Comprendre les exigences des certificats d'accès mobile et distant et l'historique ATS

Table des matières

Introduction

Ce document décrit les exigences de téléchargement de certificat sur CUCM pour l'accès mobile et à distance.

Informations générales

Cisco Expressway utilise le serveur de trafic Apache (ATS). Le serveur de trafic est un composant très important dans les solutions de traversée, principalement utilisé pour les fonctionnalités suivantes :

- Vérification du certificat : Il vérifie les certificats des noeuds de serveur Cisco Unified Communications Manager (CUCM), IM & Presence et Unity pour les services MRA.
- Proxy et mise en cache : Il agit comme un serveur proxy de mise en cache rapide et évolutif pour le trafic HTTP/HTTPS.

Sur Expressway Version 14.0.2

Le serveur de trafic (ATS) commence à voir une légère application de la « vérification de certificat » lorsqu'il communique avec CUCM pendant le provisionnement MRA.

La condition requise a été documentée sous CSCvz45074 où les certificats racine qui ont signé les certificats de serveur Expressway Core doivent être téléchargés sur CUCM en tant que Tomcat-Trust et Callmanager Trust : https://cdetsng.cisco.com/summary/#/defect/CSCvz45074.

- Traffic Server Applique La Vérification De Certificat.

- Avant de procéder à la mise à niveau vers la version X14.0.2, assurez-vous que cette condition de certificat est remplie.

Exigence - La chaîne de l'autorité de certification (CA) (racine + intermédiaire) qui a signé le certificat Expressway-C doit être ajoutée à la liste tomcat-trust et CallManager-trust de CUCM, même si Unified Communications Manager (UCM) est en mode non sécurisé.

Raison : le service de serveur de trafic dans Expressway envoie son certificat chaque fois qu'un serveur UCM le demande. Ces requêtes concernent des services exécutés sur des ports autres que 8443 (par exemple, les ports 6971, 6972, etc.). Cela permet d'appliquer la vérification de certificat même si UCM est en mode non sécurisé. Pour plus d'informations, consultez le Guide de déploiement d'un accès mobile et distant via Expressway.

Comportement sur les versions antérieures à 14.0.8

Le serveur de trafic sur Expressway-C qui gère les connexions bidirectionnelles HTTPS sécurisées entre Expressway-C et les noeuds de communications unifiées n'a pas vérifié le certificat qui a été présenté par l'extrémité distante. Dans la configuration MRA, il y a une option pour avoir la vérification du certificat TLS par la configuration du mode de vérification TLS sur 'On' quand soit CUCM, IM&P, ou les serveurs Unity sont ajoutés sous Configuration > Unified Communications > Unified CM servers/IM and Presence Service nodes/Unity Connection servers. L'option de configuration est affichée dans la capture d'écran suivante, qui indique qu'elle vérifie le nom de domaine complet ou l'adresse IP dans le SAN, ainsi que la validité du certificat et si celui-ci est signé par une autorité de certification de confiance.

Il y avait également un problème connu où deux certificats portant le même nom CN ne peuvent pas être chargés sur le magasin de confiance d'Expressway. Cette limitation a causé deux problèmes :

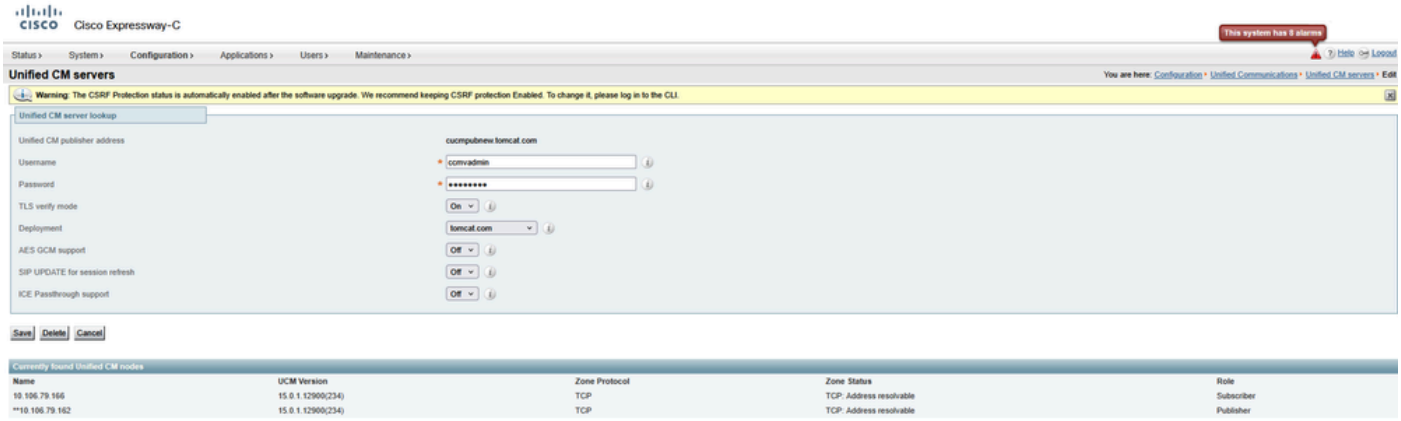
1. Si vous choisissez de charger le certificat du gestionnaire d'appels sur le magasin Expressway Trust, TLS verify 'On' échouera lors de l'ajout de CUCM.

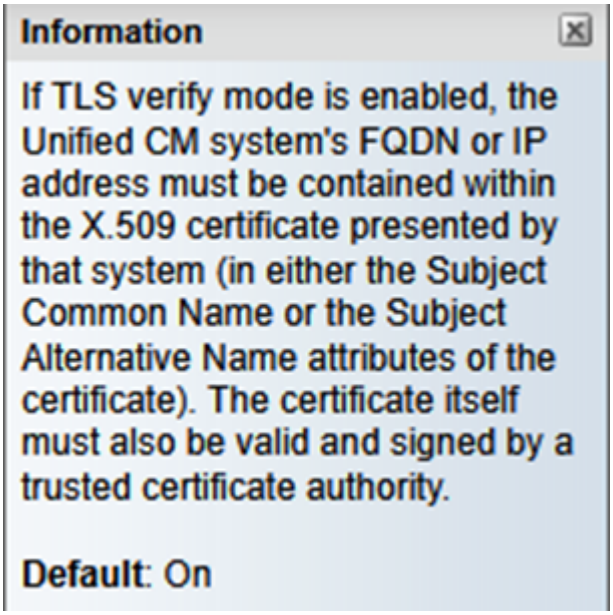
2: Si vous avez choisi de charger le certificat Tomcat sur le magasin Expressway Trust, les enregistrements SIP sécurisés sur 5061 échoueront.

Ce comportement est documenté dans CSCwa12894.

En outre, cette vérification de certificat TLS n'est effectuée qu'au moment de la découverte des serveurs CUCM/IM&P/Unity et non lors de la mise en service du client MRA.

L'inconvénient de cette configuration est qu'elle ne la vérifie que pour l'adresse d'éditeur que vous ajoutez. Il ne vérifie pas si le certificat sur les noeuds d'abonné a été correctement configuré lorsqu'il récupère les informations de noeud d'abonné (FQDN ou IP) à partir de la base de données du noeud éditeur.





Comportement sur les versions 14.0.8 et ultérieures

À partir de la version X14.0.8, le serveur Expressway procède à la vérification du certificat TLS pour chaque requête HTTPS effectuée via le serveur de trafic. Cela signifie qu'il effectue également cette opération lorsque le mode de vérification TLS est défini sur « Désactivé » lors de la détection des noeuds CUCM/IM&P/Unity. Lorsque la vérification échoue, la connexion TLS ne se termine pas et la demande échoue, ce qui peut entraîner une perte de fonctionnalité, comme la redondance, des problèmes de basculement ou des échecs de connexion complets, par exemple. De plus, si le mode de vérification TLS est activé, cela ne garantit pas que toutes les connexions fonctionnent correctement, comme indiqué dans l'exemple ci-après.

