安装并且配置口令身份供应商(IdP) Cisco身份服务的(ID)对enable (event) SSO

Contents

Introduction

本文描述在OpenAM身份供应商(IdP)的配置对enable (event)单个符号(SSO)。

Cisco IDS部署模型

产品	配置
UCCX	coresident
PCCE	与CUIC (Cisco Unified智力中心)和LD (实际数据)的共同驻留
UCCE	与CUIC和LD的共同驻留2k配置的。 独立为4k和12k配置。

Prerequisites

Requirements

Cisco 建议您了解以下主题：

- Cisco Unified Contact Center Express (UCCX)版本11.6或Cisco Unified Contact Center Enterprise Release 11.6或者被包的联系中心企业(PCCE)版本11.6如可适用。

  Note:本文参考配置关于Cisco Identitify服务(ID)和身份供应商(IdP)。 本文参考UCCX屏幕画面和示例，然而配置是类似的关于Cisco Identitify服务(UCCX/UCCE/PCCE)和IdP。

Components Used

This document is not restricted to specific software and hardware versions.

The information in this document was created from the devices in a specific lab environment.All of the devices used in this document started with a cleared (default) configuration.如果您的网络实际，请保证您了解所有命令的潜在影响。

安装

口令是提供单一登录功能并且允许站点做出保护的联机资源单个访问的消息灵通的审定决定以保密性保留的方式的开放原始码的软体项目。它支持安全主张标记语言(SAML2)。ID不是SAML2客户端和期望支持与最小的口令或在ID上的变化。在11.6， ID合格与口令IdP一起使用。

> **Note**:本文参考口令版本3.3.0作为鉴定的部分与SSO的

系统要求

组件	详细资料
口令版本	v3.3.0
下载位置	http://shibboleth.net/downloads/identity-provider/
安装平台	Ubuntu 14.0.4 Java版本"1.8.0_121"
轻量级目录访问协议(LDAP)版本	激活目录2.0
口令网络服务器	Apache Tomcat/8.5.12

为口令的安装请参考wiki

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

Configure

集成LDAP服务器

要集成LDAP服务器与口令，字段在$shibboleth_home的**$shibboleth_home/conf/ldap.properties**需要更新(**默认值**是/opt/shibboleth-idp)是指安装目录在口令的安装使用。

字段	预期值
idp.authn.LDAP.trustCertificates	装载信任锚点的资源从，通常一个本地文件以$ {idp.home} /credentials那里idp.home是作为在setenv.sh的JAVA_OPTS被导出的环境变量
idp.authn.LDAP.trustStore	装载包含信任锚点的Java keystore的资源，通常一个本地文件在% {idp/credentials
idp.authn.LDAP.returnAttributes	需要返回LDAPAttributes的逗号被分离的列表。如果要返回所有属性，
idp.authn.LDAP.baseDN	LDAP搜索需要被执行的baseDN
idp.authn.LDAP.subtreeSearch	是否递归搜索
idp.authn.LDAP.userFilter	LDAP搜索过滤器
idp.authn.LDAP.bindDN	捆绑的DN与，当搜索被执行
idp.authn.LDAP.bindDNCredential	捆绑的密码与，当搜索被执行
idp.authn.LDAP.dnFormat	生成用户Dns的一个格式化字符串验证

idp.authn.LDAP.authenticator	控制认证如何的工作流出现LDAP
idp.authn.LDAP.ldapURL	LDAP目录的连接URI

