

Microsoft AD와 CUAC 통합

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[CUAC와 AD 통합 및 AD에서 사용자 가져오기](#)

[CUAC와 AD 간의 LDAP 기능](#)

[LDAP 프로세스 요약](#)

[LDAP 프로세스 세부 정보](#)

소개

이 문서에서는 CUAC(Cisco Unified Attendant Console)와 Microsoft AD(Active Directory) 간에 LDAP(Lightweight Directory Access Protocol)가 작동하는 방식과 두 시스템을 통합하는 데 사용되는 절차에 대해 설명합니다.

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- CUCM
- CUAC
- LDAP
- 광고

사용되는 구성 요소

이 문서의 정보는 CUAC 버전 10.x를 기반으로 합니다.

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

배경 정보

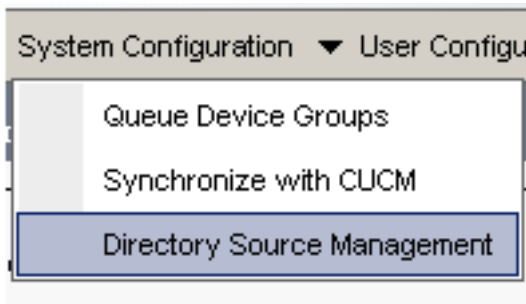
이전 CUAC 버전에서는 서버가 사전 정의된 쿼리 및 필터를 통해 Cisco CUCM(Unified Communications Manager)에서 직접 사용자를 가져옵니다. 관리자는 CUACPE Premium Edition(CUACPE)을 사용하여 AD에서 직접 사용자를 통합하고 가져올 수 있습니다. 이를 통해 관리자는 자신의 선택 및 요구 사항에 맞는 특성 및 필터를 유연하게 구현할 수 있습니다.

참고: 이제 CUACPE가 CUAC Advanced Edition for Versions 10 이상으로 교체되었습니다.

CUAC와 AD 통합 및 AD에서 사용자 가져오기

CUAC를 AD와 통합하고 AD에서 사용자를 가져오려면 다음 단계를 완료하십시오.

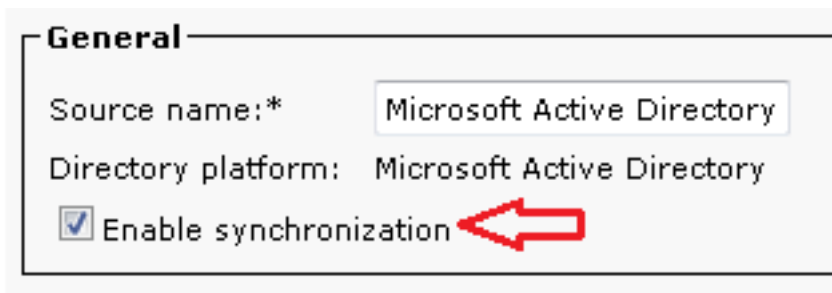
1. CUAC에서 AD에 대한 디렉터리 동기화를 활성화합니다.



2. Microsoft Active Directory를 선택하고 동기화 사용 확인란을 선택합니다.

- Directory Sources

	Source Name
Select	CCMSource
Select	Microsoft Active Directory
Select	iPlanet



3. Active Directory 서버에 대한 구성 세부 정보를 입력합니다.

Connection

Host name or IP:* 10.106.98.209

Host port:* 389 (0-65)

Use SSL

이 예에서는 administrator@aloksin.lab가 인증에 사용됩니다.

Authentication

Username:* administrator@aloksin.lab

Password:* ●●●●●●●●

4. Property Settings(속성 설정) 섹션에서 Unique(고유) 속성에 대한 구성 세부 정보를 입력합니다. 이 세부 정보는 다른 세부 정보를 입력하고 Save(저장)를 클릭하면 나타납니다.

Property Settings

Unique property: sAMAccountName ▼

Native property

참고:AD의 각 항목에 대한 고유 값입니다.중복 값이 있는 경우 CUAC는 하나의 항목만 가져옵니다.

