Licentiecode voor implementatie en probleemoplossing: Grant Flow - Verbetering in handomdraai: Cisco Collaboration-oplossingen 12.0

Inhoud

Inleiding

Dit document beschrijft hoe de overdracht van een autorisatie-code op vernieuwingtoken is gebaseerd om Jabber User Experience op verschillende apparaten te verbeteren, met name voor Jabber op mobiel

Voorwaarden

Vereisten

Cisco raadt kennis van de volgende onderwerpen aan:

- Cisco Unified Communications Manager (CUCM) 12.0-versie
- Single Sign On (SSO)/SAML
- Cisco Jabber
- Microsoft ADFS
- Identity Provider (IDP)

Raadpleeg de volgende koppelingen voor meer informatie over deze onderwerpen:

- SAML SSO-implementatiegids voor Cisco Unified Communications
- Configuratievoorbeeld van Unified Communications Manager SAML:
- AD FS versie 2.0 Instellen voor SAML SSO Configuration Voorbeeld:

Gebruikte componenten

De informatie in dit document is gebaseerd op deze software:

- Microsoft ADFS (IDP)
- LDAP actieve map
- Cisco Jabber-client
- CUCM 12.0

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk levend is, zorg er dan voor dat u de mogelijke impact van om het even welke opdracht begrijpt.

Achtergrondinformatie

Vanaf vandaag is Jabber SSO-flow met infrastructuur gebaseerd op Impliciete Grant Flow waar de CUCM Authz-service de kortstondige toegangspenningen toewijst.

Na afloop van het toegangstoken stuurt CUCM Jabber terug naar IDP voor herverificatie.

Dit leidt tot een slechte gebruikerservaring, vooral met jabber op mobiel, waar de gebruiker wordt gevraagd om vaak aanmeldingsgegevens in te voeren.

Security herarchitectuuroplossing stelt ook vergunningscode Grant Flow voor (met gebruik van Refresh Tokens benadering (verlengbaar met End Point/Other Collaboration Apps)) voor de unificatie van Jabber en End Point-in-flow voor zowel SSO- als niet-SSO-scenario's.

Functiemarkeringslichten

- Licentiecode Grant Flow is gebaseerd op verfrissingstoken (verlengbaar met End-of-life punten/andere collaboration-apps) om Jabber gebruikerservaring op verschillende apparaten te verbeteren, met name voor Jabber op mobiel.
- Ondersteunt Slaat- en Versleutelde OAuth Tokens om verschillende samenwerkingstoepassingen toe te staan om verzoeken om middelen van client te valideren en te beantwoorden.
- Het impliciete subsidie flow-model wordt behouden, hetgeen compatibiliteit in een achterwaartse richting mogelijk maakt. Dit biedt ook een naadloos pad voor andere klanten (zoals RTMT) die niet zijn overgestapt op de vergunningscode Grant flow.

Belangrijke overwegingen

- Implementatie zodat de oude jabber-cliënt met het nieuwe CUCM kan werken (aangezien het zowel impliciete subsidie- als autorisatiecode toekenningsstromen ondersteunt). Ook kan de nieuwe jabber met het oude CUCM werken. Jabber kan bepalen of CUCM de stroom van de

