Installez et configurez le fournisseur d'identité de doctrine (IDP) pour la gestion d'identité de Cisco (id) pour activer SSO

Contenu

Introduction

Ce document décrit la configuration sur le fournisseur d'identité d'OpenAM (IDP) pour activer simple se connectent (SSO).

Modèles de déploiement d'id de Cisco

Produit	Déploiement
UCCX	Co-résident
PCCE	Co-résident avec CUIC (centre d'intelligence de Cisco Unified) et LD (données vivantes)
UCCE	Co-résident avec CUIC et LD pour les déploiements 2k. Autonome pour les déploiements 4k et 12k.

Conditions préalables

Exigences

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Version 11.6 de version 11.6 ou de Cisco Unified Contact Center Enterprise du Cisco Unified Contact Center Express (UCCX) ou version 11.6 emballée du Contact Center Enterprise

(PCCE) comme applicable.

Note: Ce document met en référence la configuration en ce qui concerne le service de Cisco Identitify (id) et le fournisseur d'identité (IDP). Le document met en référence UCCX dans les captures d'écran et les exemples, toutefois la configuration est semblable en ce qui concerne le service de Cisco Identitify (UCCX/UCCE/PCCE) et l'IDP.

Composants utilisés

Ce document n'est pas limité à des versions de matériel et de logiciel spécifiques.

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est vivant, assurez-vous que vous comprenez l'impact potentiel de n'importe quelle commande.

Installez

La doctrine est un projet open source qui fournit des capacités simples d'ouverture de session et permet à des sites pour prendre des décisions au courant d'autorisation pour l'accès individuel des ressources en ligne protégées d'une manière de intimité-préservation. Il prend en charge le Langage SAML (SAML2). Les id est un client SAML2 et compté ne prendre en charge la doctrine avec minimal ou aucun changement des id. En 11.6, des id est qualifiés pour fonctionner avec l'IDP de doctrine.

Note: Ce document met en référence la version 3.3.0 de doctrine comme partie de la qualification avec SSO

Configuration système nécessaire

Composant	Détails
Version de doctrine	v3.3.0
Site de téléchargement	http://shibboleth.net/downloads/identity-pro
Installez la plate-forme	Ubuntu 14.0.4 version "1.8.0_121" de Javas
Version de Protocole LDAP (Lightweight Directory Access Protocol)	Répertoire actif 2.0
Web server de doctrine	Apache Tomcat/8.5.12

Veuillez se référer le wiki pour l'installation de la doctrine

https://wiki.shibboleth.net/confluence/display/IDP30/Installation

Configurer

Intégrez avec un serveur LDAP

Pour intégrer un serveur LDAP avec la doctrine, les champs doivent être mis à jour

dans **$shibboleth_home/conf/ldap.properties où** $shibboleth_home **(le par défaut** est /opt/shibboleth-idp) se rapporte au répertoire d'installer qui est utilisé à l'installation de la doctrine.

Champ	Valeur prévue	Description
idp.authn.LDAP.trustCertificates	Une ressource pour charger des ancres de confiance de, habituellement un fichier local dans $ {idp.home} /credentials là où idp.home est une variable d'environnement exportée comme JAVA_OPTS dans setenv.sh	% {idp.home} /credentials/ldap-server.crt
idp.authn.LDAP.trustStore	Une ressource pour charger un keystore de Javas qui contient des ancres de confiance, habituellement un fichier local dans % {idp.home} /credentials	% {idp.home} /credentials/ldap-server.truststore
idp.authn.LDAP.returnAttributes	La virgule a séparé la liste de LDAPAttributes qui doit être retourné. Si vous voulez renvoyer tous les attributs, ajoutez « * ».	*
idp.authn.LDAP.baseDN	Le baseDN auquel la recherche de LDAP doit être exécutée	CN=users, DC=cisco, DC=com
idp.authn.LDAP.subtreeSearch	Si rechercher périodiquement	vrai
idp.authn.LDAP.userFilter	Filtre de recherche de LDAP	(sAMAccountName= {utilisateur}) *
idp.authn.LDAP.bindDN	DN à lier avec quand la recherche est exécutée	administrator@cisco.com
idp.authn.LDAP.bindDNCredential	Mot de passe à lier avec quand la recherche est exécutée	
idp.authn.LDAP.dnFormat	Une chaîne de formatage pour générer les dn d'utilisateur pour authentifier	% de s@adfsserver.cisco.com (% de s@domainname)
idp.authn.LDAP.authenticator	Contrôle le processus pour la façon dont l'authentification se produit contre le LDAP	bindSearchAuthenticator
idp.authn.LDAP.ldapURL	URI de connexion pour le répertoire LDAP	