5. Container(컨테이너) 섹션에서 AD의 사용자 검색 범위인 Base DN에 대한 컨피그레이션 세부 사항을 입력합니다.

Object 클래스 필드는 요청된 검색 범위를 결정하기 위해 AD에서 사용됩니다.기본적으로 **연락처**로 설정됩니다. 즉 AD는 요청된 검색 기반에서 **연락처**(사용자가 아님)를 찾습니다. CUAC에서 **사용자**를 가져오려면 Object 클래스 설정을 **사용자**로 변경합니다.

- Container

Base DN:* dc=aloksin,dc=lab

Object class:* user (Case)

Scope: Sub Tree Level ▼

6. 설정을 저장하고 **디렉토리 필드 매핑**을 누르고 모든 사용자에게 대해 임포트하고자 하는 모든 속성을 구성합니다. 이 예에서 사용되는 구성은 다음과 같습니다.

Source Fields	Destination Fields	Default
telephoneNumber	Extension	
mail	Email	
givenName	First Name	
sn	Last Name	


7. 디렉토리 소스 페이지로 이동하고 디렉토리 규칙을 누릅니다.

inner

DN:*

class:* (Case Sensitive)

▼




8. Add **New**를 클릭하고 규칙을 생성합니다. 디렉토리 규칙을 추가하면 기본적으로 규칙 필터가 나타납니다.

Field	Operator	Value
telephoneNumber	=	*

참고: 규칙 필터를 변경할 필요가 없습니다. 전화 번호가 구성된 모든 사용자를 가져옵니다.

9. AD와 자동 동기화를 구성하려면 디렉터리 동기화 탭을 클릭합니다.

▼



10. 이제 구성이 완료되었습니다. Engineering(엔지니어링) > Service Management(서비스 관리)로 이동하고 LDAP 플러그인을 다시 시작하여 수동으로 동기화를 시작합니다.

CUAC와 AD 간의 LDAP 기능

LDAP 프로세스 요약

다음은 CUAC와 AD 간의 LDAP 프로세스 요약입니다.

1. 두 서버(CUAC 및 AD) 간에 TCP 세션이 설정됩니다.
2. CUAC는 AD에 BIND 요청을 전송하고 인증 설정에 구성된 사용자를 통해 인증합니다.

3. AD는 사용자를 인증하면 CUACPE에 BIND Success 알림을 보냅니다.
4. CUAC는 검색 범위 정보, 검색 필터, 필터링된 사용자에 대한 특성이 있는 AD에 SEARCH 요청을 보냅니다.
5. AD는 검색 기반에서 요청된 개체(Object Class 설정에서 구성)를 검색합니다.SEARCH 요청 메시지에 자세히 나와 있는 기준(필터)과 일치하는 객체를 필터링합니다.
6. AD는 검색 결과로 CUAC에 응답합니다.

다음은 이러한 단계를 보여 주는 스니퍼 캡처입니다.

```

3.208 10.106.98.209 TCP 49992 > ldap [SYN Seq=0 win=8192 Len=0 MSS=1460 WS=8
3.209 10.106.98.208 TCP ldap > 49992 [SYN, ACK] Seq=0 Ack=1 win=8192 Len=0 MS
3.208 10.106.98.209 TCP 49992 > ldap [ACK] Seq=1 Ack=1 win=65536 Len=0
3.208 10.106.98.209 LDAP bindRequest(3) "administrator@aloksin.lab" simple
3.209 10.106.98.208 LDAP bindResponse(3) success
3.208 10.106.98.209 LDAP searchRequest(4) "dc=aloksin,dc=lab" wholesubtree
3.209 10.106.98.208 LDAP searchResEntry(4) "CN=Suhail Angi,CN=Users,DC=aloksi

```

LDAP 프로세스 세부 정보

CUAC의 컨피그레이션이 완료되고 LDAP 플러그인이 다시 시작되면 CUAC 서버는 AD를 사용하여 TCP 세션을 설정합니다.

