SSO를 활성화하기 위해 Cisco Identity Service(Id)용 Shibboleth IdP(Identity Provider)를 설치하고 구성합니다

목차

소개

이 문서에서는 SSO(Single Sign On)를 활성화하기 위한 OpenAM IdP(Identity Provider)의 컨피그레이션에 대해 설명합니다.

Cisco Id 구축 모델

제품	구축
UCCX	공동 거주자
PCCE	CUIC(Cisco Unified Intelligence Center) 및 LD(Live Data)와 공동 상주
UCCE	2k 구축을 위해 CUIC 및 LD와 공동 상주 4k 및 12k 구축을 위한 독립형

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- Cisco UCCX(Unified Contact Center Express) 릴리스 11.6 또는 Cisco Unified Contact Center Enterprise 릴리스 11.6 또는 PCCE(Packaged Contact Center Enterprise) 릴리스 11.6이 해당됩니다.

참고: 이 문서는 Cisco Id(Identify Service) 및 IdP(Identity Provider)와 관련된 구성을 참조합니다. 이 문서는 스크린샷과 예에서 UCCX를 참조하지만, Cisco Id Service(UCCX/UCCE/PCCE) 및 IdP와 관련된 구성은 유사합니다.

사용되는 구성 요소

이 문서는 특정 소프트웨어 및 하드웨어 버전으로 한정되지 않습니다.

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

Install

Shibboleth는 SSO(Single Sign-On) 기능을 제공하고 사이트에서 개인 정보 보호 방식의 보호된 온라인 리소스에 대한 개별 액세스에 대해 정보에 근거한 권한 부여 결정을 내릴 수 있도록 지원하는 오픈 소스 프로젝트입니다. SAML2(Security Assertion Markup Language)를 지원합니다. IdS는 SAML2 클라이언트이며 Id의 변경을 최소화하거나 전혀 수행하지 않고 Shibboleth를 지원할 것으로 예상됩니다. 11.6에서 Id는 Shibboleth IdP를 사용할 수 있는 자격이 부여되었습니다.

참고: 이 문서에서는 Shibboleth 릴리스 3.3.0을 SSO 자격 증명의 일부로 참조합니다

시스템 요구 사항

구성 요소	세부사항
Shibboleth 버전	v3.3.0
다운로드 위치	http://shibboleth.net/downloads/identity-provider/
플랫폼 설치	Ubuntu 14.0.4 java 버전 "1.8.0_121"
LDAP(Lightweight Directory Access Protocol) 버전	Active Directory 2.0
Shibboleth 웹 서버	아파치 톰캣/8.5.12

Shibboleth의 설치를 위키를 참조하십시오.

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

구성

LDAP 서버와 통합

LDAP 서버를 shibboleth와 통합하려면 $shibboleth_home/conf/ldap.properties에서 필드를 업데이트해야 합니다. 여기서$shibboleth_home(기본값은 /opt/shibboleth-idp)은 shibboleth 설치에 사용되는 설치 디렉토리를 나타냅니다.

필드	예상 값	설명
idp.authn.LDAP.trustCertificates	일반적으로 ${idp.home}/credentials의 로컬 파일로부터 트러스트 앵커를 로드할 리소스 여기서 idp.home은 setenv.sh에서 JAVA_OPTS로 내보낸 환경 변수입니다	%{idp.home}/credentials/ldap-server.crt
idp.authn.LDAP.trustStore	트러스트 앵커를 포함하는 Java 키 저장소를 로드하는 리소스입니다. 일반적으로 %{idp.home}/credentials의 로컬 파일입니다.	%{idp.home}/credentials/ldap-server.truststore
idp.authn.LDAP.returnAttributes	반환해야 하는 쉼표로 구분된 LDAPAttributes 목록입니다. 모든 특성을 반환하려면 "*"를 추가합니다.	*
idp.authn.LDAP.baseDN	LDAP 검색을 수행해야 하는 baseDN	CN=users,DC=cisco,DC=com
idp.authn.LDAP.subtreeSearch	재귀적으로 검색할지 여부	참
idp.authn.LDAP.userFilter	LDAP 검색 필터	(sAMAccountName={user})*
idp.authn.LDAP.bindDN	검색을 수행할 때 바인딩할 DN	administrator@cisco.com
idp.authn.LDAP.bindDNCredential	검색을 수행할 때 바인딩할 비밀번호	
idp.authn.LDAP.dnFormat	인증할 사용자 DN을 생성하기 위한 서식 문자열	%s@adfsserver.cisco.com (%s@domainname)
idp.authn.LDAP.authenticator	LDAP에 대해 인증이 수행되는 방식에 대한 워크플로를 제어합니다.	바인드검색인증자