Toewijzingscode ondersteunt en alleen als dit model wordt ondersteund, switch en gebruikt het impliciete subsidiestroom.
- De AuthZ-service wordt uitgevoerd op de CUCM-server.
- AuthZ ondersteunt alleen Impliciete Grant Flow. Dit betekent dat er geen verfrissingstoken/offline toegangstoken was. Elke keer dat client een nieuw toegangstoken wilde, moet de gebruiker opnieuw authenticeren met de IDP.
- Access Tokens werden alleen uitgegeven als uw plaatsing SSO is ingeschakeld. Niet-SSO-implementaties werkten in dit geval niet en Tokens werden niet consequent op alle interfaces gebruikt.
- Access Tokens zijn niet op zichzelf gericht, maar blijven juist bewaard in het geheugen van de server die ze heeft uitgegeven. Als CUCM1 het toegangstoken heeft gegeven, kan dit alleen door CUCM1 worden geverifieerd. Als de client op CUCM2 probeert toegang te krijgen tot de service, moet CUCM2 dat token op CUCM1 valideren. Netwerkvertragingen (proxy-modus).
- Gebruikerservaring op mobiele klanten is zeer slecht omdat de gebruiker opnieuw aanmeldingsgegevens op een alfanumeriek toetsenbord moet invoeren wanneer de gebruiker opnieuw authenticeert met de IDP (meestal van 1 tot 8 uur, afhankelijk van verschillende factoren).
- Clients die meerdere toepassingen via meerdere interfaces gebruiken, moeten meerdere aanmeldingsgegevens/blokken behouden. Geen naadloze ondersteuning voor hetzelfde gebruikerslogbestand bij 2 soortgelijke klanten. Bijvoorbeeld, gebruiker A logt in van jabber instanties die op 2 verschillende iPhones lopen.
- AutoZ voor ondersteuning van zowel SSO- als niet-SSO-implementaties.
- AuthZ ter ondersteuning van impliciete subsidiestroom + autorisatiecode Omdat het **achterwaarts compatibel** is**,** stelt het klanten zoals **RTMT** in staat om te blijven werken tot ze zich aanpassen.
- Met de autorisatie-code gift flow geeft AuthZ toegangstoken op en verfrist het token. Het verfrissingstoken kan worden gebruikt om een ander toegangstoken op te halen zonder dat er een echtheidscontrole nodig is.
- Access Tokens zijn zelf-ingesloten, ondertekend en versleuteld en gebruiken de JWT (JSON-webtokens)-standaard (RFC-conform).
- Signing- en encryptiesleutels zijn gemeenschappelijk voor het cluster. Elke server in het cluster kan het toegangstoken controleren. Het is niet nodig het geheugen in stand te houden.
- de service die op CUCM 12.0 wordt uitgevoerd, is de gecentraliseerde verificatieserver in het cluster.
- Vernieuwde penningen worden opgeslagen in Database (DB). Admin moet hem indien nodig kunnen intrekken. Revocatie is gebaseerd op gebruikersnaam of gebruiker & clientID.
- Signed Access Tokens maken het mogelijk dat verschillende producten toegangspenningen valideren zonder dat ze hoeven op te slaan. Configureerbaar toegangstoken en verfrist een leven lang (standaard 1 uur en 60 dagen respectievelijk).
- Het JWT-formaat is afgestemd op Spark, dat in de toekomst synergieën met de Hybride Spark-diensten mogelijk maakt.
- Ondersteuning voor dezelfde gebruiker logt in vanaf twee soortgelijke apparaten. Bijvoorbeeld: Gebruiker A kan in van jabber instanties inloggen die op 2 verschillende iPhones draaien.

Elementen van de machtigingscode Grant Flow

- Auth Z-server
- Encryptietoetsen
- Signaaltoetsen
- Verfris Tokens

Configureren

Deze optie is standaard niet ingeschakeld.

Stap 1. Om deze functie in te schakelen, navigeer naar **System > Enterprise-parameters**.

Stap 2. Stel het param in **met de inlogstroom ververversen** naar Ingeschakeld zoals in de afbeelding.

