

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

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Installez](#)

[Configuration système nécessaire](#)

[Configurez](#)

[Intégrez avec un serveur LDAP](#)

[Fichier de configuration témoin](#)

[Permettez les demandes de tous les clients](#)

[Configurez la doctrine pour intégrer avec des id](#)

[Secure Hash Algorithm \(SHA1\) et configuration de chiffrement dans les id](#)

[Configurez l'uid et user principal à la réponse SAML](#)

[Métadonnées d'IDP](#)

[Configurez les fournisseurs de métadonnées](#)

[Davantage de configuration pour SSO](#)

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

Conditions requises

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 Identity (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 Identity (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

Version de doctrine

Site de téléchargement

Installez la plate-forme

Version de Protocole LDAP (Lightweight Directory Access Protocol)

Web server de doctrine

Détails

v3.3.0

<http://shibboleth.net/downloads/identity-pro>

Ubuntu 14.0.4

version "1.8.0_121" de Javas

Répertoire actif 2.0

Apache Tomcat/8.5.12

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

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

Configurez

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

```
#
h
e
u
r
e
e
n
q
u
e
```

l
q
u
e
s
t
i
o
n
n
e
r
e
s
p
o
n
s
e
s

i
d
p
.
a
u
t
h
n
.
L
D
A
P
.
r
e
s
p

O
H
S
E
T
I
H
E
O
U
T

=

P
H
I
L
O
S
O
P
H
Y
C
O
N
F
I
G
U
R
A
T
I
O
N
S
S
L
I
D
E

,
J
V
M
T
R
U
S
T
,
C
E
R
T
I
F
I
C
A
T
E
T
R
U
S
T

,
O
U
K
E
Y
S
T
O
R
E
T
R
U
S
t

i
d
P
.a
u
t
h
n
.L
D
A
P
.S
S
L
C
O
N
F
I
G
=
C
E
R
T
I
F
I
C
A
T
E
T
R
U
S
t

S
I
U

t
i
l
i
s
a
n
t
l
e
c
c
e
r
t
i
f
i
c
a
t
e
T
r
u
s
t
c
i
-
d
e
s
s
u
s
,
p
o
s
i
t
i
o
n
n
e
m
e
n
t
a
u
c
h
e
m
i
n
d
e
c
e
r
t

i
f
i
c
a
t
d
e
c
o
n
f
i
a
n
c
e
i
d
P
. a
u
t
h
n
. L
D
A
P
. t
r
u
s
t
C
e
r
t
i
f
i
c
a
t
e
s
=
%
{
i
d
P
. h
o
m
e
}

/ c r e d e n t i a l s / l d a p - s e r v e r . c r t # s i u t i l i s a n t l e k e y s t o r e T r u s t c i - d e s s u

S
,
P
O
S
I
T
I
O
N
N
E
M
E
N
T
A
U
C
H
E
M
I
N
D
E
T
R
U
S
T
S
T
O
R
E
I
D
P
.
A
U
T
H
N
.
L
D
A
P
.
T
R
U
S
T
S
T
O
R
E
=
=

%
{
i
d
p
.h
o
m
e
}
/
c
r
e
d
e
n
t
i
a
l
s
/
l
d
a
p
-
s
e
r
v
e
r
.t
r
u
s
t
s
t
o
r
e
A
t
t
r
i
b
u
t
s
d
e
r
e
t
o
u
r

d
e

p
e
n
d
a
n
t
l
.
a
u
t
h
e
n
t
i
f
i
c
a
t
i
o
n

i
d
p
.
a
u
t
h
n
.
L
D
A
P
.
r
e
t
u
r
n
A
t
t
r
i
b
u
t
e
s

=

u
s
e
r
P
r
i
n
c
i
p
a
l
N
a
m
e
,
s
A
M
A
c
c
o
u
n
t
N
a
m
e
i
d
P
.a
u
t
h
n
.L
D
A
P
.r
e
t
u
r
n
A
t
t
r
i
b
u
t
e
s

=
*

d
e
P
r
o
P
r
i
é
t
é
s
d
e
r
é
s
o
l
u
t
i
o
n
d
e
D
N
d
e

r
é
s
o
l
u
t
i
o
n
d
e
D
N
d
e
r
e
c
h
e
r
c
h
e
,
u

t
i
l
i
s
é
e
p
a
r
a
n
o
n
s
e
a
r
c
h
A
u
t
h
e
n
t
i
c
a
t
o
r
,
b
i
n
d
s
e
a
r
c
h
A
u
t
h
e
n
t
i
c
a
t
o
r

f
o
r
A
D

:
C
N
=
U
s
e
r
s
,
D
C
=
e
x
a
m
p
l
e
,
D
C
=
o
r
i
g
i
n
a
l
a
u
t
h
n
.
L
D
A
P
.
b
a
s
e
D
N
=
C
N
=
u
s
e
r
s
,
D
C
=
.

c
i
s
c
o
,
D
C
=
c
o
m
i
d
p
.a
u
t
h
n
.L
D
A
P
.s
u
b
t
r
e
e
s
e
a
r
c
h
=
r
e
c
t
i
f
i
e
n
t
*
i
d
p
.a
u
t
h
n
.L