idp.authn.LDAP.ldapURL	LDAP 디렉터리에 대한 연결 URI	

자세한 내용은 다음을 참조하십시오.

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

샘플 구성 파일

```
# 기다리는 시간(밀리초) 대상:응답
#idp.authn.LDAP.responseTimeout = PT3S
## SSL 구성(jvmTrust, certificateTrust 또는 keyStoreTrust)
#idp.authn.LDAP.sslConfig = certificateTrust
## 위의 certificateTrust를 사용하는 경우 신뢰할 수 있는 인증서의 경로로 설정합니다.
idp.authn.LDAP.trustCertificates = %{idp.home}/credentials/ldap-server.crt
## 위의 keyStoreTrust를 사용하는 경우 truststore 경로로 설정합니다.
idp.authn.LDAP.trustStore = %{idp.home}/credentials/ldap-server.truststore
## 인증 중 특성 반환
#idp.authn.LDAP.returnAttributes = 사용자 계정 이름, sAMAccountName
idp.authn.LDAP.returnAttributes = *
## DN 확인 속성 ##
# 검색 DN 확인, anonSearchAuthenticator, bindSearchAuthenticator에서 사용
# 대상:광고: CN=Users,DC=example,DC=org
idp.authn.LDAP.baseDN = CN=users,DC=cisco,DC=com
idp.authn.LDAP.subtreeSearch = 참
*idp.authn.LDAP.userFilter = (sAMAccountName={user})*
# bind 검색 컨피그레이션
# 대상:광고: idp.authn.LDAP.bindDN=adminuser@domain.com
idp.authn.LDAP.bindDN = 관리자@cisco.com
idp.authn.LDAP.bindDNCredential = Cisco@123
# directAuthenticator 및 adAuthenticator에서 사용되는 형식 DN 확인
# 대상:AD는 idp.authn.LDAP.dnFormat=%s을(를) 사용합니다.@domain.com
#idp.authn.LDAP.dnFormat = %s@adfsserver.cisco.com
# LDAP 특성 구성, attribute-resolver.xml 참조
# 참고, 이것레거시 V2 레졸버 컨피그레이션의 사용에는 적용되지 않을 수 있습니다.
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:undefined}
idp.attribute.resolver.LDAP.bindDN = %{idp.authn.LDAP.bindDN:undefined}
idp.attribute.resolver.LDAP.bindDNCredential = %{idp.authn.LDAP.bindDNCredential:undefined}
idp.attribute.resolver.LDAP.useStartTLS = %{idp.authn.LDAP.useStartTLS:참}
idp.attribute.resolver.LDAP.trustCertificates = %{idp.authn.LDAP.trustCertificates:undefined}
idp.attribute.resolver.LDAP.searchFilter = (sAMAccountName=$resolutionContext.principal)
```

모든 클라이언트의 요청 허용

모든 클라이언트의 요청이 도달할 수 있도록 하려면 "$shibboleth_home/conf/access-control.xml"에서 변경해야 합니다.

<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>

허용되는 범위에 '0.0.0.0/0'을 추가합니다. 이는 모든 ip 범위에서 요청을 허용합니다.

