# Integração do CUAC com o Microsoft AD

## Contents



## Introduction

Este documento descreve a forma como o LDAP (Lightweight Diretory Access Protocol) funciona entre o Cisco Unified Attendant Console (CUAC) e o Microsoft Ative Diretory (AD) e os procedimentos usados para integrar os dois sistemas.

## Prerequisites

### Requirements

A Cisco recomenda que você tenha conhecimento destes tópicos:

- CUCM
- CUAC
- LDAP
- AD

### Componentes Utilizados

As informações neste documento são baseadas na versão 10.x do CUAC.

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. If your network is live, make sure that you understand the potential impact of any command.

# Informações de Apoio

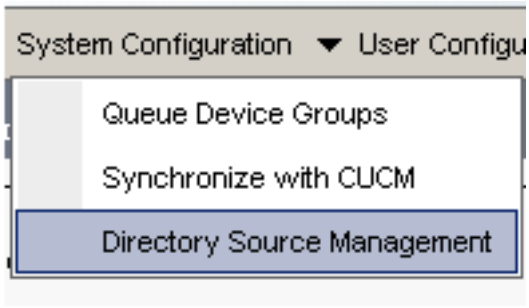
Nas versões anteriores do CUAC, o servidor obtém usuários diretamente do Cisco Unified Communications Manager (CUCM) por meio de consultas e filtros predefinidos. Com o CUAC Premium Edition (CUACPE), os administradores podem integrar e importar usuários diretamente do AD. Isso concede flexibilidade aos administradores para a implementação de atributos e filtros de sua própria escolha e requisitos.

> **Note**: O CUACPE foi substituído pelo CUAC Advanced Edition para versões 10 e posteriores.

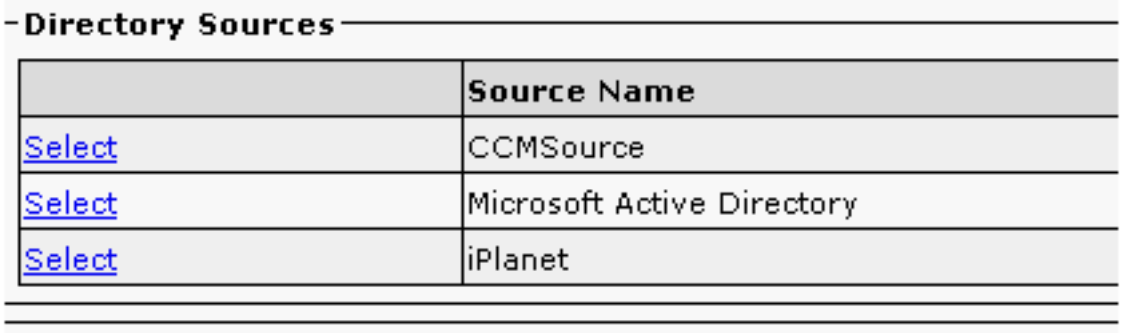
# Integre o AD ao CUAC e importe usuários do AD

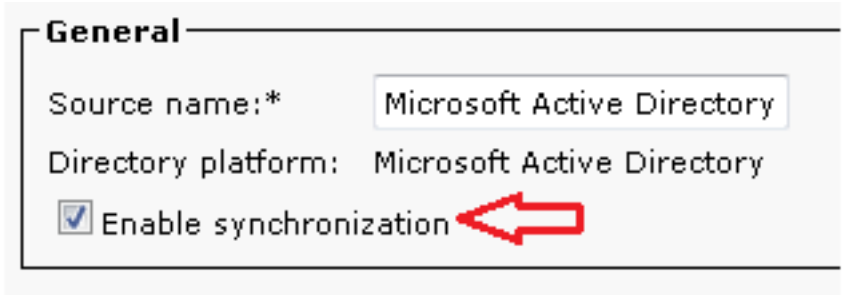
Conclua estes passos para integrar o CUAC ao AD e importar usuários do AD:

1. Ative a sincronização de diretórios para AD no CUAC.

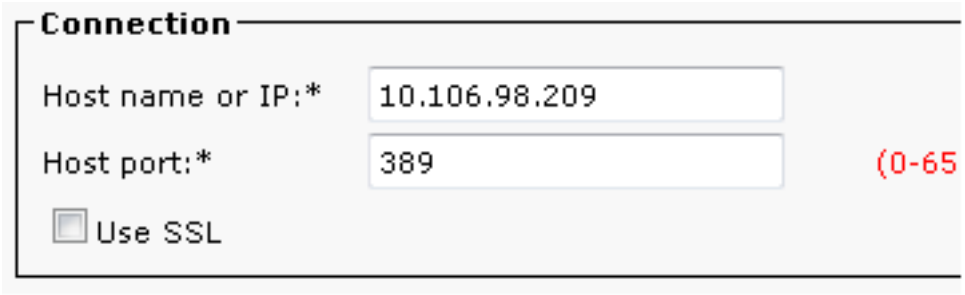



2. Selecione **Microsoft Ative Diretory** e marque a caixa de seleção **Habilitar sincronização**:

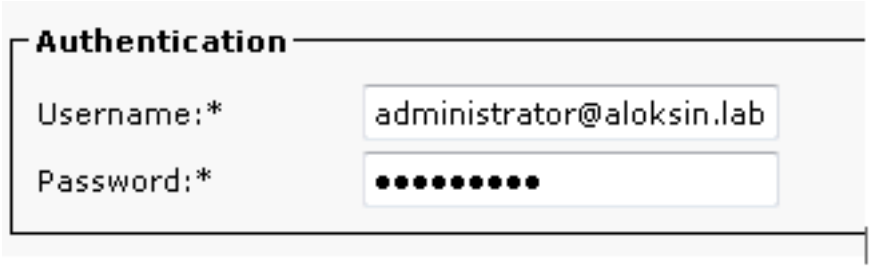


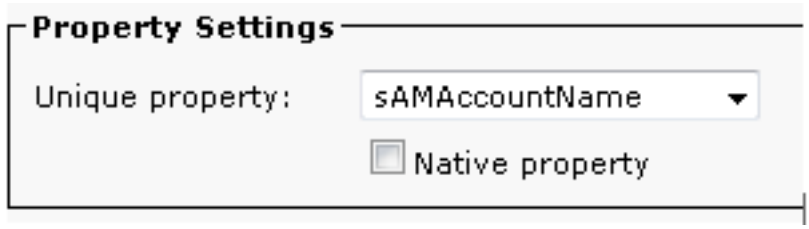

3. Insira os detalhes da configuração do servidor Ative Diretory:




Para este exemplo, **administrator@aloksin.lab** é usado para autenticação:



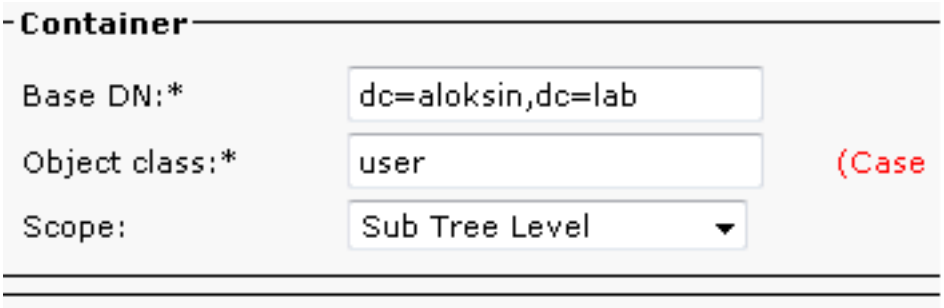

4. Na seção Configurações da propriedade, insira os detalhes da configuração da propriedade Exclusiva, que é exibida quando você digita os outros detalhes e clique em **Salvar**.




**Note**: Este é um valor exclusivo para cada entrada no AD. Se houver valores duplicados, o CUAC extrai apenas uma entrada.

5. Na seção Contêiner, insira os detalhes de configuração do DN base, que é o escopo de pesquisa do usuário no AD.

   O campo *Object class* é usado pelo AD para determinar o escopo de pesquisa solicitado. Por padrão, ele é definido como *contato*, o que significa que o AD procura *contatos* (não usuários) na base de pesquisa solicitada. Para importar *usuários* no CUAC, altere a configuração de classe Object para **usuário**:

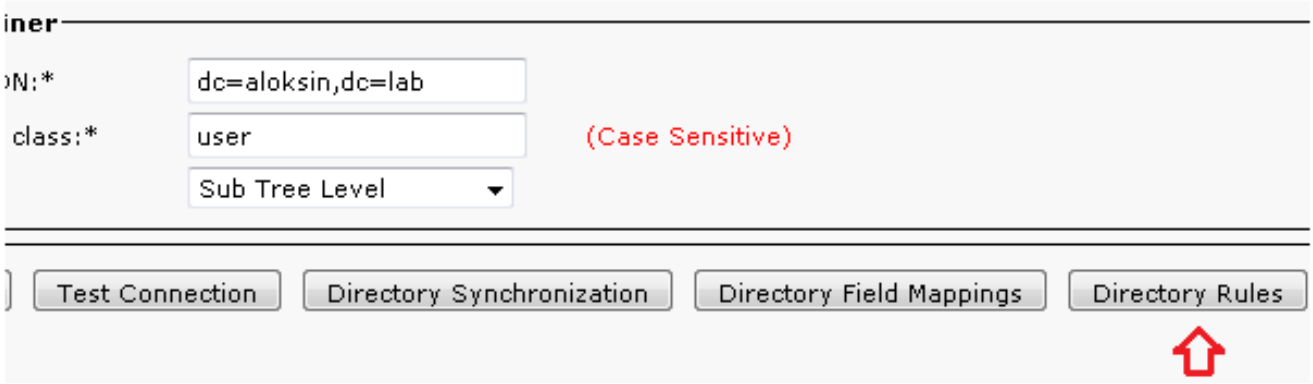



6. Salve as configurações, clique em **Mapeamentos de campos de diretório** e configure todos os atributos que você gostaria de importar para qualquer usuário. Aqui está a configuração

usada neste exemplo:

| Source Fields | Destination Fields | Defaul |
|---|---|---|
| telephoneNumber | Extension | |
| mail | Email | |
| givenName | First Name | |
| sn | Last Name | |

7. Navegue até a página de origem do diretório e clique em **Regras de diretório**:




8. Clique em **Adicionar novo** e crie uma regra. Quando você adiciona uma regra de diretório, um filtro de regra é exibido por padrão.

| | Field | Operator | Value |
|---|---|---|---|
| | telephoneNumber | = | * |

**Note**: Não há necessidade de alterar o filtro de regras. Importa todos os usuários que têm um número de telefone configurado.