Les certificats exacts que l'Expressway vérifie vers les noeuds CUCM/IM&P/Unity sont comme indiqué dans la section du guide d'ARM.

https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/expressway/config_guide/X15-0/mra/exwy_b_mra-deployment-guide-x150.pdf

Profilé

Conditions requises pour le certificat > Conditions requises pour l'échange de certificats

En raison de ces changements dans la façon dont la communication s'effectue entre Expressway-Core et CUCM, il faut s'assurer que :

1. Il est recommandé d'utiliser des certificats signés par une autorité de certification pour l'accès mobile et distant.

2. Chaque cluster Unified CM doit faire confiance au certificat Expressway-C. Pour chaque grappe, assurez-vous que :

- Si le mode mixte est activé — Le certificat Expressway-C doit être installé dans le magasin CallManager-trust et Tomcat-trust sur Unified CM.
- Si le mode mixte est désactivé — Le certificat de l'autorité de certification racine qui signe le certificat Expressway-C doit être installé dans le magasin CallManager-trust et Tomcat-trust sur Unified CM. Ensuite, redémarrez les éléments suivants : · Service Tomcat · Service CallManager · Service proxy HA (si vous utilisez TLS sur Tomcat).

Sur Expressway - Core, assurez-vous que ces mesures sont prises :

- Expressway-C doit faire confiance aux certificats présentés par chaque cluster Unified CM et IM and Presence Service.

Le magasin de confiance d'Expressway-C doit inclure le certificat CA racine qui signe les certificats Unified CM et IM and Presence Service pour tous les clusters UC.

Remarque : Assurez-vous d'ajouter tous les certificats d'autorité de certification racine et intermédiaire ou la chaîne d'autorité de certification complète utilisés pour signer le certificat Expressway-C à la liste de confiance Tomcat et CallManager de Cisco Unified Communications Manager (UCM), même si l'UCM fonctionne en mode non sécurisé.

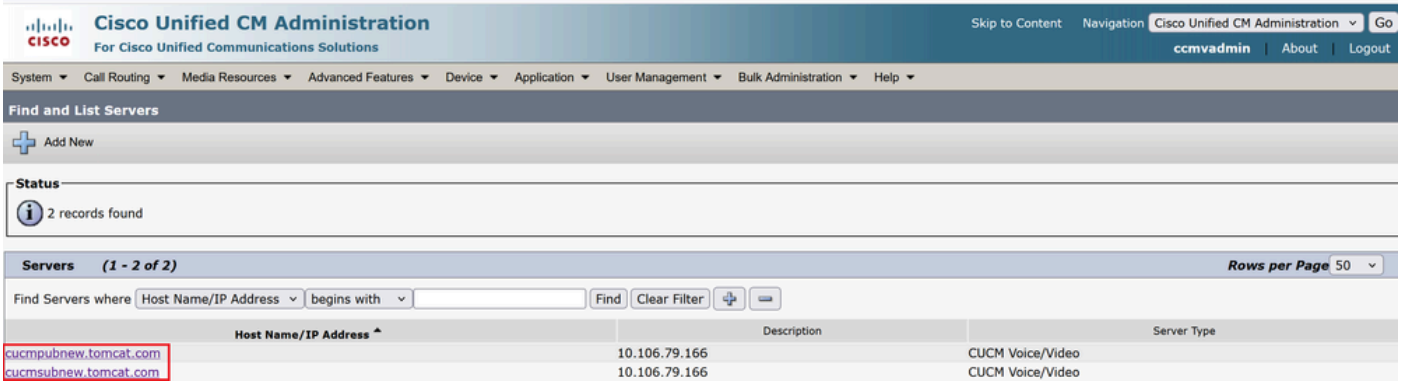
Raison : le service de serveur de trafic dans Expressway envoie son certificat chaque fois qu'un serveur (UCM) le demande. Ces requêtes concernent des services exécutés sur des ports autres que 8443 (par exemple, les ports 6971, 6972, etc.). Cela permet d'appliquer la vérification de certificat même si UCM est en mode non sécurisé.

La façon dont l'adresse CUCM est ajoutée sous System > Server joue un rôle très important dans l'ajout de CUCM/IMP sur Expressway core sous Configuration > Unified Communications > Unified CM servers/IM and Presence Service nodes.

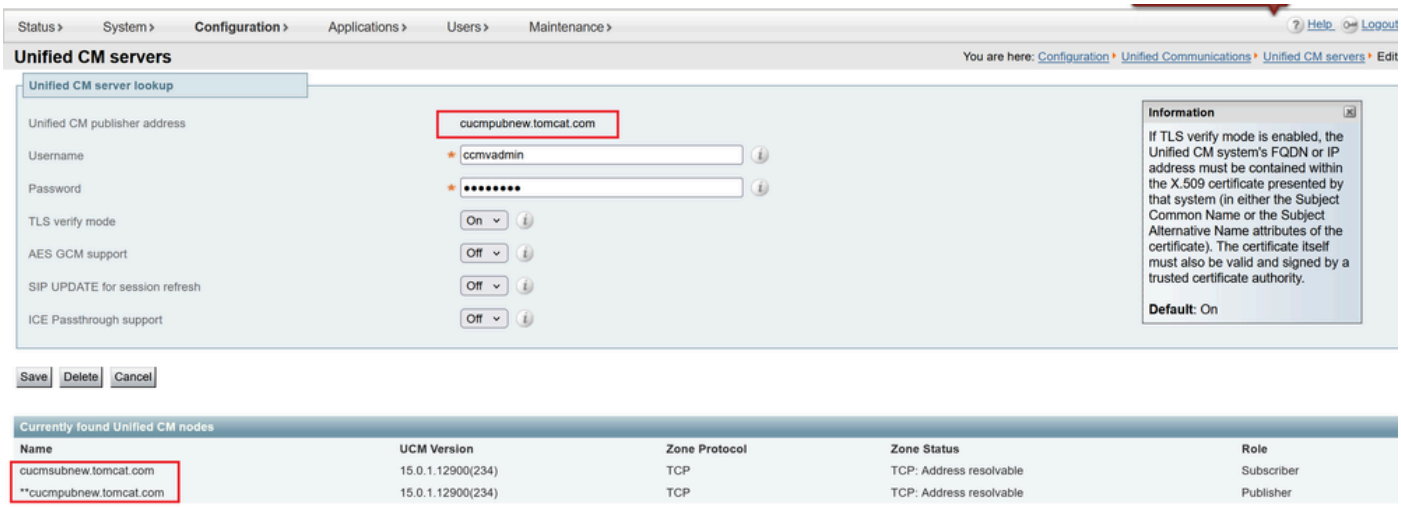
CUCM doit toujours être ajouté avec un nom de domaine complet et non un nom d'hôte ou une adresse IP. S'il est vu que CUCM est ajouté sous System > Server comme nom d'hôte/adresse IP

pendant la connexion TLS, la vérification TLS 'On' échouera et le cluster CUCM ne sera pas ajouté sur Expressway-Core.