- Toegangstoken is ondertekend en versleuteld. Signing- en encryptiesleutel is gemeenschappelijk voor de cluster. Dit betekent dat elk knooppunt in het cluster het toegangstoken kan valideren.
- Het toegangstoken is in JWT-formaat (RFC 7519).
- Access Tokens hergebruiken een enterprise parameter (OAuth Access Token Expiry timer), die van toepassing is op zowel oude token als nieuwe token formaten.
- Standaardwaarde - 60 minuten.
- Minimale waarde - 1 min.
- Maximumwaarde - 1440 minuten


```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6IjhkMGQ1MzI0LWY0ZjAtNGIwYi04MTFlLTRhNTlmZGI2YjcyMjpj
Mjc3MGM5N2JkYTlkMzRmZDA1YTdlYTFhZWQzZTU0Y2E4MGJkZDdlZTM1ZDk3MDNiNjBiNTQ5MTBiZDQ0ODRiIn0.eyJwcml2
YXRlIjoiZXlKaGJHY2lPaUprYVhJaUxDSmpkSGtpT2lKS1YxUWlMQ0psYm1NaU9pSkJNVEk0UTBKRExVaFRNalUySWl3aWEy
bGtJam9pT0dRd1pEVXpNalF0WmpSbU1DMDBZakJpTFRneE1XVXROR0UxT1daa1lqWmlOekl5T21Vd1ptUm1ZMk16WlRRMU5E
RTFOV0ZpTkRJek5tRTJOMlV4T0RCbU1qWmxZMkl3WXpJeE56SXlOREJtWlRFellXWXlOak14TkRkalpHVXpNR1l3TjJJaWZR
Li5xQWd6aGdRaTVMMkdlaDl5V2RvN25nLmdMTHNpaTRjQk50c1NEUXRJTE51RWRnWTl4WkJVczJ4YzBaeTFGQjZQNmNzWWJf
ZkRnaDRZby04V1NaNjUzdXowbnFOalpXT1E1dGdnYW9qMlp6ZFk2ZzN2SWFHbF9JWUtNdkNIWWNscmt4YUFGTk5MWExLQlJm
aTA2LVk2V3l1dUdxNmpNWk5DbnlKX1pTbUpkVFQwc1Z4RTdGTXVxaUJsMElrRGdyVDdvOFNXMEY5cXFadndEZDJSaDdqNkRJ
WGdkS3VtOWltU2xNU1pjejhueVdic01Udk5yMWY0M25VenJzMHk5WWN6NnBDX0czZmlWYjJsX2VWLVFkcFh4TUo2bnZodXcy
djRiUGVkM3VMQlpaVW1oQ3B6TUVDdW5NMlh1TVBrTGdlS1NqWG44aGhPRFNVcW1WQ0Uta3RZdnRBc2Q0RnJxcGNxWlZiS0Zi
VTFRbU0wV2pMYVJtUk9IVllQVkc0a3FBdTRWalVMUzVCRWszNnZ4Nmp3U3BMUy1IdTcwbVRNcmR3dmV5Q2ZOYkhyT0FlVmVv
ekFIR3JqdGlmaFpmSFVUTWZiNkMtX2tOQVJGQWdDclZTZy0wUzlxb1JvTWVkUENETEE4MDJiaWwtNDJjOC15MWo4X1FVaC02
UUtCV2dodVd4VWtBODRpekFFaWl0QTlsSHFKM3Nxd2JFNURkZmhIay05bTJfTTN5MWlWVkdoRVQ3ZW9XVDBqWllnRGRBQjFz
UGwxLTlaSFNYYmsydTE3SkJVRV9FOXI0V0tWMnBqWGtiN0lQSWgtQ3JWQTZkcVdQRHVIbmx1V19wblNLYnYtTkZVbGQ0WEY3
cmZLYmQySlg4eUhhX05pOVVVUnUwZVdsNWxGRUVabklubmFKZEdHLUZrb3VuN2xHSFlwSE4ydXVudmRnOHZVZzZsa0JPbmoz
eUFjc1ZTMGxKc1NWdUxFYldwd2c4YjdBdDM3d3AtMWt2Y1ZQaWpCQ1lCV181d2JzbTFYd2k4MVc2WHVpNzMzQVg3cEJVQnBf
T2VRNzQ2ZXJJekNUUFZCYUpZUGJuZWEtdFhsU3RmZzBGeVRmbnhnX1Vzazl3QXJkemE4c204T0FQaWMxZmFQOG0uUTdFN0FV
X2xUVnNmZFI2bnkydUdhQSJ9.u2fJrVA55NQC3esPb4kcodt5rnjc1o-5uEDdUf-
KnCYEPBZ7t2CTsMMVVE3nfRhM39MfTlNS-qVOVpuoW_51NYaENXQMxfxlU9aXp944QiU1OeFQKj_g-
n2dEINRStbtUc3KMKqtz38BFf1g2Z51sdlnBn4XyVWPgGCf4XSfsFIa9fF051awQ0LcCv6YQTGer_6nk7t6F1MzPzBZzja1a
bpm--6LNSzjPftEiexpD2oXvW8Vl0Z9ggNk5Pn3Ne4RzqK09J9WChaJSXkTTE5G39EZcePmVNtcbayq-
L2pAK5weDa2k4uYMfAQAwcTOhUrwK3yilwqjHAamcG-CoiPZQ
```

```
OAuth Refresh Token Expiry Timer” parameter in enterprise parameters page in CUCM.
Path: System -> Enterprise parameters
Values are integers ranging from 1 – 90
Minimum lifetime = 1 Day
Default lifetime = 60 days
Maximum lifetime = 90 days
```

Elke keer dat de klant om een nieuw toegangstoken vraagt, wordt er een nieuw toegangstoken uitgegeven. Het oude blijft geldig zolang:

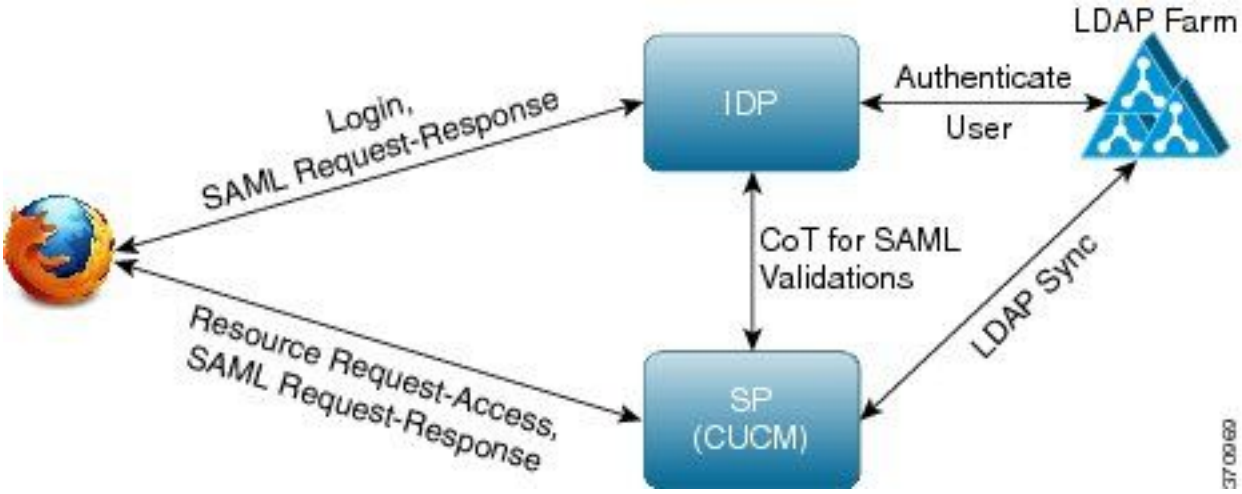
- Toetsen voor signalering/encryptie zijn niet gewijzigd
- Geldigheid (opgeslagen in de token) breekt.
- JSON web-tokens: bestaan uit drie delen, gescheiden door punten, die: Kop, payload en handtekening.

Steekproef-toegangstoken:

- Aan het begin van het token dat in vet is gemarkeerd, verschijnt de header.
- Midden is de payload.
- Als het teken vet is, wordt het einde gemarkeerd met de handtekening.

Netwerkdiagram

Hier is een overzicht op hoog niveau van de betrokken callflow:



Verfris Tokens

- Vernieuwtoken zijn getekend.
- Het ververftoken wordt opgeslagen in de tabel **met** vernieuwde **details** in de database als hashwaarde van zichzelf. Dit is het voorkomen van replicatie door DB omdat het door iemand kan worden gekozen. U kunt de tabel als volgt bekijken:
  `run sql select * from refreshtokendetails`
  of met een leesbare datum:

```
run sql select pkid,refreshtokenindex,userid,clientid,dbinfo('utc_to_datetime',validity) as
validity,state from refreshtokendetails
```

```
admin:run sql select * from refreshtokendetails
pkid          refreshtokenindex userid   clientid validity            state
============= ================= ======== ======== ================== ==========
173e2283-1... 65483476618891... bvanturn C1b4b... 2019-01-05 14:11:46 1080686546
cd2c634c-7... 0bf6b2989db114... bvanturn C1b4b... 2019-01-05 14:28:41 569144456
a3706858-b... b4800f20dbfe0e... bvanturn C1b4b... 2019-01-05 14:38:12 1146722445
```

> **Waarschuwing**: Het verftoken wordt van DB gespoeld wanneer de geldigheid is verlopen. Timer-thread loopt om 2.00 uur elke dag (niet aanpasbaar via UI, maar kan via externe ondersteuningsaccount worden gewijzigd). Als de tabel een groot aantal toegangspenningen heeft, die ongeldig zijn en moeten worden gemarkeerd. Dit kan een CPU-punt veroorzaken.

