** pour lancer la procédure de renouvellement :

```
Enter "yes" to proceed: yes
Proceeding.
Renewed Dogtag sslserver certificate:
  Subject: CN=ca1.example.com,O=EXAMPLE.COM 201905222205
  Serial:  268369925
  Expires: 2021-08-14 02:19:33
...
```

```
Becoming renewal master.
The ipa-cert-fix command was successful
```

Il peut s'écouler jusqu'à une minute avant que **ipa-cert-fix** ne renouvelle tous les certificats expirés.

3. Si vous le souhaitez, vérifiez que tous les services sont en cours d'exécution :

```
# ipactl status
Directory Service: RUNNING
krb5kdc Service: RUNNING
kadmin Service: RUNNING
httpd Service: RUNNING
ipa-custodia Service: RUNNING
pki-tomcatd Service: RUNNING
ipa-otpd Service: RUNNING
ipa: INFO: The ipactl command was successful
```

À ce stade, les certificats ont été renouvelés et les services fonctionnent. L'étape suivante consiste à vérifier les autres serveurs du domaine IdM.

NOTE

Si vous devez réparer des certificats sur plusieurs serveurs d'autorité de certification :

1. Après avoir vérifié que la réplication LDAP fonctionne dans la topologie, exécutez d'abord **ipa-cert-fix** sur un serveur d'autorité de certification, conformément à la procédure ci-dessus.

2. Avant d'exécuter **ipa-cert-fix** sur un autre serveur d'autorité de certification, déclenchez les renouvellements de Certmonger pour les certificats partagés via **getcert-resubmit** (sur l'autre serveur d'autorité de certification), afin d'éviter le renouvellement inutile des certificats partagés.

12.2. VÉRIFICATION DES AUTRES SERVEURS IDM DANS LE DOMAINE IDM APRÈS LE RENOUVELLEMENT

Après avoir renouvelé les certificats du serveur de renouvellement de l'autorité de certification à l'aide de l'outil **ipa-cert-fix**, vous devez :

- Redémarrer tous les autres serveurs de gestion des identités (IdM) du domaine.
- Vérifier si le certificateur a renouvelé les certificats.
- S'il existe d'autres répliques d'autorité de certification (CA) dont les certificats système ont expiré, renouvelez également ces certificats à l'aide de l'outil **ipa-cert-fix**.

Conditions préalables

- Connectez-vous au serveur avec des droits d'administration.

Procédure

1. Redémarrer IdM avec le paramètre **--force**:

```
# ipactl restart --force
```

Avec le paramètre **--force**, l'utilitaire **ipactl** ignore les échecs de démarrage des services individuels. Par exemple, si le serveur est également une autorité de certification dont les certificats ont expiré, le service **pki-tomcat** ne démarre pas. Ce phénomène est attendu et ignoré en raison de l'utilisation du paramètre **--force**.

2. Après le redémarrage, vérifiez que le service **certmonger** a renouvelé les certificats (l'état des certificats indique MONITORING) :

```
# getcert list | egrep '^Request|status:|subject:'
Request ID '20190522120745':
        status: MONITORING
        subject: CN=IPA RA,O=EXAMPLE.COM 201905222205
Request ID '20190522120834':
        status: MONITORING
        subject: CN=Certificate Authority,O=EXAMPLE.COM 201905222205
...
```

Il peut s'écouler un certain temps avant que **certmonger** ne renouvelle les certificats partagés sur la réplique.

3. Si le serveur est également une autorité de certification, la commande précédente indique **CA_UNREACHABLE** pour le certificat utilisé par le service **pki-tomcat**:

```
Request ID '20190522120835':
        status: CA_UNREACHABLE
        subject: CN=ca2.example.com,O=EXAMPLE.COM 201905222205
...
```

4. Pour renouveler ce certificat, utilisez l'utilitaire **ipa-cert-fix**:

```
# ipa-cert-fix
Dogtag sslserver certificate:
  Subject: CN=ca2.example.com,O=EXAMPLE.COM
  Serial:  3
  Expires: 2019-05-11 12:07:11

Enter "yes" to proceed: yes
Proceeding.
Renewed Dogtag sslserver certificate:
  Subject: CN=ca2.example.com,O=EXAMPLE.COM 201905222205
  Serial:  15
  Expires: 2019-08-14 04:25:05

The ipa-cert-fix command was successful
```

Désormais, tous les certificats IdM ont été renouvelés et fonctionnent correctement.

CHAPITRE 13. REMPLACEMENT DES CERTIFICATS DU SERVEUR WEB ET DU SERVEUR LDAP S'ILS N'ONT PAS ENCORE EXPIRÉ SUR UNE RÉPLIQUE IDM

En tant qu'administrateur du système de gestion des identités (IdM), vous pouvez remplacer manuellement les certificats des services Web (ou **httpd**) et LDAP (ou **Directory**) fonctionnant sur un serveur IdM. Par exemple, cela peut être nécessaire si les certificats arrivent à expiration et si l'utilitaire **certmonger** n'est pas configuré pour renouveler les certificats automatiquement ou si les certificats sont signés par une autorité de certification (AC) externe.

L'exemple installe les certificats pour les services fonctionnant sur le serveur IdM **server.idm.example.com**. Vous obtenez les certificats auprès d'une autorité de certification externe.

NOTE

Les certificats des services HTTP et LDAP ont des paires de clés et des noms d'objet différents sur les différents serveurs IdM et vous devez donc renouveler les certificats sur chaque serveur IdM individuellement.

Conditions préalables

- Sur au moins une autre réplique IdM dans la topologie avec laquelle le serveur IdM a un accord de réplication, les certificats web et LDAP sont toujours valides. Il s'agit d'une condition préalable à la commande **ipa-server-certinstall**. Cette commande nécessite une connexion **TLS** pour communiquer avec d'autres répliques IdM. Toutefois, si les certificats ne sont pas valides, cette connexion ne peut pas être établie et la commande **ipa-server-certinstall** échoue. Dans ce cas, voir Remplacement des certificats du serveur web et du serveur LDAP s'ils ont expiré dans l'ensemble du déploiement IdM.
- Vous avez un accès **root** au serveur IdM.
- Vous connaissez le mot de passe de **Directory Manager**.
- Vous avez accès à un fichier contenant la chaîne de certificats de l'autorité de certification externe, *ca_certificate_chain_file.crt*.

Procédure

1. Installez les certificats contenus dans *ca_certificate_chain_file.crt* en tant que certificats d'autorité de certification supplémentaires pour IdM :

```
# ipa-cacert-manage install
```

2. Mettre à jour les bases de données de certificats IdM locales avec les certificats provenant de *ca_certicate_chain_file.crt*:

```
# ipa-certupdate
```

3. Générez une clé privée et une demande de signature de certificat (CSR) à l'aide de l'utilitaire **OpenSSL**:

```
$ openssl req -new -newkey rsa:2048 -days 365 -nodes -keyout new.key -out new.csr -
addext "subjectAltName = DNS:ipa-ca.idm.example.test" -subj
'/CN=server.idm.example.com,O=IDM.EXAMPLE.COM'
```

Soumettre la CSR à l'autorité de certification externe. La procédure diffère selon le service utilisé comme autorité de certification externe. Une fois que l'autorité de certification a signé le certificat, importez le certificat sur le serveur IdM.

4. Sur le serveur IdM, remplacez l'ancienne clé privée et l'ancien certificat du serveur web Apache par la nouvelle clé et le nouveau certificat signé :

```
# ipa-server-certinstall -w --pin=password new.key new.crt
```

Dans la commande ci-dessus :

- L'option **-w** indique que vous installez un certificat dans le serveur web.
- L'option **--pin** spécifie le mot de passe protégeant la clé privée.

5. Lorsque vous y êtes invité, saisissez le mot de passe **Directory Manager**.
6. Remplacez l'ancienne clé privée et l'ancien certificat du serveur LDAP par la nouvelle clé et le nouveau certificat signé :

```
# ipa-server-certinstall -d --pin=password new.key new.cert
```

Dans la commande ci-dessus :

- L'option **-d** indique que vous installez un certificat dans le serveur LDAP.
- L'option **--pin** spécifie le mot de passe protégeant la clé privée.

7. Lorsque vous y êtes invité, saisissez le mot de passe **Directory Manager**.
8. Redémarrez le service **httpd**:

```
# systemctl restart httpd.service
```

9. Redémarrez le service **Directory**:

```
# systemctl restart dirsrv@IDM.EXAMPLE.COM.service
```

Ressources supplémentaires

- Convertir les formats de certificats pour qu'ils fonctionnent avec l'IdM
- La page de manuel **ipa-server-certinstall(1)**

CHAPITRE 14. REMPLACEMENT DES CERTIFICATS DU SERVEUR WEB ET DU SERVEUR LDAP S'ILS ONT EXPIRÉ DANS L'ENSEMBLE DU DÉPLOIEMENT IDM

La gestion de l'identité (IdM) utilise les certificats de service suivants :

- Le certificat du serveur LDAP (ou **Directory**)
- Le certificat du serveur web (ou **httpd**)
- Le certificat PKINIT

Dans un déploiement IdM sans autorité de certification, **certmonger** n'assure pas par défaut le suivi des certificats de service IdM et ne notifie pas leur expiration. Si l'administrateur du système IdM ne met pas en place manuellement des notifications pour ces certificats ou ne configure pas **certmonger** pour qu'il les suive, les certificats expireront sans préavis.

Cette section décrit comment remplacer manuellement les certificats expirés pour les services **httpd** et LDAP fonctionnant sur le serveur IdM **server.idm.example.com**.

NOTE

Les certificats des services HTTP et LDAP ont des paires de clés et des noms de sujets différents sur les différents serveurs IdM. Par conséquent, vous devez renouveler les certificats sur chaque serveur IdM individuellement.

Conditions préalables

- Les certificats HTTP et LDAP ont expiré sur *all* IdM replicas dans la topologie. Si ce n'est pas le cas, voir Remplacement des certificats du serveur Web et du serveur LDAP s'ils n'ont pas encore expiré sur une réplique IdM.
- Vous avez un accès **root** au serveur IdM et aux répliques.
- Vous connaissez le mot de passe de **Directory Manager**.
- Vous avez créé des sauvegardes des répertoires et fichiers suivants :
  - **/etc/dirsrv/slapd-*IDM-EXAMPLE-COM*/**
  - **/etc/httpd/alias**
  - **/var/lib/certmonger**
  - **/var/lib/ipa/certs/**

Procédure

1. Si vous n'utilisez pas la même autorité de certification pour signer les nouveaux certificats ou si le certificat d'autorité de certification déjà installé n'est plus valide, mettez à jour les informations relatives à l'autorité de certification externe dans votre base de données locale à l'aide d'un fichier contenant une chaîne de certificats d'autorité de certification valide de l'autorité de certification externe. Le fichier est accepté dans les formats suivants : certificat PEM et DER, chaîne de certificats PKCS#7, PKCS#8, clé privée brute et PKCS#12.

a. Installer les certificats disponibles sur *ca_certificate_chain_file.crt* en tant que certificats d'autorité de certification supplémentaires dans IdM :

```
# ipa-cacert-manage install ca_certificate_chain_file.crt
```

b. Mettre à jour les bases de données de certificats IdM locales avec les certificats provenant de *ca_certicate_chain_file.crt*:

```
# ipa-certupdate
```

2. Demander les certificats pour **httpd** et LDAP :

a. Créez une demande de signature de certificat (CSR) pour le serveur web Apache fonctionnant sur vos instances IdM auprès de votre autorité de certification tierce à l'aide de l'utilitaire **OpenSSL**:

```
$ openssl req -new -newkey rsa:2048 -nodes -keyout /var/lib/ipa/private/httpd.key -
out /tmp/http.csr -addext 'subjectAltName = DNS:server.idm.example.com,
otherName:1.3.6.1.4.1.311.20.2.3;UTF8:HTTP/server.idm.example.com@IDM.EXAM
PLE.COM' -subj '/O=IDM.EXAMPLE.COM/CN=server.idm.example.com'
```

La création d'une nouvelle clé privée est facultative. Si vous disposez toujours de la clé privée originale, vous pouvez utiliser l'option **-in** avec la commande **openssl req** pour spécifier le nom du fichier d'entrée à partir duquel la demande doit être lue.

b. Créez une demande de signature de certificat (CSR) pour le serveur LDAP fonctionnant sur vos instances IdM auprès de votre autorité de certification tierce à l'aide de l'utilitaire **OpenSSL**:

```
$ openssl req -new -newkey rsa:2048 -nodes -keyout ~/ldap.key -out /tmp/ldap.csr -
addext 'subjectAltName = DNS:server.idm.example.com,
otherName:1.3.6.1.4.1.311.20.2.3;UTF8:ldap/server.idm.example.com@IDM.EXAMP
LE.COM' -subj '/O=IDM.EXAMPLE.COM/CN=server.idm.example.com'
```

La création d'une nouvelle clé privée est facultative. Si vous disposez toujours de la clé privée originale, vous pouvez utiliser l'option **-in** avec la commande **openssl req** pour spécifier le nom du fichier d'entrée à partir duquel la demande doit être lue.

c. Soumettez les CSR, **/tmp/http.csr** et **tmp/ldap.csr**, à l'autorité de certification externe et obtenez un certificat pour **httpd** et un certificat pour LDAP. La procédure diffère selon le service utilisé comme autorité de certification externe.

3. Installer le certificat pour **httpd**:

```
# cp /path/to/httpd.crt /var/lib/ipa/certs/
```

4. Installer le certificat LDAP dans une base de données NSS :

a. [Facultatif] Dresser la liste des certificats disponibles :

```
# certutil -d /etc/dirsrv/slapd-IDM-EXAMPLE-COM/ -L
Certificate Nickname                                         Trust Attributes
                                                             SSL,S/MIME,JAR/XPI

Server-Cert                                                  u,u,u
```

–

Le surnom du certificat par défaut est **Server-Cert**, mais il est possible qu'un nom différent ait été appliqué.

b. Supprimez l'ancien certificat non valide de la base de données NSS (**NSSDB**) en utilisant le pseudonyme du certificat de l'étape précédente :

```
# certutil -D -d /etc/dirsrv/slapd-IDM-EXAMPLE-COM/ -n 'Server-Cert' -f
/etc/dirsrv/slapd-IDM-EXAMPLE-COM/pwdfile.txt
```

c. Créer un fichier PKCS12 pour faciliter le processus d'importation dans **NSSDB**:

```
# openssl pkcs12 -export -in ldap.crt -inkey ldap.key -out ldap.p12 -name Server-
Cert
```

d. Installer le fichier PKCS#12 créé dans le site **NSSDB**:

```
# pk12util -i ldap.p12 -d /etc/dirsrv/slapd-IDM-EXAMPLE-COM/ -k
/etc/dirsrv/slapd-IDM-EXAMPLE-COM/pwdfile.txt
```

e. Vérifiez que le nouveau certificat a été importé avec succès :

```
# certutil -L -d /etc/dirsrv/slapd-IDM-EXAMPLE-COM/
```

5. Redémarrez le service **httpd**:

```
# systemctl restart httpd.service
```

6. Redémarrez le service **Directory**:

```
# systemctl restart dirsrv@IDM-EXAMPLE-COM.service
```

7. Effectuez toutes les étapes précédentes sur toutes vos répliques IdM. Il s'agit d'une condition préalable à l'établissement de connexions **TLS** entre les répliques.

8. Enrôler les nouveaux certificats dans le stockage LDAP :

a. Remplacez l'ancienne clé privée et l'ancien certificat du serveur web Apache par la nouvelle clé et le nouveau certificat signé :

```
# ipa-server-certinstall -w --pin=password /var/lib/ipa/private/httpd.key
/var/lib/ipa/certs/httpd.crt
```

Dans la commande ci-dessus :

- L'option **-w** indique que vous installez un certificat dans le serveur web.
- L'option **--pin** spécifie le mot de passe protégeant la clé privée.

b. Lorsque vous y êtes invité, saisissez le mot de passe **Directory Manager**.

c. Remplacez l'ancienne clé privée et l'ancien certificat du serveur LDAP par la nouvelle clé et le nouveau certificat signé :

```
# ipa-server-certinstall -d --pin=password /etc/dirsrv/slapd-IDM-EXAMPLE-COM/ldap.key /path/to/ldap.crt
```

Dans la commande ci-dessus :

- L'option **-d** indique que vous installez un certificat dans le serveur LDAP.
- L'option **--pin** spécifie le mot de passe protégeant la clé privée.

d. Lorsque vous y êtes invité, saisissez le mot de passe **Directory Manager**.

e. Redémarrez le service **httpd**:

```
# systemctl restart httpd.service
```

f. Redémarrez le service **Directory**:

```
# systemctl restart dirsrv@IDM-EXAMPLE-COM.service
```

9. Exécutez les commandes de l'étape précédente sur toutes les autres répliques concernées.

Ressources supplémentaires

- Convertir les formats de certificats pour qu'ils fonctionnent avec l'IdM
- homme **ipa-server-certinstall(1)**
- Comment renouveler manuellement les certificats de gestion des identités (IPA) sur RHEL 8 après leur expiration (IPA sans autorité de certification) ?

CHAPITRE 15. GÉNÉRATION DE CRL SUR LE SERVEUR DE L'AUTORITÉ DE CERTIFICATION IDM

Si votre déploiement IdM utilise une autorité de certification (CA) intégrée, il se peut que vous deviez déplacer la génération de la liste de révocation des certificats (CRL) d'un serveur de gestion des identités (IdM) à un autre. Cela peut s'avérer nécessaire, par exemple, lorsque vous souhaitez migrer le serveur vers un autre système.

Ne configurez qu'un seul serveur pour générer la CRL. Le serveur IdM qui joue le rôle d'éditeur de CRL est généralement le même que celui qui joue le rôle de serveur de renouvellement de l'autorité de certification, mais ce n'est pas obligatoire. Avant de mettre hors service le serveur éditeur de CRL, sélectionnez et configurez un autre serveur pour jouer le rôle de serveur éditeur de CRL.

Ce chapitre décrit

- Arrêt de la génération de CRL sur le serveur IdM.
- Démarrage de la génération de CRL sur la réplique IdM.

15.1. ARRÊT DE LA GÉNÉRATION DE CRL SUR UN SERVEUR IDM

Pour arrêter la génération de la liste de révocation de certificats (CRL) sur le serveur IdM CRL publisher, utilisez la commande **ipa-crlgen-manage**. Avant de désactiver la génération, vérifiez que le serveur génère réellement des CRL. Vous pouvez ensuite le désactiver.

Conditions préalables

- Vous devez être connecté en tant que root.

Procédure

1. Vérifiez si votre serveur génère la CRL :

```
[root@server ~]# ipa-crlgen-manage status
CRL generation: enabled
Last CRL update: 2019-10-31 12:00:00
Last CRL Number: 6
The ipa-crlgen-manage command was successful
```

2. Arrêter la génération de la CRL sur le serveur :

```
[root@server ~]# ipa-crlgen-manage disable
Stopping pki-tomcatd
Editing /var/lib/pki/pki-tomcat/conf/ca/CS.cfg
Starting pki-tomcatd
Editing /etc/httpd/conf.d/ipa-pki-proxy.conf
Restarting httpd
CRL generation disabled on the local host. Please make sure to configure CRL generation on another master with ipa-crlgen-manage enable.
The ipa-crlgen-manage command was successful
```

3. Vérifier si le serveur a cessé de générer des CRL :

```
[root@server ~]# ipa-crlgen-manage status
```

Le serveur a cessé de générer la CRL. L'étape suivante consiste à activer la génération de CRL sur la réplique IdM.

15.2. DÉMARRER LA GÉNÉRATION DE CRL SUR UN SERVEUR RÉPLIQUE IDM

Vous pouvez commencer à générer la liste de révocation des certificats (CRL) sur un serveur d'autorité de certification IdM à l'aide de la commande **ipa-crlgen-manage**.

Conditions préalables

- Le système RHEL doit être un serveur d'autorité de certification IdM.
- Vous devez être connecté en tant que root.

Procédure

1. Commencez à générer la CRL :

```
[root@replica1 ~]# ipa-crlgen-manage enable
Stopping pki-tomcatd
Editing /var/lib/pki/pki-tomcat/conf/ca/CS.cfg
Starting pki-tomcatd
Editing /etc/httpd/conf.d/ipa-pki-proxy.conf
Restarting httpd
Forcing CRL update
CRL generation enabled on the local host. Please make sure to have only a single CRL generation master.
The ipa-crlgen-manage command was successful
```

2. Vérifier si la CRL est générée :

```
[root@replica1 ~]# ipa-crlgen-manage status
CRL generation: enabled
Last CRL update: 2019-10-31 12:10:00
Last CRL Number: 7
The ipa-crlgen-manage command was successful
```

CHAPITRE 16. MISE HORS SERVICE D'UN SERVEUR QUI JOUE LE RÔLE DE SERVEUR DE RENOUVELLEMENT DE L'AUTORITÉ DE CERTIFICATION ET D'ÉDITEUR DE CRL

Il se peut qu'un serveur joue à la fois le rôle de serveur de renouvellement de l'autorité de certification (CA) et celui d'éditeur de la liste de révocation des certificats (CRL). Si vous devez mettre ce serveur hors ligne ou le déclasser, sélectionnez et configurez un autre serveur d'autorité de certification pour remplir ces rôles.