Pour plus de détails, référez-vous :

https://wiki.shibboleth.net/confluence/display/IDP30/LDAPAuthnConfiguration

Fichier de configuration témoin

```
# heure en quelques millisecondes d'attendre des
forresponses
#idp.authn.LDAP.responseTimeout = PT3S
Configuration SSL de ##, jvmTrust, certificateTrust, ou
keyStoreTrust
#idp.authn.LDAP.sslConfig = certificateTrust
## si utilisant le certificateTrust ci-dessus,
positionnement au chemin de certificat de confiance
idp.authn.LDAP.trustCertificates = % {idp.home}
/credentials/ldap-server.crt
## si utilisant le keyStoreTrust ci-dessus, positionnement
au chemin de truststore
idp.authn.LDAP.trustStore = % {idp.home} /credentials/ldap-
server.truststore
```

```
Attributs de retour de ## pendant l'authentification
#idp.authn.LDAP.returnAttributes = userPrincipalName,
sAMAccountName
idp.authn.LDAP.returnAttributes = *
## de propriétés de résolution de DN de ##
# résolution de DN de recherche, utilisée par
anonSearchAuthenticator, bindSearchAuthenticator
# forAD : CN=Users, DC=example, DC=org
idp.authn.LDAP.baseDN = CN=users, DC=cisco, DC=com
idp.authn.LDAP.subtreeSearch = rectifient
*idp.authn.LDAP.userFilter = (sAMAccountName= {utilisateur})
*
# configuration de recherche de grippage
# forAD : idp.authn.LDAP.bindDN= adminuser@domain.com
idp.authn.LDAP.bindDN = administrator@cisco.com
idp.authn.LDAP.bindDNCredential = Cisco@123
# résolution de DN de format, utilisée par
directAuthenticator, adAuthenticator
# utilisation idp.authn.LDAP.dnFormat=% s@domain.com de
forAD
#idp.authn.LDAP.dnFormat = % de s@adfsserver.cisco.com
# la configuration d'attribut de LDAP, voient attribute-
resolver.xml
# la note, thislikely ne s'appliquera pas à l'utilisation
des configurations existantes du résolveur V2
idp.attribute.resolver.LDAP.ldapURL = %
{idp.authn.LDAP.ldapURL}
idp.attribute.resolver.LDAP.connectTimeout =
%{idp.authn.LDAP.connectTimeout:PT3S}
idp.attribute.resolver.LDAP.responseTimeout =
%{idp.authn.LDAP.responseTimeout:PT3S}
idp.attribute.resolver.LDAP.baseDN = %
{idp.authn.LDAP.baseDN : non défini}
idp.attribute.resolver.LDAP.bindDN = %
{idp.authn.LDAP.bindDN : non défini}
idp.attribute.resolver.LDAP.bindDNCredential = %
{idp.authn.LDAP.bindDNCredential : non défini}
idp.attribute.resolver.LDAP.useStartTLS = %
{idp.authn.LDAP.useStartTLS : vrai}
idp.attribute.resolver.LDAP.trustCertificates = %
{idp.authn.LDAP.trustCertificates : non défini}
idp.attribute.resolver.LDAP.searchFilter =
(sAMAccountName=$resolutionContext.principal)
```

Permettez les demandes de tous les clients

Pour s'assurer que les demandes de tous les clients atteignent, des changements sont exigés de « $shibboleth_home/conf/access-control.xml »

key= <entry " AccessByIPAddress " >
parent= <bean " shibboleth.IPRangeAccessControl » de " AccessByIPAddress » d'id=
p : allowedRanges= " # {{'127.0.0.1/32', '0.0.0.0/0', '::1/128', '10.78.93.103/32'}} »/>
</entry>

Ajoutez '0.0.0.0/0' aux plages permises. Ceci permet des demandes de n'importe quelle plage d'IP.

Configurez la doctrine pour intégrer avec des id

Secure Hash Algorithm (SHA1) et configuration de chiffrement dans les id

Pour configurer des id pour se transférer sur SHA1, « $shibboleth_home/conf/idp.properties » ouvert et positionnement :

idp.signing.config = shibboleth.SigningConfiguration.SHA1

Cette configuration peut également être changée :

idp.encryption.optional = rectifient

Si vous le placez pour rectifier, le manque de localiser une clé de chiffrement pour utiliser, quand activé, n'aura pas comme conséquence la panne de demande. Ceci aide à faire le cryptage « opportuniste », c.-à-d., pour chiffrer autant que possible (une clé compatible s'avère dans les métadonnées du pair pour chiffrer avec) mais pour ignorer le cryptage autrement.

Configurez l'uid et user_principal à la réponse SAML

L'AttributeDefinition est ajouté dans **« $shibboleth_home/conf/attribute-resolver.xml »** pour tracer le sAMAccountName et l'userPrincipalName au à l'uid et user_principal dans la réponse SAML.

En outre, ajoutez les configurations de connecteur de LDAP avec le <DataConnector> de balise.

> **Note**: ReturnAttributes doit être spécifié avec la valeur « **userPrincipalName de sAMAccountName** ».