Sample refresh token:


```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6IjhkMGQ1MzI0LWY0ZjAtNGIwYi04MTFlLTRhNTlmZGI2YjcyMjpj
Mjc3MGM5N2JkYTlkMzRmZDA1YTdlYTFhZWQzZTU0Y2E4MGJkZDdlZTM1ZDk3MDNiNjBiNTQ5MTBiZDQ0ODRiIn0.eyJleHAi
OjE1MDI2MjAwNTIsImlzcyI6IjhkMGQ1MzI0LWY0ZjAtNGIwYi04MTFlLTRhNTlmZGI2YjcyMiIsInR5cCI6InVzZXIiLCJ0
aWQiOiJiOTkxMjIxZi1mNDJlLTRlNTItODg3MS1jODc2ZTYzNWRkNWIiLCJjdHlwIjoicmVmcmVzaCIsImNjaWQiOiJDM2Iw
YWZmZWZlMTQzOTA0MTY4M2U5YzJjMzdkMzZmNDM4ZWYwZWYyN2MwOTM4YWRjNjIyNmUwYzAzZDE2OWYyYSJ9.creRusfwSYA
MAtttS2FIPAgIVvCiREvnzlouxeyGVndalJlMa-ZpRqv8FOBrsYwqEyulrl-
TeM8XGGQCUvFaqO9IkhJqSYz3zvFvvySWzDhl_pPyWIQteAhL1GaQkue6a5ZegeHRp1sjEczKMLC6H68CHCfletn5-
j2FNrAUOX99Vg5h4mHvlhfjJEel3dU_rciAIni12e3LOKajkzFxF6W0cXzzujyi2yPbY9gZsp9HoBbkkfThaZQbSlCEpvB3t
7yRfEMIEaHhEUU4M3-uSybuvitUWJnUIdTONiWGRh_fOFR9LV3Iv9J54dbsecpsncc369pYhu5IHwvsglNKEQ
```


Verfris Tokens herroepen

Admin heeft de mogelijkheid om alle verfrissingspenningen voor een gebruiker of apparaat-enige verfrissingspenningen voor een gebruiker te herroepen via **gebruikerID** of **gebruikerID** en **ClientID**.

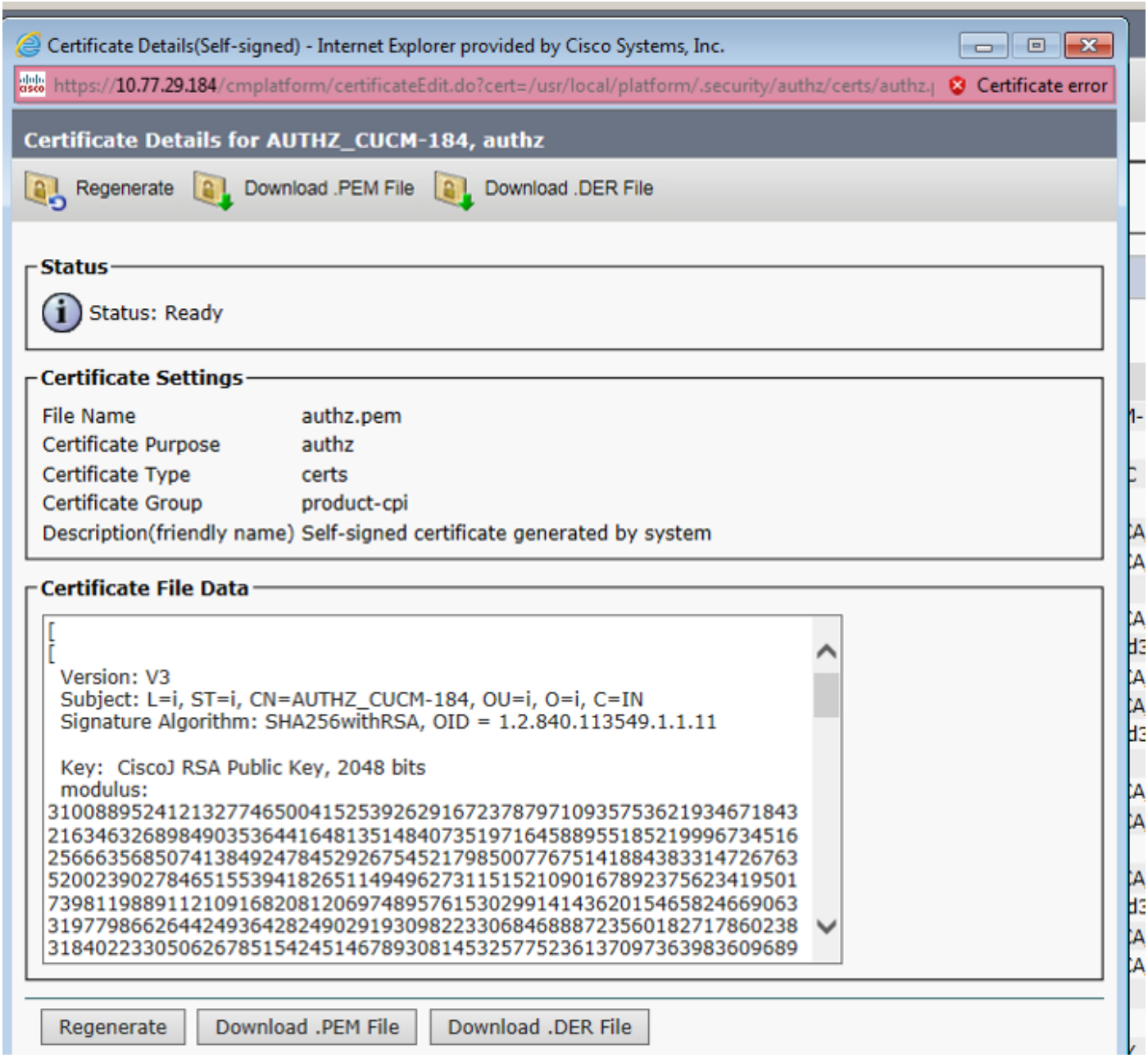
Zo trekt u op een apparaat gebaseerde RTs voor een gebruiker in:

- herroepen RT voor gebruiker xyz en apparaat geïdentificeerd door client_id abc.
- https://cucm-193:8443/ssosp/token/revoke?user_id=xyz&client_id=abc

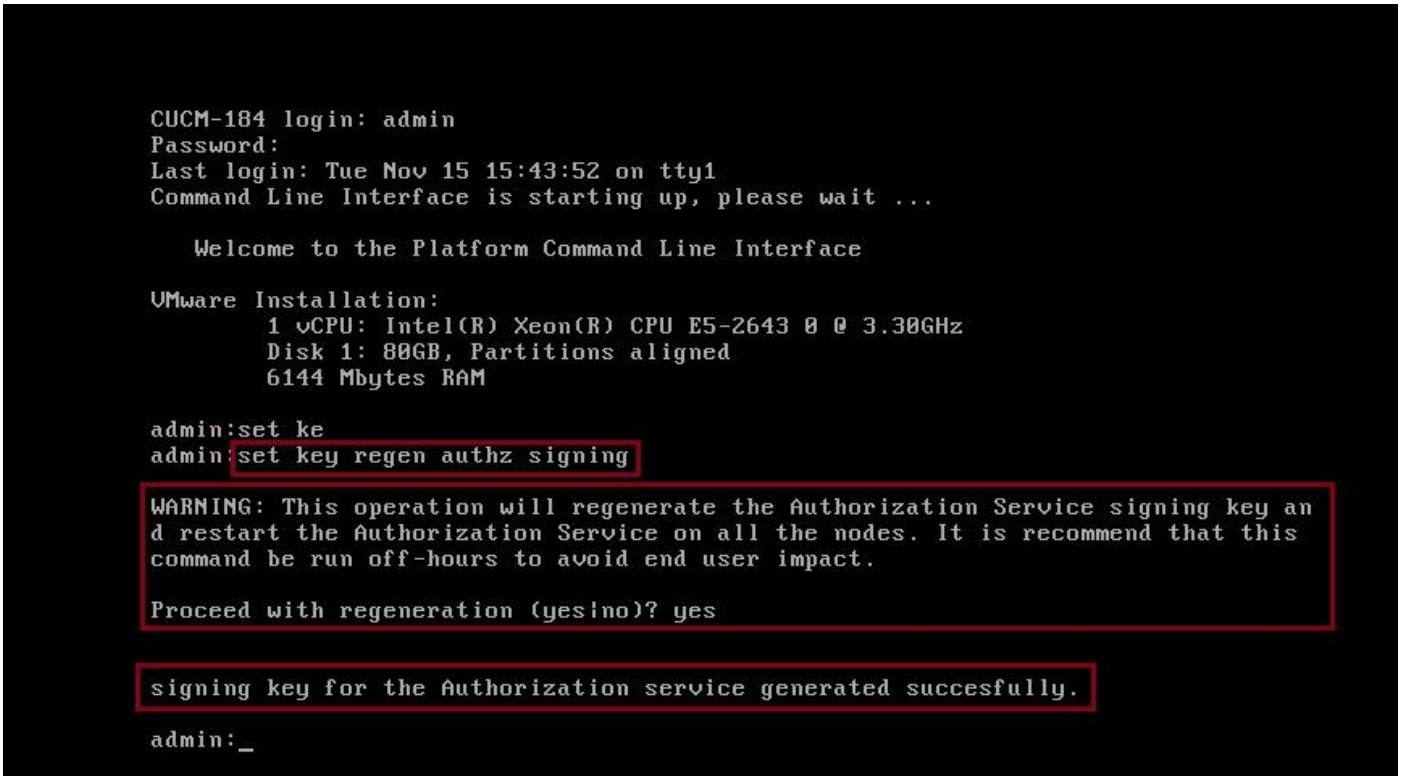
Toetsen voor signalering en encryptie

- Signing-toets is gebaseerd op RSA, die privaat/openbaar sleutelpaar heeft.
- De encryptiesleutel is een symmetrische sleutel.
- Deze toetsen worden alleen op de uitgever gemaakt en worden verspreid over alle knooppunten in de cluster.
- Zowel de ondertekenende sleutel als de coderingssleutel kunnen opnieuw worden gegenereerd met behulp van de opties die zijn vermeld. Dit moet echter alleen gebeuren als de beheerder van mening is dat de toetsen zijn aangetast. De impact van de hergeneratie van een van deze toetsen is dat alle door AuthZ-diensten uitgegeven toegangspenningen ongeldig worden.
- Signaaltoetsen kunnen opnieuw gegenereerd worden met UI en CLI.
- Encryptietoetsen kunnen alleen met CLI worden geregenereerd.

De regeneratie van Auteur certs (ondertekenende sleutel) van de **Cisco Unified OS**-beheerpagina op CUCM is zoals in de afbeelding weergegeven.