D
A
P
.
u
s
e
r
F
i
l
t
e
r

=

(
S
A
M
A
c
c
o
u
n
t
N
a
m
e
=

{
u
t
i
l
i
s
a
t
e
u
r
}
)
*

c
o
n
f
i
g
u
r
a
t
i
o
n
d

e
r
e
c
h
e
r
c
h
e
d
e
g
r
i
p
p
a
g
e
#

f
o
r
A
D
:
i
d
p
.a
u
t
h
n
.L
D
A
P
.b
i
n
d
N
=
a
d
m
i
n
u
s
e
r
@
d
o
m
a

i
n
.
c
o
m
i
d
p
.
a
u
t
h
n
.
L
D
A
P
.
b
i
n
d
D
N

=

a
d
m
i
n
i
s
t
r
a
t
o
r
@
c
i
s
c
o
.
c
o
m
i
d
p
.
a
u
t
h
n
.
L
D

A
P
.
b
i
n
d
D
N
C
r
e
d
e
n
t
i
a
l

=

C
i
s
c
o
@
1
2
3

r
é
s
o
l
u
t
i
o
n
d
e
D
N
d
e
f
o
r
m
a
t
,
u
t
i
l
i
s
é
e
p

a
r
d
i
r
e
c
t
A
u
t
h
e
n
t
i
c
a
t
o
r
,
a
d
A
u
t
h
e
n
t
i
c
a
t
o
r
#

u
t
i
l
i
s
a
t
i
o
n
i
d
P
.a
u
t
h
n
.L
D
A
P

.
d
n
F
O
r
m
a
t
=
%
s
@
d
o
m
a
i
n
.c
o
m
d
e
f
o
r
A
D

i
d
P
.a
u
t
h
n
.L
D
A
P
.d
n
F
O
r
m
a
t
=
%
d
e
s
@
a

d
f
s
s
e
r
v
e
r
.
c
i
s
c
o
.
c
o
m

l
a
c
o
n
f
i
g
u
r
a
t
i
o
n
d
.
a
t
t
r
i
b
u
t
d
e
L
D
A
P
.
v
o
i
e
n
t
a
t
t
r
i
b

u
t
e
-
r
e
s
o
l
v
e
r
.
x
m
l

l
a
n
o
t
e
,

t
h
i
s
l
i
k
e
l
y
n
e
s
.
a
p
p
l
i
q
u
e
r
a
p
a
s
s
w
o
r
d
.
u
t
i
l
i
s
a
t
i

O
n
d
e
s
c
o
n
f
i
g
u
r
a
t
i
o
n
s
e
x
i
s
t
a
n
t
e
s
d
u
r
e
s
o
l
v
e
u
r
y
2
i
d
p
.a
t
t
r
i
b
u
t
e
.r
e
s
o
l
v
e
r

.
L
D
A
P
. l
d
a
P
U
R
L
=

%
{
i
d
P
. a
u
t
h
n
. L
D
A
P
. l
d
a
P
U
R
L
}
i
d
P
. a
t
t
r
i
b
u
t
e
. r
e
s
o
l
v
e
r
. L

D
A
P
.
c
o
n
n
e
c
t
T
i
m
e
o
u
t
=
%
{
i
d
P
.
a
u
t
h
n
.
L
D
A
P
.
c
o
n
n
e
c
t
T
i
m
e
o
u
t
:
P
T
3
S
y
d
P
.
a
t
t
r

i
b
u
t
e
.
r
e
s
o
l
v
e
r
.
L
D
A
P
.
r
e
s
p
o
n
s
e
T
i
m
e
o
u
t
=
%
{
i
d
p
.
a
u
t
h
n
.
L
D
A
P
.
r
e
s
p
o
n
s
e
T
i
m

e
o
u
t
:
P
T
3
S
}l
d
P
.a
t
t
r
i
b
u
t
e
.r
e
s
o
l
v
e
r
.L
D
A
P
.b
a
s
e
D
N
=
%
{
i
d
P
.a
u
t
h
n
.L
D
A
P
.

b
a
s
e
D
N

:
n
o
n
d
e
f
i
n
i
l
i
d
P
. a
t
t
r
i
b
u
t
e
. r
e
s
o
l
v
e
r
. L
D
A
P
. b
i
n
d
D
N

=

%
{
i
d
P
. a
u
t

h
n
.
L
D
A
P
.
b
i
n
d
D
N

:
n
o
n
d
ē
f
i
n
i
l
l
i
d
P
.
a
t
t
r
i
b
u
t
e
.
r
e
s
o
l
v
e
r
.
L
D
A
P
.
b
i
n
d
D
N
C
r
e
d

e
n
t
i
a
l
=
%
{
i
d
P
.a
u
t
h
n
.L
D
A
P
.b
i
n
d
D
N
C
r
e
d
e
n
t
i
a
l
:
n
o
n
d
e
f
i
n
i
l
i
d
P
.a
t
t
r
i
b
u
t

e
.
r
e
s
o
l
v
e
r
.
L
D
A
P
.
u
s
e
s
t
a
r
t
T
L
S
=
%
{
i
d
p
.
a
u
t
h
n
.
L
D
A
P
.
u
s
e
s
t
a
r
t
T
L
S
:
v
r
a
i
}

d
p
. a
t
t
r
i
b
u
t
e
. r
e
s
o
l
v
e
r
. L
D
A
P
. t
r
u
s
t
C
e
r
t
i
f
i
c
a
t
e
s
=
%
{
i
d
p
. a
u
t
h
n
. L
D
A
P
. t
r

U
S
T
C
E
R
T
I
F
I
C
A
T
E
S

:
N
O
N
D
E
F
I
N
I
T
I
A
L
P
A
T
T
R
I
B
U
T
E
.
R
E
S
O
L
V
E
R
.
L
D
A
P
.
S
E
A
R
C
H
F
I
L

```
t  
e  
r  
=  
(  
s  
A  
M  
A  
c  
c  
o  
u  
n  
t  
N  
a  
m  
e  
=  
$  
r  
e  
s  
o  
l  
u  
t  
i  
o  
n  
C  
o  
n  
t  
e  
x  
t  
.  
p  
r  
i  
n  
c  
i  
p  
a  
l  
)
```

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 = rectifiant
```

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://adfserver.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 certficate 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](#)