Dans cet exemple, l'hôte **server.idm.example.com**, qui remplit les rôles de serveur de renouvellement de l'autorité de certification et d'éditeur de CRL, doit être mis hors service. Cette procédure transfère les rôles de serveur de renouvellement de l'autorité de certification et d'éditeur de CRL à l'hôte **replica.idm.example.com** et supprime **server.idm.example.com** de l'environnement IdM.

NOTE

Il n'est pas nécessaire de configurer le même serveur pour qu'il joue à la fois le rôle de serveur de renouvellement de l'autorité de certification et celui d'éditeur de CRL.

Conditions préalables

- Vous disposez des informations d'identification de l'administrateur IdM.
- Vous avez le mot de passe root du serveur que vous déclassez.
- Vous avez au moins deux répliques CA dans votre environnement IdM.

Procédure

1. Obtenir les informations d'identification de l'administrateur IdM :

```
[user@server ~]$ kinit admin
Password for admin@IDM.EXAMPLE.COM:
```

2. *(Optional)* Si vous n'êtes pas sûr des serveurs qui jouent le rôle de serveur de renouvellement de l'autorité de certification et d'éditeur de CRL :

   a. Affichez le serveur de renouvellement de l'autorité de certification actuel. Vous pouvez exécuter la commande suivante à partir de n'importe quel serveur IdM :

```
[user@server ~]$ ipa config-show | grep 'CA renewal'
  IPA CA renewal master: server.idm.example.com
```

   b. Teste si un hôte est l'éditeur actuel de la CRL.

```
[user@server ~]$ ipa-crlgen-manage status
CRL generation: enabled
Last CRL update: 2019-10-31 12:00:00
Last CRL Number: 6
The ipa-crlgen-manage command was successful
```

   Un serveur CA qui ne génère pas de CRL affiche **CRL generation: disabled**.

```
[user@replica ~]$ ipa-crlgen-manage status
CRL generation: disabled
The ipa-crlgen-manage command was successful
```

Continuez à saisir cette commande sur les serveurs de l'autorité de certification jusqu'à ce que vous trouviez le serveur d'édition de la CRL.

c. Affichez tous les autres serveurs CA que vous pouvez promouvoir pour remplir ces rôles. Cet environnement possède deux serveurs CA.

```
[user@server ~]$ ipa server-role-find --role 'CA server'
----------------------
2 server roles matched
----------------------
  Server name: server.idm.example.com
  Role name: CA server
  Role status: enabled
  Server name: replica.idm.example.com
  Role name: CA server
  Role status: enabled
----------------------------
Number of entries returned 2
----------------------------
```

3. Définissez **replica.idm.example.com** comme serveur de renouvellement de l'autorité de certification.

```
[user@server ~]$ ipa config-mod --ca-renewal-master-server replica.idm.example.com
```

4. Sur **server.idm.example.com**:

a. Désactiver la tâche de mise à jour des certificats :

```
[root@server ~]# pki-server ca-config-set ca.certStatusUpdateInterval 0
```

b. Redémarrer les services IdM :

```
[user@server ~]$ ipactl restart
```

5. Sur **replica.idm.example.com**:

a. Activer la tâche de mise à jour des certificats :

```
[root@server ~]# pki-server ca-config-unset ca.certStatusUpdateInterval
```

b. Redémarrer les services IdM :

```
[user@replica ~]$ ipactl restart
```

6. Sur **server.idm.example.com**, arrêtez de générer la CRL.

```
[user@server ~]$ ipa-crlgen-manage disable
Stopping pki-tomcatd
```

```
Editing /var/lib/pki/pki-tomcat/conf/ca/CS.cfg
Starting pki-tomcatd
Editing /etc/httpd/conf.d/ipa-pki-proxy.conf
Restarting httpd
CRL generation disabled on the local host. Please make sure to configure CRL generation on
another master with ipa-crlgen-manage enable.
The ipa-crlgen-manage command was successful
```

7. Sur **replica.idm.example.com**, commencez à générer la CRL.

```
[user@replica ~]$ ipa-crlgen-manage enable
Stopping pki-tomcatd
Editing /var/lib/pki/pki-tomcat/conf/ca/CS.cfg
Starting pki-tomcatd
Editing /etc/httpd/conf.d/ipa-pki-proxy.conf
Restarting httpd
Forcing CRL update
CRL generation enabled on the local host. Please make sure to have only a single CRL
generation master.
The ipa-crlgen-manage command was successful
```

8. Arrêter les services IdM sur **server.idm.example.com**:

```
[user@server ~]$ ipactl stop
```

9. Sur **replica.idm.example.com**, supprimez **server.idm.example.com** de l'environnement IdM.

```
[user@replica ~]$ ipa server-del server.idm.example.com
```

10. Sur **server.idm.example.com**, utilisez la commande **ipa-server-install --uninstall** en tant que compte root :

```
[root@server ~]# ipa-server-install --uninstall
...
Are you sure you want to continue with the uninstall procedure? [no]: yes
```

Verification steps

- Afficher le serveur de renouvellement de l'autorité de certification actuel.

```
[user@replica ~]$ ipa config-show | grep 'CA renewal'
  IPA CA renewal master: replica.idm.example.com
```

- Confirmez que l'hôte **replica.idm.example.com** génère la CRL.

```
[user@replica ~]$ ipa-crlgen-manage status
CRL generation: enabled
Last CRL update: 2019-10-31 12:10:00
Last CRL Number: 7
The ipa-crlgen-manage command was successful
```

Ressources supplémentaires

- Modification et réinitialisation du serveur de renouvellement de l'autorité de certification IdM
- Génération de CRL sur le serveur de l'autorité de certification IdM
- Désinstallation d'une réplique IdM

CHAPITRE 17. OBTENTION D'UN CERTIFICAT IDM POUR UN SERVICE À L'AIDE DE CERTMONGER

17.1. APERÇU DE CERTMONGER

Lorsque la gestion des identités (IdM) est installée avec une autorité de certification (AC) IdM intégrée, elle utilise le service **certmonger** pour suivre et renouveler les certificats de système et de service. Lorsque le certificat arrive à sa date d'expiration, **certmonger** gère le processus de renouvellement :

- régénérer une demande de signature de certificat (CSR) en utilisant les options fournies dans la demande initiale.
- soumettre la CSR à l'autorité de certification IdM à l'aide de la commande IdM API **cert-request**.
- réception du certificat de l'autorité de certification IdM.
- l'exécution d'une commande de pré-sauvegarde si elle est spécifiée dans la demande initiale.
- installer le nouveau certificat à l'emplacement spécifié dans la demande de renouvellement : soit dans une base de données **NSS**, soit dans un fichier.
- l'exécution d'une commande post-save si elle est spécifiée dans la demande initiale. Par exemple, la commande post-sauvetage peut demander à **certmonger** de redémarrer un service pertinent, de sorte que le service prenne en charge le nouveau certificat.

Types de certificats certmonger tracks

Les certificats peuvent être divisés en certificats de système et de service.

Contrairement aux certificats de service (par exemple, pour **HTTP**, **LDAP** et **PKINIT**), qui ont des paires de clés et des noms de sujets différents sur des serveurs différents, les certificats du système IdM et leurs clés sont partagés par toutes les répliques de l'autorité de certification. Les certificats du système IdM sont les suivants

- **IdM CA** certificat
- **OCSP** certificat de signature
- **IdM CA subsystem** certificats
- **IdM CA audit signing** certificat
- **IdM renewal agent** (RA) certificat
- **KRA** certificats de transport et de stockage

Le service **certmonger** assure le suivi des certificats du système et des services IdM qui ont été demandés lors de l'installation de l'environnement IdM avec une autorité de certification intégrée. **Certmonger** assure également le suivi des certificats qui ont été demandés manuellement par l'administrateur du système pour d'autres services fonctionnant sur l'hôte IdM. **Certmonger** n'assure pas le suivi des certificats d'autorité de certification externe ou des certificats d'utilisateur.

Composants Certmonger

Le service **certmonger** se compose de deux éléments principaux :

- Le site **certmonger daemon**, qui est le moteur de suivi de la liste des certificats et de lancement des commandes de renouvellement
- L'utilitaire **getcert** pour **command-line interface** (CLI), qui permet à l'administrateur système d'envoyer activement des commandes au démon **certmonger**.

Plus précisément, l'administrateur système peut utiliser l'utilitaire **getcert** pour :

- Demander un nouveau certificat
- Consulter la liste des certificats suivis par **certmonger**
- Démarrer ou arrêter le suivi d'un certificat
- Renouveler un certificat

17.2. OBTENTION D'UN CERTIFICAT IDM POUR UN SERVICE À L'AIDE DE CERTMONGER

Pour garantir que la communication entre les navigateurs et le service web exécuté sur votre client de gestion d'identité (IdM) est sécurisée et cryptée, utilisez un certificat TLS. Obtenez le certificat TLS pour votre service web auprès de l'autorité de certification (AC) IdM.

Cette section décrit comment utiliser **certmonger** pour obtenir un certificat IdM pour un service (**HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM**) fonctionnant sur un client IdM.

L'utilisation de **certmonger** pour demander le certificat automatiquement signifie que **certmonger** gère et renouvelle le certificat lorsqu'il doit être renouvelé.

Pour une représentation visuelle de ce qui se passe lorsque **certmonger** demande un certificat de service, voir Section 17.3, « Flux de communication pour le demandeur de certificat demandant un certificat de service ».

Conditions préalables

- Le serveur web est enregistré en tant que client IdM.
- Vous avez un accès root au client IdM sur lequel vous exécutez la procédure.
- Le service pour lequel vous demandez un certificat ne doit pas nécessairement préexister dans l'IdM.

Procédure

1. Sur le client **my_company.idm.example.com** IdM sur lequel le service **HTTP** est exécuté, demandez un certificat pour le service correspondant au principal **HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM** et spécifiez que
   - Le certificat doit être stocké dans le fichier local **/etc/pki/tls/certs/httpd.pem**
   - La clé privée doit être stockée dans le fichier local **/etc/pki/tls/private/httpd.key**
   - Qu'une demande d'extension pour **SubjectAltName** soit ajoutée à la demande de signature avec le nom DNS de **my_company.idm.example.com**:

     ```
     # ipa-getcert request -K HTTP/my_company.idm.example.com -k
     ```

```
/etc/pki/tls/private/httpd.key -f /etc/pki/tls/certs/httpd.pem -g 2048 -D
my_company.idm.example.com -C "systemctl restart httpd"
New signing request "20190604065735" added.
```

Dans la commande ci-dessus :

- La commande **ipa-getcert request** spécifie que le certificat doit être obtenu auprès de l'autorité de certification IdM. La commande **ipa-getcert request** est un raccourci pour **getcert request -c IPA**.
- L'option **-g** spécifie la taille de la clé à générer s'il n'y en a pas déjà une.
- L'option **-D** spécifie la valeur DNS **SubjectAltName** à ajouter à la demande.
- L'option **-C** demande à **certmonger** de redémarrer le service **httpd** après avoir obtenu le certificat.
- Pour spécifier que le certificat doit être émis avec un profil particulier, utilisez l'option **-T**.
- Pour demander à l'autorité de certification spécifiée un certificat utilisant l'émetteur nommé, utilisez l'option **-X ISSUER**.

2. Optionnellement, pour vérifier le statut de votre demande :

```
# ipa-getcert list -f /etc/pki/tls/certs/httpd.pem
Number of certificates and requests being tracked: 3.
Request ID '20190604065735':
    status: MONITORING
    stuck: no
    key pair storage: type=FILE,location='/etc/pki/tls/private/httpd.key'
    certificate: type=FILE,location='/etc/pki/tls/certs/httpd.crt'
    CA: IPA
[...]
```

La sortie montre que la demande est à l'état **MONITORING**, ce qui signifie qu'un certificat a été obtenu. Les emplacements de la paire de clés et du certificat sont ceux demandés.

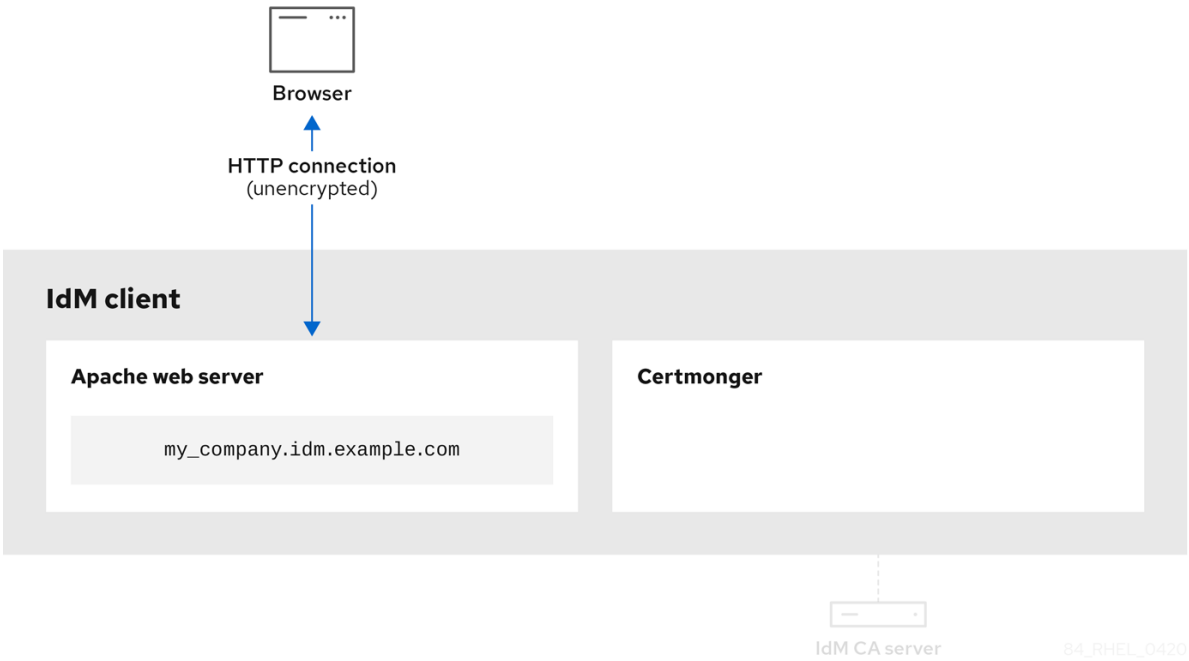
17.3. FLUX DE COMMUNICATION POUR LE DEMANDEUR DE CERTIFICAT DEMANDANT UN CERTIFICAT DE SERVICE

Les diagrammes de cette section montrent les étapes de ce qui se passe lorsque **certmonger** demande un certificat de service au serveur de l'autorité de certification (CA) de la gestion des identités (IdM). La séquence est constituée de ces diagrammes :

- Communication non cryptée
- Certmonger demande un certificat de service
- CA IdM délivrant le certificat de service
- Le certificateur qui applique le certificat de service
- Certmonger demande un nouveau certificat lorsque l'ancien est proche de l'expiration

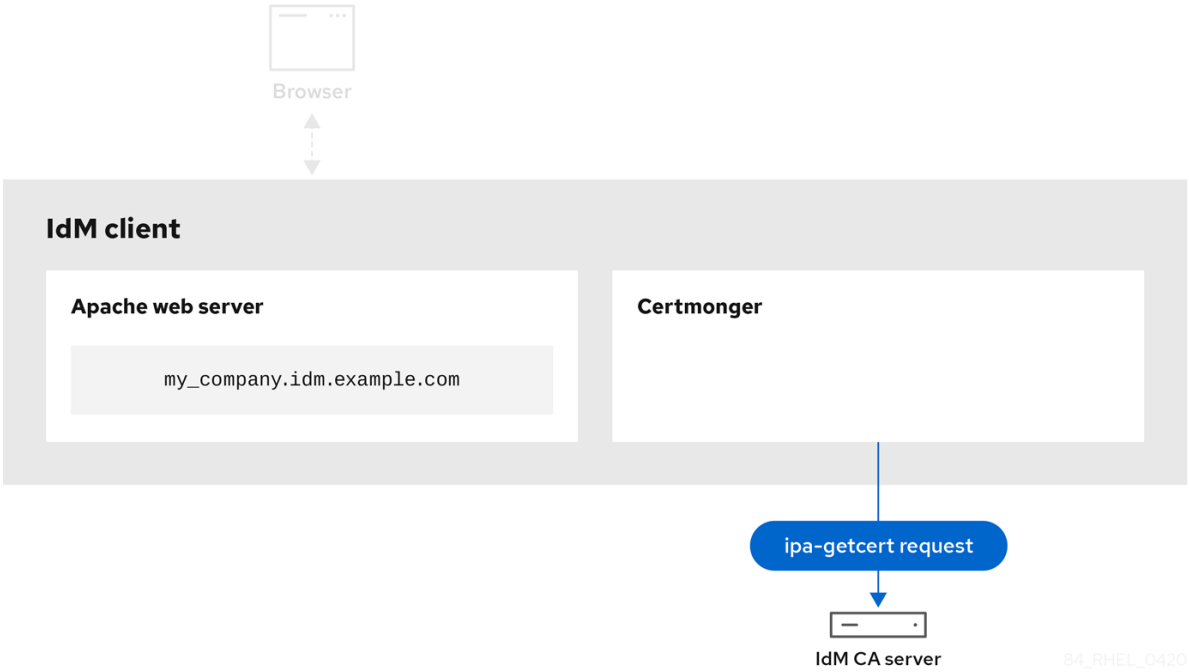
La communication non chiffrée illustre la situation initiale : sans certificat HTTPS, la communication entre le serveur web et le navigateur n'est pas chiffrée.

Figure 17.1. Communication non cryptée



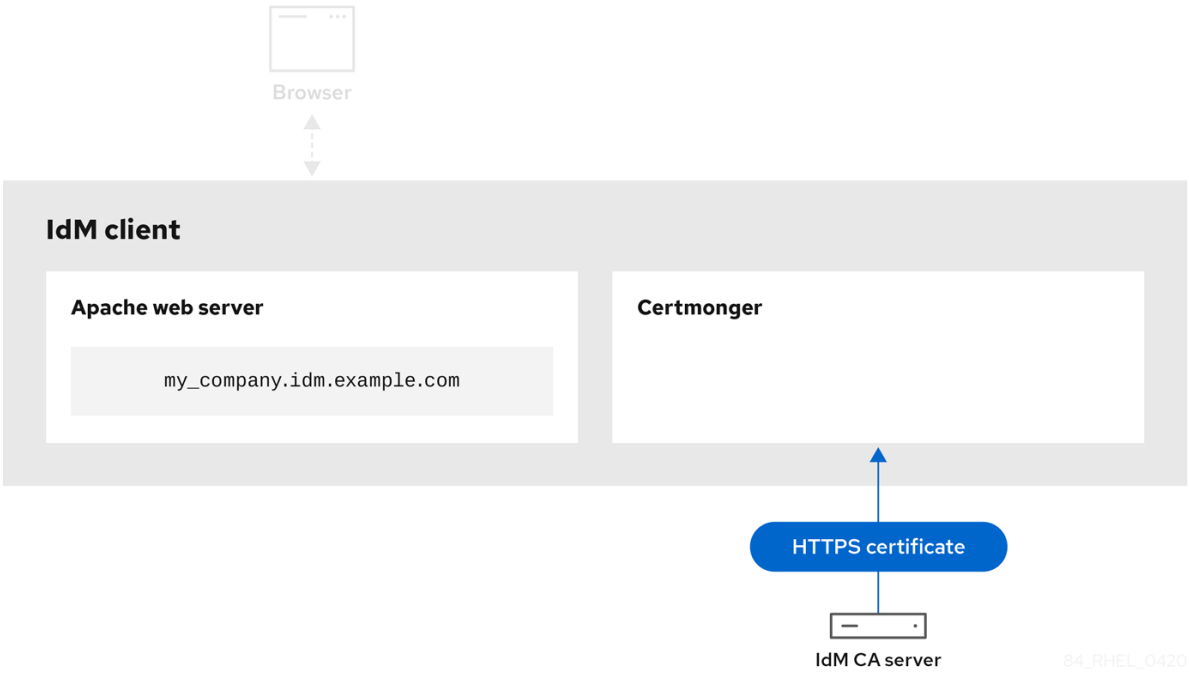
Certmonger demandant un certificat de service montre l'administrateur système utilisant **certmonger** pour demander manuellement un certificat HTTPS pour le serveur web Apache. Notez que lors de la demande d'un certificat de serveur web, certmonger ne communique pas directement avec l'autorité de certification. Il passe par IdM.

