

承認コード付与フローの導入とトラブルシューティング – OAuth機能拡張 : Cisco Collaboration Solutions 12.0

内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[背景説明](#)

[主な機能](#)

[重要な考慮事項](#)

[承認コード付与フローの要素](#)

[設定](#)

[ネットワーク図](#)

[トークンの更新](#)

[更新トークンの取り消し](#)

[確認](#)

[トラブルシューティング](#)

[関連情報](#)

概要

このドキュメントでは、特にモバイル上のJabberで、さまざまなデバイス間のJabberユーザーエクスペリエンスを向上させるために、認証コード許可(AUTHORIZATION)フローが更新トークンに基づくしくみを説明します。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- Cisco Unified Communications Manager(CUCM)12.0バージョン
- シングルサインオン(SSO)/SAML
- Cisco Jabber
- Microsoft ADFS
- アイデンティティプロバイダー(IdP)

これらのトピックの詳細については、次のリンクを参照してください。

- [Cisco Unified Communications用SAML SSO導入ガイド](#)
- [Unified Communications Manager SAML SSOの設定例 :](#)

- [SAML SSO 向け AD FS 2.0 バージョンのセットアップ例:](#)

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアに基づいています。

- Microsoft ADFS(IdP)
- LDAP Active Directory
- Cisco Jabber クライアント
- CUCM 12.0

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

背景説明

現在、インフラストラクチャを使用したJabber SSOフローは、CUCM Authzサービスが短期アクセストークンを割り当てる暗黙的許可フローに基づいています。

ポストアクセストークンの期限切れ、CUCMは再認証のためにJabberをIdPにリダイレクトします。

これにより、ユーザエクスペリエンスが低下します。特に、ユーザがクレデンシャルを頻繁に入力するように求められるモバイル上のjabberでは問題が発生します。

また、Security Re-Architecture Solutionでは、SSOと非SSOの両方のシナリオでJabberとエンドポイントのログインフローを統合するための、認証コード付与フロー(Refresh Tokensアプローチ (エンドポイント/他のコラボレーションアプリケーションに拡張可能)も提案しています。

主な機能

- 承認コードの付与フローは、更新トークン (エンドポイントやその他のコラボレーションアプリケーションに拡張可能) に基づいて、さまざまなデバイス (特にモバイル上のJabber) のJabberユーザエクスペリエンスを向上させます。
- 自己完結型および暗号化OAuthトークンをサポートし、さまざまなコラボレーションアプリケーションがクライアントのリソース要求を検証および応答できるようにします。
- 暗黙的な認可フローモデルが保持され、後方互換性が確保されます。これにより、認証コード許可フローに移動していない他のクライアント (RTMTなど) のシームレスなパスも可能になります。

重要な考慮事項

- 古いJabberクライアントが新しいCUCMと連携できるようにする実装 (暗黙的な認可と認可コードの認可フローの両方をサポートするため)。また、新しいJabberは古いCUCMと連携できます。Jabberは、CUCMが認証コード認可フローをサポートしているかどうかを判別できます。また、このモデルをサポートしている場合にのみ、暗黙的な認可フローを切り替えて使用します。