Cette figure montre CUCM ajouté en tant que nom d'hôte :



Cette figure montre CUCM ajouté sur Expressway-Core avec FQDN avec TLS verify Mode = ON :



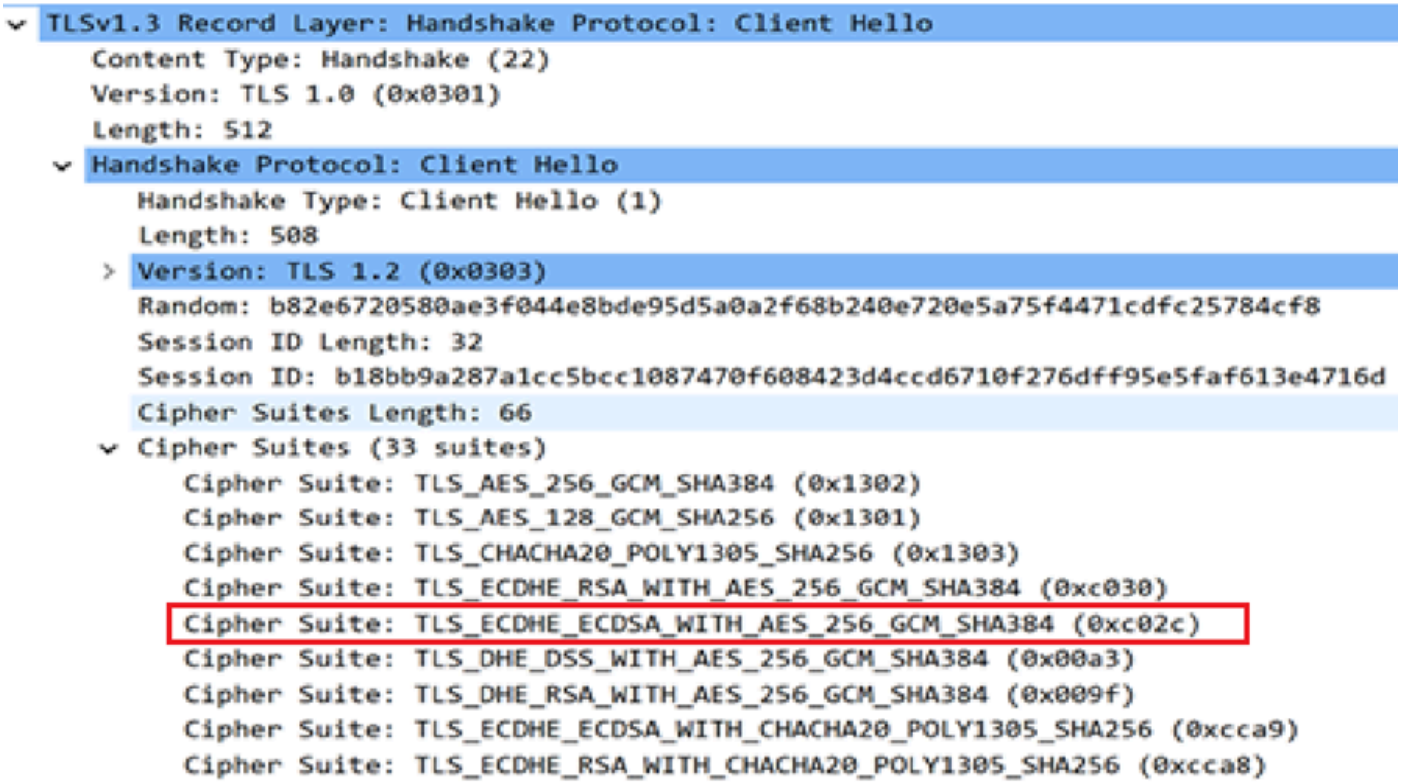
Il y a également eu un changement introduit dans X14.2 qui présentera les chiffrements lors d'une connexion TLS (client hello) dans un ordre de préférence différent. Cela dépendait du chemin de mise à niveau et provoquait des connexions TLS inattendues après une mise à niveau logicielle. Il se peut qu'avant la mise à niveau pendant la connexion TLS, il ait demandé le certificat Cisco Tomcat ou Cisco CallManager de CUCM. Mais qu'après la mise à niveau, il a demandé la variante ECDSA (qui est la variante de chiffrement plus sécurisée que RSA). Les certificats Cisco Tomcat-ECDSA ou Cisco CallManager-ECDSA peuvent être signés par une autre autorité de certification ou simplement par des certificats auto-signés (par défaut).

Cette modification de l'ordre de préférence de chiffrement n'est pas toujours pertinente pour vous, car elle dépend du chemin de mise à niveau indiqué dans les notes de version d'Expressway X14.2.1. En bref, vous pouvez voir à partir de Maintenance > Security > Ciphers pour chacune des listes de chiffrement si elle ne précède pas ECDHE-RSA-AES256-GCM-SHA384 ou non. Si ce n'est pas le cas, il préfère le chiffrement ECDSA plus récent au chiffrement RSA. Si c'est le cas, vous avez le comportement précédent avec RSA qui a la préférence la plus élevée.

La capture d'écran suivante montre dans la zone rouge le chiffrement ECDSA annoncé par le coeur d'Expressway pendant le message de négociation TLS dans le Hello du client, #IF TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 est choisi par le répondeur distant (CUCM) dans le Hello du serveur, alors la négociation TLS échouera si :

Les certificats d'autorité de certification racine ou les certificats ECDSA réels du répondeur, c'est-

à-dire que CUCM n'est pas installé dans le magasin Expressway Trust dans ce cas.



Vous pouvez également modifier les Chiffres Expressway pour que l'ECDSA ne soit pas prioritaire.

1. Modifiez le chiffrement SIP en ajoutant la chaîne SSL ouverte GCM-Sha384.

"ECDHE-RSA-AES256-GCM-SHA384:EECDH:EDH:HIGH:.....:!MD5:!PSK:!eNULL:!aNULL:!aDH"

2. Ajoutez + afin de déplacer le chiffre à la dernière préférence ou ajoutez ! afin de désactiver ECDSA de façon permanente.

Chiffrement : "EECDH:EDH:HIGH:-AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!eNULL:!aNULL:!aDH:+ECDSA"

3. Ajoutez le certificat CA racine et intermédiaire qui a signé le certificat ECDSA sur CUCM ou ajoutez le certificat Tomcat-ECDSA sur le magasin de confiance Expressway (dans certains cas).

Cependant, en raison de la modification de la priorité de chiffrement, après la mise à niveau, les déploiements MRA peuvent échouer. Le TAC devra donc effectuer la solution de contournement mentionnée précédemment pour que les choses fonctionnent à nouveau.

Avec l'introduction de TLS 1.3, il devient encore plus difficile de vérifier quels certificats sont échangés dans wireshark.

Comportement sur les versions x15.3

Pour l'interface SIP uniquement, vous pouvez choisir d'avoir des chiffrements RSA ou ECDSA.

Avec X15.x, TLS 1.3 a été appliqué. Comme on le voit sur le terrain, l'algorithme RSA est choisi principalement sur ECDSA. Les clients qui effectuent la mise à niveau vers x15.2 peuvent désormais choisir

entre RSA et l'algorithme ECDSA avec cet ensemble de commandes :

xConfiguration SIP Advanced TlsSignatureAlgoPrefRsa : Activé/Désactivé

TlssignatureAlgoPrefRSA ne fonctionnera que si l'interface SIP a TLS 1.3

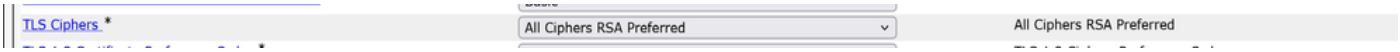
xConfiguration SIP Advanced SipTlsVersions : « TLSv1.3 »

Remarque : Cette option n'est éligible pour l'interface SIP qu'à partir de maintenant. Les considérations relatives à Traffic Server et à Tomcat sur le 8443 restent inchangées, comme indiqué précédemment.

Les combinaisons de chiffrement envoyées par Expressway à CUCM au cours du « bonjour client » seront comme indiqué lorsque RSA est choisi.

- Algorithme de signature : rsa_pss_rsae_sha512 (0x0806)
- Algorithme de signature : rsa_pss_rsae_sha384 (0x0805)
- Algorithme de signature : rsa_pss_rsae_sha256 (0x0804)
- Algorithme de signature : ecdsa_secp521r1_sha512 (0x0603)
- Algorithme de signature : ecdsa_secp384r1_sha384 (0x0503)
- Algorithme de signature : ecdsa_secp256r1_sha256 (0x0403)

La configuration précédente fonctionnera en tandem sur la configuration que vous avez choisie sur les chiffrements CUCM à TLS sous Enterprise Parameters > Security Parameters.