Figure 17.2. Certmonger demande un certificat de service



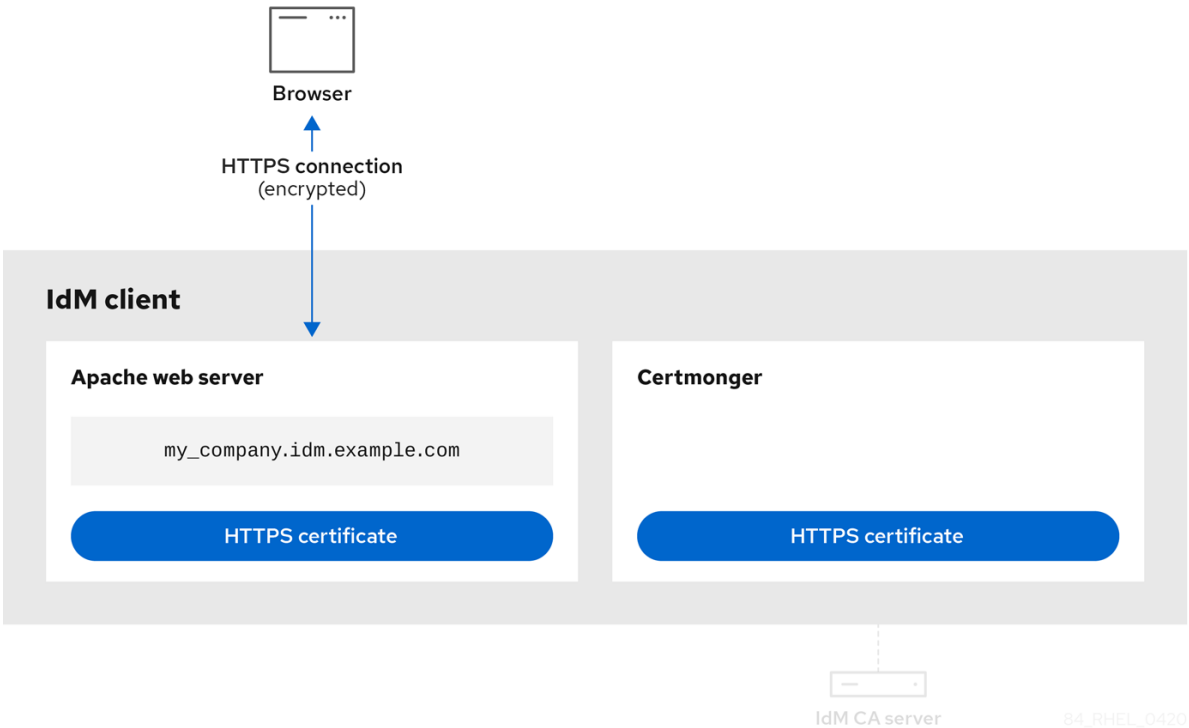
L'autorité de certification IdM délivrant le certificat de service montre une autorité de certification IdM délivrant un certificat HTTPS pour le serveur web.

Figure 17.3. CA IdM délivrant le certificat de service



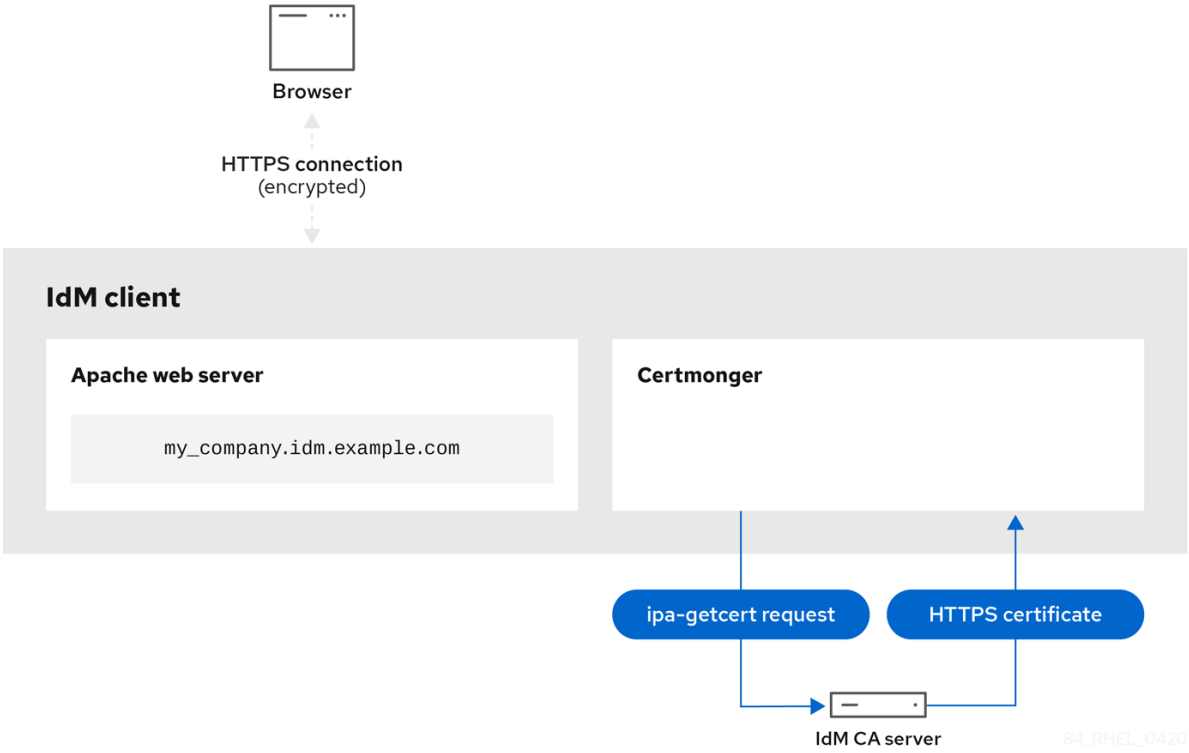
Certmonger l'application du certificat de service montre que **certmonger** place le certificat HTTPS aux endroits appropriés sur le client IdM et, si cela lui est demandé, redémarre le service **httpd**. Le serveur Apache utilise ensuite le certificat HTTPS pour crypter le trafic entre lui-même et le navigateur.

Figure 17.4. Le certificateur qui applique le certificat de service



Certmonger demandant un nouveau certificat lorsque l'ancien est proche de l' expiration montre que **certmonger** demande automatiquement le renouvellement du certificat de service auprès de l'autorité de certification IdM avant l'expiration du certificat. L'AC IdM délivre un nouveau certificat.

Figure 17.5. Certmonger demande un nouveau certificat lorsque l'ancien est proche de l'expiration



17.4. AFFICHER LES DÉTAILS D'UNE DEMANDE DE CERTIFICAT SUIVIE PAR CERTMONGER

Le service **certmonger** surveille les demandes de certificat. Lorsqu'une demande de certificat est signée avec succès, il en résulte un certificat. **Certmonger** gère les demandes de certificat, y compris les certificats qui en résultent. Cette section décrit comment afficher les détails d'une demande de certificat particulière gérée par **certmonger**.

Procédure

- Si vous savez comment spécifier la demande de certificat, énumérez les détails de cette demande de certificat en particulier. Vous pouvez, par exemple, spécifier :
  - L'identifiant de la demande
  - L'emplacement du certificat
  - Le surnom du certificat
    Par exemple, pour afficher les détails du certificat dont l'ID de demande est 20190408143846, en utilisant l'option **-v** pour afficher tous les détails des erreurs au cas où votre demande de certificat n'aurait pas abouti :

```
# getcert list -i 20190408143846 -v
Number of certificates and requests being tracked: 16.
Request ID '20190408143846':
 status: MONITORING
 stuck: no
 key pair storage: type=NSSDB,location='/etc/dirsrv/slapd-IDM-EXAMPLE-
COM',nickname='Server-Cert',token='NSS Certificate DB',pinfile='/etc/dirsrv/slapd-IDM-
EXAMPLE-COM/pwdfile.txt'
 certificate: type=NSSDB,location='/etc/dirsrv/slapd-IDM-EXAMPLE-
COM',nickname='Server-Cert',token='NSS Certificate DB'
 CA: IPA
 issuer: CN=Certificate Authority,O=IDM.EXAMPLE.COM
 subject: CN=r8server.idm.example.com,O=IDM.EXAMPLE.COM
 expires: 2021-04-08 16:38:47 CEST
 dns: r8server.idm.example.com
 principal name: ldap/server.idm.example.com@IDM.EXAMPLE.COM
 key usage: digitalSignature,nonRepudiation,keyEncipherment,dataEncipherment
 eku: id-kp-serverAuth,id-kp-clientAuth
 pre-save command:
 post-save command: /usr/libexec/ipa/certmonger/restart_dirsrv IDM-EXAMPLE-COM
 track: yes
 auto-renew: yes
```

La sortie affiche plusieurs informations sur le certificat, par exemple :

- l'emplacement du certificat ; dans l'exemple ci-dessus, il s'agit de la base de données NSS dans le répertoire **/etc/dirsrv/slapd-IDM-EXAMPLE-COM**
- le surnom du certificat ; dans l'exemple ci-dessus, il s'agit de **Server-Cert**
- le fichier qui stocke l'épingle ; dans l'exemple ci-dessus, il s'agit de **/etc/dirsrv/slapd-IDM-EXAMPLE-COM/pwdfile.txt**
- l'autorité de certification (AC) qui sera utilisée pour renouveler le certificat ; dans l'exemple ci-dessus, il s'agit de l'AC **IPA**
- la date d'expiration ; dans l'exemple ci-dessus, il s'agit de **2021-04-08 16:38:47 CEST**
- l'état du certificat ; dans l'exemple ci-dessus, l'état **MONITORING** signifie que le certificat est valide et qu'il fait l'objet d'un suivi
- la commande postérieure à la sauvegarde ; dans l'exemple ci-dessus, il s'agit du redémarrage du service **LDAP**

- Si vous ne savez pas comment spécifier la demande de certificat, indiquez les détails de tous les certificats que **certmonger** contrôle ou tente d'obtenir :

```
# getcert list
```

Ressources supplémentaires

- Voir la page de manuel **getcert list**.

17.5. DÉMARRAGE ET ARRÊT DU SUIVI DES CERTIFICATS

Cette section décrit comment utiliser les commandes **getcert stop-tracking** et **getcert start-tracking** pour contrôler les certificats. Ces deux commandes sont fournies par le service **certmonger**. L'activation du suivi des certificats est particulièrement utile si vous avez importé un certificat émis par l'autorité de certification (AC) de la gestion des identités (IdM) sur la machine à partir d'un autre client IdM. L'activation du suivi des certificats peut également constituer la dernière étape du scénario de provisionnement suivant :

1. Sur le serveur IdM, vous créez un certificat pour un système qui n'existe pas encore.
2. Vous créez le nouveau système.
3. Vous inscrivez le nouveau système en tant que client IdM.
4. Vous importez le certificat et la clé du serveur IdM sur le client IdM.
5. Vous commencez à suivre le certificat à l'aide de **certmonger** pour vous assurer qu'il est renouvelé lorsqu'il arrive à expiration.

Procédure

- Pour désactiver la surveillance d'un certificat dont l'ID de demande est 20190408143846 :

  ```
  # getcert stop-tracking -i 20190408143846
  ```

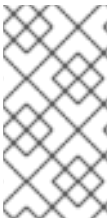
  Pour plus d'options, voir la page de manuel **getcert stop-tracking**.

- Pour permettre la surveillance d'un certificat stocké dans le fichier **/tmp/some_cert.crt**, dont la clé privée est stockée dans le fichier **/tmp/some_key.key**:

  ```
  # getcert start-tracking -c IPA -f /tmp/some_cert.crt -k /tmp/some_key.key
  ```

  Certmonger ne peut pas identifier automatiquement le type d'autorité de certification qui a émis le certificat. Pour cette raison, ajoutez l'option **-c** avec la valeur **IPA** à la commande **getcert start-tracking** si le certificat a été émis par l'autorité de certification IdM. Si l'option **-c** n'est pas ajoutée, **certmonger** entre dans l'état NEED_CA.

  Pour plus d'options, voir la page de manuel **getcert start-tracking**.



NOTE

Les deux commandes ne manipulent pas le certificat. Par exemple, **getcert stop-tracking** ne supprime pas le certificat ni ne le retire de la base de données NSS ou du système de fichiers, mais le retire simplement de la liste des certificats surveillés. De même, **getcert start-tracking** ne fait qu'ajouter un certificat à la liste des certificats surveillés.

17.6. RENOUVELLEMENT MANUEL D'UN CERTIFICAT

Lorsqu'un certificat est proche de sa date d'expiration, le démon **certmonger** émet automatiquement une commande de renouvellement en utilisant l'aide de l'autorité de certification (CA), obtient un certificat renouvelé et remplace le certificat précédent par le nouveau.

Il est également possible de renouveler manuellement un certificat à l'avance en utilisant la commande **getcert resubmit**. De cette manière, vous pouvez mettre à jour les informations contenues dans le certificat, par exemple en ajoutant un Subject Alternative Name (SAN).

Cette section décrit comment renouveler un certificat manuellement.

Procédure

- Pour renouveler un certificat dont l'identifiant de demande est 20190408143846 :

```
# getcert resubmit -i 20190408143846
```

  Pour obtenir l'identifiant de demande d'un certificat spécifique, utilisez la commande **getcert list**. Pour plus d'informations, voir la page de manuel **getcert list**.

17.7. FAIRE EN SORTE QUE CERTMONGER REPRENNE LE SUIVI DES CERTIFICATS IDM SUR UNE RÉPLIQUE D'AC

Cette procédure montre comment faire en sorte que **certmonger** reprenne le suivi des certificats de système de gestion d'identité (IdM) qui sont essentiels pour un déploiement IdM avec une autorité de certification intégrée après que le suivi des certificats a été interrompu. L'interruption peut être due au fait que l'hôte IdM a été désinscrit de l'IdM pendant le renouvellement des certificats système ou que la topologie de réplication ne fonctionne pas correctement. La procédure montre également comment faire en sorte que **certmonger** reprenne le suivi des certificats du service IdM, à savoir les certificats **HTTP**, **LDAP** et **PKINIT**.

Conditions préalables

- L'hôte sur lequel vous souhaitez reprendre les certificats du système de suivi est un serveur IdM qui est également une autorité de certification IdM, mais pas le serveur de renouvellement de l'autorité de certification IdM.

Procédure

1. Obtenir le code PIN pour les certificats de l'autorité de certification du sous-système :

```
# grep 'internal=' /var/lib/pki/pki-tomcat/conf/password.conf
```

2. Ajouter le suivi aux certificats CA du sous-système, en remplaçant **[internal PIN]** dans les commandes ci-dessous par le PIN obtenu à l'étape précédente :

```
# getcert start-tracking -d /etc/pki/pki-tomcat/alias -n "caSigningCert cert-pki-ca" -c
'dogtag-ipa-ca-renew-agent' -P [internal PIN] -B
/usr/libexec/ipa/certmonger/stop_pkicad -C '/usr/libexec/ipa/certmonger/renew_ca_cert
"caSigningCert cert-pki-ca"' -T caCACert

# getcert start-tracking -d /etc/pki/pki-tomcat/alias -n "auditSigningCert cert-pki-ca" -c
'dogtag-ipa-ca-renew-agent' -P [internal PIN] -B
/usr/libexec/ipa/certmonger/stop_pkicad -C '/usr/libexec/ipa/certmonger/renew_ca_cert
"auditSigningCert cert-pki-ca"' -T caSignedLogCert

# getcert start-tracking -d /etc/pki/pki-tomcat/alias -n "ocspSigningCert cert-pki-ca" -c
'dogtag-ipa-ca-renew-agent' -P [internal PIN] -B
/usr/libexec/ipa/certmonger/stop_pkicad -C '/usr/libexec/ipa/certmonger/renew_ca_cert
"ocspSigningCert cert-pki-ca"' -T caOCSPCert

# getcert start-tracking -d /etc/pki/pki-tomcat/alias -n "subsystemCert cert-pki-ca" -c
'dogtag-ipa-ca-renew-agent' -P [internal PIN] -B
```

```
/usr/libexec/ipa/certmonger/stop_pkicad -C '/usr/libexec/ipa/certmonger/renew_ca_cert
"subsystemCert cert-pki-ca"' -T caSubsystemCert

# getcert start-tracking -d /etc/pki/pki-tomcat/alias -n "Server-Cert cert-pki-ca" -c
'dogtag-ipa-ca-renew-agent' -P [internal PIN] -B
/usr/libexec/ipa/certmonger/stop_pkicad -C '/usr/libexec/ipa/certmonger/renew_ca_cert
"Server-Cert cert-pki-ca"' -T caServerCert
```

3. Ajouter le suivi des certificats IdM restants, les certificats **HTTP**, **LDAP**, **IPA renewal agent** et **PKINIT**:

```
# getcert start-tracking -f /var/lib/ipa/certs/httpd.crt -k /var/lib/ipa/private/httpd.key -p
/var/lib/ipa/passwds/idm.example.com-443-RSA -c IPA -C
/usr/libexec/ipa/certmonger/restart_httpd -T calPAserviceCert

# getcert start-tracking -d /etc/dirsrv/slapd-IDM-EXAMPLE-COM -n "Server-Cert" -c IPA
-p /etc/dirsrv/slapd-IDM-EXAMPLE-COM/pwdfile.txt -C
'/usr/libexec/ipa/certmonger/restart_dirsrv "IDM-EXAMPLE-COM"' -T calPAserviceCert

# getcert start-tracking -f /var/lib/ipa/ra-agent.pem -k /var/lib/ipa/ra-agent.key -c
dogtag-ipa-ca-renew-agent -B /usr/libexec/ipa/certmonger/renew_ra_cert_pre -C
/usr/libexec/ipa/certmonger/renew_ra_cert -T caSubsystemCert

# getcert start-tracking -f /var/kerberos/krb5kdc/kdc.crt -k
/var/kerberos/krb5kdc/kdc.key -c dogtag-ipa-ca-renew-agent -B
/usr/libexec/ipa/certmonger/renew_ra_cert_pre -C
/usr/libexec/ipa/certmonger/renew_kdc_cert -T KDCs_PKINIT_Certs
```

4. Redémarrer **certmonger**:

```
# systemctl restart certmonger
```

5. Attendez une minute après le démarrage de **certmonger**, puis vérifiez l'état des nouveaux certificats :

```
# getcert list
```

Ressources supplémentaires

- Si tous les certificats de votre système IdM ont expiré, consultez cette solution KCS (Knowledge Centered Support) pour renouveler manuellement les certificats du système IdM sur le serveur de l'autorité de certification IdM qui est également le serveur de renouvellement de l'autorité de certification et le serveur d'édition des CRL. Suivez ensuite la procédure décrite dans cette solution KCS pour renouveler manuellement les certificats du système IdM sur tous les autres serveurs d'autorité de certification de la topologie.

17.8. UTILISATION DE SCEP AVEC CERTMONGER

Le Simple Certificate Enrollment Protocol (SCEP) est un protocole de gestion des certificats que vous pouvez utiliser sur différents appareils et systèmes d'exploitation. Si vous utilisez un serveur SCEP en tant qu'autorité de certification (CA) externe dans votre environnement, vous pouvez utiliser **certmonger** pour obtenir un certificat pour un client Identity Management (IdM).

17.8.1. Vue d'ensemble de SCEP

Le Simple Certificate Enrollment Protocol (SCEP) est un protocole de gestion des certificats que vous pouvez utiliser sur différents appareils et systèmes d'exploitation. Vous pouvez utiliser un serveur SCEP comme autorité de certification (CA) externe.

Vous pouvez configurer un client de gestion d'identité (IdM) pour qu'il demande et récupère un certificat via HTTP directement auprès du service SCEP de l'autorité de certification. Ce processus est sécurisé par un secret partagé qui n'est généralement valable que pour une durée limitée.

Du côté client, SCEP exige que vous fournissiez les composants suivants :

- URL SCEP : l'URL de l'interface SCEP de l'AC.
- Secret partagé SCEP : un code PIN **challengePassword** partagé entre l'autorité de certification et le client SCEP, utilisé pour obtenir le certificat.

Le client récupère ensuite la chaîne de certificats de l'autorité de certification via SCEP et envoie une demande de signature de certificat à l'autorité de certification.

Lors de la configuration de SCEP avec **certmonger**, vous créez un nouveau profil de configuration de l'autorité de certification qui spécifie les paramètres du certificat émis.

17.8.2. Demande d'un certificat signé par l'AC IdM via SCEP

L'exemple suivant ajoute une configuration d'autorité de certification **SCEP_example** SCEP à **certmonger** et demande un nouveau certificat au client IdM **client.idm.example.com**. **certmonger** prend en charge à la fois le format de base de données de certificats NSS et les formats basés sur des fichiers (PEM), tels que OpenSSL.

Conditions préalables

- Vous connaissez l'URL de SCEP.
- Vous disposez du secret partagé **challengePassword** PIN.

Procédure

1. Ajouter la configuration de l'autorité de certification à **certmonger**:

```
[root@client.idm.example.com ~]# getcert add-scep-ca -c SCEP_example -u SCEP_URL
```

   - **-c**: Pseudonyme obligatoire pour la configuration de l'autorité de certification. La même valeur peut être utilisée ultérieurement avec d'autres commandes **getcert**.
   - **-u**: URL de l'interface SCEP du serveur.

IMPORTANT

Lorsque vous utilisez une URL HTTPS, vous devez également spécifier l'emplacement de la copie formatée PEM du certificat CA du serveur SCEP à l'aide de l'option **-R**.

2. Vérifiez que la configuration de l'autorité de certification a été ajoutée avec succès :

```
[root@client.idm.example.com ~]# getcert list-cas -c SCEP_example
CA 'SCEP_example':
        is-default: no
        ca-type: EXTERNAL
        helper-location: /usr/libexec/certmonger/scep-submit -u
http://SCEP_server_enrollment_interface_URL
        SCEP CA certificate thumbprint (MD5): A67C2D4B 771AC186 FCCA654A 5E55AAF7
        SCEP CA certificate thumbprint (SHA1): FBFF096C 6455E8E9 BD55F4A5 5787C43F
1F512279
```

Si la configuration a été ajoutée avec succès, certmonger récupère la chaîne d'autorité de certification de l'autorité de certification distante. La chaîne d'autorité de certification apparaît alors sous forme d'empreintes dans la sortie de la commande. Lors de l'accès au serveur via HTTP non chiffré, comparez manuellement les empreintes avec celles affichées sur le serveur SCEP afin d'éviter une attaque de type "man-in-the-middle".