그런 다음 CUAC는 AD 서버를 인증하기 위해 BIND 요청을 보냅니다.인증에 성공하면 AD는 CUAC에 BIND Success 응답을 보냅니다.이를 통해 두 서버 모두 사용자 및 사용자 정보를 동기화 하기 위해 포트 389에 세션을 설정하려고 합니다.

다음은 BIND 트랜잭션의 인증에 사용되는 고유 이름을 정의하는 서버의 컨피그레이션입니다.

Authentication

Username:*

Password:*

이러한 메시지는 패킷 캡처에 나타납니다.

- 다음은 TCP 핸드셰이크와 BIND 요청입니다.

```

98.208 10.106.98.209 TCP 50190 > ldap [SYN Seq=0 win=8192 Len=0 MSS=1460 WS=8
98.209 10.106.98.208 TCP ldap > 50190 [SYN, ACK] Seq=0 Ack=1 win=8192 Len=0 MS
98.208 10.106.98.209 TCP 50190 > ldap [ACK] Seq=1 Ack=1 win=65536 Len=0
98.208 10.106.98.209 LDAP bindRequest(3) "administrator@aloksin.lab" simple
98.209 10.106.98.208 LDAP bindResponse(3) success

```

- 다음은 BIND 요청의 확장입니다.

```

⊖ Lightweight Directory Access Protocol
  ⊖ LDAPMessage bindRequest(3) "administrator@aloksin.lab" simple
    messageID: 3
  ⊖ protocolOp: bindRequest (0)
    ⊖ bindRequest
      version: 3
      name: administrator@aloksin.lab
    ⊖ authentication: simple (0)
      simple: 633173633031323321
    [Response To: 81]

```

- 다음은 BIND 응답의 확장입니다. 이는 사용자(이 예에서 **administrator**)의 성공적인 인증을 나타냅니다.

```

⊖ Lightweight Directory Access Protocol
  ⊖ LDAPMessage bindResponse(3) success
    messageID: 3
  ⊖ protocolOp: bindResponse (1)
    ⊖ bindResponse
      resultCode: success (0)
      matchedDN:
      errorMessage:
    [Response To: 80]
    [Time: 0.002073000 seconds]

```

바인딩이 성공하면 서버는 사용자를 가져오기 위해 AD에 SEARCH 요청을 보냅니다. 이 검색 요청에는 AD에서 사용하는 필터 및 특성이 포함되어 있습니다. 그런 다음 AD는 정의된 검색 기준 (SEARCH 요청 메시지에 자세히 설명되어 있음)에서 사용자를 검색합니다. 이는 필터와 속성 확인의 기준을 충족합니다.

다음은 CUCM에서 전송하는 SEARCH 요청의 예입니다.

```

Lightweight Directory Access Protocol
LDAPMessage searchRequest(2) "dc=aloksin,dc=lab" wholeSubtree
messageID: 2
protocolOp: searchRequest (3)
searchRequest
  baseObject: dc=aloksin,dc=lab
  scope: wholeSubtree (2)
  derefAliases: derefAlways (3)
  sizeLimit: 0
  timeLimit: 0
  typesOnly: False
  Filter: (&(&(objectclass=user)!(objectclass=Computer)))
  (! (UserAccountControl:1.2.840.113556.1.4.803:=2))
  filter: and (0)
    and: (&(&(objectclass=user)!(objectclass=Computer)))
  (! (UserAccountControl:1.2.840.113556.1.4.803:=2))
    and: 3 items
      Filter: (objectclass=user)
        and item: equalityMatch (3)
          equalityMatch
            attributeDesc: objectclass
            assertionValue: user