De plus, il est important de noter que lors d'une connexion TLS rompue sur TLS 1.3 entre Expressway-C et CUCM, les erreurs imprimées dans les journaux de diagnostic ou PCAP ne sont pas très utiles. Il vaut la peine d'activer ces débogages tout en travaillant avec le TAC, de sorte que le composant imprime des erreurs claires à dépanner.

xConfiguration Logger Développeur développeur.trafficServer.http Niveau : "DÉBOGAGE"
xConfiguration Logger Développeur développeur.traffic_server.http_trans Niveau : "DÉBOGAGE"
xConfiguration Logger Développeur developer.traffic.server.iocore Niveau : "DÉBOGAGE"
xConfiguration Logger Développeur developer.traffic.ssl Niveau : "DÉBOGAGE"

À quoi s'attendre lorsque Callmanager partage un certificat avec plusieurs services

Les choses changent légèrement avec la réutilisation du certificat sur CUCM.

À partir de CUCM 14.0, vous pouvez réutiliser les certificats ECDSA Tomcat et Tomcat en tant

que Call manager et Call manager ECDSA.

Le certificat Tomcat peut être réutilisé en tant que certificat Callmanager.

Le certificat Tomcat-ECDSA peut être réutilisé en tant que certificat Callmanager-ECDSA.

Ça rend la vie facile.

1. Plusieurs services sur CUCM utilisent désormais un seul certificat, ce qui réduit le coût du certificat.

2. Moins de gestion des certificats.

3. Si vous devez télécharger le certificat Tomcat/Callmanager ou Tomcat-ECDSA/Callmanager-ECDSA (pour une raison quelconque) sur le magasin de confiance Expressway-Core, il s'agira d'un seul certificat que vous devez télécharger. Il n'y aura pas de problème d'avoir le même problème de nom CN (mentionné plus tôt dans ce document).

Remarque : La réutilisation du certificat n'aura lieu que lorsque Tomcat et Tomcat-ECDSA sont des certificats multi-san.

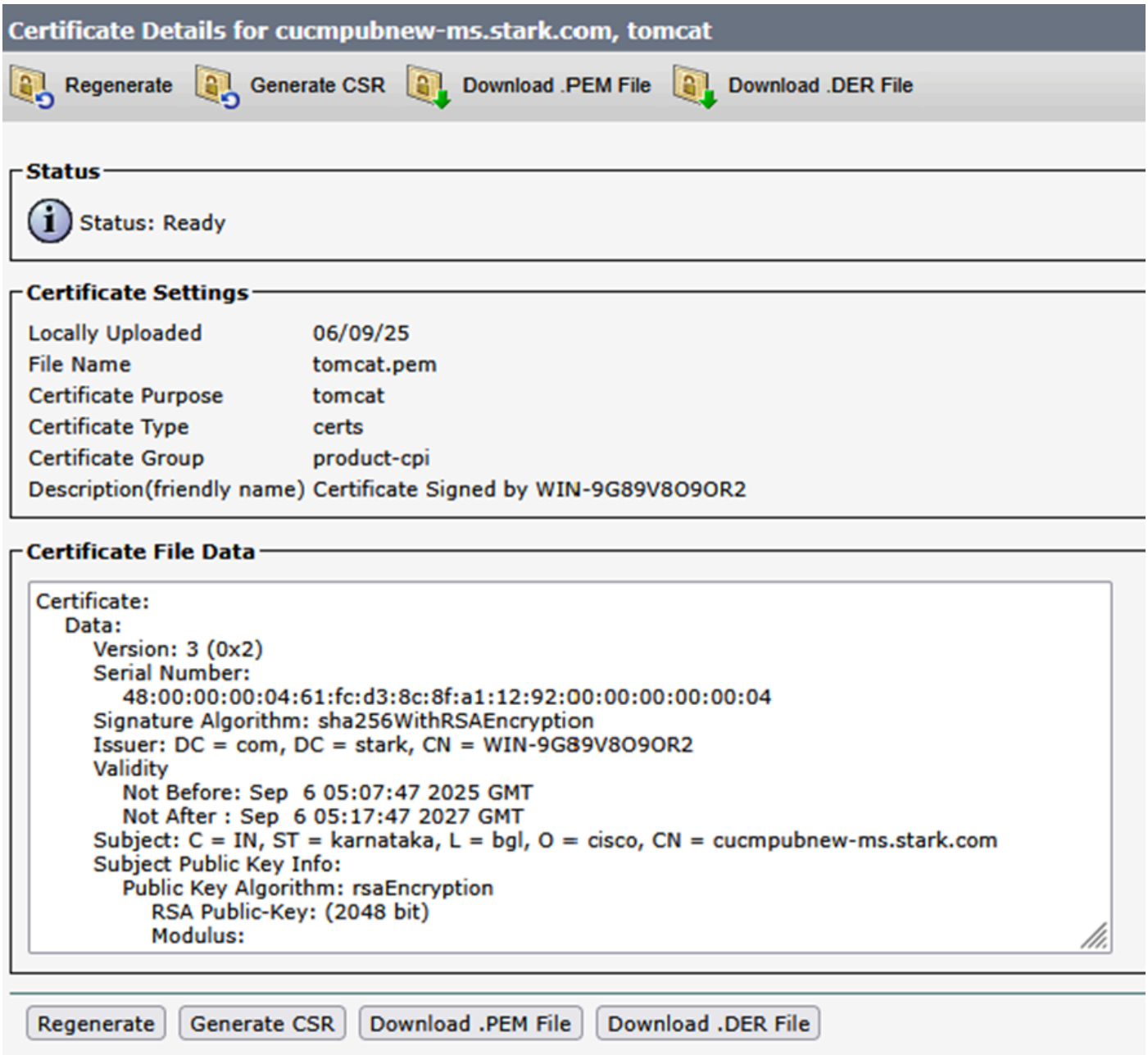
Les certificats de serveur ECDSA Post Reuse, Callmanager et Callmanager ne sont pas visibles sur le magasin d'approbation CUCM. Vous pouvez valider la réutilisation des certificats à partir de l'interface CLI en exécutant les commandes suivantes :