3. Demander un certificat à l'autorité de certification :

   - Si vous utilisez NSS :

     ```
     [root@client.idm.example.com ~]# getcert request -I Example_Task -c SCEP_example -
     d /etc/pki/nssdb -n ExampleCert -N cn="client.idm.example.com" -L one-time_PIN -D
     client.idm.example.com
     ```

     Les options permettent de spécifier les paramètres suivants de la demande de certificat :

     - **-I**: *(Optional)* Nom de la tâche : l'ID de suivi de la demande. La même valeur peut être utilisée ultérieurement avec la commande **getcert list**.
     - **-c**: Configuration de l'autorité de certification à laquelle soumettre la demande.
     - **-d**: Répertoire avec la base de données NSS pour stocker le certificat et la clé.
     - **-n**: Pseudonyme du certificat, utilisé dans la base de données du SSN.
     - **-N**: Nom du sujet dans le CSR.
     - **-L**: Code PIN **challengePassword** à usage unique et limité dans le temps, délivré par l'AC.
     - **-D**: Nom alternatif du sujet pour le certificat, généralement le même que le nom d'hôte.

   - Si vous utilisez OpenSSL :

     ```
     [root@client.idm.example.com ~]# getcert request -I Example_Task -c SCEP_example -f
     /etc/pki/tls/certs/server.crt -k /etc/pki/tls/private/private.key -N
     cn="client.idm.example.com" -L one-time_PIN -D client.idm.example.com
     ```

     Les options permettent de spécifier les paramètres suivants de la demande de certificat :

     - **-I**: *(Optional)* Nom de la tâche : l'ID de suivi de la demande. La même valeur peut être utilisée ultérieurement avec la commande **getcert list**.
     - **-c**: Configuration de l'autorité de certification à laquelle soumettre la demande.
     - **-f**: Chemin d'accès au certificat.

- **-k**: Chemin d'accès à la clé.
- **-N**: Nom du sujet dans le CSR.
- **-L**: Code PIN **challengePassword** à usage unique et limité dans le temps, délivré par l'AC.
- **-D**: Nom alternatif du sujet pour le certificat, généralement le même que le nom d'hôte.

Vérification

1. Vérifiez qu'un certificat a été émis et correctement stocké dans la base de données locale :

- Si vous avez utilisé NSS, entrez :

```
[root@client.idm.example.com ~]# getcert list -I Example_Task
 Request ID 'Example_Task':
        status: MONITORING
        stuck: no
        key pair storage:
type=NSSDB,location='/etc/pki/nssdb',nickname='ExampleCert',token='NSS Certificate
DB'
        certificate:
type=NSSDB,location='/etc/pki/nssdb',nickname='ExampleCert',token='NSS Certificate
DB'
        signing request thumbprint (MD5): 503A8EDD DE2BE17E 5BAA3A57 D68C9C1B
        signing request thumbprint (SHA1): B411ECE4 D45B883A 75A6F14D 7E3037F1
D53625F4
        CA: IPA
        issuer: CN=Certificate Authority,O=EXAMPLE.COM
        subject: CN=client.idm.example.com,O=EXAMPLE.COM
        expires: 2018-05-06 10:28:06 UTC
        key usage: digitalSignature,keyEncipherment
        eku: iso.org.dod.internet.security.mechanisms.8.2.2
        certificate template/profile: IPSECIntermediateOffline
        pre-save command:
        post-save command:
        track: yes
 auto-renew: yes
```

- Si vous avez utilisé OpenSSL, entrez :

```
[root@client.idm.example.com ~]# getcert list -I Example_Task
Request ID 'Example_Task':
        status: MONITORING
        stuck: no
        key pair storage: type=FILE,location='/etc/pki/tls/private/private.key'
        certificate: type=FILE,location='/etc/pki/tls/certs/server.crt'
        CA: IPA
        issuer: CN=Certificate Authority,O=EXAMPLE.COM
        subject: CN=client.idm.example.com,O=EXAMPLE.COM
        expires: 2018-05-06 10:28:06 UTC
        eku: id-kp-serverAuth,id-kp-clientAuth
        pre-save command:
```

```
post-save command:
track: yes
auto-renew: yes
```

L'état **MONITORING** signifie que le certificat délivré a été récupéré avec succès. La page de manuel **getcert-list(1)** répertorie les autres états possibles et leur signification.

Ressources supplémentaires

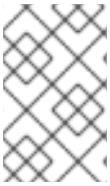
- Pour plus d'options lors de la demande d'un certificat, voir la page de manuel **getcert-request(1)**.

17.8.3. Renouvellement automatique des certificats AD SCEP avec certmonger

Lorsque **certmonger** envoie une demande de renouvellement de certificat SCEP, cette demande est signée à l'aide de la clé privée du certificat existant. Cependant, les demandes de renouvellement envoyées par **certmonger** incluent par défaut le PIN **challengePassword** qui a été utilisé pour obtenir les certificats à l'origine.

Un serveur Active Directory (AD) Network Device Enrollment Service (NDES) qui fonctionne comme serveur SCEP rejette automatiquement toutes les demandes de renouvellement qui contiennent le PIN original **challengePassword**. Par conséquent, le renouvellement échoue.

Pour que le renouvellement avec AD fonctionne, vous devez configurer **certmonger** pour qu'il envoie les demandes de renouvellement signées sans le code PIN de **challengePassword**. Vous devez également configurer le serveur AD de manière à ce qu'il ne compare pas le nom du sujet lors du renouvellement.



NOTE

Il se peut que des serveurs SCEP autres qu'AD refusent également les demandes contenant l'adresse **challengePassword**. Dans ce cas, il se peut que vous deviez également modifier la configuration de **certmonger** de cette manière.

Conditions préalables

- Le serveur RHEL doit être équipé de RHEL 8.6 ou d'une version plus récente.

Procédure

1. Ouvrez **regedit** sur le serveur AD.
2. Dans la sous-clé **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\MSCEP**, ajoutez une nouvelle entrée REG_DWORD de 32 bits **DisableRenewalSubjectNameMatch** et définissez sa valeur à **1**.
3. Sur le serveur où **certmonger** est exécuté, ouvrez le fichier **/etc/certmonger/certmonger.conf** et ajoutez la section suivante :

```
[scep]
challenge_password_otp = yes
```

4. Redémarrer certmonger :

```
# systemctl restart certmonger
```

CHAPITRE 18. DÉPLOYER ET GÉRER LE SERVICE ACME DANS IDM

L'environnement de gestion automatisée des certificats (ACME) est un protocole de validation automatisée des identifiants et d'émission de certificats. Son objectif est d'améliorer la sécurité en réduisant la durée de vie des certificats et en évitant les processus manuels dans la gestion du cycle de vie des certificats.

Grâce à RHEL Identity Management (IdM), l'administrateur peut facilement déployer et gérer le service ACME à l'échelle de la topologie à partir d'un seul système.

18.1. LE SERVICE ACME DANS L'IDM

ACME utilise un mécanisme d'authentification par défi et réponse pour prouver qu'un client a le contrôle d'un identifiant. Dans ACME, un identifiant est une preuve de propriété utilisée pour obtenir un certificat en résolvant un défi. Dans le cadre de la gestion de l'identité (IdM), ACME prend actuellement en charge les défis suivants :

- **dns-01** où le client crée des enregistrements DNS pour prouver qu'il a le contrôle de l'identifiant
- **http-01** où le client fournit une ressource HTTP pour prouver qu'il a le contrôle de l'identifiant

Dans IdM, le service ACME utilise le répondeur ACME de l'ICP. Le sous-système ACME est automatiquement déployé sur chaque serveur d'autorité de certification dans le déploiement IdM, mais il ne répondra pas aux demandes tant que l'administrateur ne l'aura pas activé. Les serveurs sont découverts à l'aide du nom **ipa-ca.DOMAIN**. Tous les serveurs CA IdM sont enregistrés sous ce nom DNS, de sorte que les demandes leur sont adressées de manière équilibrée par round-robin.

L'ACME est également déployé, mais désactivé, lorsque l'administrateur met à niveau un serveur à l'aide de la commande **ipa-server-upgrade**.

ACME fonctionne comme un service distinct au sein d'Apache Tomcat. Les fichiers de configuration d'ACME sont stockés à l'adresse **/etc/pki/pki-tomcat/acme** et l'ICP enregistre les informations relatives à ACME à l'adresse **/var/log/pki/pki-tomcat/acme/**.

IdM utilise le profil **acmeIPAServerCert** lors de l'émission de certificats ACME. La période de validité des certificats émis est de 90 jours. Pour cette raison, il est fortement recommandé de configurer ACME pour qu'il supprime automatiquement les certificats expirés afin qu'ils ne s'accumulent pas dans l'autorité de certification, ce qui pourrait avoir un effet négatif sur les performances.

Il existe différents clients ACME. Pour une utilisation avec RHEL, le client choisi doit prendre en charge les défis **dns-01** et **http-01**. Actuellement, les clients suivants ont été testés et sont connus pour fonctionner avec ACME dans RHEL :

- **certbot** avec les défis **http-01** et **dns-01**
- **mod_md**qui ne prend en charge que le défi **http-01**

18.2. ACTIVATION DU SERVICE ACME DANS IDM

Par défaut, le service ACME est déployé, mais désactivé. L'activation du service ACME permet de l'activer sur tous les serveurs d'AC IdM dans l'ensemble du déploiement IdM. Cette opération s'effectue par réplication.

Dans cet exemple, vous activez ACME et vous le configurez pour qu'il supprime automatiquement les certificats expirés le premier jour de chaque mois à minuit.

Conditions préalables

- Les serveurs du déploiement IdM fonctionnent sous RHEL 9.2 ou une version plus récente, avec l'option Random Certificate Serial Numbers (RSNv3) activée.
- Vous disposez des droits de root sur le serveur IdM sur lequel vous exécutez la procédure.

Procédure

1. Activer l'ACME dans l'ensemble du déploiement IdM :

```
# ipa-acme-manage enable
The ipa-acme-manage command was successful
```

2. Configurer ACME pour qu'il supprime automatiquement les certificats expirés de l'autorité de certification :

```
# ipa-acme-manage pruning --enable --cron "0 0 1 * *"
```

NOTE

Les certificats expirés sont supprimés après leur période de conservation. Par défaut, cette période est de 30 jours après l'expiration.

Verification steps

- Pour vérifier si le service ACME est installé et activé, utilisez la commande **ipa-acme-manage status**:

```
# ipa-acme-manage status
ACME is enabled
The ipa-acme-manage command was successful
```

18.3. DÉSACTIVATION DU SERVICE ACME DANS IDM

La désactivation du service ACME le désactive dans l'ensemble du déploiement IdM. Cette désactivation est gérée par la réplication.

Conditions préalables

- Les serveurs du déploiement IdM fonctionnent sous RHEL 9.2 ou une version plus récente, avec l'option Random Certificate Serial Numbers (RSNv3) activée.
- Vous disposez des droits de root sur le serveur IdM sur lequel vous exécutez la procédure.

Procédure

1. Désactiver ACME dans l'ensemble du déploiement IdM :

```
# ipa-acme-manage disable
The ipa-acme-manage command was successful
```

2. (Facultatif) Désactiver la suppression automatique des certificats expirés :

```
ipa-acme-manage pruning --disable
```

Verification steps

- Pour vérifier si le service ACME est installé, mais désactivé, utilisez la commande **ipa-acme-manage status**:

```
# ipa-acme-manage status
ACME is disabled
The ipa-acme-manage command was successful
```

CHAPITRE 19. DEMANDE DE CERTIFICATS À L'AIDE DES RÔLES SYSTÈME RHEL

Vous pouvez utiliser le rôle de système **certificate** pour émettre et gérer des certificats.

Ce chapitre couvre les sujets suivants :

- Le rôle du système **certificate**
- Demande d'un nouveau certificat auto-signé à l'aide du rôle de système **certificate**
- Demande d'un nouveau certificat à l'autorité de certification IdM à l'aide du rôle de système **certificate**

19.1. LE RÔLE DU SYSTÈME CERTIFICATE

En utilisant le rôle de système **certificate**, vous pouvez gérer l'émission et le renouvellement des certificats TLS et SSL à l'aide d'Ansible Core.

Ce rôle utilise **certmonger** comme fournisseur de certificats et prend actuellement en charge l'émission et le renouvellement de certificats auto-signés et l'utilisation de l'autorité de certification (AC) intégrée à IdM.

Vous pouvez utiliser les variables suivantes dans votre playbook Ansible avec le rôle de système **certificate**:

certificate_wait

pour indiquer si la tâche doit attendre que le certificat soit délivré.

certificate_requests

pour représenter chaque certificat à délivrer et ses paramètres.

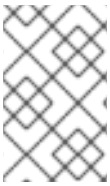
Ressources supplémentaires

- Voir le fichier **/usr/share/ansible/roles/rhel-system-roles.certificate/README.md**.
- Préparation d'un nœud de contrôle et de nœuds gérés à l'utilisation des rôles système RHEL

19.2. DEMANDE D'UN NOUVEAU CERTIFICAT AUTO-SIGNÉ À L'AIDE DU RÔLE DE SYSTÈME CERTIFICATE

Avec le rôle de système **certificate**, vous pouvez utiliser Ansible Core pour émettre des certificats auto-signés.

Ce processus utilise le fournisseur **certmonger** et demande le certificat par le biais de la commande **getcert**.



NOTE

Par défaut, **certmonger** essaie automatiquement de renouveler le certificat avant qu'il n'expire. Vous pouvez désactiver cette fonction en définissant le paramètre **auto_renew** dans le manuel de jeu Ansible sur **no**.

Conditions préalables

- Le paquetage Ansible Core est installé sur la machine de contrôle.
- Le paquetage **rhel-system-roles** est installé sur le système à partir duquel vous souhaitez exécuter le playbook.

Procédure

1. *Optional:* Créer un fichier d'inventaire, par exemple **inventory.file**:

```
$ *touch inventory.file* (toucher le fichier d'inventaire)
```

2. Ouvrez votre fichier d'inventaire et définissez les hôtes sur lesquels vous souhaitez demander le certificat, par exemple :

```
[webserver]
server.idm.example.com
```

3. Créez un fichier playbook, par exemple **request-certificate.yml**:
   - Définissez **hosts** pour inclure les hôtes sur lesquels vous souhaitez demander le certificat, par exemple **webserver**.
   - Définissez la variable **certificate_requests** de manière à ce qu'elle contienne les éléments suivants :
     - Attribuez au paramètre **name** le nom souhaité pour le certificat, par exemple **mycert**.
     - Le paramètre **dns** correspond au domaine à inclure dans le certificat, par exemple ***.example.com**.
     - Réglez le paramètre **ca** sur **self-sign**.
   - Définir le rôle de **rhel-system-roles.certificate** sous **roles**.
     Il s'agit du fichier du playbook pour cet exemple :

```
---
- hosts: webserver

  vars:
    certificate_requests:
      - name: mycert
        dns: "*.example.com"
        ca: self-sign

  roles:
    - rhel-system-roles.certificate
```

4. Enregistrer le fichier.
5. Exécutez le manuel de jeu :

```
*ansible-playbook -i inventory.file request-certificate.yml* $ *ansible-playbook -i inventory.file request-certificate.yml* $ *ansible-playbook -i inventory.file
```

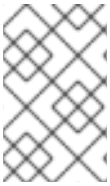
Ressources supplémentaires

- Voir le fichier **/usr/share/ansible/roles/rhel-system-roles.certificate/README.md**.
- Voir la page de manuel **ansible-playbook(1)**.

19.3. DEMANDE D'UN NOUVEAU CERTIFICAT À L'AUTORITÉ DE CERTIFICATION IDM À L'AIDE DU RÔLE DE SYSTÈME CERTIFICATE

Avec le rôle de système **certificate**, vous pouvez utiliser **anible-core** pour émettre des certificats tout en utilisant un serveur IdM avec une autorité de certification (CA) intégrée. Vous pouvez donc gérer efficacement et de manière cohérente la chaîne de confiance des certificats pour plusieurs systèmes lorsque vous utilisez IdM comme autorité de certification.

Ce processus utilise le fournisseur **certmonger** et demande le certificat par le biais de la commande **getcert**.



NOTE

Par défaut, **certmonger** essaie automatiquement de renouveler le certificat avant qu'il n'expire. Vous pouvez désactiver cette fonction en définissant le paramètre **auto_renew** dans le manuel de jeu Ansible sur **no**.

Conditions préalables

- Le paquetage Ansible Core est installé sur la machine de contrôle.
- Le paquetage **rhel-system-roles** est installé sur le système à partir duquel vous souhaitez exécuter le playbook.

Procédure

1. *Optional:* Créer un fichier d'inventaire, par exemple **inventory.file**:

```
$ *touch inventory.file* (toucher le fichier d'inventaire)
```

2. Ouvrez votre fichier d'inventaire et définissez les hôtes sur lesquels vous souhaitez demander le certificat, par exemple :

```
[webserver]
server.idm.example.com
```

3. Créez un fichier playbook, par exemple **request-certificate.yml**:
   - Définissez **hosts** pour inclure les hôtes sur lesquels vous souhaitez demander le certificat, par exemple **webserver**.
   - Définissez la variable **certificate_requests** de manière à ce qu'elle contienne les éléments suivants :
     - Attribuez au paramètre **name** le nom souhaité pour le certificat, par exemple **mycert**.
     - Le paramètre **dns** correspond au domaine à inclure dans le certificat, par exemple **www.example.com**.

- Le paramètre **principal** indique le principal Kerberos, par exemple **HTTP/www.example.com@EXAMPLE.COM**.
- Réglez le paramètre **ca** sur **ipa**.

- Définir le rôle de **rhel-system-roles.certificate** sous **roles**.
  Il s'agit du fichier du playbook pour cet exemple :

```
---
- hosts: webserver
  vars:
    certificate_requests:
      - name: mycert
        dns: www.example.com
        principal: HTTP/www.example.com@EXAMPLE.COM
        ca: ipa

  roles:
    - rhel-system-roles.certificate
```

4. Enregistrer le fichier.
5. Exécutez le manuel de jeu :

```
*ansible-playbook -i inventory.file request-certificate.yml* $ *ansible-playbook -i inventory.file request-certificate.yml* $ *ansible-playbook -i inventory.file
```

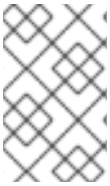
Ressources supplémentaires

- Voir le fichier **/usr/share/ansible/roles/rhel-system-roles.certificate/README.md**.
- Voir la page de manuel **ansible-playbook(1)**.

19.4. SPÉCIFICATION DES COMMANDES À EXÉCUTER AVANT OU APRÈS L'ÉMISSION D'UN CERTIFICAT À L'AIDE DU RÔLE DE SYSTÈME CERTIFICATE

Avec le rôle **certificate**, vous pouvez utiliser Ansible Core pour exécuter une commande avant et après l'émission ou le renouvellement d'un certificat.

Dans l'exemple suivant, l'administrateur veille à arrêter le service **httpd** avant l'émission ou le renouvellement d'un certificat auto-signé pour **www.example.com**, et à le redémarrer ensuite.



NOTE

Par défaut, **certmonger** essaie automatiquement de renouveler le certificat avant qu'il n'expire. Vous pouvez désactiver cette fonction en définissant le paramètre **auto_renew** dans le manuel de jeu Ansible sur **no**.

Conditions préalables

- Le paquetage Ansible Core est installé sur la machine de contrôle.

- Le paquetage **rhel-system-roles** est installé sur le système à partir duquel vous souhaitez exécuter le playbook.

Procédure

1. *Optional:* Créer un fichier d'inventaire, par exemple **inventory.file**:

```
$ *touch inventory.file* (toucher le fichier d'inventaire)
```

2. Ouvrez votre fichier d'inventaire et définissez les hôtes sur lesquels vous souhaitez demander le certificat, par exemple :

```
[webserver]
server.idm.example.com
```

3. Créez un fichier playbook, par exemple **request-certificate.yml**:

   - Définissez **hosts** pour inclure les hôtes sur lesquels vous souhaitez demander le certificat, par exemple **webserver**.
   - Définissez la variable **certificate_requests** de manière à ce qu'elle contienne les éléments suivants :
     - Attribuez au paramètre **name** le nom souhaité pour le certificat, par exemple **mycert**.
     - Le paramètre **dns** correspond au domaine à inclure dans le certificat, par exemple **www.example.com**.
     - Définissez le paramètre **ca** en fonction de l'autorité de certification que vous souhaitez utiliser pour émettre le certificat, par exemple **self-sign**.
     - Attribuez au paramètre **run_before** la valeur de la commande que vous souhaitez exécuter avant l'émission ou le renouvellement de ce certificat, par exemple **systemctl stop httpd.service**.
     - Attribuez au paramètre **run_after** la commande que vous souhaitez exécuter après l'émission ou le renouvellement de ce certificat, par exemple **systemctl start httpd.service**.
   - Définir le rôle de **rhel-system-roles.certificate** sous **roles**.
     Il s'agit du fichier du playbook pour cet exemple :

```
---
- hosts: webserver
  vars:
    certificate_requests:
      - name: mycert
        dns: www.example.com
        ca: self-sign
        run_before: systemctl stop httpd.service
        run_after: systemctl start httpd.service

  roles:
    - rhel-system-roles.certificate
```

4. Enregistrer le fichier.

5. Exécutez le manuel de jeu :

> *ansible-playbook -i inventory.file request-certificate.yml* $ *ansible-playbook -i inventory.file request-certificate.yml* $ *ansible-playbook -i inventory.file

Ressources supplémentaires

- Voir le fichier **/usr/share/ansible/roles/rhel-system-roles.certificate/README.md**.
- Voir la page de manuel **ansible-playbook(1)**.