```

```

Filter: (!(objectclass=Computer))
  and item: not (2)
    Filter: (objectclass=Computer)
      not: equalityMatch (3)
        equalityMatch
          attributeDesc: objectclass
          assertionValue: Computer
Filter: (!(UserAccountControl:1.2.840.113556.1.4.
803:=2))
  and item: not (2)
    Filter: (UserAccountControl:1.2.840.113556
.1.4.803:=2)
      not: extensibleMatch (9)
        extensibleMatch UserAccountControl
          matchingRule: 1.2.840.113556.
1.4.803
          type: UserAccountControl
          matchValue: 2
          dnAttributes: False

```

attributes: 15 items

```

AttributeDescription: objectguid
AttributeDescription: samaccountname
AttributeDescription: givenname
AttributeDescription: middlename
AttributeDescription: sn
AttributeDescription: manager
AttributeDescription: department
AttributeDescription: telephonenumber
AttributeDescription: mail
AttributeDescription: title
AttributeDescription: homephone
AttributeDescription: mobile
AttributeDescription: pager
AttributeDescription: msrtcsip-primaryuseraddress
AttributeDescription: msrtcsip-primaryuseraddress

```

[Response In: 103]

controls: 1 item

Control

```

controlType: 1.2.840.113556.1.4.319 (pagedResultsControl)
criticality: True
SearchControlValue
  size: 250
  cookie: <MISSING>

```

AD는 CUCM에서 이 요청을 받으면 baseObject에서 사용자를 검색합니다.dc=aloksin,dc=lab - 필터를 만족합니다.필터에 자세히 설명된 요구 사항을 충족하지 않는 사용자는 제외됩니다.AD는 필터링된 모든 사용자로 CUCM에 응답하고 요청된 특성의 값을 전송합니다.

참고:개체를 가져올 수 없습니다.사용자만 가져옵니다.이는 SEARCH 요청 메시지에서 보낸 필터에 objectclass=user가 포함되기 때문입니다.따라서 AD는 연락처가 아닌 사용자만 검색합니다.CUCM에는 기본적으로 이러한 모든 매핑과 필터가 있습니다.

CUAC는 기본적으로 구성되지 않습니다.사용자에 대한 특성을 가져오기 위해 구성된 매핑 세부 정보가 없으므로 이러한 세부 정보를 수동으로 입력해야 합니다.이러한 매핑을 생성하려면 System Configuration(시스템 컨피그레이션) > Directory Source Management(디렉토리 소스 관리) > Active Directory > Directory Field Mapping(디렉토리 필드 매핑)으로 이동합니다.

관리자는 자신의 요구 사항에 따라 필드를 매핑할 수 있습니다.예를 들면 다음과 같습니다.

Directory Source				
Microsoft Active Directory				
Field Mappings				
		Source Fields	Destination Fields	Default Value
<input type="checkbox"/>	Select	telephoneNumber	Extension	
<input type="checkbox"/>	Select	mail	Email	
<input type="checkbox"/>	Select	givenName	First Name	
<input type="checkbox"/>	Select	sn	Last Name	

소스 필드 정보는 SEARCH 요청 메시지에서 AD로 전송됩니다. AD가 SEARCH 응답 메시지를 보낼 때 이러한 값은 CUACPE의 Destination Fields에 저장됩니다.

기본적으로 CUAC에는 Object Class(개체 클래스)가 *연락처*로 설정되어 있습니다. 이 기본 설정을 사용하면 AD로 전송되는 필터가 다음과 같이 나타납니다.

Filter: (&(&(objectclass=**contact**)(.....))

이 필터를 사용하면 AD는 *사용자*가 아니라 검색 기반에서 *연락처*를 검색하므로 CUACPE에 어떤 사용자도 반환하지 않습니다. 따라서 객체 클래스를 *사용자*로 변경해야 합니다.

Container

Base DN:*

Object class:* (Case Sensitive)

Scope:

이 시점까지 CUAC에서 이러한 설정이 구성되었습니다.