Id와 통합하도록 Shibboleth 구성

SHA1(Secure Hash Algorithm) 및 Id의 암호화 컨피그레이션

ID를 SHA1의 기본값으로 구성하려면 "$shibboleth_home/conf/idp.properties"을 열고 다음을 설정합니다.

idp.signing.config = shibboleth.SigningConfiguration.SHA1

이 컨피그레이션도 변경할 수 있습니다.

idp.encryption.optional = true

true로 설정하면 사용할 암호화 키를 찾지 못하면 요청이 실패합니다. 이 h암호화를 "기회주의적"으로 수행할 수 있습니다. 즉, 가능하면 암호화하되(암호화할 피어의 메타데이터에서 호환되는 키가 발견됨), 그렇지 않으면 암호화를 건너뜁니다.

SAML 응답에 uid 및 user_principal 구성

AttributeDefinition이 "$shibboleth_home/conf/attribute-resolver.xml"에 추가되어 sAMAccountName 및 userPrincipalName을 SAML 응답의 uid 및 user_principal에 매핑합니다.

또한 <DataConnector> 태그와 함께 ldap 커넥터 설정을 추가합니다.

참고: ReturnAttributes는 "sAMAccountName userPrincipalName" 값으로 지정해야 합니다.

참고: AD(Active Directory)와의 통합이 있는 경우 LDAPProperty는 필수입니다.

<#root>

%{idp.attribute.resolver.LDAP.searchFilter}

sAMAccountName userPrincipalName

"$shibboleth_home/conf/attribute-filter.xml"의 변경 사항 통합

<#root>

"$shibboleth_home/conf/saml-nameid.xml"을 (를) 다음으로 변경합니다

<#root>

IdP 메타데이터

IdP 메타데이터는 "$shibboleth_home/metadata" 폴더에서 사용할 수 있습니다. idp-metadata.xml 파일은 API(Application Programming Interface)를 통해 ID에 업로드할 수 있습니다

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

여기서 idsport는 구성 가능한 엔티티가 아니며 값은 "8553"입니다.

> 경고: Shibboleth 메타데이터는 2개의 서명 인증서, 일반 서명 인증서 및 백채널을 포함할 수 있습니다. "$shibboleth_home/credentials"의 idp-backchannel.crt 파일로 이동하여 백채널 인증서를 식별합니다. 메타데이터에서 백채널 인증서를 사용할 수 있는 경우 IdS로 업로드하기 전에 메타데이터 xml에서 백채널 인증서를 제거해야 합니다. 이는 IdS가 사용하는 Fedlet 12.0 라이브러리가 메타데이터에서 하나의 인증서만 지원하기 때문입니다. 둘 이상의 서명 인증서를 사용할 수 있는 경우, fedlet은 사용 가능한 첫 번째 인증서를 사용합니다.

메타데이터 공급자 구성

$shibboleth_home/metadata-providers.xml의 항목으로 메타데이터 공급자를 구성해야 합니다.

```
<MetadataProvider id="smart-86"  xsi:type="FilesystemMetadataProvider" metadataFile="/opt/shibboleth-id
```

여기서"id" 특성은 고유한 이름일 수 있습니다.

이 항목은 메타데이터 공급자가 지정된 ID로 등록되었으며 지정된 파일 /opt/shibboleth-idp/SP/sp.xml에서 메타데이터를 사용할 수 있음을 나타냅니다.

Id의 SP(서비스 공급자) 메타데이터를 항목에 지정된 metadataFile에 복사해야 합니다.

> 참고: ID의 SP 메타데이터는 GET https://<ishost>:<idsport>/ids/v1/config/spmetadata를 통해 검색할 수 있습니다. 여기서 idsport는 구성 가능한 엔티티가 아니며 값은 "8553"입니다.

SSO에 대한 추가 컨피그레이션

이 문서에서는 SSO가 Cisco Identity Service와 통합되도록 IdP 측면에서 구성하는 방법을 설명합니다. 자세한 내용은 개별 제품 컨피그레이션 가이드를 참조하십시오.

- UCCX
- UCCE
- PCCE

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.