欲了解更详细的信息，请参考：

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

示例配置文件

```
#forresponses
#idp.authn.LDAP.responseTimeout = PT3S
## SSL jvmTrustcertificateTrustkeyStoreTrust
#idp.authn.LDAP.sslConfig = certificateTrust
##certificateTrust
idp.authn.LDAP.trustCertificates = % {idp.home}
/credentials/ldap_server.crt
##keyStoreTrusttruststore
idp.authn.LDAP.trustStore = % {idp.home}
/credentials/ldap-server.truststore
##
#idp.authn.LDAP.returnAttributes = userPrincipalName
sAMAccountName
idp.authn.LDAP.returnAttributes = *
## DN##
#DNanonSearchAuthenticator bindSearchAuthenticator
# forAD Cn=users DC=example DC=org
idp.authn.LDAP.baseDN = Cn=users Dc=cisco Dc=com
idp.authn.LDAP.subtreeSearch =
*idp.authn.LDAP.userFilter = (sAMAccountName= {}) *
#
# forAD idp.authn.LDAP.bindDN= adminuser@domain.com
idp.authn.LDAP.bindDN = administrator@cisco.com
idp.authn.LDAP.bindDNCredential = Cisco@123
#DNdirectAuthenticator adAuthenticator
# forADidp.authn.LDAP.dnFormat=% s@domain.com
#idp.authn.LDAP.dnFormat = % s@adfsserver.cisco.com
# LDAPresolver.xml
# thislikelyV2
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 }
idp.attribute.resolver.LDAP.bindDN = %
{idp.authn.LDAP.bindDN }
idp.attribute.resolver.LDAP.bindDNCredential = %
{idp.authn.LDAP.bindDNCredential }
idp.attribute.resolver.LDAP.useStartTLS = %
{idp.authn.LDAP.useStartTLS }
idp.attribute.resolver.LDAP.trustCertificates = %
{idp.authn.LDAP.trustCertificates }
idp.attribute.resolver.LDAP.searchFilter =
(sAMAccountName=$resolutionContext.principal)
```

允许自所有客户端的请求

要保证自所有客户端的请求到达，需要变化在**“$shibboleth_home/conf/access-control.xml上”**

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

添加**'0.0.0.0/0'**到允许的范围。这允许自所有ip范围的请求。

配置口令集成ID

安全散列算法(SHA1)和在ID的加密配置

配置ID默认为SHA1、开放**"$shibboleth_home/conf/idp.properties"和**集：

idp.signing.config = shibboleth.SigningConfiguration.SHA1

可能也更改此配置：

idp.encryption.optional =真

如果设置它对真，疏忽找出加密密钥使用，当启用，不会导致请求故障。即，这帮助执行加密"机会主义"加密若情况许可(一个兼容的键在对等体的元数据被找到加密与)，但是否则跳过加密。

配置uid和user_principal对SAML回应

AttributeDefinition在**"$shibboleth_home/conf/attribute-resolver.xml"**被添加映射sAMAccountName和userPrincipalName到到uid和user_principal在SAML回应。

另外，请添加与标记<DataConnector>的ldap连接器设置。

> **Note**: ReturnAttributes需要用值"**sAMAccountName userPrincipalName**"指定。

> **Note**: 如果有与激活目录(AD)的集成LDAPProperty是必须的，万一。

```
<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>
```

合并在**"$shibboleth_home/conf/attribute-filter.xml上的"**变化

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

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

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

更改 **“$shibboleth_home/conf/saml-nameid.xml” toinclude**

```
<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'} }" />
```

IdP元数据

IdP元数据是可用的在文件夹“**$shibboleth_home/metadata**”。idp-metadata.xml文件可以被加载到ID通过Application Programming Interface (API)

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

那里**idsport不**一个可配置实体和值是**"8553"**

> **警告**：口令元数据**能**包含2签署的证书、一般签署的认证和backchannel。连接对**在**“$shibboleth_home/credentials的**”**文件**idp-backchannel.crt**识别backchannel认证。 **如果**反向信道认证是可用的在元数据，您应该从元数据xml删除反向信道认证在加载前到ID。这是因为ID在元数据使用技术支持仅一certifcate的fedlet 12.0库。如果超过一个签署的认证是可用的，fedlet使用第一个可用的认证。

配置元数据供应商

我们需要用在**$shibboleth_home/metadata-providers.xml的**条目配置元数据供应商**。**

```
<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'} }" />
```

那里**" id "**属性可以是所有唯一名字。

此条目表明元数据供应商向特定id登记，并且元数据是可用的在指定的文件/opt/shibboleth-idp/SP/sp.xml。

必须复制ID服务提供商(SP)元数据到在条目指定的metadataFile。

> **Note**: ID SP元数据可以通过**GET https://<idshost>:<idsport>/ids/v1/config/spmetadata**被检索**，idsport不**是一个可配置实体，并且值是**"8553"。**

SSO的进一步配置

本文描述从IdP方面的配置SSO的能集成Cisco身份服务。关于更详细的资料，请参见单个产品配置指南：

- UCCX
- UCCE
- PCCE