- AuthZサービスはCUCMサーバで実行されます。
- AuthZは暗黙的な許可フローのみをサポートします。これは、更新トークン/オフラインアクセストークンがなかったことを意味します。クライアントが新しいアクセストークンを必要とするたびに、ユーザはIdPで再認証する必要があります。
- アクセストークンは、展開がSSO対応の場合にのみ発行されます。この場合、非SSO展開は機能せず、すべてのインターフェイスで一貫してアクセストークンが使用されませんでした。
- アクセストークンは自己完結型ではなく、トークンを発行したサーバーのメモリに保持されます。CUCM1がアクセストークンを発行した場合は、CUCM1によってのみ確認できます。クライアントがCUCM2のサービスにアクセスしようとする、CUCM2はCUCM1でそのトークンを検証する必要があります。ネットワーク遅延(プロキシモード)。
- ユーザがIdPで再認証を行う場合(通常、いくつかの要因に応じて1時間から8時間の間で実行される)、英数字のキーパッドでクレデンシャルを再入力する必要があるため、モバイルクライアントでのユーザエクスペリエンスは非常に悪いです。
- 複数のインターフェイスを介して複数のアプリケーションと通信するクライアントは、複数のクレデンシャル/ブロックを維持する必要があります。2つの類似クライアントから同じユーザがログインするシームレスなサポートはありません。たとえば、ユーザAは2つの異なるiPhoneで実行されるjabberインスタンスからログインします。
- AuthZ:SSOと非SSOの両方の導入をサポートします。
- 暗黙的な認可フロー+認可コード認可フローをサポートするAuthZ。下位互換性があるため、RTMTなどのクライアントも適応するまで作業を継続できます。
- 認証コード認可フローでは、AuthZはアクセストークンとリフレッシュトークンを発行します。refreshトークンを使用すると、認証を必要とせずに、別のアクセストークンを取得できます。
- アクセストークンは、自己完結型、署名型、暗号化型であり、JWT(JSON Webトークン)標準(RFC準拠)を使用します。
- 署名キーと暗号化キーは、クラスタに共通です。クラスタ内の任意のサーバがアクセストークンを確認できます。メモリ内で維持する必要はありません。
- CUCM 12.0で実行されるサービスは、クラスタ内の中央集中型の認証サーバです。
- 更新トークンはデータベース(DB)に保存されます。管理者は、必要に応じて取り消す必要があります。失効は、useridまたはuseridとclientIDに基づいています。
- 署名付きアクセストークンを使用すると、異なる製品がアクセストークンを保存しなくても検証できます。設定可能なアクセストークンと更新トークンの有効期間(デフォルトは1時間と60日)。
- JWTフォーマットはSparkと連携しており、将来的にSparkハイブリッドサービスとの相乗効果が期待できます。
- 同じユーザが2台の類似デバイスからログインできます。例:ユーザAは、2つの異なるiPhoneで実行されるjabberインスタンスからログインできます。

承認コード付与フローの要素

- 認証Zサーバ
- 暗号キー
- 署名キー
- トークンの更新

設定

この機能はデフォルトでは有効になっていません。

ステップ1：この機能を有効にするには、[System] > [Enterprise Parameters]に移動します。

ステップ2：図に示すように、Refresh Login Flowを使用したパラメータOAuthを[Enabled]に設定します。

SSO and OAuth Configuration		
OAuth Access Token Expiry Timer (minutes) *	60	60
OAuth Refresh Token Expiry Timer (days) *	60	60
Redirect URIs for Third Party SSO Client		
SSO Login Behavior for iOS *	Use embedded browser (WebView)	Use embedded browser (WebView)
OAuth with Refresh Login Flow *	Enabled	Disabled
Use SSO for RTMT *	True	True

- アクセストークンは署名され、暗号化されます。署名と暗号化キーはクラスタに共通です。つまり、クラスタ内の任意のノードがアクセストークンを検証できます。
- アクセストークンはJWT形式(RFC 7519)です。
- アクセストークンは、古いトークン形式と新しいトークン形式の両方に適用されるエンタープライズパラメータ(OAuth Access Token Expiry timer)を再利用します。
- デフォルト値：60分
- 最小値：1分
- 最大値：1440分

eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6IjpkMGQ1MzI0LWY0ZjAtNGIwYi04MTF1LlRlRmNTlmZGI2YjcyMjppMjc3MGM5N2JkYTlkMzRmZDA1YtdlYTFhZWQzZTU0Y2E4MGJkZDdlZTMlZDk3MDNiNjBiNTQ5MTBiZDQ0ODRiIn0.eyJwcm12YXRlIjoizXlKaGJHY2lPaUprYVhJaUxDSmpkSGtpT2lKS1YxUWlMQ0psYmlNaU9pSkJNVEk0UTBKRExVaFRNalUySWl3aWEybgGtJam9pT0dRd1pEVXpNalF0WmpSbU1DMDBZakJpTFRneE1XVXROR0UxT1daa1lqWmlOek15T21Vd1ptUm1ZMk16WlRRMU5ERTFOV0ZpTkrJek5tRTJOM1V4T0RCbU1qWmxZMk13WXPJeE56SX1OREJtWlRFe1lXWl1Oak14TkRkalpHVXpNR113TjJJaWZRLi5xQWd6aGdRaTVMMkdlaDl5V2RvN25nLmdMTHNpaTRjQk50c1NEUXRJT51RWRnWTl4WkJVczJ4YzBaeTFGQjZQNmNzWWJfZkRnaDRzby04V1NaNjUzdXowbnFOalpXT1E1dGdnYW9qMlp6ZFk2ZzN2SWFhbF9JWUtNdKNIWWNscmt4YUFGTk5MWExLQlJmaTA2LVk2V311dUdxNmpNwk5DbnlKXlpTbUpkVFQwc1Z4RTdGTxVxaUJSMElrRGdyVDdvOFNXMEY5cXFadndEzDJSaDdqNkRjWGdks3VtOWltU2xNU1pjejhueVdic01Udk5yMWY0M25VenJzMHk5WwN6NnBDX0czZmlWYjJsX2VWLVFkcFh4TUo2bnZodXcydjRiUGVkm3VMQ1paVWl0Q3B6TUVdDw5Nmlh1TVBrTGD1S1NqWG44aGhPRFNvcWlWQ0Uta3RzdNRCBc2Q0RnJxcGNxWlZiS0ZiVTRbU0wV2pMYVJtUk9IVl1lQVkc0a3FBdTRWalVMUzVCRWszNnZ4Nmp3U3BMUy1IdTcwbVRNcmR3dmV5Q2ZOYkhyT0FlVmVvekFIR3JqdG1maFpmSfVUTWziNkMtX2tOQVJGQWdDclZTZy0wUz1xb1JvTWVkuENETEE4MDJiaWwtNDJjOC15Mwo4X1FVaC02UUtCV2dodVd4VWtBODRpekFFaWl0QT1sSHFKM3Nxd2JFNURkZmhIay05bTJfTTN5Mw1WVkdORVQ3ZW9XVDBqW1lnRGRBQjFzUGwxLTlaSFNYmsydtE3SkJVRV9FOXIOV0tWmNqWGTin01QSwgTQ3JWQTZkcVdQRHVlbnx1V19wblNlYnYtTkZVbGQ0WEY3cmZLYmQyS1g4eUhhX05pOVVVUnUwZVdsNWxGRUVabklubmFKZEdHLUZrb3VuN2xHSFlwSE4ydXVudmRnOHZVZzZsa0JpbmozUUFjclZTMGxKc1NwdUxYfYldwd2c4YjdBdDM3d3AtMwT2Y1ZQaWpCQ11CV181d2JzbTFYd2k4MVC2WHVpNzZmZmV3cEJjVQnBfT2VRNzQ2ZXJjJekNUUFZCYUpZUGJuzWETdFhsU3RmZzBGeVRmbnbnX1Vzaz13QXJkeme4c204T0FQaWmXZmFQOG0uUtDFN0FVX2xUVnNmZFI2bnkydUdhQSJ9.u2fJrVa55NQC3esPb4kcodt5rnjcl0-5uEDdUf-KnCYEPBZ7t2CTsMMVVE3nfrhm39mft1NS-qVOVpuow_51NYaENXQMxfxlU9aXp944QiU1OeFQKj_g-n2dEINRStbtUc3KMKqtz38BFflg2Z51sdlnBn4XyVWPgGcf4XSfsFIa9ff051awQ0LcCv6YQTGer_6nk7t6F1MzPzBzZjala bpm--6LNSzjPftEiexpD2oXvW8V10Z9ggNk5Pn3Ne4RzqK09J9WChaJSXkTTE5G39EZcePmVNTcbayq-L2pAK5weDa2k4uYmFAQAwcTOhUrwK3yilwqjHAamG-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