show cert own CallManager

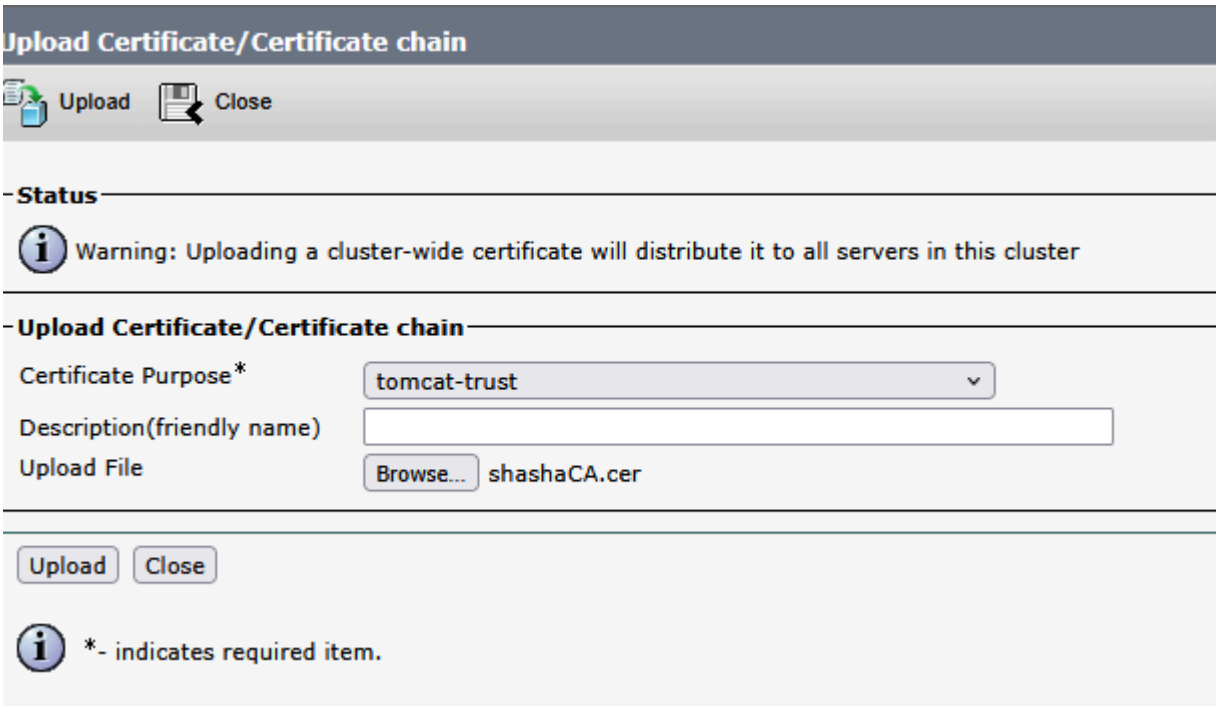
show cert own tomcat

Étapes de réutilisation du certificat

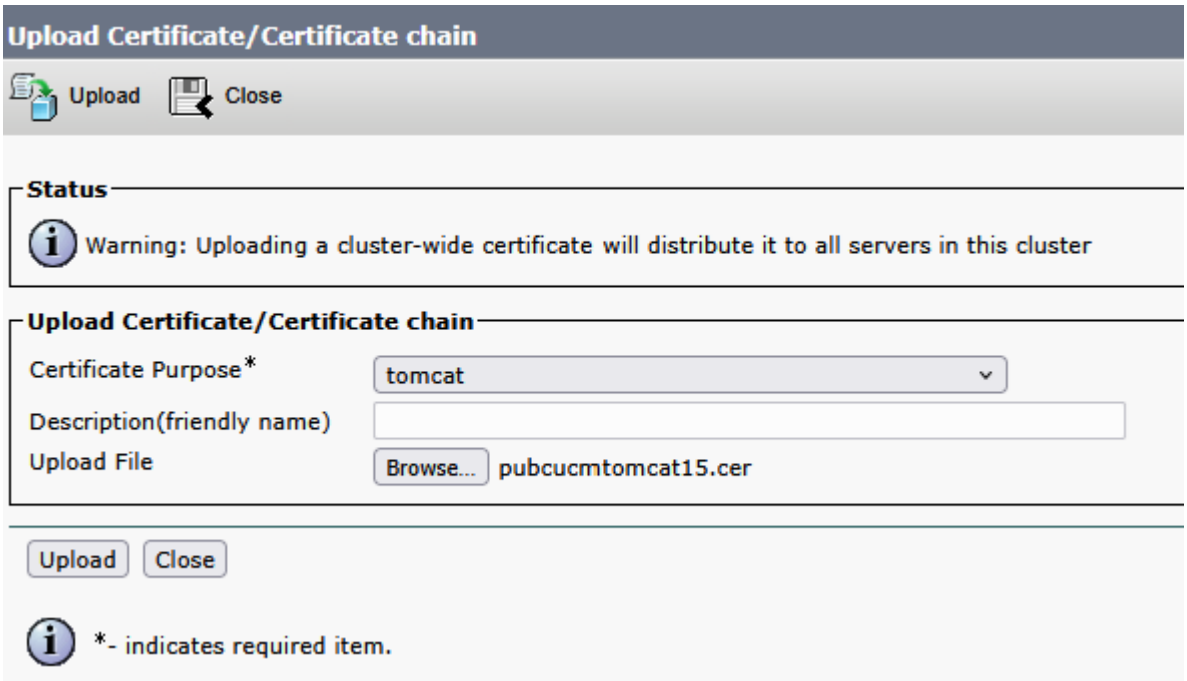
Génération de l'ajout pub CSR Tomcat.



Téléchargez le certificat CA qui signera le certificat Tomcat sur CUCM en tant que Tomcat-trust.



Une fois le certificat Tomcat signé, téléchargez-le sur l'éditeur. Redémarrez les services appropriés lorsque vous y êtes invité.



Une fois le certificat Tomcat signé, téléchargez-le sur l'éditeur. Redémarrez les services appropriés lorsque vous y êtes invité.

Réussite : Certificat téléchargé. Effectuez une sauvegarde de reprise après sinistre pour que la dernière sauvegarde contienne le certificat téléchargé.
Redémarrez le service Web Cisco Tomcat à l'aide de l'interface de ligne de commande « utils service restart Cisco Tomcat » sur tous les noeuds de cluster (UCM/IMP). Redémarrez les

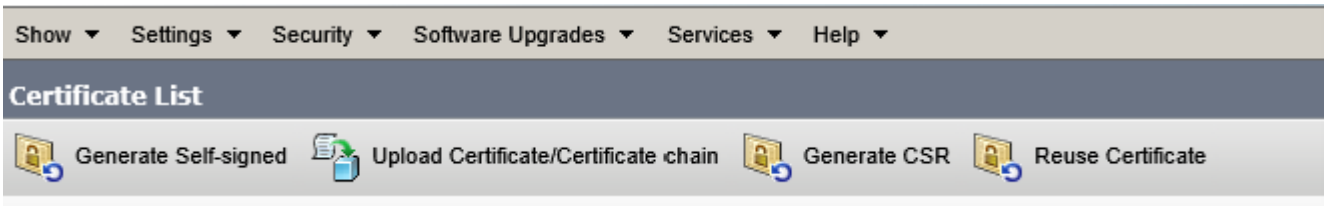
services Web Cisco UDS Tomcat et Cisco AXL Tomcat à l'aide de l'interface de ligne de commande « utils service restart Cisco UDS Tomcat and utils service restart Cisco AXL Tomcat » sur tous les noeuds de cluster UCM. Redémarrez également les services Cisco DRF Master et Cisco DRF Local sur le noeud éditeur. Redémarrez uniquement le service local DRF Cisco sur le ou les noeuds de l'abonné.

Le certificat Tomcat est maintenant signé par l'autorité de certification.

tomcat	cucmpubnew-ms.stark.com_51dc40f400000000000b	Identity	CA-signed	RSA	Multi-server(SAN)	RICKY200-TMS-CA	10/23/2027Certificate Signed by RICKY200-TMS-CA

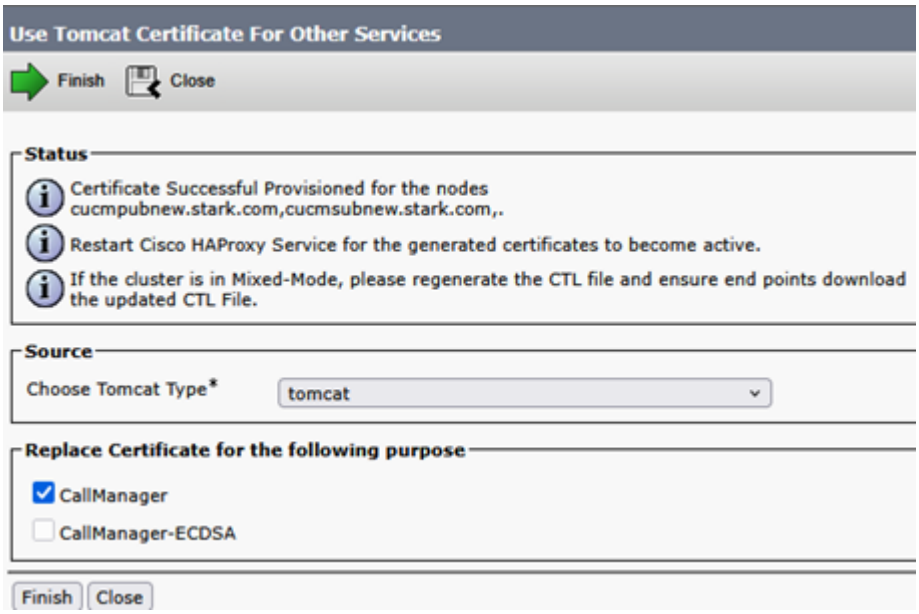
Afin de réutiliser le certificat Tomcat comme certificat Callmanager maintenant.