De regeneratie van de Authz-signaaltoets met het gebruik van de CLI-opdracht is zoals in de afbeelding weergegeven.



Admin kan autorisatie- en encryptiesleutels weergeven met behulp van CLI. De hash van de toets wordt weergegeven in plaats van de originele toets.

Opdracht om toetsen weer te geven is:

Signaaltoets: **Laat belangrijke auz** tekenen en tonen zoals in de afbeelding.

```
admin:show key authz signing

authz signing key with checksum: a155d81be734850226f990a62816f1ae last synced on: 06/09/2017 13:04:47
```

Encryptiesleutel: **Laat een belangrijke auz-encryptie** en zoals in de afbeelding zien.

```
admin:show key authz encryption

authz encryption key with checksum: 88edce92173e33f9cedbbfb09cd0e8c4 last synced on: 06/14/2017 16:22:06
```

Opmerking: Het signatuur en de encryptie auz zijn altijd verschillend.

Verifiëren

Gebruik dit gedeelte om te bevestigen dat de configuratie correct werkt.

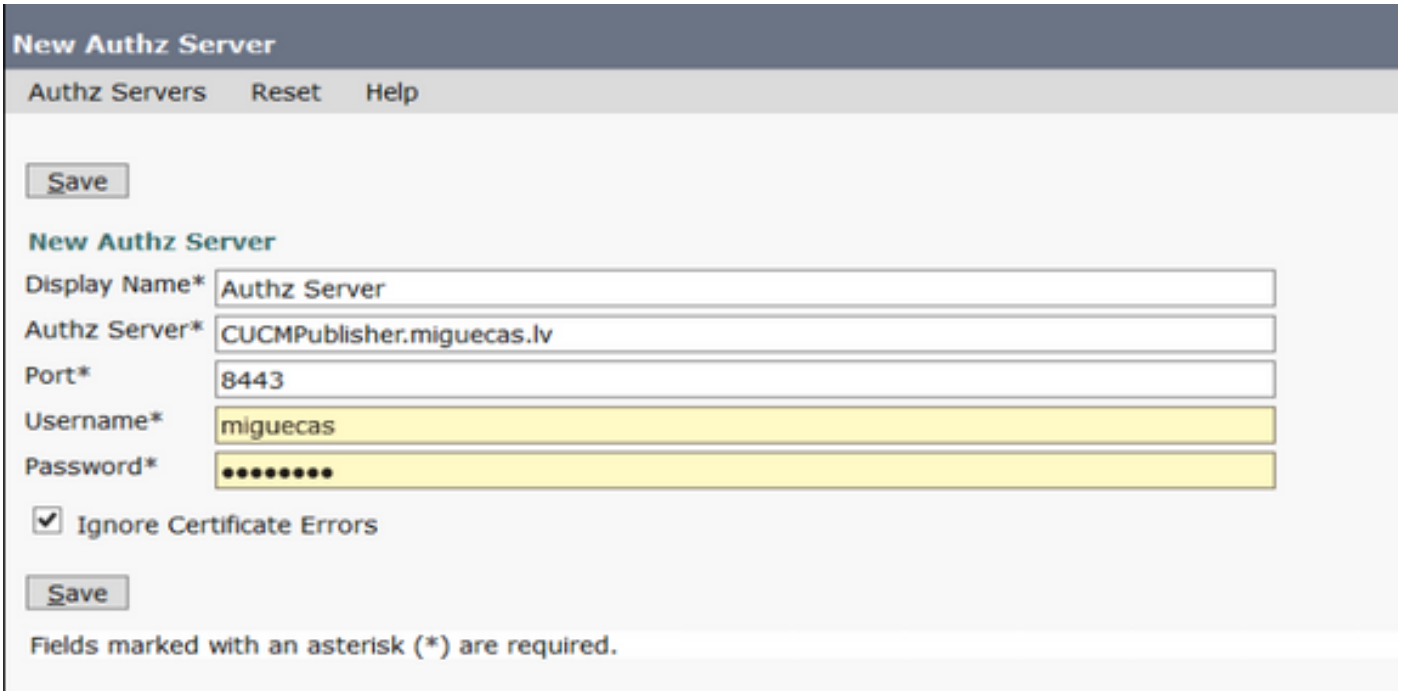
Wanneer de netwerkbeheerder is bedoeld om 0 te gebruiken op de Cisco Unity Connection-server (CUC), moet de netwerkbeheerder twee stappen uitvoeren.