新しいアクセストークンは、クライアントが1つのアクセストークンを要求するたびに発行されます。古いバージョンは、次の限り有効です。

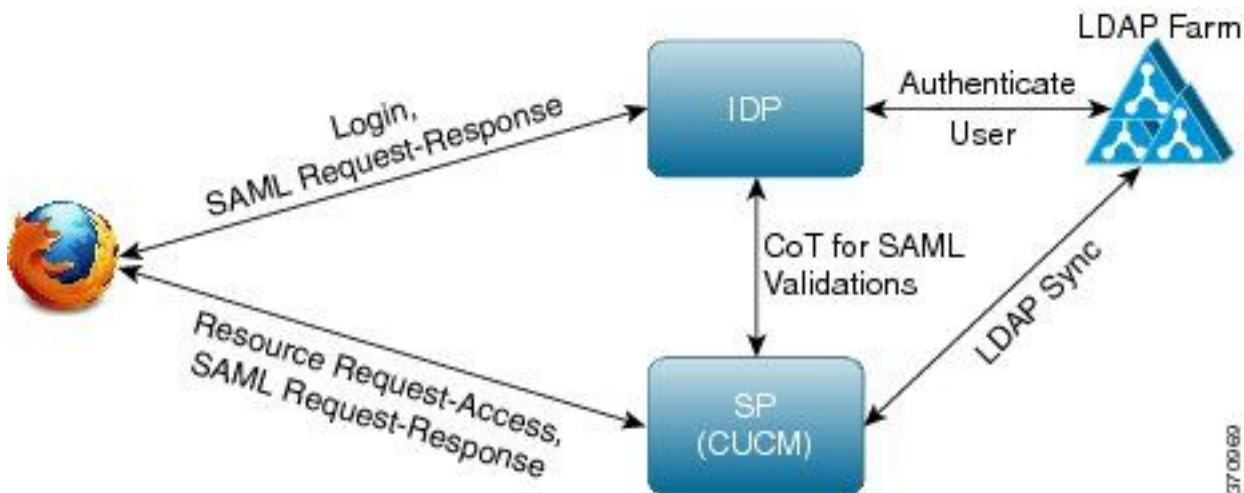
- 署名/暗号化キーは変更されていません
- 有効性 (トークン内に格納) が壊れます。
- JSON Webトークン : 次の3つの部分で構成されます。ドットで区切られます。ヘッダー、ペイロード、および署名。

アクセストークンの例 :

- 太字で強調表示されているトークンの先頭にヘッダーがあります。
- 中央の部分がペイロードです。
- 最後に、トークンが太字で強調表示されている場合はシグニチャです。

ネットワーク図

関連するコールフローの概要を次に示します。



トークンの更新

- 更新トークンが署名されています。
- Refreshトークンは、データベース内の`refreshtokendetails`テーブルに、自身のハッシュ値として格納されます。これは、DBによるレプリケーションを防止するためのものです。これは、誰かが選択できるためです。テーブルを確認するには、次のコマンドを実行します。

```
run sql select * from refreshtokendetails
```

または判読可能な有効日付を使用して :

```
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  Clb4b... 2019-01-05 14:11:46 1080686546
cd2c634c-7... 0bf6b2989db114... bvanturn  Clb4b... 2019-01-05 14:28:41 569144456
a3706858-b... b4800f20dbfe0e... bvanturn  Clb4b... 2019-01-05 14:38:12 1146722445
```

警告 : 有効期限が切れると、更新トークンがDBからフラッシュされます。タイマースレッドは毎日午前2時に実行されます (UIでは構成できませんが、リモートサポートアカウントで変更できます)。テーブルに多数のアクセストークンがある場合は、無効であり、フラッシュする必要があります。これにより、CPUスパイクが発生する可能性があります。

Certificate Details(Self-signed) - Internet Explorer provided by Cisco Systems, Inc.

https://10.77.29.184/cmplatform/certificateEdit.do?cert=/usr/local/platform/.security/authz/certs/authz.j Certificate error

Certificate Details for AUTHZ_CUCM-184, authz

Regenerate Download .PEM File Download .DER File

Status

Status: Ready

Certificate Settings

File Name	authz.pem
Certificate Purpose	authz
Certificate Type	certs
Certificate Group	product-cpi
Description(friendly name)	Self-signed certificate generated by system