CHAPITRE 20. RESTREINDRE UNE APPLICATION À NE FAIRE CONFIANCE QU'À UN SOUS-ENSEMBLE DE CERTIFICATS

Si votre installation de gestion des identités (IdM) est configurée avec l'autorité de certification (AC) intégrée Certificate System (CS), vous pouvez créer des sous-AAC légères. Toutes les sous-AAC que vous créez sont subordonnées à l'AC principale du système de certificats, l'AC **ipa**.

Dans ce contexte, *lightweight sub-CA* signifie *a sub-CA issuing certificates for a specific purpose* . Par exemple, une AC secondaire légère vous permet de configurer un service, tel qu'une passerelle de réseau privé virtuel (VPN) et un navigateur web, de manière à ce qu'il n'accepte que les certificats délivrés par *sub-CA A*. En configurant d'autres services pour qu'ils n'acceptent que les certificats émis par *sub-CA B*, vous les empêchez d'accepter les certificats émis par *sub-CA A*, l'autorité de certification primaire, c'est-à-dire l'autorité de certification **ipa**, et toute sous-AC intermédiaire entre les deux.

Si vous révoquez le certificat intermédiaire d'une sous-AC, tous les certificats émis par cette sous-AC sont automatiquement considérés comme invalides par les clients correctement configurés. Tous les autres certificats émis directement par l'autorité de certification racine, **ipa**, ou par une autre autorité de certification secondaire, restent valides.

Cette section utilise l'exemple du serveur web Apache pour illustrer comment limiter une application à un sous-ensemble de certificats. Complétez cette section pour restreindre le serveur web fonctionnant sur votre client IdM à l'utilisation d'un certificat émis par la sous-CA **webserver-ca** IdM, et pour exiger des utilisateurs qu'ils s'authentifient sur le serveur web à l'aide de certificats d'utilisateur émis par la sous-CA **webclient-ca** IdM.

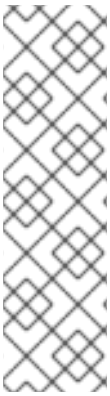
Les étapes à suivre sont les suivantes :

1. Créer une sous-CA IdM
2. Télécharger le certificat du sous-CA à partir de l'interface Web de l'IdM
3. Créez une liste de contrôle d'accès à l'autorité de certification en spécifiant la combinaison correcte d'utilisateurs, de services et d'autorités de certification, ainsi que le profil de certificat utilisé
4. Demander à la sous-CA IdM un certificat pour le service web fonctionnant sur un client IdM
5. Mise en place d'une instance unique du serveur HTTP Apache
6. Ajouter le cryptage TLS au serveur HTTP Apache
7. Définir les versions du protocole TLS prises en charge sur un serveur HTTP Apache
8. Définir les algorithmes de chiffrement pris en charge par le serveur HTTP Apache
9. Configurer l'authentification du certificat client TLS sur le serveur web
10. Demander un certificat pour l'utilisateur à la sous-CA IdM et l'exporter vers le client
11. Importer le certificat de l'utilisateur dans le navigateur et configurer le navigateur pour qu'il fasse confiance au certificat du sous-CA

20.1. GESTION DES SOUS-CA LÉGERS

Cette section décrit comment gérer les autorités de certification subordonnées légères (sub-CA). Toutes les autorités de certification subordonnées que vous créez sont subordonnées à l'autorité de

certification principale du système de certification, l'autorité de certification **ipa**. Vous pouvez également désactiver et supprimer des autorités de certification subordonnées.



NOTE

- Si vous supprimez un sous-CA, le contrôle de révocation pour ce sous-CA ne fonctionnera plus. Ne supprimez un sous-CA que lorsqu'il n'y a plus de certificats émis par ce sous-CA et dont l'heure d'expiration **notAfter** se situe dans le futur.
- Vous ne devez désactiver les sous-CA que s'il reste des certificats non expirés émis par ce sous-CA. Si tous les certificats émis par un sous-CA ont expiré, vous pouvez supprimer ce sous-CA.
- Vous ne pouvez pas désactiver ou supprimer l'autorité de certification IdM.

Pour plus de détails sur la gestion des sous-CA, voir :

- Création d'un sous-CA à partir de l'interface Web IdM
- Suppression d'un sous-CA à partir de l'interface Web IdM
- Création d'un sous-CA à partir de la CLI IdM
- Désactivation d'un sous-CA à partir de la CLI IdM
- Suppression d'un sous-CA à partir de la CLI IdM

20.1.1. Création d'un sous-CA à partir de l'interface Web IdM

Cette procédure décrit comment utiliser l'interface Web IdM pour créer de nouveaux sous-CA nommés **webserver-ca** et **webclient-ca**.

Conditions préalables

- Assurez-vous d'avoir obtenu les informations d'identification de l'administrateur.

Procédure

1. Dans le menu **Authentication**, cliquez sur **Certificates**.
2. Sélectionnez **Certificate Authorities** et cliquez sur **Add**.
3. Saisissez le nom de la sous-CA **webserver-ca**. Entrez le DN du sujet, par exemple **CN=WEBSERVER,O=IDM.EXAMPLE.COM**, dans le champ DN du sujet. Notez que le Subject DN doit être unique dans l'infrastructure de l'AC IdM.
4. Saisissez le nom du sous-CA **webclient-ca**. Saisissez le Subject DN **CN=WEBCLIENT,O=IDM.EXAMPLE.COM** dans le champ Subject DN.
5. Dans l'interface de ligne de commande, exécutez la commande **ipa-certupdate** pour créer une demande de suivi **certmonger** pour les certificats des sous-CA **webserver-ca** et **webclient-ca**:

```
[root@ipaserver ~]# ipa-certupdate
```

IMPORTANT

Si vous oubliez d'exécuter la commande **ipa-certupdate** après avoir créé un sous-CA, si le certificat du sous-CA expire, les certificats d'entité finale émis par le sous-CA sont considérés comme invalides, même si le certificat de l'entité finale n'a pas expiré.

Vérification

- Vérifier que le certificat de signature du nouveau sous-CA a été ajouté à la base de données IdM :

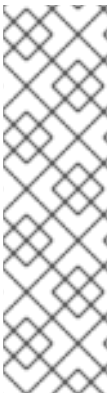
```
[root@ipaserver ~]# certutil -d /etc/pki/pki-tomcat/alias/ -L

Certificate Nickname                    Trust Attributes
                                        SSL,S/MIME,JAR/XPI

caSigningCert cert-pki-ca               CTu,Cu,Cu
Server-Cert cert-pki-ca                 u,u,u
auditSigningCert cert-pki-ca            u,u,Pu
caSigningCert cert-pki-ca ba83f324-5e50-4114-b109-acca05d6f1dc u,u,u
ocspSigningCert cert-pki-ca             u,u,u
subsystemCert cert-pki-ca               u,u,u
```

NOTE

Le nouveau certificat de sous-CA est automatiquement transféré à toutes les répliques sur lesquelles une instance de système de certificats est installée.

20.1.2. Suppression d'un sous-CA à partir de l'interface Web IdM

Cette procédure décrit comment supprimer des sous-CA légers dans l'interface Web IdM.



NOTE

- Si vous supprimez un sous-CA, le contrôle de révocation pour ce sous-CA ne fonctionnera plus. Ne supprimez un sous-CA que lorsqu'il n'y a plus de certificats émis par ce sous-CA et dont l'heure d'expiration **notAfter** se situe dans le futur.
- Vous ne devez désactiver les sous-CA que s'il reste des certificats non expirés émis par ce sous-CA. Si tous les certificats émis par un sous-CA ont expiré, vous pouvez supprimer ce sous-CA.
- Vous ne pouvez pas désactiver ou supprimer l'autorité de certification IdM.

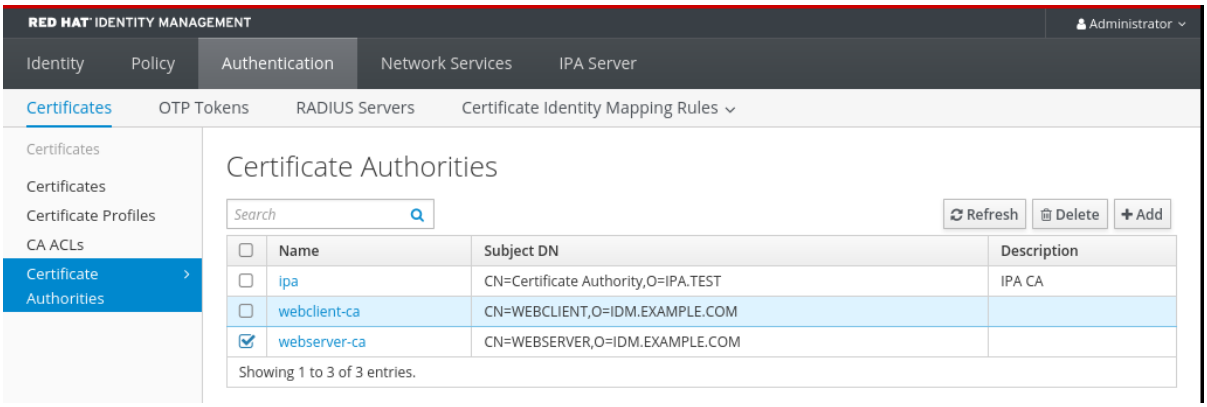
Conditions préalables

- Assurez-vous d'avoir obtenu les informations d'identification de l'administrateur.
- Vous avez désactivé le sous-CA dans le CLI IdM. Voir Désactivation d'un sous-CA à partir de la CLI IdM

Procédure

1. Dans l'IdM WebUI, ouvrez l'onglet **Authentication** et sélectionnez le sous-onglet **Certificates**.
2. Sélectionnez **Certificate Authorities**.
3. Sélectionnez le sous-CA à supprimer et cliquez sur **Delete**.

Figure 20.1. Suppression d'un sous-CA dans l'interface Web IdM



4. Cliquez sur **Delete** pour confirmer.

Le sous-CA est retiré de la liste des **Certificate Authorities**.

20.1.3. Création d'un sous-CA à partir de la CLI IdM

Cette procédure décrit comment utiliser le CLI IdM pour créer de nouveaux sous-CA nommés **webserver-ca** et **webclient-ca**.

Conditions préalables

- Assurez-vous d'avoir obtenu les informations d'identification de l'administrateur.
- Assurez-vous que vous êtes connecté à un serveur IdM qui est un serveur CA.

Procédure

1. Entrez la commande **ipa ca-add** et indiquez le nom du sous-CA **webserver-ca** et son Subject Distinguished Name (DN) :

```
[root@ipaserver ~]# ipa ca-add webserver-ca --
subject="CN=WEBSERVER,O=IDM.EXAMPLE.COM"
-------------------
Created CA "webserver-ca"
-------------------
  Name: webserver-ca
  Authority ID: ba83f324-5e50-4114-b109-acca05d6f1dc
  Subject DN: CN=WEBSERVER,O=IDM.EXAMPLE.COM
  Issuer DN: CN=Certificate Authority,O=IDM.EXAMPLE.COM
```

Nom

Nom de l'AC.

ID de l'autorité

ID individuel créé automatiquement pour l'AC.

Sujet DN

Nom distinctif du sujet (DN). Le DN du sujet doit être unique dans l'infrastructure de l'AC IdM.

DN de l'émetteur

L'autorité de certification mère qui a délivré le certificat de l'autorité de certification secondaire. Toutes les sous-AAC sont créées en tant qu'enfants de l'AC racine IdM.

2. Créez le sous-CA **webclient-ca** pour délivrer des certificats aux clients web :

```
[root@ipaserver ~]# ipa ca-add webclient-ca --
subject="CN=WEBCLIENT,O=IDM.EXAMPLE.COM"
-------------------
Created CA "webclient-ca"
-------------------
  Name: webclient-ca
  Authority ID: 8a479f3a-0454-4a4d-8ade-fd3b5a54ab2e
  Subject DN: CN=WEBCLIENT,O=IDM.EXAMPLE.COM
  Issuer DN: CN=Certificate Authority,O=IDM.EXAMPLE.COM
```

3. Exécutez la commande **ipa-certupdate** pour créer une demande de suivi **certmonger** pour les certificats des sous-CA **webserver-ca** et **webclient-ca**:

```
[root@ipaserver ~]# ipa-certupdate
```

IMPORTANT

Si vous oubliez d'exécuter la commande **ipa-certupdate** après avoir créé un sous-CA et que le certificat du sous-CA expire, les certificats d'entité finale émis par ce sous-CA sont considérés comme non valides, même si le certificat d'entité finale n'a pas expiré.

Verification steps

- Vérifier que le certificat de signature du nouveau sous-CA a été ajouté à la base de données IdM :

```
[root@ipaserver ~]# certutil -d /etc/pki/pki-tomcat/alias/ -L

Certificate Nickname                    Trust Attributes
                                        SSL,S/MIME,JAR/XPI

caSigningCert cert-pki-ca               CTu,Cu,Cu
Server-Cert cert-pki-ca                 u,u,u
auditSigningCert cert-pki-ca            u,u,Pu
caSigningCert cert-pki-ca ba83f324-5e50-4114-b109-acca05d6f1dc u,u,u
ocspSigningCert cert-pki-ca             u,u,u
subsystemCert cert-pki-ca               u,u,u
```

NOTE

Le nouveau certificat de sous-CA est automatiquement transféré à toutes les répliques sur lesquelles une instance de système de certificats est installée.

20.1.4. Désactivation d'un sous-CA à partir de la CLI IdM

Cette procédure décrit comment désactiver un sous-CA à partir de la CLI IdM. S'il existe encore des certificats non expirés qui ont été émis par un sous-CA, vous ne devez pas le supprimer mais vous pouvez le désactiver. Si vous supprimez le sous-CA, le contrôle de révocation pour ce sous-CA ne fonctionnera plus.

Conditions préalables

- Assurez-vous d'avoir obtenu les informations d'identification de l'administrateur.

Procédure

1. Exécutez la commande **ipa ca-find** pour déterminer le nom du sous-CA que vous supprimez :

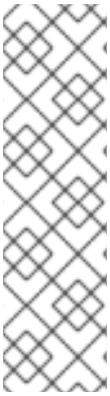
```
[root@ipaserver ~]# ipa ca-find
-------------
3 CAs matched
-------------
  Name: ipa
  Description: IPA CA
  Authority ID: 5195deaf-3b61-4aab-b608-317aff38497c
  Subject DN: CN=Certificate Authority,O=IPA.TEST
  Issuer DN: CN=Certificate Authority,O=IPA.TEST

  Name: webclient-ca
  Authority ID: 605a472c-9c6e-425e-b959-f1955209b092
  Subject DN: CN=WEBCLIENT,O=IDM.EXAMPLE.COM
  Issuer DN: CN=Certificate Authority,O=IPA.TEST

 Name: webserver-ca
  Authority ID: 02d537f9-c178-4433-98ea-53aa92126fc3
  Subject DN: CN=WEBSERVER,O=IDM.EXAMPLE.COM
  Issuer DN: CN=Certificate Authority,O=IPA.TEST
----------------------------
Number of entries returned 3
----------------------------
```

2. Exécutez la commande **ipa ca-disable** pour désactiver votre sous-CA, dans cet exemple, le **webserver-ca**:

```
ipa ca-disable webserver-ca
---------------------------
Disabled CA "webserver-ca"
---------------------------
```

20.1.5. Suppression d'un sous-CA à partir de la CLI IdM

Cette procédure décrit comment supprimer des sous-CA légers à partir de la CLI IdM.



NOTE

- Si vous supprimez un sous-CA, le contrôle de révocation pour ce sous-CA ne fonctionnera plus. Ne supprimez un sous-CA que lorsqu'il n'y a plus de certificats émis par ce sous-CA et dont l'heure d'expiration **notAfter** se situe dans le futur.
- Vous ne devez désactiver les sous-CA que s'il reste des certificats non expirés émis par ce sous-CA. Si tous les certificats émis par un sous-CA ont expiré, vous pouvez supprimer ce sous-CA.
- Vous ne pouvez pas désactiver ou supprimer l'autorité de certification IdM.

Conditions préalables

- Assurez-vous d'avoir obtenu les informations d'identification de l'administrateur.

Procédure

1. Pour afficher la liste des sous-CA et des CA, exécutez la commande **ipa ca-find**:

```
# ipa ca-find
----------
3 CAs matched
----------
  Name: ipa
  Description: IPA CA
  Authority ID: 5195deaf-3b61-4aab-b608-317aff38497c
  Subject DN: CN=Certificate Authority,O=IPA.TEST
  Issuer DN: CN=Certificate Authority,O=IPA.TEST

  Name: webclient-ca
  Authority ID: 605a472c-9c6e-425e-b959-f1955209b092
  Subject DN: CN=WEBCLIENT,O=IDM.EXAMPLE.COM
  Issuer DN: CN=Certificate Authority,O=IPA.TEST

 Name: webserver-ca
  Authority ID: 02d537f9-c178-4433-98ea-53aa92126fc3
  Subject DN: CN=WEBSERVER,O=IDM.EXAMPLE.COM
  Issuer DN: CN=Certificate Authority,O=IPA.TEST
----------------------------
Number of entries returned 3
----------------------------
```

2. Exécutez la commande **ipa ca-disable** pour désactiver votre sous-CA, dans cet exemple, le **webserver-ca**:

```
# ipa ca-disable webserver-ca
-------------------------
Disabled CA "webserver-ca"
-------------------------
```

3. Supprimer le sous-CA, dans cet exemple, le **webserver-ca**:

```
# ipa ca-del webserver-ca
-------------------------
```

```
Deleted CA "webserver-ca"
-------------------------
```

Vérification

- Exécutez **ipa ca-find** pour afficher la liste des CA et des sous-CA. Le site **webserver-ca** ne figure plus dans la liste.

```
# ipa ca-find
-------------
2 CAs matched
-------------
  Name: ipa
  Description: IPA CA
  Authority ID: 5195deaf-3b61-4aab-b608-317aff38497c
  Subject DN: CN=Certificate Authority,O=IPA.TEST
  Issuer DN: CN=Certificate Authority,O=IPA.TEST

  Name: webclient-ca
  Authority ID: 605a472c-9c6e-425e-b959-f1955209b092
  Subject DN: CN=WEBCLIENT,O=IDM.EXAMPLE.COM
  Issuer DN: CN=Certificate Authority,O=IPA.TEST
----------------------------
Number of entries returned 2
----------------------------
```

20.2. TÉLÉCHARGEMENT DU CERTIFICAT DU SOUS-CA À PARTIR DE L'INTERFACE WEB DE L'IDM

Conditions préalables

- Assurez-vous d'avoir obtenu les informations d'identification de l'administrateur IdM.

Procédure

1. Dans le menu **Authentication**, cliquez sur **Certificates** > **Certificates**.

   Figure 20.2. certificat de sous-CA dans la liste des certificats

2. Cliquez sur le numéro de série du certificat sous-CA pour ouvrir la page d'information sur le certificat.
3. Dans la page d'information sur le certificat, cliquez sur **Actions** > **Download**.
4. Dans le CLI, déplacez le certificat sub-CA dans le répertoire **/etc/pki/tls/private/**:

```
# mv path/to/the/downloaded/certificate /etc/pki/tls/private/sub-ca.crt
```

20.3. CRÉATION D'ACL POUR L'AUTHENTIFICATION DU SERVEUR WEB ET DU CLIENT

Les règles de la liste de contrôle d'accès de l'autorité de certification (CA ACL) définissent quels profils peuvent être utilisés pour délivrer des certificats à quels utilisateurs, services ou hôtes. En associant des profils, des mandants et des groupes, les CA ACL permettent à des mandants ou à des groupes de demander des certificats à l'aide de profils particuliers.

Par exemple, à l'aide d'ACL, l'administrateur peut restreindre l'utilisation d'un profil destiné aux employés travaillant dans un bureau situé à Londres aux seuls utilisateurs membres du groupe lié au bureau de Londres.

20.3.1. Affichage des listes de contrôle d'accès aux CA dans l'interface de gestion de l'IdM

Remplissez cette section pour afficher la liste des listes de contrôle d'accès des autorités de certification (CA ACL) disponibles dans votre déploiement IdM et les détails d'une CA ACL spécifique.

Procédure

1. Pour afficher toutes les ACL de votre environnement IdM, entrez la commande **ipa caacl-find**:

```
$ ipa caacl-find
-----------------
1 CA ACL matched
-----------------
  ACL name: hosts_services_caIPAserviceCert
  Enabled: TRUE
```

2. Pour afficher les détails d'une CA ACL, entrez la commande **ipa caacl-show** et indiquez le nom de la CA ACL. Par exemple, pour afficher les détails de la CA ACL **hosts_services_caIPAserviceCert**, entrez :

```
$ ipa caacl-show hosts_services_caIPAserviceCert
  ACL name: hosts_services_caIPAserviceCert
  Enabled: TRUE
  Host category: all
  Service category: all
  CAs: ipa
  Profiles: caIPAserviceCert
  Users: admin
```

20.3.2. Création d'une liste CA ACL pour les serveurs web s'authentifiant auprès des clients web à l'aide de certificats émis par webserver-ca

Cette section explique comment créer une CA ACL qui oblige l'administrateur système à utiliser la sous-CA **webserver-ca** et le profil **caIPAserviceCert** lorsqu'il demande un certificat pour le service **HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM**. Si l'utilisateur demande un certificat à partir d'un sous-CA différent ou d'un profil différent, la demande échoue. La seule exception est l'activation d'une autre CA ACL correspondante. Pour afficher les CA ACL disponibles, voir Affichage des CA ACL dans IdM CLI.