Cliquez sur Réutiliser le certificat.



Choisissez Tomcat dans la liste déroulante et cochez la case Certificat Callmanager.

Cliquez sur Terminer.



Le certificat Tomcat est maintenant réutilisé en tant que certificat Callmanager. Vous pouvez le valider à partir de la CLI.

Numéro de série (SN) du certificat Callmanager : 56:ff:6c:71:00:00:00:00:00:0d

```
admin:show cert own CallManager
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            56:ff:6c:71:00:00:00:00:00:0d
        Signature Algorithm: sha384WithRSAEncryption
        Issuer: DC = com, DC = RICKY200, CN = RICKY200-TMS-CA
        Validity
            Not Before: Oct 24 08:44:34 2025 GMT
            Not After : Oct 24 08:54:34 2027 GMT
        Subject: C = IN, ST = karnataka, L = bgl, O = cisco, CN = cucmpubnew-ms.
tomcat.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (2048 bit)
                Modulus:
                    00:b4:a6:fa:8f:9a:c3:32:02:74:fa:e9:92:30:de:
                    6e:3b:70:cd:d7:4e:64:e4:71:04:fe:17:80:0d:5b:
                    44:d1:7f:00:63:69:4a:5c:1a:1b:75:0c:1a:d6:ce:
                    10:3f:01:e2:d0:f1:75:33:57:b7:0a:71:e1:60:d1:
                    89:3c:e8:a4:8c:3e:30:69:4d:4e:98:da:b8:5d:dd:
                    23:8c:4d:69:90:69:9d:43:74:84:20:a8:9f:45:dc:
                    5a:aa:7b:c8:d1:d0:6f:05:13:d8:99:58:0e:49:7b:
Press <enter> for 1 line, <space> for one page, or <q> to quit
```

Numéro de certificat Tomcat : 56:ff:6c:71:00:00:00:00:00:0d

```
admin:show cert own tomcat
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            56:ff:6c:71:00:00:00:00:00:0d
        Signature Algorithm: sha384WithRSAEncryption
        Issuer: DC = com, DC = RICKY200, CN = RICKY200-TMS-CA
        Validity
            Not Before: Oct 24 08:44:34 2025 GMT
            Not After : Oct 24 08:54:34 2027 GMT
        Subject: C = IN, ST = karnataka, L = bgl, O = cisco, CN = cucmpubnew-ms.tomcat.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (2048 bit)
                Modulus:
                    00:b4:a6:fa:8f:9a:c3:32:02:74:fa:e9:92:30:de:
                    6e:3b:70:cd:d7:4e:64:e4:71:04:fe:17:80:0d:5b:
                    44:d1:7f:00:63:69:4a:5c:1a:1b:75:0c:1a:d6:ce:
                    10:3f:01:e2:d0:f1:75:33:57:b7:0a:71:e1:60:d1:
                    89:3c:e8:a4:8c:3e:30:69:4d:4e:98:da:b8:5d:dd:
                    23:8c:4d:69:90:69:9d:43:74:84:20:a8:9f:45:dc:
                    5a:aa:7b:c8:d1:d0:6f:05:13:d8:99:58:0e:49:7b:
Press <enter> for 1 line, <space> for one page, or <q> to quit
```

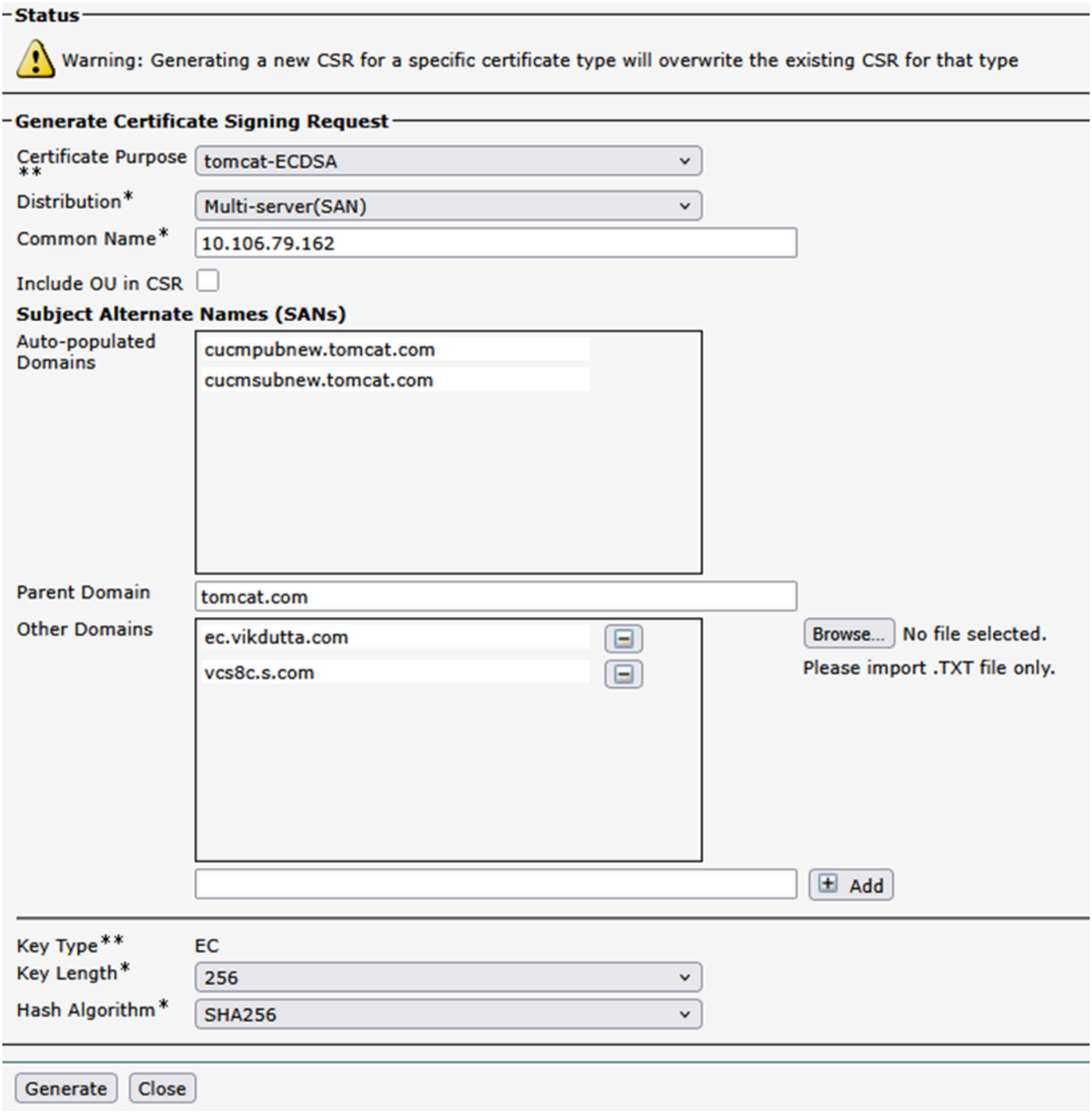
Effectuez les mêmes étapes sur l'Abonné.

Signons maintenant le certificat ECDSA afin qu'il puisse être réutilisé comme Callmanager-ECDSA.