Certificate File Data

```
[
[
Version: V3
Subject: L=i, ST=i, CN=AUTHZ_CUCM-184, OU=i, O=i, C=IN
Signature Algorithm: SHA256withRSA, OID = 1.2.840.113549.1.1.11

Key: CiscoJ RSA Public Key, 2048 bits
modulus:
310088952412132774650041525392629167237879710935753621934671843
216346326898490353644164813514840735197164588955185219996734516
256663568507413849247845292675452179850077675141884383314726763
520023902784651553941826511494962731151521090167892375623419501
739811988911210916820812069748957615302991414362015465824669063
319779866264424936428249029193098223306846888723560182717860238
318402233050626785154245146789308145325775236137097363983609689
```

CLIコマンドを使用したAuthz署名キーの再生成を図に示します。

```
CUCM-184 login: admin
Password:
Last login: Tue Nov 15 15:43:52 on tty1
Command Line Interface is starting up, please wait ...
```

```
Welcome to the Platform Command Line Interface
```

```
VMware Installation:
 1 vCPU: Intel(R) Xeon(R) CPU E5-2643 0 @ 3.30GHz
Disk 1: 80GB, Partitions aligned
6144 Mbytes RAM
```

```
admin:set ke
admin:set key regen authz signing
```

```
WARNING: This operation will regenerate the Authorization Service signing key and restart the Authorization Service on all the nodes. It is recommend that this command be run off-hours to avoid end user impact.
```

```
Proceed with regeneration (yes/no)? yes
```

```
signing key for the Authorization service generated succesfully.
```

```
admin:_
```

管理者は、CLIを使用して認証キーと暗号化キーを表示できます。キーのハッシュは、元のキーではなく表示されます。

キーを表示するコマンドは次のとおりです。

署名キー：show key authz signingと図に示すように。

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

暗号キー:show key authz encryptionと図に示すように。

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

注：署名authzと暗号化authzは常に異なります。

確認

ここでは、設定が正常に機能しているかどうかを確認します。

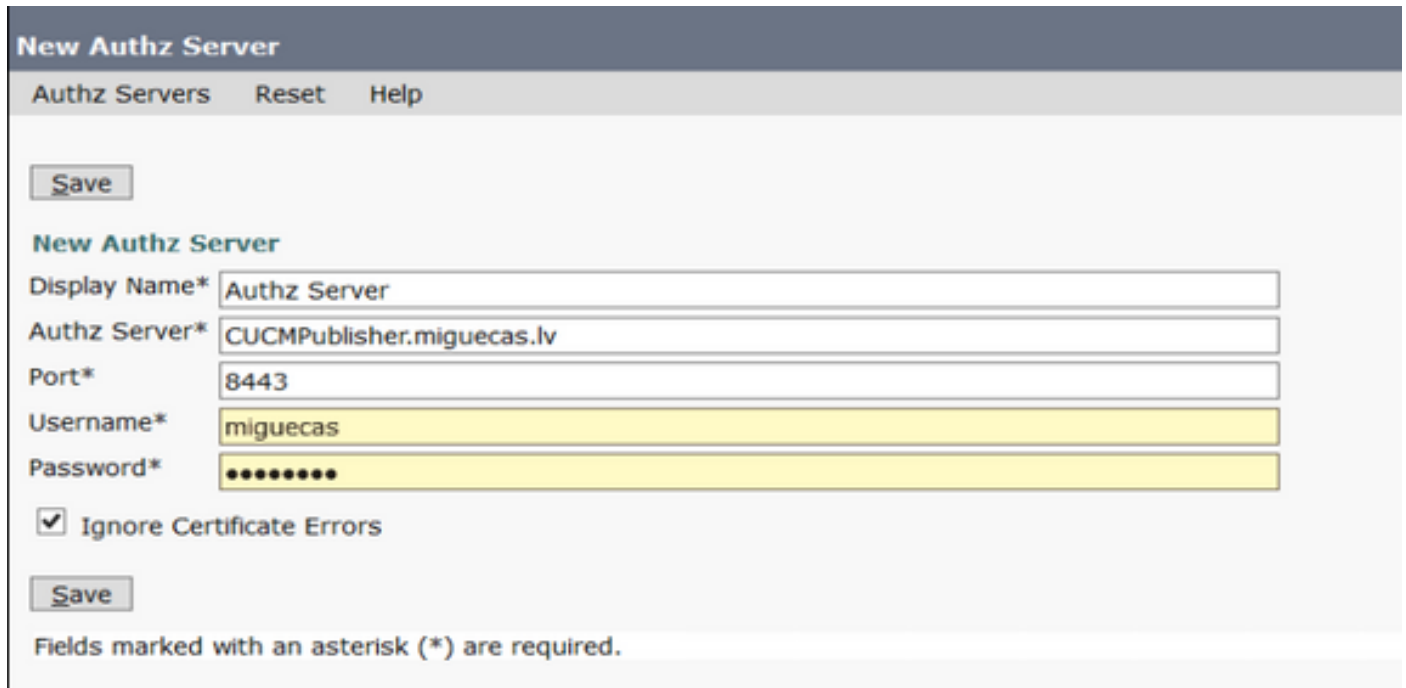
Cisco Unity Connection(CUC)サーバでOAuthを使用する場合、ネットワーク管理者は2つの手順を実行する必要があります。