Conditions préalables

- Assurez-vous que le service **HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM** fait partie de l'IdM.
- Assurez-vous d'avoir obtenu les informations d'identification de l'administrateur de l'IdM.

Procédure

1. Créez une CAL à l'aide de la commande **ipa caacl** et indiquez son nom :

```
$ ipa caacl-add TLS_web_server_authentication
----------------------------------------------
Added CA ACL "TLS_web_server_authentication"
----------------------------------------------
  ACL name: TLS_web_server_authentication
  Enabled: TRUE
```

2. Modifiez la CA ACL en utilisant la commande **ipa caacl-mod** pour spécifier la description de la CA ACL :

```
$ ipa caacl-mod TLS_web_server_authentication --desc="CAACL for web servers
authenticating to web clients using certificates issued by webserver-ca"
-----------------------------------------------
Modified CA ACL "TLS_web_server_authentication"
-----------------------------------------------
  ACL name: TLS_web_server_authentication
  Description: CAACL for web servers authenticating to web clients using certificates issued
by webserver-ca
  Enabled: TRUE
```

3. Ajoutez la sous-CA **webserver-ca** à la CA ACL :

```
$ ipa caacl-add-ca TLS_web_server_authentication --ca=webserver-ca
  ACL name: TLS_web_server_authentication
  Description: CAACL for web servers authenticating to web clients using certificates issued
by webserver-ca
  Enabled: TRUE
  CAs: webserver-ca
-------------------------
Number of members added 1
-------------------------
```

4. Utilisez l'adresse **ipa caacl-add-service** pour spécifier le service dont le principal pourra demander un certificat :

```
$ ipa caacl-add-service TLS_web_server_authentication --
service=HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM
  ACL name: TLS_web_server_authentication
  Description: CAACL for web servers authenticating to web clients using certificates issued
by webserver-ca
  Enabled: TRUE
  CAs: webserver-ca
  Services: HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM
-------------------------
Number of members added 1
-------------------------
```

5. Utilisez la commande **ipa caacl-add-profile** pour spécifier le profil de certificat pour le certificat demandé :

```
$ ipa caacl-add-profile TLS_web_server_authentication --
certprofiles=caIPAserviceCert
  ACL name: TLS_web_server_authentication
  Description: CAACL for web servers authenticating to web clients using certificates issued
by webserver-ca
  Enabled: TRUE
  CAs: webserver-ca
  Profiles: caIPAserviceCert
  Services: HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM
-------------------------
Number of members added 1
-------------------------
```

Vous pouvez utiliser immédiatement l'ACL CA nouvellement créée. Elle est activée par défaut après sa création.

NOTE

L'objectif des listes de contrôle d'accès est de spécifier les combinaisons d'autorités de certification et de profils autorisées pour les demandes émanant de mandants ou de groupes particuliers. Les CA ACL n'affectent pas la validation ou la confiance dans les certificats. Elles n'affectent pas la manière dont les certificats émis seront utilisés.

20.3.3. Création d'un CA ACL pour les navigateurs web des utilisateurs s'authentifiant auprès des serveurs web à l'aide de certificats émis par webclient-ca

Cette section explique comment créer une CA ACL qui oblige l'administrateur système à utiliser la sous-CA **webclient-ca** et le profil **IECUserRoles** lorsqu'il demande un certificat. Si l'utilisateur demande un certificat à partir d'un sous-CA différent ou d'un profil différent, la demande échoue. La seule exception est l'activation d'une autre CA ACL correspondante. Pour afficher les CA ACL disponibles, voir Affichage des CA ACL dans IdM CLI .

Conditions préalables

- Assurez-vous d'avoir obtenu les informations d'identification de l'administrateur IdM.

Procédure

1. Créez une CAL à l'aide de la commande **ipa caacl** et spécifiez son nom :

```
$ ipa caacl-add TLS_web_client_authentication
-----------------------------------------
Added CA ACL "TLS_web_client_authentication"
-----------------------------------------
  ACL name: TLS_web_client_authentication
  Enabled: TRUE
```

2. Modifiez la CA ACL en utilisant la commande **ipa caacl-mod** pour spécifier la description de la CA ACL :

```
$ ipa caacl-mod TLS_web_client_authentication --desc="CAACL for user web
browsers authenticating to web servers using certificates issued by webclient-ca"
---------------------------------------------
Modified CA ACL "TLS_web_client_authentication"
---------------------------------------------
  ACL name: TLS_web_client_authentication
  Description: CAACL for user web browsers authenticating to web servers using certificates
issued by webclient-ca
  Enabled: TRUE
```

3. Ajoutez la sous-CA **webclient-ca** à la CA ACL :

```
$ ipa caacl-add-ca TLS_web_client_authentication --ca=webclient-ca
  ACL name: TLS_web_client_authentication
  Description: CAACL for user web browsers authenticating to web servers using certificates
issued by webclient-ca
  Enabled: TRUE
  CAs: webclient-ca
-------------------------
Number of members added 1
-------------------------
```

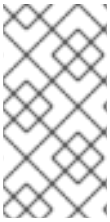
4. Utilisez la commande **ipa caacl-add-profile** pour spécifier le profil de certificat pour le certificat demandé :

```
$ ipa caacl-add-profile TLS_web_client_authentication --certprofiles=IECUserRoles
  ACL name: TLS_web_client_authentication
  Description: CAACL for user web browsers authenticating to web servers using certificates
issued by webclient-ca
  Enabled: TRUE
  CAs: webclient-ca
  Profiles: IECUserRoles
-------------------------
Number of members added 1
-------------------------
```

5. Modifiez l'ACL CA à l'aide de la commande **ipa caacl-mod** pour spécifier que l'ACL CA s'applique à tous les utilisateurs IdM :

```
$ ipa caacl-mod TLS_web_client_authentication --usercat=all
---------------------------------------------
Modified CA ACL "TLS_web_client_authentication"
---------------------------------------------
  ACL name: TLS_web_client_authentication
  Description: CAACL for user web browsers authenticating to web servers using certificates
issued by webclient-ca
  Enabled: TRUE
  User category: all
  CAs: webclient-ca
  Profiles: IECUserRoles
```

Vous pouvez utiliser immédiatement l'ACL CA nouvellement créée. Elle est activée par défaut après sa création.



NOTE

L'objectif des listes de contrôle d'accès est de spécifier les combinaisons d'autorités de certification et de profils autorisées pour les demandes émanant de mandants ou de groupes particuliers. Les CA ACL n'affectent pas la validation ou la confiance dans les certificats. Elles n'affectent pas la manière dont les certificats émis seront utilisés.

20.4. OBTENTION D'UN CERTIFICAT IDM POUR UN SERVICE À L'AIDE DE CERTMONGER

Pour garantir que la communication entre les navigateurs et le service web fonctionnant sur votre client IdM est sécurisée et cryptée, utilisez un certificat TLS. Si vous voulez restreindre les navigateurs web à faire confiance aux certificats émis par la sous-CA **webserver-ca** mais par aucune autre sous-CA IdM, obtenez le certificat TLS pour votre service web auprès de la sous-CA **webserver-ca**.

Cette section décrit comment utiliser **certmonger** pour obtenir un certificat IdM pour un service (**HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM**) fonctionnant sur un client IdM.

L'utilisation de **certmonger** pour demander le certificat automatiquement signifie que **certmonger** gère et renouvelle le certificat lorsqu'il doit être renouvelé.

Pour une représentation visuelle de ce qui se passe lorsque **certmonger** demande un certificat de service, voir Section 20.5, « Flux de communication pour le demandeur de certificat demandant un certificat de service ».

Conditions préalables

- Le serveur web est enregistré en tant que client IdM.
- Vous avez un accès root au client IdM sur lequel vous exécutez la procédure.
- Le service pour lequel vous demandez un certificat ne doit pas nécessairement préexister dans l'IdM.

Procédure

1. Sur le client **my_company.idm.example.com** IdM sur lequel le service **HTTP** est exécuté, demandez un certificat pour le service correspondant au principal **HTTP/my_company.idm.example.com@IDM.EXAMPLE.COM** et spécifiez que
   - Le certificat doit être stocké dans le fichier local **/etc/pki/tls/certs/httpd.pem**
   - La clé privée doit être stockée dans le fichier local **/etc/pki/tls/private/httpd.key**
   - La sous-CA **webserver-ca** doit être l'autorité de certification émettrice
   - Qu'une demande d'extension pour **SubjectAltName** soit ajoutée à la demande de signature avec le nom DNS de **my_company.idm.example.com**:

     ```
     # ipa-getcert request -K HTTP/my_company.idm.example.com -k
     /etc/pki/tls/private/httpd.key -f /etc/pki/tls/certs/httpd.pem -g 2048 -D
     my_company.idm.example.com -X webserver-ca -C "systemctl restart httpd"
     New signing request "20190604065735" added.
     ```

     Dans la commande ci-dessus :

- La commande **ipa-getcert request** spécifie que le certificat doit être obtenu auprès de l'autorité de certification IdM. La commande **ipa-getcert request** est un raccourci pour **getcert request -c IPA**.
- L'option **-g** spécifie la taille de la clé à générer s'il n'y en a pas déjà une.
- L'option **-D** spécifie la valeur DNS **SubjectAltName** à ajouter à la demande.
- L'option **-X** précise que l'émetteur du certificat doit être **webserver-ca** et non **ipa**.
- L'option **-C** demande à **certmonger** de redémarrer le service **httpd** après avoir obtenu le certificat.
- Pour spécifier que le certificat doit être émis avec un profil particulier, utilisez l'option **-T**.

2. Optionnellement, pour vérifier le statut de votre demande :

```
# ipa-getcert list -f /etc/pki/tls/certs/httpd.pem
Number of certificates and requests being tracked: 3.
Request ID '20190604065735':
   status: MONITORING
   stuck: no
   key pair storage: type=FILE,location='/etc/pki/tls/private/httpd.key'
   certificate: type=FILE,location='/etc/pki/tls/certs/httpd.crt'
   CA: IPA
   issuer: CN=WEBSERVER,O=IDM.EXAMPLE.COM

[...]
```

La sortie montre que la demande est à l'état **MONITORING**, ce qui signifie qu'un certificat a été obtenu. Les emplacements de la paire de clés et du certificat sont ceux demandés.

20.5. FLUX DE COMMUNICATION POUR LE DEMANDEUR DE CERTIFICAT DEMANDANT UN CERTIFICAT DE SERVICE

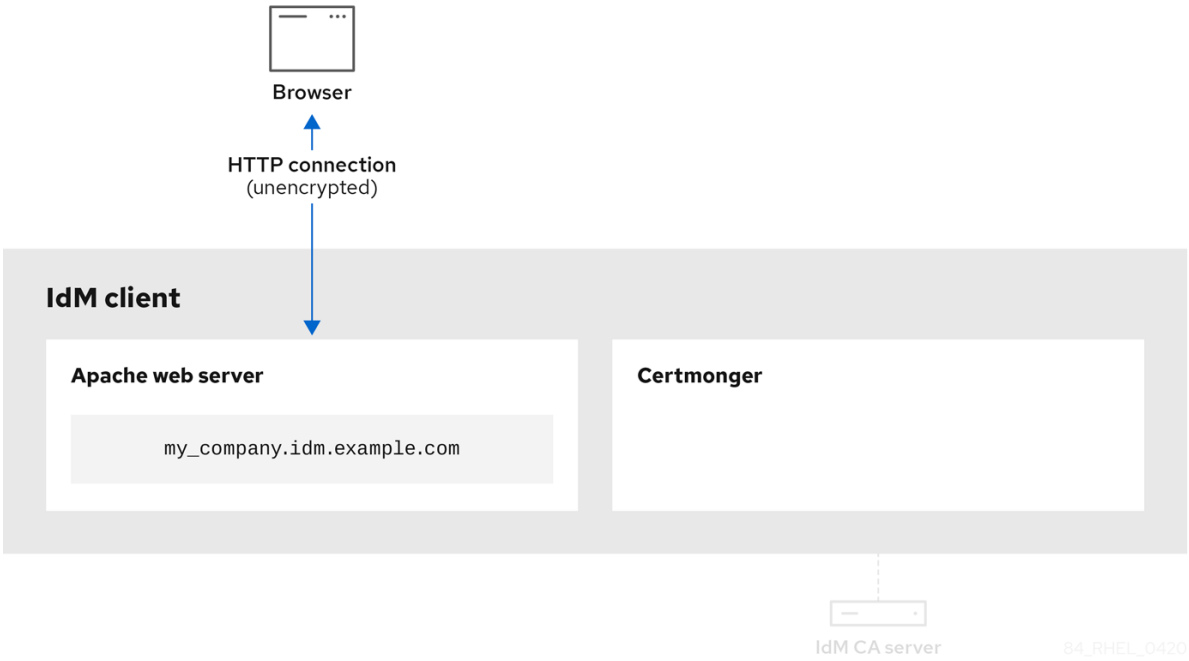
Les diagrammes de cette section montrent les étapes de ce qui se passe lorsque **certmonger** demande un certificat de service au serveur de l'autorité de certification (CA) de la gestion des identités (IdM). La séquence est constituée de ces diagrammes :

- Communication non cryptée
- Certmonger demande un certificat de service
- CA IdM délivrant le certificat de service
- Le certificateur qui applique le certificat de service
- Certmonger demande un nouveau certificat lorsque l'ancien est proche de l'expiration

Dans les diagrammes, le sous-CA **webserver-ca** est représenté par le générique **IdM CA server**.

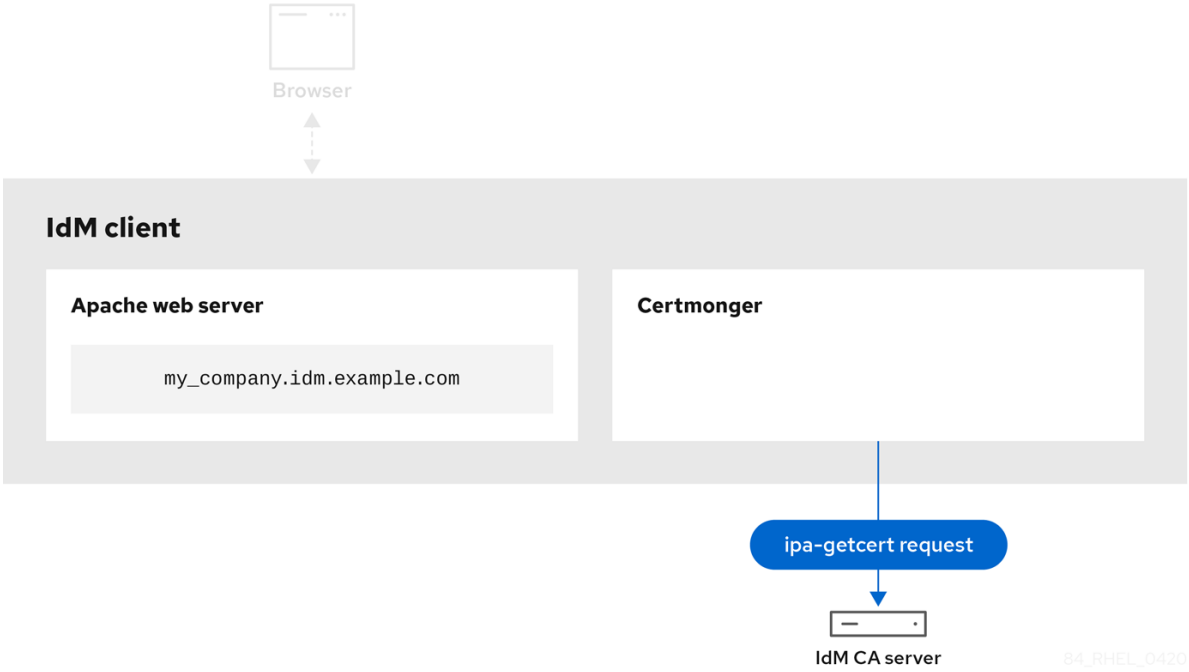
La communication non chiffrée illustre la situation initiale : sans certificat HTTPS, la communication entre le serveur web et le navigateur n'est pas chiffrée.

Figure 20.3. Communication non cryptée



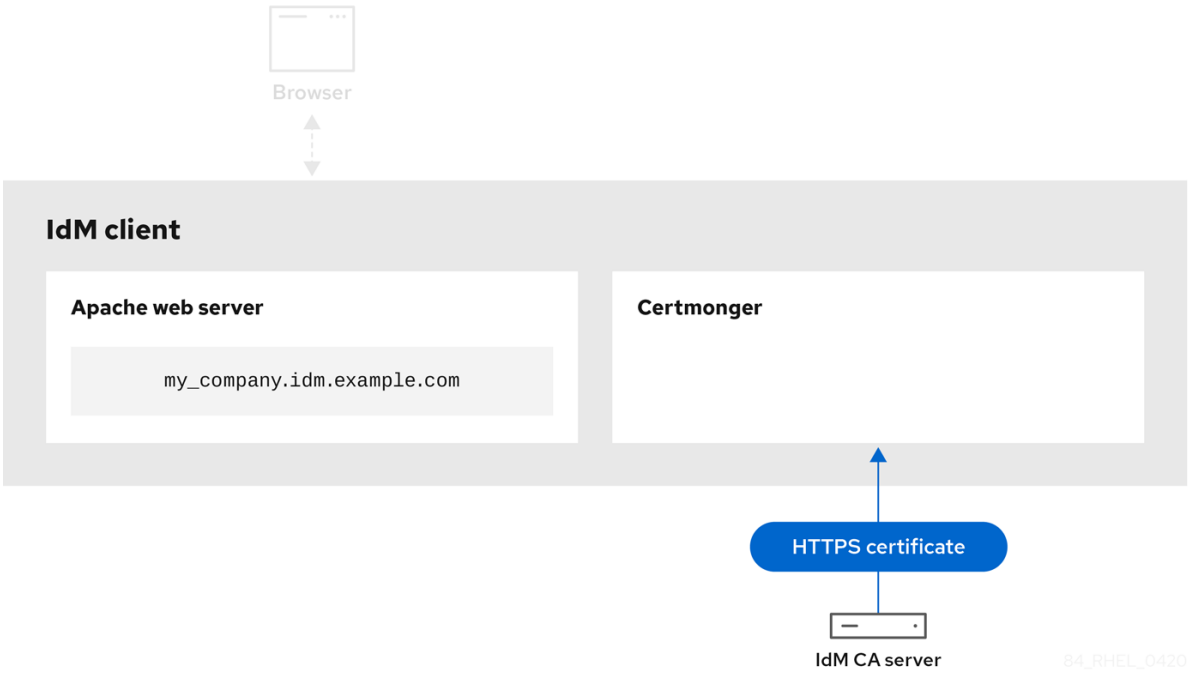
Certmonger demandant un certificat de service montre l'administrateur système utilisant **certmonger** pour demander manuellement un certificat HTTPS pour le serveur web Apache. Notez que lors de la demande d'un certificat de serveur web, certmonger ne communique pas directement avec l'autorité de certification. Il passe par IdM.

Figure 20.4. Certmonger demande un certificat de service



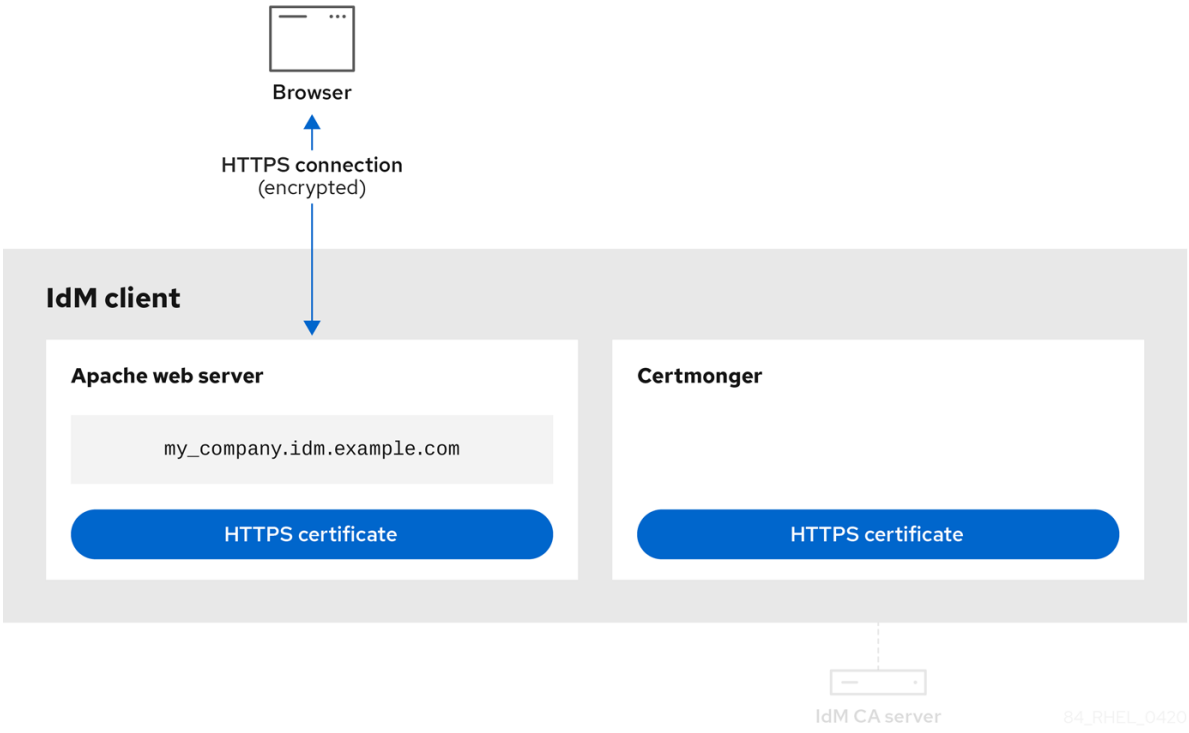
L'autorité de certification IdM délivrant le certificat de service montre une autorité de certification IdM délivrant un certificat HTTPS pour le serveur web.