Stap 1. Configureer de Unity Connection Server om de OAuth Token-signalering en encryptiesleutels uit de CUCM te halen.

Stap 2. Schakel gifteserver in.

Opmerking: Om de ondertekenings- en encryptiesleutels te halen, moet Unity worden geconfigureerd met de CUCM host-gegevens en een gebruikersaccount dat is ingeschakeld van de CUCM AXL Access. Als dit niet is ingesteld, kan de Unity Server de OAuth Token niet van het CUCM ophalen en kan het voicemail-logbestand voor de gebruikers niet beschikbaar zijn.

Navigeren in naar **Cisco Unity Connection Management > systeeminstellingen > Auteur servers**



Problemen oplossen

Deze sectie verschaft de informatie die u kunt gebruiken om problemen met uw configuratie op te lossen.

Opmerking: Als Auto wordt gebruikt en de Cisco Jabber-gebruikers niet kunnen inloggen, herzie altijd de ondertekening en encryptiesleutels van CUCM en Instant Messaging and Presence (IM&P) servers.

De netwerkbeheerders moeten deze twee opdrachten op alle CUCM- en IM&P-knooppunten **uitvoeren**:

- **€ ? belangrijke auteurs tonen**
- **zeer belangrijke autorisatie-encryptie tonen**

Als de signaaluitvoer en de encryptie auz-output niet over alle knooppunten overeenkomen, moeten ze worden gereproduceerd. Om dat te kunnen doen, moeten deze twee opdrachten op alle CUCM- en IM&P-knooppunten worden uitgevoerd:

- **reeks - encryptie met regen**
- **ondertekening van regen .**

Daarna moet de **Cisco Tomcat-**service op alle knooppunten opnieuw worden gestart.

Gelijktijdig gebruik van de toetsen en mismatch, is deze foutlijn te vinden in de Cisco Jabber-

loggen:

```
2021-03-30 14:21:49,631 WARN  [0x0000264c] [vices\impl\system\SingleSignOn.cpp(1186)] [Single-Sign-On-Logger] [CSFUnified::SingleSignOn::Impl::handleRefreshTokenFailure] - Failed to get valid access token from refresh token, maybe server issue.
```

Op deze locaties worden de Sso-app-logs gegenereerd:

- **Activelog platform/log/ssoApp.log** Dit vereist geen spoorconfiguratie voor logverzameling. Elke keer dat de SSO-app wordt uitgevoerd, wordt er een nieuw logbestand gegenereerd in ssoApp.log-bestand.
- SSOSP-logbestanden: **lijst van bestanden met activelog tomcat/logs/ssosp/log4j** Telkens als deze optie is ingeschakeld, wordt er een nieuw logbestand met de naam **ssosp00XXX.log** op deze locatie aangemaakt. Alle andere SSO-bewerkingen en alle Oauth-bewerkingen worden ook in dit bestand ingelogd.
- Certificaatdocumenten: **bestandlijst activelogplatform/log/certMgmt*.log** Elke keer dat het AuthZ certificaat wordt geregenereerd (UI of CLI) wordt er een nieuw logbestand gegenereerd voor deze gebeurtenis.
  Voor de regeneratie van auz-encryptiesleutel wordt een nieuw logbestand gegenereerd voor deze gebeurtenis.

Gerelateerde informatie

[Invoerend geluid met Cisco Collaboration Solution release 12.0]