ステップ1:OAuthトークン署名と暗号化キーをCUCMから取得するようにUnity Connectionサーバを設定します。

ステップ2:CUCサーバでOAuthサービスを有効にします。

注：署名キーと暗号化キーを取得するには、CUCMホストの詳細とCUCM AXLアクセスが有効なユーザアカウントを使用してUnityを設定する必要があります。これが設定されていない場合、UnityサーバはCUCMからOAuthトークンを取得できず、ユーザのボイスメールログインを使用できません。

[Cisco Unity Connection Administration] > [System Settings] > [Authz Servers]に移動します



The screenshot shows the 'New Authz Server' configuration page. At the top, there are navigation links: 'Authz Servers', 'Reset', and 'Help'. Below these is a 'Save' button. The main form is titled 'New Authz Server' and contains the following fields:

- Display Name*: Authz Server
- Authz Server*: CUCMPublisher.miguecas.lv
- Port*: 8443
- Username*: miguecas
- Password*: [masked with dots]

Below the fields is a checkbox labeled 'Ignore Certificate Errors' which is checked. There is another 'Save' button at the bottom of the form. A note at the bottom states: 'Fields marked with an asterisk (*) are required.'

トラブルシューティング

ここでは、設定のトラブルシューティングに使用できる情報を示します。

注：OAuthを使用していて、Cisco Jabberユーザがログインできない場合は、CUCMおよびインスタントメッセージングおよびプレゼンス(IM&P)サーバから署名および暗号化キーを必ず確認してください。

ネットワーク管理者は、すべてのCUCMおよびIM&Pノードで次の2つのコマンドを実行する必要があります。

- show key authz signing
- show key authz encryption

署名authzと暗号化authzの出力がすべてのノードで一致しない場合は、再生成する必要があります。これを実行するには、次の2つのコマンドをすべてのCUCMおよびIM&Pノードで実行する必要があります。

- set key regen authz encryption
- set key regen authz signing

その後、すべてのノードでCisco Tomcatサービスを再起動する必要があります。

キーの不一致に加えて、次のエラー行がCisco Jabberのログに表示されます。

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.

ssoアプリケーションログは次の場所で生成されます。

- **file view activelog platform/log/ssoApp.log** ログ収集のトレース設定は不要です。SSOアプリケーションの操作が実行されるたびに、ssoApp.logファイルに新しいログエントリが生成されます。
- **SSOSPログ** : **file list activelog tomcat/logs/ssosp/log4j**
ssoが有効になるたびに、この場所にssosp00XXX.logという名前の新しいログ・ファイルが作成されます。その他のSSO操作とすべてのOAuth操作もこのファイルにログインします。
- **証明書ログ** : **file list activelog platform/log/certMgmt*.log**
AuthZ証明書が再生成されるたびに (UIまたはCLI)、このイベントに対して新しいログファイルが生成されます。
authz暗号化キーの再生成では、このイベントに対して新しいログファイルが生成されます。

関連情報

[Cisco Collaboration Solution リリース12.0によるOAuthの導入](#)