Figure 20.5. CA IdM délivrant le certificat de service



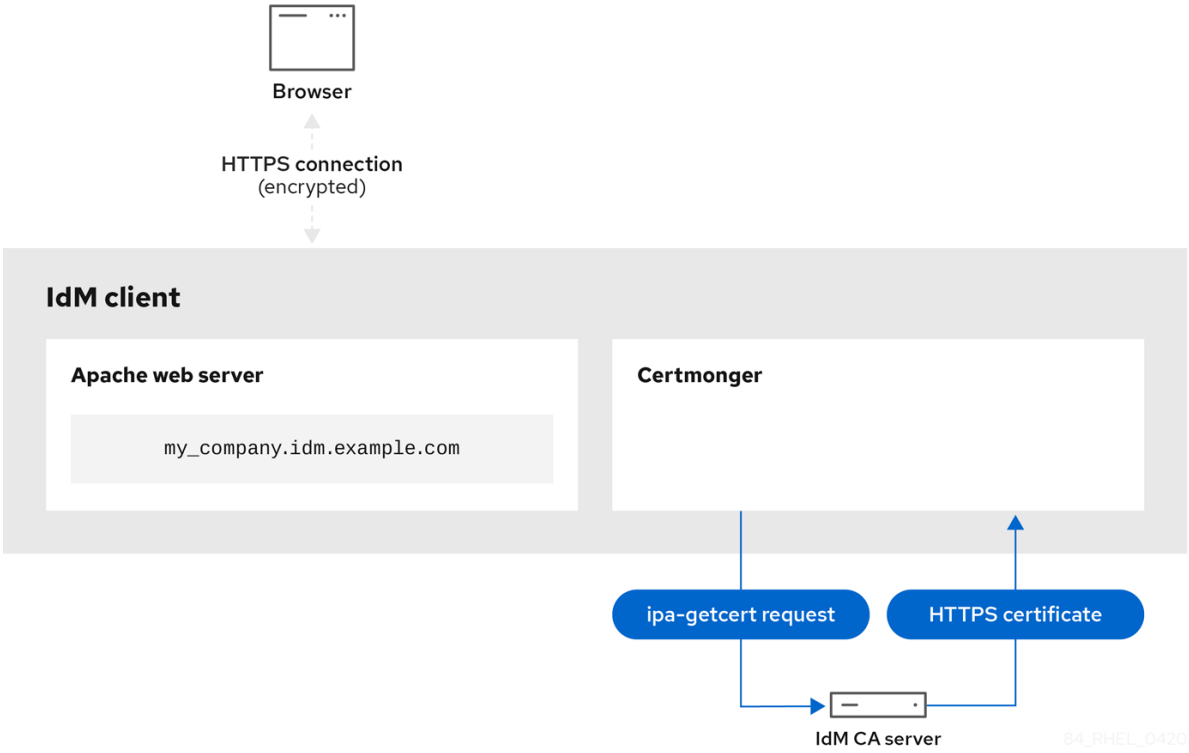
Certmonger l'application du certificat de service montre que **certmonger** place le certificat HTTPS aux endroits appropriés sur le client IdM et, si on lui demande de le faire, redémarre le service **httpd**. Le serveur Apache utilise ensuite le certificat HTTPS pour crypter le trafic entre lui-même et le navigateur.

Figure 20.6. Le certificateur qui applique le certificat de service



Certmonger demandant un nouveau certificat lorsque l'ancien est proche de l' expiration montre que **certmonger** demande automatiquement le renouvellement du certificat de service auprès de l'autorité de certification IdM avant l'expiration du certificat. L'AC IdM délivre un nouveau certificat.

Figure 20.7. Certmonger demande un nouveau certificat lorsque l'ancien est proche de l'expiration



20.6. CONFIGURATION D'UNE INSTANCE UNIQUE DU SERVEUR HTTP APACHE

Cette section décrit comment configurer un serveur HTTP Apache à instance unique pour servir du contenu HTML statique.

Suivez la procédure décrite dans cette section si le serveur web doit fournir le même contenu à tous les domaines associés au serveur. Si vous souhaitez fournir un contenu différent pour différents domaines, configurez des hôtes virtuels basés sur le nom. Pour plus d'informations, voir Configuration des hôtes virtuels Apache basés sur le nom.

Procédure

1. Installez le paquetage **httpd**:

```
# dnf install httpd
```

2. Si vous utilisez **firewalld**, ouvrez le port TCP **80** dans le pare-feu local :

```
# firewall-cmd --permanent --add-port=80/tcp
# firewall-cmd --reload
```

3. Activez et démarrez le service **httpd**:

```
# systemctl enable --now httpd
```

4. Facultatif : Ajoutez des fichiers HTML au répertoire **/var/www/html/**.

NOTE

Lors de l'ajout de contenu à **/var/www/html/**, les fichiers et les répertoires doivent être lisibles par l'utilisateur sous lequel **httpd** s'exécute par défaut. Le propriétaire du contenu peut être soit l'utilisateur **root** et le groupe d'utilisateurs **root**, soit un autre utilisateur ou groupe au choix de l'administrateur. Si le propriétaire du contenu est l'utilisateur **root** et le groupe d'utilisateurs **root**, les fichiers doivent pouvoir être lus par d'autres utilisateurs. Le contexte SELinux pour tous les fichiers et répertoires doit être **httpd_sys_content_t**, qui est appliqué par défaut à tout le contenu du répertoire **/var/www**.

Verification steps

- Connectez-vous à l'aide d'un navigateur web à **http://my_company.idm.example.com/** ou **http://*server_IP*/**.
  Si le répertoire **/var/www/html/** est vide ou ne contient pas de fichier **index.html** ou **index.htm**, Apache affiche **Red Hat Enterprise Linux Test Page**. Si **/var/www/html/** contient des fichiers HTML portant un nom différent, vous pouvez les charger en saisissant l'URL de ce fichier, par exemple **http://*server_IP*/*example.html*** ou **http://my_company.idm.example.com/*example.html***.

Ressources supplémentaires

- Manuel Apache : Manuel d'installation du serveur HTTP Apache .
- Voir la page de manuel **httpd.service(8)**.

20.7. AJOUTER LE CRYPTAGE TLS À UN SERVEUR HTTP APACHE

Cette section décrit comment activer le cryptage TLS sur le serveur HTTP Apache **my_company.idm.example.com** pour le domaine **idm.example.com**.

Conditions préalables

- Le serveur Apache HTTP **my_company.idm.example.com** est installé et fonctionne.
- Vous avez obtenu le certificat TLS de la sous-CA **webserver-ca** et l'avez stocké dans le fichier **/etc/pki/tls/certs/httpd.pem** comme décrit dans Obtaining an IdM certificate for a service using certmonger (Obtention d'un certificat IdM pour un service à l'aide de certmonger). Si vous utilisez un chemin différent, adaptez les étapes correspondantes de la procédure.
- La clé privée correspondante est stockée dans le fichier **/etc/pki/tls/private/httpd.key**. Si vous utilisez un chemin différent, adaptez les étapes correspondantes de la procédure.
- Le certificat CA **webserver-ca** est stocké dans le fichier **/etc/pki/tls/private/sub-ca.crt**. Si vous utilisez un chemin différent, adaptez les étapes correspondantes de la procédure.
- Les clients et le serveur web **my_company.idm.example.com** résolvent le nom d'hôte du serveur en adresse IP du serveur web.

Procédure

1. Installez le paquetage **mod_ssl**:

```
# dnf install mod_ssl
```

2. Modifiez le fichier **/etc/httpd/conf.d/ssl.conf** et ajoutez les paramètres suivants à la directive **<VirtualHost _default_:443>**:

   a. Définir le nom du serveur :

```
ServerName my_company.idm.example.com
```

IMPORTANT

Le nom du serveur doit correspondre à l'entrée définie dans le champ **Common Name** du certificat.

   b. Facultatif : Si le certificat contient des noms d'hôtes supplémentaires dans le champ **Subject Alt Names** (SAN), vous pouvez configurer **mod_ssl** pour qu'il fournisse également un cryptage TLS pour ces noms d'hôtes. Pour ce faire, ajoutez le paramètre **ServerAliases** avec les noms correspondants :

```
ServerAlias www.my_company.idm.example.com
server.my_company.idm.example.com
```

   c. Définissez les chemins d'accès à la clé privée, au certificat du serveur et au certificat de l'autorité de certification :

```
SSLCertificateKeyFile "/etc/pki/tls/private/httpd.key"
SSLCertificateFile "/etc/pki/tls/certs/httpd.pem"
SSLCACertificateFile "/etc/pki/tls/certs/ca.crt"
```

3. Pour des raisons de sécurité, configurez l'accès au fichier de la clé privée uniquement pour l'utilisateur **root**:

```
# chown root:root /etc/pki/tls/private/httpd.key
# chmod 600 //etc/pki/tls/private/httpd.key
```

AVERTISSEMENT

Si des utilisateurs non autorisés ont eu accès à la clé privée, révoquez le certificat, créez une nouvelle clé privée et demandez un nouveau certificat. Sinon, la connexion TLS n'est plus sécurisée.

4. Si vous utilisez **firewalld**, ouvrez le port **443** dans le pare-feu local :

```
# firewall-cmd --permanent --add-port=443/tcp
# firewall-cmd --reload
```

5. Redémarrez le service **httpd**:

```
# systemctl restart httpd
```

NOTE

Si vous avez protégé le fichier de clé privée par un mot de passe, vous devez saisir ce mot de passe à chaque démarrage du service **httpd**.

- Utilisez un navigateur et connectez-vous à **https://my_company.idm.example.com**.

Ressources supplémentaires

- Cryptage SSL/TLS.
- Considérations de sécurité pour TLS dans RHEL 8

20.8. DÉFINITION DES VERSIONS DU PROTOCOLE TLS PRISES EN CHARGE SUR UN SERVEUR HTTP APACHE

Par défaut, le serveur HTTP Apache sur RHEL utilise la politique cryptographique du système qui définit des valeurs par défaut sûres, qui sont également compatibles avec les navigateurs récents. Par exemple, la politique **DEFAULT** définit que seules les versions des protocoles **TLSv1.2** et **TLSv1.3** sont activées dans Apache.

Cette section décrit comment configurer manuellement les versions du protocole TLS prises en charge par votre serveur HTTP Apache **my_company.idm.example.com**. Suivez la procédure si votre environnement exige de n'activer que certaines versions du protocole TLS, par exemple :

- Si votre environnement l'exige, les clients peuvent également utiliser le protocole faible **TLS1** (TLSv1.0) ou **TLS1.1**.
- Si vous souhaitez configurer Apache pour qu'il ne prenne en charge que le protocole **TLSv1.2** ou **TLSv1.3**.

Conditions préalables

- Le cryptage TLS est activé sur le serveur **my_company.idm.example.com** comme décrit dans Ajouter le cryptage TLS à un serveur HTTP Apache .

Procédure

1. Modifiez le fichier **/etc/httpd/conf/httpd.conf** et ajoutez le paramètre suivant à la directive **<VirtualHost>** pour laquelle vous souhaitez définir la version du protocole TLS. Par exemple, pour activer uniquement le protocole **TLSv1.3**:

```
SSLProtocol -All TLSv1.3
```

2. Redémarrez le service **httpd**:

```
# systemctl restart httpd
```

Verification steps

1. Utilisez la commande suivante pour vérifier que le serveur prend en charge **TLSv1.3**:

```
# openssl s_client -connect example.com:443 -tls1_3
```

2. Utilisez la commande suivante pour vérifier que le serveur ne prend pas en charge **TLSv1.2**:

```
# openssl s_client -connect example.com:443 -tls1_2
```

Si le serveur ne prend pas en charge le protocole, la commande renvoie une erreur :

```
140111600609088:error:1409442E:Routines SSL:ssl3_read_bytes:version du protocole
d'alerte tlsv1:ssl/record/rec_layer_s3.c:1543:alerte SSL numéro 70
```

3. Facultatif : Répétez la commande pour d'autres versions du protocole TLS.

Ressources supplémentaires

- **update-crypto-policies(8)** page de manuel
- Utilisation de politiques cryptographiques à l'échelle du système .
- Pour plus de détails sur le paramètre **SSLProtocol**, reportez-vous à la documentation **mod_ssl** dans le manuel Apache : Installation du serveur HTTP Ap ache.

20.9. DÉFINITION DES ALGORITHMES DE CHIFFREMENT PRIS EN CHARGE SUR UN SERVEUR HTTP APACHE

Par défaut, le serveur HTTP Apache utilise la politique cryptographique du système qui définit des valeurs par défaut sûres, également compatibles avec les navigateurs récents. Pour obtenir la liste des algorithmes de chiffrement autorisés par la politique cryptographique du système, consultez le fichier **/etc/crypto-policies/back-ends/openssl.config**.

Cette section décrit comment configurer manuellement les algorithmes de chiffrement pris en charge par le serveur HTTP Apache **my_company.idm.example.com**. Suivez la procédure si votre environnement requiert des algorithmes de chiffrement spécifiques.

Conditions préalables

- Le cryptage TLS est activé sur le serveur **my_company.idm.example.com** comme décrit dans Ajouter le cryptage TLS à un serveur HTTP Apache .

Procédure

1. Modifiez le fichier **/etc/httpd/conf/httpd.conf** et ajoutez le paramètre **SSLCipherSuite** à la directive **<VirtualHost>** pour laquelle vous souhaitez définir les algorithmes TLS :

```
SSLCipherSuite "EECDH AESGCM:EDH AESGCM:AES256 EECDH:AES256
EDH:!SHA1:!SHA256"
```

Cet exemple n'active que les algorithmes de chiffrement **EECDH AESGCM**, **EDH AESGCM**, **AES256 EECDH** et **AES256 EDH** et désactive tous les algorithmes de chiffrement qui utilisent les codes d'authentification des messages (MAC) **SHA1** et **SHA256**.

2. Redémarrez le service **httpd**:

```
# systemctl restart httpd
```

Verification steps

1. Pour afficher la liste des algorithmes de chiffrement pris en charge par le serveur HTTP Apache :

   a. Installez le paquetage **nmap**:

   ```
   # dnf install nmap
   ```

   b. Utilisez l'utilitaire **nmap** pour afficher les algorithmes de chiffrement pris en charge :

   ```
   # nmap --script ssl-enum-ciphers -p 443 example.com
   ...
   PORT    STATE SERVICE
   443/tcp open  https
   | ssl-enum-ciphers:
   |   TLSv1.2:
   |     ciphers:
   |       TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (ecdh_x25519) - A
   |       TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (dh 2048) - A
   |       TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (ecdh_x25519) - A
   ...
   ```

Ressources supplémentaires

- **update-crypto-policies(8)** page de manuel
- Utilisation de politiques cryptographiques à l'échelle du système .
- Manuel d'installation du serveur HTTP Apache - SSLCipherSuite

20.10. CONFIGURATION DE L'AUTHENTIFICATION DU CERTIFICAT CLIENT TLS

L'authentification par certificat client permet aux administrateurs d'autoriser uniquement les utilisateurs qui s'authentifient à l'aide d'un certificat à accéder aux ressources du serveur web **my_company.idm.example.com**. Cette section explique comment configurer l'authentification par certificat client pour l'annuaire **/var/www/html/Example/**.

IMPORTANT

Si le serveur **my_company.idm.example.com** Apache utilise le protocole TLS 1.3, certains clients nécessitent une configuration supplémentaire. Par exemple, dans Firefox, définissez le paramètre **security.tls.enable_post_handshake_auth** dans le menu **about:config** sur **true**. Pour plus de détails, voir Transport Layer Security version 1.3 dans Red Hat Enterprise Linux 8.

Conditions préalables

- Le cryptage TLS est activé sur le serveur **my_company.idm.example.com** comme décrit dans Ajouter le cryptage TLS à un serveur HTTP Apache .

Procédure

1. Modifiez le fichier **/etc/httpd/conf/httpd.conf** et ajoutez les paramètres suivants à la directive **<VirtualHost>** pour laquelle vous souhaitez configurer l'authentification du client :

```
<Directory "/var/www/html/Example/">
  SSLVerifyClient require
</Directory>
```

   Le paramètre **SSLVerifyClient require** définit que le serveur doit valider avec succès le certificat du client avant que ce dernier puisse accéder au contenu du répertoire **/var/www/html/Example/**.

2. Redémarrez le service **httpd**:

```
# systemctl restart httpd
```

Verification steps

1. Utilisez l'utilitaire **curl** pour accéder à l'URL **https://my_company.idm.example.com/Example/** sans authentification du client :

```
$ curl https://my_company.idm.example.com/Example/
curl: (56) OpenSSL SSL_read: error:1409445C:SSL routines:ssl3_read_bytes:tlsv13 **alert
certificate required**, errno 0
```

   L'erreur indique que le serveur web **my_company.idm.example.com** nécessite une authentification par certificat client.

2. Transmettez la clé privée et le certificat du client, ainsi que le certificat de l'autorité de certification à **curl** pour accéder à la même URL avec l'authentification du client :

```
$ curl --cacert ca.crt --key client.key --cert client.crt
https://my_company.idm.example.com/Example/
```

   Si la demande aboutit, **curl** affiche le fichier **index.html** stocké dans le répertoire **/var/www/html/Example/**.

Ressources supplémentaires

- Manuel d'installation du serveur HTTP Apache - configuration mod_ssl

20.11. DEMANDER UN NOUVEAU CERTIFICAT D'UTILISATEUR ET L'EXPORTER VERS LE CLIENT

En tant qu'administrateur Identity Management (IdM), vous pouvez configurer un serveur web fonctionnant sur un client IdM pour demander aux utilisateurs qui utilisent des navigateurs web pour accéder au serveur de s'authentifier avec des certificats émis par un sous-CA IdM spécifique. Complétez cette section pour demander un certificat d'utilisateur à un sous-CA IdM spécifique et pour exporter le certificat et la clé privée correspondante sur l'hôte à partir duquel l'utilisateur souhaite accéder au serveur web à l'aide d'un navigateur web. Ensuite, il faut importer le certificat et la clé privée dans le navigateur.

Procédure

1. Si vous le souhaitez, créez un nouveau répertoire, par exemple **~/certdb/**, et faites-en une base de données de certificats temporaire. Si on vous le demande, créez un mot de passe pour la base de données des certificats NSS afin de crypter les clés du certificat qui sera généré lors d'une étape ultérieure :

```
# mkdir ~/certdb/
# certutil -N -d ~/certdb/
Enter a password which will be used to encrypt your keys.
The password should be at least 8 characters long,
and should contain at least one non-alphabetic character.

Enter new password:
Re-enter password:
```

2. Créez la demande de signature de certificat (CSR) et redirigez la sortie vers un fichier. Par exemple, pour créer une CSR avec le nom **certificate_request.csr** pour un certificat bit **4096** pour l'utilisateur **idm_user** dans le domaine **IDM.EXAMPLE.COM**, en définissant le surnom des clés privées du certificat à **idm_user** pour faciliter la recherche, et en définissant le sujet à **CN=idm_user,O=IDM.EXAMPLE.COM**:

```
# certutil -R -d ~/certdb/ -a -g 4096 -n idm_user -s "CN=idm_user,O=IDM.EXAMPLE.COM"
> certificate_request.csr
```

3. À l'invite, saisissez le même mot de passe que celui que vous avez saisi lorsque vous avez utilisé **certutil** pour créer la base de données temporaire. Continuez ensuite à taper randlomly jusqu'à ce qu'on vous dise d'arrêter :

```
Enter Password or Pin for "NSS Certificate DB":

A random seed must be generated that will be used in the
creation of your key.  One of the easiest ways to create a
random seed is to use the timing of keystrokes on a keyboard.

To begin, type keys on the keyboard until this progress meter
is full.  DO NOT USE THE AUTOREPEAT FUNCTION ON YOUR KEYBOARD!


Continue typing until the progress meter is full:
```

4. Soumettez le fichier de demande de certificat au serveur. Indiquez le principal Kerberos à associer au certificat nouvellement émis, le fichier de sortie pour stocker le certificat et,

éventuellement, le profil de certificat. Indiquez la sous-CA IdM à laquelle vous souhaitez délivrer le certificat. Par exemple, pour obtenir un certificat du profil **IECUserRoles**, un profil avec l'extension des rôles d'utilisateur ajoutés, pour le principal **idm_user**@**IDM.EXAMPLE.COM** à partir de **webclient-ca**, et enregistrer le certificat dans le fichier ~/**idm_user.pem**:

```
# ipa cert-request certificate_request.csr --principal=idm_user@IDM.EXAMPLE.COM --
profile-id=IECUserRoles --ca=webclient-ca --certificate-out=~/idm_user.pem
```

5. Ajoutez le certificat à la base de données NSS. Utilisez l'option **-n** pour définir le même surnom que celui que vous avez utilisé lors de la création de la CSR, afin que le certificat corresponde à la clé privée dans la base de données NSS. L'option **-t** définit le niveau de confiance. Pour plus de détails, voir la page de manuel certutil(1). L'option **-i** spécifie le fichier de certificat d'entrée. Par exemple, pour ajouter à la base de données NSS un certificat avec le pseudonyme **idm_user** qui est stocké dans le fichier ~/**idm_user.pem** de la base de données ~/**certdb**/:

```
# certutil -A -d ~/certdb/ -n idm_user -t "P,," -i ~/idm_user.pem
```

6. Vérifiez que la clé dans la base de données NSS n'indique pas **(orphan)** comme surnom. Par exemple, pour vérifier que le certificat stocké dans la base de données ~/**certdb**/ n'est pas orphelin :

```
# certutil -K -d ~/certdb/
< 0> rsa      5ad14d41463b87a095b1896cf0068ccc467df395   NSS Certificate
DB:idm_user
```

7. Utilisez la commande **pk12util** pour exporter le certificat de la base de données NSS au format PKCS12. Par exemple, pour exporter le certificat avec le pseudonyme **idm_user** de la base de données NSS /**root**/**certdb** vers le fichier ~/**idm_user.p12**:

```
# pk12util -d ~/certdb -o ~/idm_user.p12 -n idm_user
Enter Password or Pin for "NSS Certificate DB":
Enter password for PKCS12 file:
Re-enter password:
pk12util: PKCS12 EXPORT SUCCESSFUL
```

8. Transférez le certificat vers l'hôte sur lequel vous souhaitez activer l'authentification par certificat pour **idm_user**:

```
# scp ~/idm_user.p12 idm_user@client.idm.example.com:/home/idm_user/
```

9. Sur l'hôte vers lequel le certificat a été transféré, rendez le répertoire dans lequel le fichier .pkcs12 est stocké inaccessible au groupe "other" pour des raisons de sécurité :

```
# chmod o-rwx /home/idm_user/
```

10. Pour des raisons de sécurité, supprimez la base de données NSS temporaire et le fichier .pkcs12 du serveur :

```
# rm ~/certdb/
# rm ~/idm_user.p12
```

20.12. CONFIGURATION D'UN NAVIGATEUR POUR ACTIVER L'AUTHENTIFICATION PAR CERTIFICAT

Pour pouvoir s'authentifier à l'aide d'un certificat lors de l'utilisation de l'interface WebUI pour se connecter à la gestion des identités (IdM), vous devez importer l'utilisateur et les certificats de l'autorité de certification (AC) concernés dans le navigateur Mozilla Firefox ou Google Chrome. L'hôte sur lequel le navigateur est exécuté ne doit pas nécessairement faire partie du domaine IdM.