> **Note**: LDAPProperty serait obligatoire au cas où s'il y a une intégration avec un Répertoire actif (AD).

```
<AttributeDefinition xsi:type="Simple" id="ciscoUPN" sourceAttributeID="userPrincipalName">
 <Dependency ref="LDAP" />
 <AttributeEncoder xsi:type="SAML1String" name="user_principal" />
 <AttributeEncoder xsi:type="SAML2String" name="user_principal" friendlyName="user_principal" />
</AttributeDefinition>

<AttributeDefinition xsi:type="Simple" id="ciscoUID" sourceAttributeID="sAMAccountName">
 <Dependency ref="LDAP" />
 <AttributeEncoder xsi:type="SAML1String" name="uid" />
 <AttributeEncoder xsi:type="SAML2String" name="uid" friendlyName="uid" />
</AttributeDefinition>

    <DataConnector id="LDAP" xsi:type="LDAPDirectory"
     ldapURL="ldap://adfsserver.cisco.com"
     baseDN="CN=users,DC=cisco,DC=com"
     principal="administrator@cisco.com"
     principalCredential="<cred>">
        <FilterTemplate>
            <![CDATA[
                %{idp.attribute.resolver.LDAP.searchFilter}
            ]]>
        </FilterTemplate>
      <ReturnAttributes>sAMAccountName userPrincipalName</ReturnAttributes>
      <LDAPProperty name="java.naming.referral" value="follow"/>
    </DataConnector>
```

Incorporez les changements de **« $shibboleth_home/conf/attribute-filter.xml »**

```
<PolicyRequirementRule xsi:type="ANY" />


        <AttributeRule attributeID="ciscoUID">
            <PermitValueRule xsi:type="ANY" />
        </AttributeRule>
```

```
        <AttributeRule attributeID="ciscoUPN">
            <PermitValueRule xsi:type="ANY" />
        </AttributeRule>
```

Changez le **toinclude « $shibboleth_home/conf/saml-nameid.xml »**

```
<bean parent="shibboleth.SAML2AttributeSourcedGenerator"
        p:format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"
        p:attributeSourceIds="#{ {'ciscoUPN'} }" />
<bean parent="shibboleth.SAML2AttributeSourcedGenerator"
        p:format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"
        p:attributeSourceIds="#{ {'ciscoUID'} }" />
```

Métadonnées d'IDP

Les métadonnées d'IDP est disponible dans le répertoire « **$shibboleth_home/metadata** ». Le fichier idp-metadata.xml peut être téléchargé aux id par l'intermédiaire de l'interface de programmation (l'API)

METTEZ https://<idshost>:<idsport>/ids/v1/config/idpmetadata

là où l'**idsport** n'est pas une entité configurable et la valeur est **"8553"**

> **Avertissement :** Les métadonnées de doctrine **peuvent** contenir 2 Certificats de signature, le certificat de signature général et le backchannel. Naviguez vers le fichier **idp-backchannel.crt dans** « $shibboleth_home/credentials **»** pour identifier le certificat de backchannel. **Si le** certificat de canal de retour est disponible dans les métadonnées, vous devriez retirer le certificat de canal de retour du xml de métadonnées avant téléchargement sur des id. C'est parce que la bibliothèque du fedlet 12.0 que les id utilise des supports seulement un certifcate dans les métadonnées. Si plus d'un certificat de signature est disponible, le fedlet utilise le premier certificat disponible.

Configurez les fournisseurs de métadonnées

Nous devons configurer les fournisseurs de métadonnées
avec l'entrée dans **$shibboleth_home/metadata-providers.xml.**

```
<bean parent="shibboleth.SAML2AttributeSourcedGenerator"
        p:format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"
        p:attributeSourceIds="#{ {'ciscoUPN'} }" />
<bean parent="shibboleth.SAML2AttributeSourcedGenerator"
        p:format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"
        p:attributeSourceIds="#{ {'ciscoUID'} }" />
```

là où l'attribut de **" id "** peut être n'importe quel nom unique.

Cette entrée indique qu'un fournisseur de métadonnées est inscrit à l'id donné et les métadonnées est disponible dans le fichier spécifié /opt/shibboleth-idp/SP/sp.xml.

Des métadonnées de fournisseur de services (fournisseur de services) des id doivent être copiées sur le metadataFile spécifié dans l'entrée.

> **Note**: Des métadonnées de fournisseur de services des id peuvent être récupérées par

l'intermédiaire
de **OBTIENNENT https://<idshost>:<idsport>/ids/v1/config/spmetadata,** où l'**idsport** n'est pas une entité configurable et la valeur est **"8553".**

Davantage de configuration pour SSO

Ce document décrit la configuration de l'aspect d'IDP pour que SSO intègre avec la gestion d'identité de Cisco. Pour d'autres détails, référez-vous aux différents guides de configuration de produit :

- UCCX
- UCCE
- PCCE