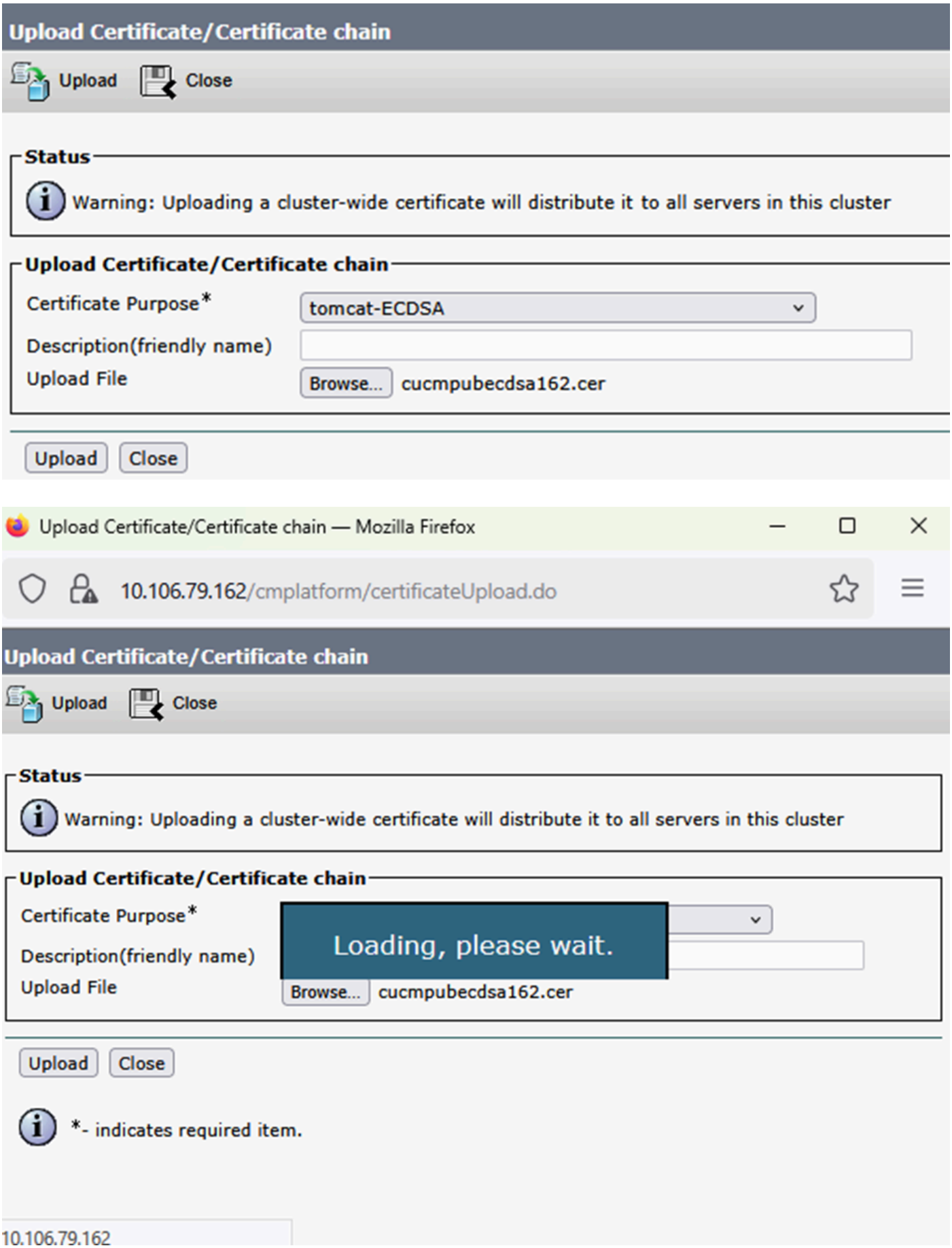
Le certificat Tomcat-ECDSA actuel est auto-signé.

tomcat	10.106.79.162_5a...f	Identity	CA-signed	RSA	Multi-server(SAN)	RICKY200-TMS-CA	10/25/2027	Certificate Signed by RICKY200-TMS-CA
tomcat-ECDSA	cucmpubnew-EC.tomcat.com_4b4...	Identity	self-signed	EC	cucmpubnew.tomcat.com	cucmpubnew-EC.tomcat.com	10/23/2030	Self-signed certificate generated by system

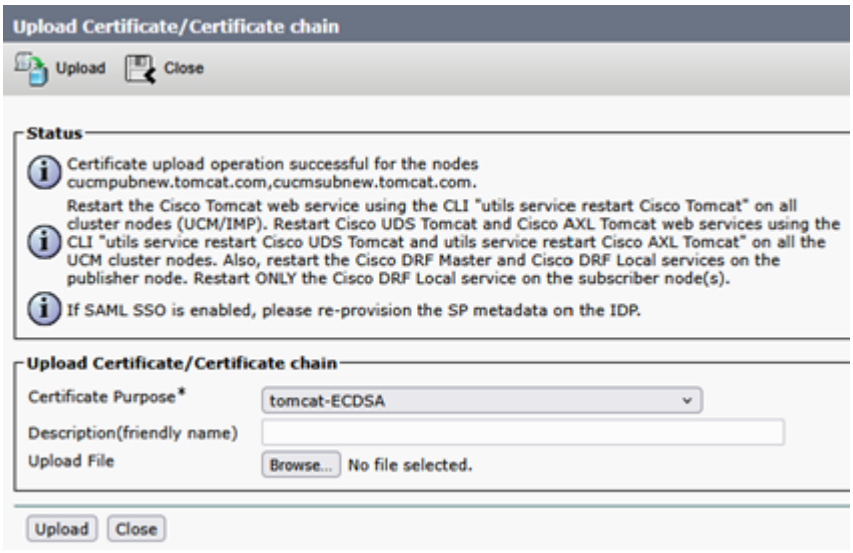
Signez un CSR multisan pour le certificat Tomcat-ECDSA.



Signez le certificat à l'aide de CSR et téléchargez.

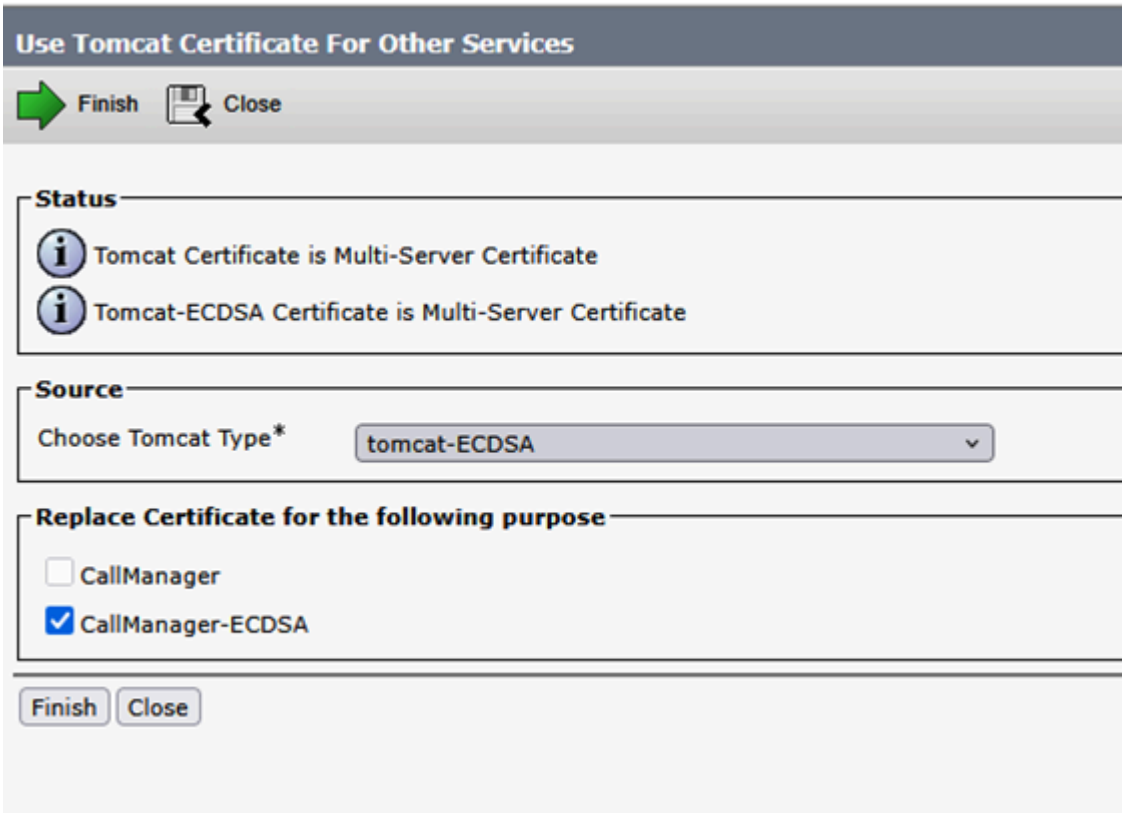


Téléchargement réussi. Redémarrez les services appropriés, comme demandé.