IdM prend en charge les navigateurs suivants pour se connecter à l'interface WebUI :

- Mozilla Firefox 38 et versions ultérieures
- Google Chrome 46 et versions ultérieures

La procédure suivante montre comment configurer le navigateur Mozilla Firefox 57.0.1.

Conditions préalables

- Vous disposez du certificat d'utilisateur que vous souhaitez importer dans le navigateur au format PKCS#12.
- Vous avez téléchargé le certificat de sous-CA et vous l'avez à votre disposition au format PEM.

Procédure

1. Ouvrez Firefox, puis naviguez vers **Preferences → Privacy & Security**.

Figure 20.8. Section Vie privée et sécurité dans les préférences

2. Cliquez sur **Afficher les certificats**.

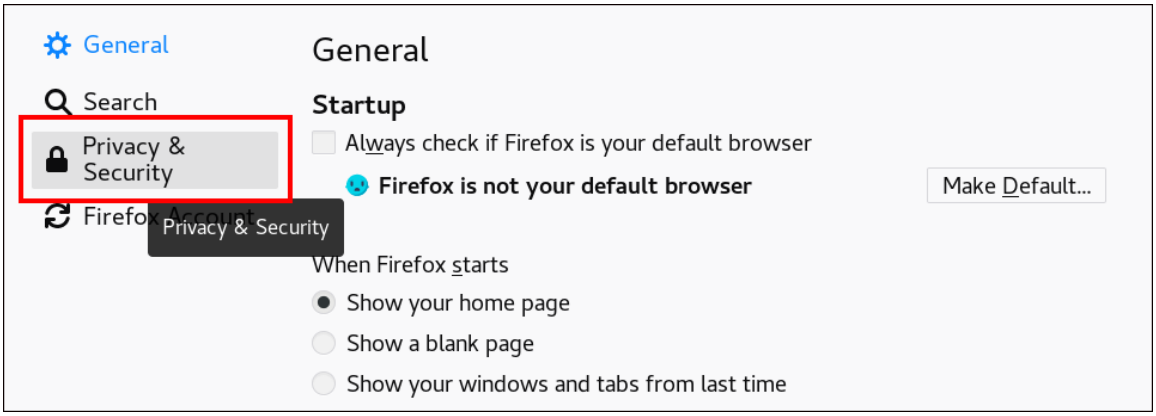
Figure 20.9. Voir les certificats en protection de la vie privée et en sécurité

3. Dans l'onglet **Your Certificates**, cliquez sur **Importer**. Localisez et ouvrez le certificat de l'utilisateur au format PKCS12, puis cliquez sur **OK** et **OK**.

4. Pour vous assurer que votre sous-CA IdM est reconnu par Firefox comme une autorité de confiance, importez le certificat du sous-CA IdM que vous avez sauvegardé dans Télécharger le certificat du sous-CA à partir de l'interface Web de l'IdM comme un certificat d'autorité de confiance :

   a. Ouvrez Firefox, allez dans Préférences et cliquez sur **Confidentialité & Sécurité**.

   Figure 20.10. Section Vie privée et sécurité dans les préférences



   b. Cliquez sur **Afficher les certificats**.

   Figure 20.11. Voir les certificats en protection de la vie privée et en sécurité

   c. Dans l'onglet **Authorities**, cliquez sur **Importer**. Localisez et ouvrez le certificat sub-CA. Faites confiance au certificat pour identifier les sites web, puis cliquez sur **OK** et **OK**.

CHAPITRE 21. INVALIDER RAPIDEMENT UN GROUPE SPÉCIFIQUE DE CERTIFICATS APPARENTÉS

En tant qu'administrateur système, si vous souhaitez pouvoir invalider rapidement un groupe spécifique de certificats apparentés :

- Concevez vos applications de manière à ce qu'elles ne fassent confiance qu'aux certificats émis par une sous-CA de gestion d'identité (IdM) légère spécifique. Par la suite, vous pourrez invalider tous ces certificats en révoquant uniquement le certificat de la sous-CA de gestion d'identité (IdM) qui a émis ces certificats. Pour plus de détails sur la création et l'utilisation d'un sous-CA léger dans IdM, voir rapidement Invalider un groupe spécifique de certificats liés .

- Pour garantir que tous les certificats émis par la sous-CA IdM à révoquer sont immédiatement invalides, configurez les applications qui s'appuient sur ces certificats pour qu'elles utilisent les répondeurs OCSP de l'IdM. Par exemple, pour configurer le navigateur Firefox afin qu'il utilise les répondeurs OCSP, assurez-vous que la case **Query OCSP responder servers to confirm the current validity of certificates** est cochée dans les préférences de Firefox.
  Dans l'IdM, la liste de révocation des certificats (CRL) est mise à jour toutes les quatre heures. d Pour invalider tous les certificats émis par une sous-CA de l'IdM, révoquez le certificat de la sous-CA de l'IdM. En outre, désactivez les ACL de l'autorité de certification concernée et envisagez de désactiver la sous-AAC IdM. La désactivation de l'AC secondaire empêche l'AC secondaire d'émettre de nouveaux certificats, mais permet de produire des réponses OCSP (Online Certificate Status Protocol) pour les certificats précédemment émis, car les clés de signature de l'AC secondaire sont conservées.

IMPORTANT

Ne supprimez pas le sous-CA si vous utilisez OCSP dans votre environnement. La suppression du sous-CA supprime les clés de signature du sous-CA, ce qui empêche la production de réponses OCSP pour les certificats émis par ce sous-CA.

Le seul cas où il est préférable de supprimer un sous-CA plutôt que de le désactiver est celui où l'on souhaite créer un nouveau sous-CA avec le même Subject distinguished name (DN) mais une nouvelle clé de signature.

21.1. DÉSACTIVATION DES LISTES DE CONTRÔLE D'ACCÈS AUX CA DANS L'INTERFACE DE GESTION DE L'IDM

Lorsque vous souhaitez retirer un service IdM ou un groupe de services IdM, envisagez de désactiver toutes les ACL correspondantes existantes.

Complétez cette section pour désactiver l'ACL TLS_web_server_authentication CA qui empêche le serveur web fonctionnant sur votre client IdM de demander un certificat devant être émis par la sous-CA IdM **webserver-ca**, et pour désactiver l'ACL TLS_web_client_authentication CA qui empêche les utilisateurs IdM de demander un certificat d'utilisateur devant être émis par la sous-CA IdM **webclient-ca**.

Procédure

1. En option, pour afficher toutes les ACL de votre environnement IdM, entrez la commande **ipa caacl-find**:

```
$ ipa caacl-find
----------------
```

```
3 CA ACLs matched
-----------------
  ACL name: hosts_services_caIPAserviceCert
  Enabled: TRUE

  ACL name: TLS_web_server_authentication
  Enabled: TRUE

  ACL name: TLS_web_client_authentication
  Enabled: TRUE
```

2. En option, pour afficher les détails d'une CA ACL, entrez la commande **ipa caacl-show** et indiquez le nom de la CA ACL :

```
$ ipa caacl-show TLS_web_server_authentication
  ACL name: TLS_web_server_authentication
  Description: CAACL for web servers authenticating to web clients using certificates issued
by webserver-ca
  Enabled: TRUE
  CAs: webserver-ca
  Profiles: caIPAserviceCert
  Services: HTTP/rhel8server.idm.example.com@IDM.EXAMPLE.COM
```

3. Pour désactiver une CA ACL, entrez la commande **ipa caacl-disable** et indiquez le nom de la CA ACL.

   - Pour désactiver l'ACL **TLS_web_server_authentication** CA, entrez :

```
$ ipa caacl-disable TLS_web_server_authentication
-----------------------------------------------
Disabled CA ACL "TLS_web_server_authentication"
-----------------------------------------------
```

   - Pour désactiver l'ACL **TLS_web_client_authentication** CA, entrez :

```
$ ipa caacl-disable TLS_web_client_authentication
-----------------------------------------------
Disabled CA ACL "TLS_web_client_authentication"
-----------------------------------------------
```

   La seule CA ACL activée actuellement est la CA ACL **hosts_services_caIPAserviceCert**.

IMPORTANT

Soyez extrêmement prudent lorsque vous désactivez l'ACL **hosts_services_caIPAserviceCert**. La désactivation de **hosts_services_caIPAserviceCert**, sans une autre ACL CA accordant aux serveurs IdM l'utilisation de l'AC **ipa** avec le profil **caIPAserviceCert** signifie que le renouvellement des certificats IdM **HTTP** et **LDAP** échouera. Les certificats expirés des serveurs IdM **HTTP** et **LDAP** finiront par provoquer une défaillance du système IdM.

21.2. DÉSACTIVATION D'UN SOUS-CA IDM

Après avoir révoqué le certificat CA d'une sous-CA IdM pour invalider tous les certificats émis par cette sous-CA, envisagez de désactiver la sous-CA IdM si vous n'en avez plus besoin. Vous pourrez réactiver le sous-CA ultérieurement.

La désactivation du sous-CA empêche le sous-CA de délivrer de nouveaux certificats, mais permet de produire des réponses OCSP (Online Certificate Status Protocol) pour les certificats précédemment délivrés, car les clés de signature du sous-CA sont conservées.

Conditions préalables

- Vous êtes connecté en tant qu'administrateur IdM.

Procédure

- Entrez la commande **ipa ca-disable** et spécifiez le nom du sous-CA :

```
$ ipa ca-disable webserver-CA
--------------------
Disabled CA "webserver-CA"
--------------------
```

CHAPITRE 22. VÉRIFICATION DES CERTIFICATS À L'AIDE DE IDM HEALTHCHECK

Cette section aide à comprendre et à utiliser l'outil Healthcheck de la gestion des identités (IdM) pour identifier les problèmes liés aux certificats IPA gérés par certmonger.

Pour plus de détails, voir Healthcheck in IdM.

22.1. CERTIFICATS IDM TESTS DE CONTRÔLE DE SANTÉ

L'outil Healthcheck comprend plusieurs tests permettant de vérifier l'état des certificats gérés par certmonger dans Identity Management (IdM). Pour plus d'informations sur certmonger, voir Obtention d'un certificat IdM pour un service à l'aide de certmonger.

Cette série de tests vérifie l'expiration, la validation, la confiance et d'autres aspects. Plusieurs erreurs peuvent être générées pour le même problème sous-jacent.

Pour voir tous les tests de certificats, exécutez le programme **ipa-healthcheck** avec l'option **--list-sources**:

```
# ipa-healthcheck --list-sources
```

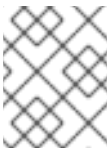
Vous trouverez tous les tests sous la source **ipahealthcheck.ipa.certs**:

IPACertmongerExpirationCheck (contrôle d'expiration)

Ce test vérifie les expirations dans **certmonger**.
Si une erreur est signalée, le certificat a expiré.

Si un avertissement apparaît, cela signifie que le certificat va bientôt expirer. Par défaut, ce test s'applique dans un délai de 28 jours ou moins avant l'expiration du certificat.

Vous pouvez configurer le nombre de jours dans le fichier **/etc/ipahealthcheck/ipahealthcheck.conf**. Après avoir ouvert le fichier, modifiez l'option **cert_expiration_days** située dans la section default.

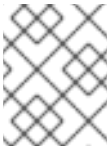


NOTE

Certmonger charge et maintient sa propre vue de l'expiration du certificat. Cette vérification ne valide pas le certificat sur disque.

IPACertfileExpirationCheck (vérification de l'expiration du fichier de certification)

Ce test vérifie si le fichier de certificat ou la base de données NSS ne peut pas être ouvert. Ce test vérifie également l'expiration. Par conséquent, lisez attentivement l'attribut **msg** dans le message d'erreur ou d'avertissement. Le message précise le problème.



NOTE

Ce test vérifie le certificat sur disque. Si un certificat est manquant, illisible, etc., une erreur distincte peut également être soulevée.

IPACertNSSTrust

Ce test compare la confiance dans les certificats stockés dans les bases de données du NSS. Pour les certificats suivis attendus dans les bases de données du SSN, la confiance est comparée à une valeur attendue et une erreur est soulevée en cas de non-concordance.

IPANSSChainValidation

Ce test valide la chaîne de certificats des certificats NSS. Le test s'exécute : **certutil -V -u V -e -d [dbdir] -n [nickname]**

IPAOpenSSLChainValidation

Ce test valide la chaîne des certificats OpenSSL. Pour être comparable à la validation **NSSChain**, voici la commande OpenSSL que nous exécutons :

```
openssl verify -verbose -show_chain -CAfile /etc/ipa/ca.crt [fichier cert]
```

IPARAAgent

Ce test compare le certificat sur disque avec l'enregistrement équivalent dans LDAP à l'adresse **uid=ipara,ou=People,o=ipaca**.

IPACertRevocation

Ce test utilise certmonger pour vérifier que les certificats n'ont pas été révoqués. Par conséquent, le test peut détecter les problèmes liés aux certificats gérés par certmonger uniquement.

IPACertmongerCA

Ce test permet de vérifier la configuration de l'autorité de certification (AC) de certmonger. L'IdM ne peut pas délivrer de certificats sans autorité de certification.
Certmonger gère un ensemble d'aides d'AC. Dans IdM, il existe une autorité de certification nommée IPA qui délivre des certificats par l'intermédiaire d'IdM, en s'authentifiant en tant qu'hôte ou utilisateur principal, pour des certificats d'hôte ou de service.

Il y a aussi **dogtag-ipa-ca-renew-agent** et **dogtag-ipa-ca-renew-agent-reuse** qui renouvellent les certificats du sous-système de l'autorité de certification.

NOTE

Exécutez ces tests sur tous les serveurs IdM lorsque vous essayez de vérifier s'il y a des problèmes.

22.2. CERTIFICATS DE DÉPISTAGE À L'AIDE DE L'OUTIL HEALTHCHECK

Cette section décrit un test manuel autonome du contrôle de santé d'un certificat de gestion d'identité (IdM) à l'aide de l'outil Healthcheck.

L'outil Healthcheck comprend de nombreux tests, ce qui vous permet d'abréger les résultats :

- à l'exclusion de tout test réussi : **--failures-only**
- ne comprenant que des tests de certificats : **--source=ipahealthcheck.ipa.certs**

Conditions préalables

- Vous devez effectuer les tests Healthcheck en tant qu'utilisateur **root**.

Procédure

- Pour exécuter le contrôle de santé avec les avertissements, les erreurs et les problèmes critiques concernant les certificats, entrez :

```
# ipa-healthcheck --source=ipahealthcheck.ipa.certs --failures-only
```

Un test réussi affiche des parenthèses vides :

```
[]
```

L'échec du test se traduit par la sortie suivante :

```
{
  "source": "ipahealthcheck.ipa.certs",
  "check": "IPACertfileExpirationCheck",
  "result": "ERROR",
  "kw": {
    "key": 1234,
    "dbdir": "/path/to/nssdb",
    "error": [error],
    "msg": "Unable to open NSS database '/path/to/nssdb': [error]"
  }
}
```

Ce test **IPACertfileExpirationCheck** a échoué lors de l'ouverture de la base de données NSS.

Ressources supplémentaires

- Voir **man ipa-healthcheck**.

CHAPITRE 23. VÉRIFICATION DES CERTIFICATS SYSTÈME À L'AIDE DE IDM HEALTHCHECK

Cette section décrit un outil de contrôle de santé dans la gestion des identités (IdM) qui permet d'identifier les problèmes liés aux certificats du système.

For details, see

Contrôle de santé dans l'IdM.

23.1. CERTIFICATS DE SYSTÈME TESTS DE CONTRÔLE DE SANTÉ

L'outil Healthcheck comprend plusieurs tests de vérification des certificats du système (DogTag).

Pour voir tous les tests, exécutez le programme **ipa-healthcheck** avec l'option **--list-sources**:

```
# ipa-healthcheck --list-sources
```

Vous trouverez tous les tests sous la source **ipahealthcheck.dogtag.ca**:

DogtagCertsConfigCheck

Ce test compare les certificats de l'autorité de certification (CA) dans sa base de données NSS aux mêmes valeurs stockées dans **CS.cfg**. S'ils ne correspondent pas, l'autorité de certification ne démarre pas.
Plus précisément, il vérifie :

- **auditSigningCert cert-pki-ca** contre **ca.audit_signing.cert**
- **ocspSigningCert cert-pki-ca** contre **ca.ocsp_signing.cert**
- **caSigningCert cert-pki-ca** contre **ca.signing.cert**
- **subsystemCert cert-pki-ca** contre **ca.subsystem.cert**
- **Server-Cert cert-pki-ca** contre **ca.sslserver.cert**

Si Key Recovery Authority (KRA) est installé :

- **transportCert cert-pki-kra** contre **ca.connector.KRA.transportCert**

DogtagCertsConnectivityCheck (contrôle de connectivité)

Ce test vérifie la connectivité. Ce test est équivalent à la commande **ipa cert-show 1** qui vérifie :

- La configuration du proxy PKI dans Apache
- IdM capable de trouver une autorité de certification
- Le certificat du client de l'agent RA
- Exactitude des réponses de l'AC aux demandes

Notez que le test vérifie un certificat avec le numéro de série #1 parce que vous voulez vérifier qu'un **cert-show** peut être exécuté et obtenir un résultat attendu de la part de l'autorité de certification (soit le certificat, soit un résultat non trouvé).

NOTE

Exécutez ces tests sur tous les serveurs IdM lorsque vous essayez de trouver un problème.

23.2. CONTRÔLE DES CERTIFICATS DE SYSTÈME À L'AIDE DE HEALTHCHECK

Cette section décrit un test manuel autonome des certificats de gestion d'identité (IdM) à l'aide de l'outil Healthcheck.

Comme l'outil Healthcheck comprend de nombreux tests, vous pouvez restreindre les résultats en n'incluant que les tests DogTag : **--source=ipahealthcheck.dogtag.ca**

Procédure

- Pour lancer un contrôle de santé limité aux certificats DogTag, entrez :

```
# ipa-healthcheck --source=ipahealthcheck.dogtag.ca
```

Un exemple de test réussi :

```
{
  "source: ipahealthcheck.dogtag.ca",
  "check: DogtagCertsConfigCheck",
  "result: SUCCESS",
  "uuid: 9b366200-9ec8-4bd9-bb5e-9a280c803a9c",
  "when: 20191008135826Z",
  "duration: 0.252280",
  "kw:" {
    "key": "Server-Cert cert-pki-ca",
    "configfile":  "/var/lib/pki/pki-tomcat/conf/ca/CS.cfg"
    }
}
```

Exemple d'échec d'un test :

```
{
  "source: ipahealthcheck.dogtag.ca",
  "check: DogtagCertsConfigCheck",
  "result: CRITICAL",
  "uuid: 59d66200-1447-4b3b-be01-89810c803a98",
  "when: 20191008135912Z",
  "duration: 0.002022",
  "kw:" {
    "exception": "NSDB /etc/pki/pki-tomcat/alias not initialized",
    }
}
```

Ressources supplémentaires

- Voir **man ipa-healthcheck**.