- 연결 세부 정보
- 인증(바인딩용 고유 사용자)
- 컨테이너 설정
- 디렉터리 매핑

이 예에서 Unique 속성은 sAMAccountName으로 구성됩니다. CUAC에서 LDAP 플러그인을 다시 시작하고 SEARCH 요청 메시지를 확인하면 ObjectClass=user를 제외한 특성 또는 필터가 포함되지 않습니다.

```

Lightweight Directory Access Protocol
  LDAPMessage searchRequest(224) "dc=aloksin,dc=lab" wholeSubtree
    messageID: 224
    protocolOp: searchRequest (3)
      searchRequest
        baseObject: dc=aloksin,dc=lab
        scope: wholeSubtree (2)
        derefAliases: neverDerefAliases (0)
        sizeLimit: 1
        timeLimit: 0
        typesOnly: True
        Filter: (ObjectClass=user)
          filter: equalityMatch (3)
            equalityMatch
              attributeDesc: ObjectClass
              assertionValue: user
          attributes: 0 items
  
```


[Response In: 43]

디렉토리 규칙은 여기에 없습니다.연락처를 AD와 동기화하려면 규칙을 만들어야 합니다.기본적으로 구성된 디렉토리 규칙이 없습니다.필터가 만들어지면 바로 필터가 이미 있습니다. 전화 번호가 있는 모든 사용자를 가져와야 하므로 필터를 변경할 필요가 없습니다.

Field	Operator	Value
telephoneNumber	=	*

AD와의 동기화를 시작하고 사용자를 가져오려면 LDAP 플러그인을 다시 시작합니다.다음은 CUAC의 SEARCH 요청입니다.

```

Lightweight Directory Access Protocol
  LDAPMessage searchRequest(4) "dc=aloksin,dc=lab" wholeSubtree
    messageID: 4
    protocolOp: searchRequest (3)
    searchRequest
      baseObject: dc=aloksin,dc=lab
      scope: wholeSubtree (2)
      derefAliases: neverDerefAliases (0)
      sizeLimit: 0
      timeLimit: 15
      typesOnly: False
      Filter: (&(&(objectclass=user)(telephoneNumber=*))
(!((UserAccountControl:1.2.840.113556.1.4.803:=2)))
        filter: and (0)
          and: (&(&(objectclass=user)(telephoneNumber=*))
(!((UserAccountControl:1.2.840.113556.1.4.803:=2)))
            and: 3 items
              Filter: (objectclass=user)
                and item: equalityMatch (3)
                  equalityMatch
                    attributeDesc: objectclass
                    assertionValue: user
              Filter: (telephoneNumber=*)
                and item: present (7)
                  present: telephoneNumber
              Filter: (!(UserAccountControl:1.2.840.113556.
1.4.803:=2))
                and item: not (2)
                  Filter: (UserAccountControl:1.2.840.113556.
1.4.803:=2)
                    not: extensibleMatch (9)
                      extensibleMatch UserAccountControl
                        matchingRule: 1.2.840.113556.1.
4.803
                        type: UserAccountControl
                        matchValue: 2
                        dnAttributes: False
            attributes: 10 items
              AttributeDescription: TELEPHONENUMBER
              AttributeDescription: MAIL
              AttributeDescription: GIVENNAME
              AttributeDescription: SN
              AttributeDescription: sAMAccountName
              AttributeDescription: ObjectClass
              AttributeDescription: whenCreated
              AttributeDescription: whenChanged
              AttributeDescription: uSNCreated
              AttributeDescription: uSNChanged

```

[Response In: 11405]

```
controls: 1 item
  Control
    controlType: 1.2.840.113556.1.4.319 (pagedResultsControl)
    SearchControlValue
      size: 500
      cookie: <MISSING>
```

AD가 SEARCH 요청 메시지에 자세히 나와 있는 기준과 일치하는 사용자를 찾으면 사용자 정보가 포함된 *SearchResEntry* 메시지를 보냅니다.