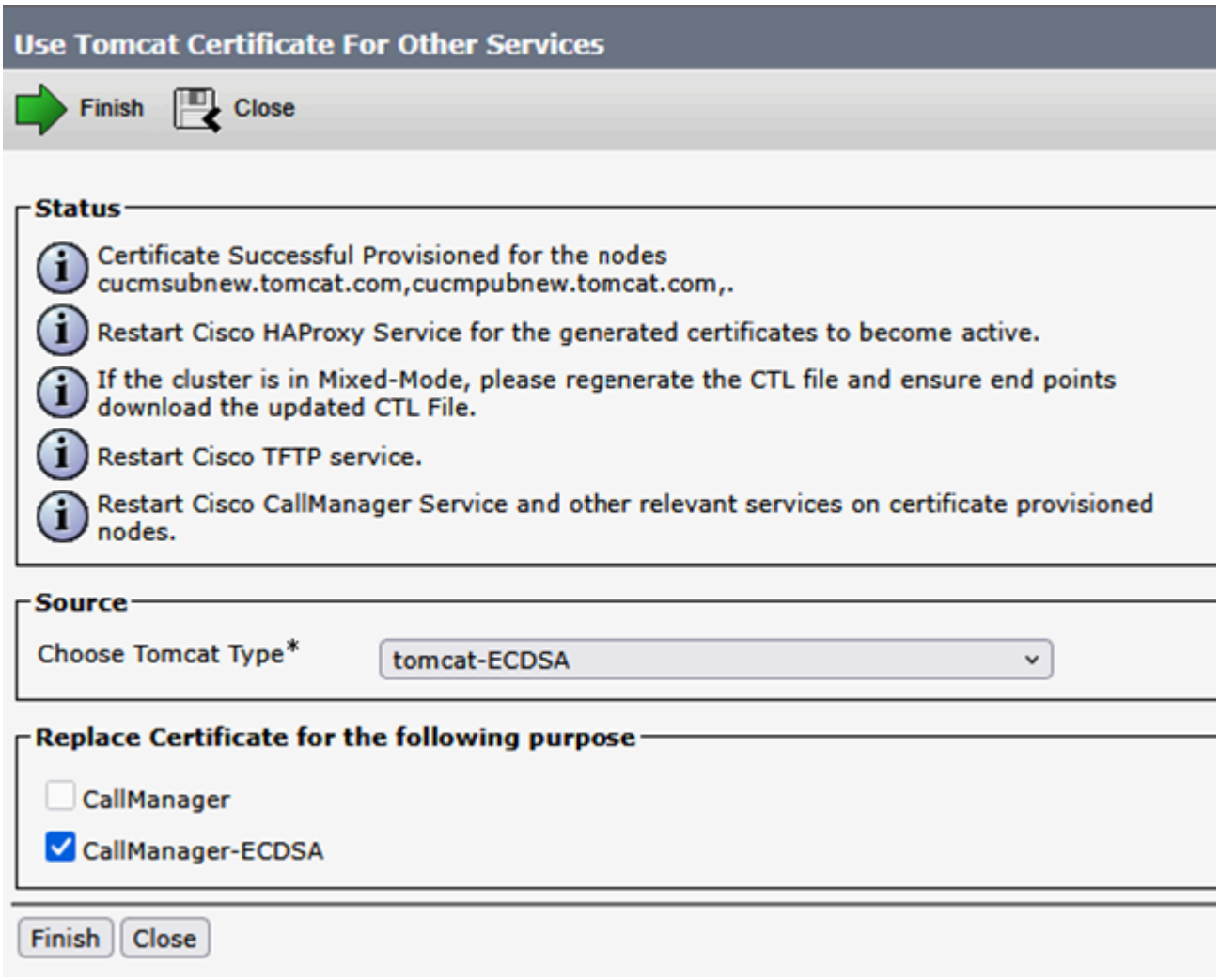


Tomcat et Tomcat-ECDSA signés par CA.

Réutilisez maintenant Tomcat-ECDSA comme certificat Callmanager-ECDSA.



Téléchargement réussi. Redémarrez les services appropriés lorsque vous y êtes invité.



Vérifiez les certificats à partir de CLI.

SN du certificat Callmanager-ECDSA :
2f:00:00:00:38:80:be:cc:a8:a1:8e:8f:23:00:00:00:00:00:00:38

```
admin:show cert own CallManager-ecdsa
Invalid Certificate Name. Certificate Not Found.

admin:show cert own CallManager-Ecdsa
Invalid Certificate Name. Certificate Not Found.

admin:show cert own tomcat-ECDSA
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            2f:00:00:00:38:80:be:cc:a8:a1:8e:8f:23:00:00:00:00:00:38
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: DC = com, DC = bgluclab, CN = bgluclab-WIN-DC-01-CA
        Validity
            Not Before: Oct 25 06:46:37 2025 GMT
            Not After : Oct 25 06:46:37 2026 GMT
```

SN du certificat Tomcat-ECDSA : 2f:00:00:00:38:80:be:cc:a8:a1:8e:8f:23:00:00:00:00:00:00:38.

```
admin:show cert own tomcat-ECDSA
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            2f:00:00:00:38:80:be:cc:a8:a1:8e:8f:23:00:00:00:00:00:38
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: DC = com, DC = bgluclab, CN = bgluclab-WIN-DC-01-CA
        Validity
            Not Before: Oct 25 06:46:37 2025 GMT
            Not After : Oct 25 06:46:37 2026 GMT
        Subject: C = IN, ST = karnataka, L = bgl, O = cisco, CN = cucmpubnew-EC-ms.tomcat.com
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (256 bit)
```

Puisque vous utilisez maintenant un certificat pour deux services, c'est-à-dire le certificat Tomcat pour les services Tomcat et Callmanager, et Tomcat-ECDSA pour les services Tomcat-ECDSA et Callmanager-ECDSA, il est devenu moins fastidieux de télécharger des certificats sur le magasin de confiance Expressway (si nécessaire).

Faire vérifier 'On' par TLS lors de l'ajout d'UCM sur expressway-core pour MRA, a été plus facile que jamais. Il suffit d'ajouter un certificat d'autorité de certification ou un certificat de serveur Tomcat pour effectuer le travail (car le certificat est maintenant partagé entre Callmanager et le service Tomcat).

Si une mise à niveau vers x14.2 ou une version ultérieure a provoqué une panne pour l'accès à distance mobile, vous pouvez également vous reporter à ce document complet pour résoudre le problème.

Historique des versions du serveur Apache Traffic Server

Afin de vérifier la version sur votre serveur, connectez-vous à root et exécutez ~ # /apache2/bin/httpd -v.

Expressway x8.11.4
Version du serveur : Apache/2.4.34 (Unix)
Serveur construit : 12 nov. 2018 19:04:23

Expressway x12.6

Version du serveur : Apache/2.4.43 (Unix)
Serveur construit : 26 mai 2020 18:27:21

Expressway x14.0.8

Version du serveur : Apache/2.4.53 (Unix)
Serveur construit : 4 mai 2022 08:52:57

Expressway x15.3

Version du serveur : Apache/2.4.62 (Unix)
Serveur construit : 16 juillet 2025 12:10:19

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.