8.208	10.106.98.209	TCP	49992 > 1dap [SYN] Seq=0 win=8192 Len=0 MSS=1460 wS=8 SACK_PERM=1
8.209	10.106.98.208	TCP	1dap > 49992 [SYN, ACK] Seq=0 Ack=1 win=8192 Len=0 MSS=1460 wS=8 SACK_PERM=1
8.208	10.106.98.209	TCP	49992 > 1dap [ACK] Seq=1 Ack=1 win=65536 Len=0
8.208	10.106.98.209	LDAP	bindRequest(3) "administrator@aloksin.lab" simple
8.209	10.106.98.208	LDAP	bindResponse(3) success
8.208	10.106.98.209	LDAP	searchRequest(4) "dc=aloksin,dc=lab" wholeSubtree
8.209	10.106.98.208	LDAP	searchResEntry(4) "CN=Suhail Angi,CN=Users,DC=aloksin,DC=lab" searchResEntry(4) "CN=Pra
8.209	10.106.98.208	LDAP	searchResRef(4)
8.208	10.106.98.209	TCP	49992 > 1dap [ACK] Seq=389 Ack=1555 win=65536 Len=0

SearchResEntry 메시지는 다음과 같습니다.

```
Lightweight Directory Access Protocol
  LDAPMessage searchResEntry(4) "CN=Suhail Angi,CN=Users,DC=aloksin,DC=lab" [4 results]
    messageID: 4
    protocolOp: searchResEntry (4)
      searchResEntry
        objectName: CN=Suhail Angi,CN=Users,DC=aloksin,DC=lab
        attributes: 9 items
          PartialAttributeList item objectClass
            type: objectClass
            vals: 4 items
              top
              person
              organizationalPerson
              user
          PartialAttributeList item sn
            type: sn
            vals: 1 item
              Angi
          PartialAttributeList item telephoneNumber
            type: telephoneNumber
            vals: 1 item
              1002
          PartialAttributeList item givenName
            type: givenName
            vals: 1 item
              Suhail
          PartialAttributeList item whenCreated
            type: whenCreated
            vals: 1 item
              20131222000850.0Z
          PartialAttributeList item whenChanged
            type: whenChanged
            vals: 1 item
              20131222023413.0Z
          PartialAttributeList item uSNCreated
            type: uSNCreated
            vals: 1 item
              12802
          PartialAttributeList item uSNChanged
            type: uSNChanged
            vals: 1 item
              12843
          PartialAttributeList item SAMAccountName
```

```

        type: sAMAccountName
        vals: 1 item
            sangi
[Response To: 11404]
[Time: 0.001565000 seconds]
Lightweight Directory Access Protocol
LDAPMessage searchResEntry(4) "CN=Pragathi NS,CN=Users,DC=aloksin,DC=lab" [5 results]
messageID: 4
protocolOp: searchResEntry (4)
searchResEntry
    objectName: CN=Pragathi NS,CN=Users,DC=aloksin,DC=lab
    attributes: 9 items
        PartialAttributeList item objectClass
            type: objectClass
            vals: 4 items
                top
                person
                organizationalPerson
                user
        PartialAttributeList item sn
            type: sn
            vals: 1 item
                NS
        PartialAttributeList item telephoneNumber
            type: telephoneNumber
            vals: 1 item
                1000
            .....
            ....{message truncated}.....
            .....

```

참고:이 특성이 요청되더라도 응답에 MAIL이 없습니다. 이는 AD의 사용자에게 대해 MAIL ID가 구성되지 않았기 때문입니다.

CUAC에서 이러한 값을 수신하면 SQL(Structured Query Language) 테이블에 저장됩니다. 그런 다음 콘솔에 로그인하면 콘솔이 CUACPE 서버의 이 SQL 테이블에서 사용자 목록을 가져옵니다.