9. Para configurar a sincronização automática com o AD, clique na guia **Diretory Synchronization**.


10. A configuração agora está concluída. Navegue até **Engineering > Service Management** e reinicie o plug-in LDAP para iniciar a sincronização manualmente.

# Funcionalidade LDAP entre CUAC e AD

## Resumo do processo LDAP

Aqui está um resumo do processo LDAP entre o CUAC e o AD:

1. Uma sessão TCP é estabelecida entre os dois servidores (CUAC e AD).

2. O CUAC envia uma solicitação BIND ao AD e se autentica por meio do usuário configurado nas configurações de Autenticação.

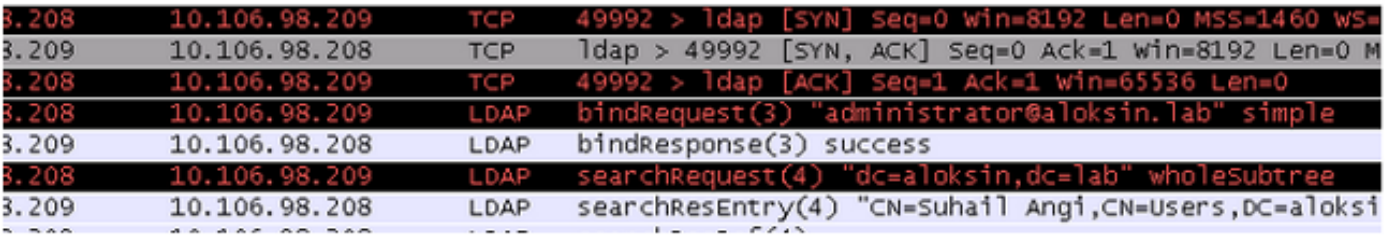
3. Quando o AD autentica o usuário com êxito, ele envia uma notificação de Êxito de BIND ao CUACPE.

4. O CUAC envia uma solicitação de PESQUISA ao AD, que tem as informações do escopo da pesquisa, os filtros para a pesquisa e os atributos para qualquer usuário filtrado.

5. O AD verifica o objeto solicitado (configurado nas configurações da Classe de objeto) na base de pesquisa. Filtra objetos que correspondem aos critérios (filtro) detalhados na mensagem de solicitação de PESQUISA.

6. O AD responde ao CUAC com os resultados da pesquisa.

Aqui está uma captura de farejador que ilustra estes passos:

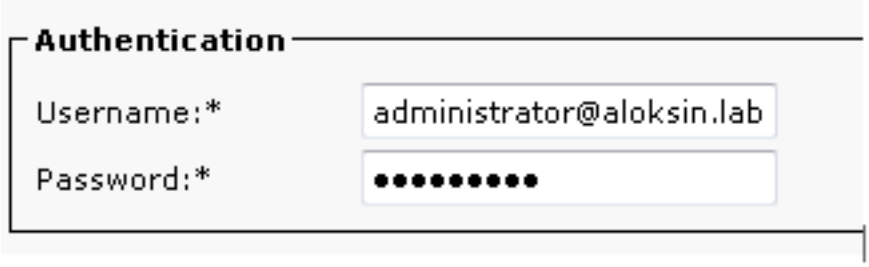



## Detalhes do processo LDAP

Quando a configuração no CUAC é concluída e o plug-in LDAP é reiniciado, o servidor CUAC configura uma sessão TCP com o AD.

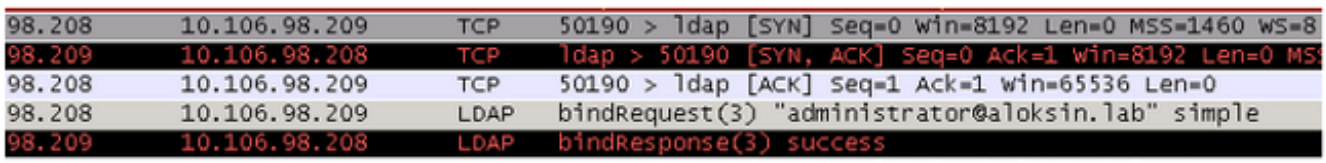
Em seguida, o CUAC envia uma solicitação BIND para autenticar com o servidor AD. Se a autenticação for bem-sucedida, o AD enviará uma resposta BIND Success ao CUAC. Com isso, ambos os servidores tentam configurar uma sessão na porta 389 para sincronizar usuários e suas informações.

Aqui está a configuração no servidor que define o nome distinto, que é usado para autenticação na transação BIND:

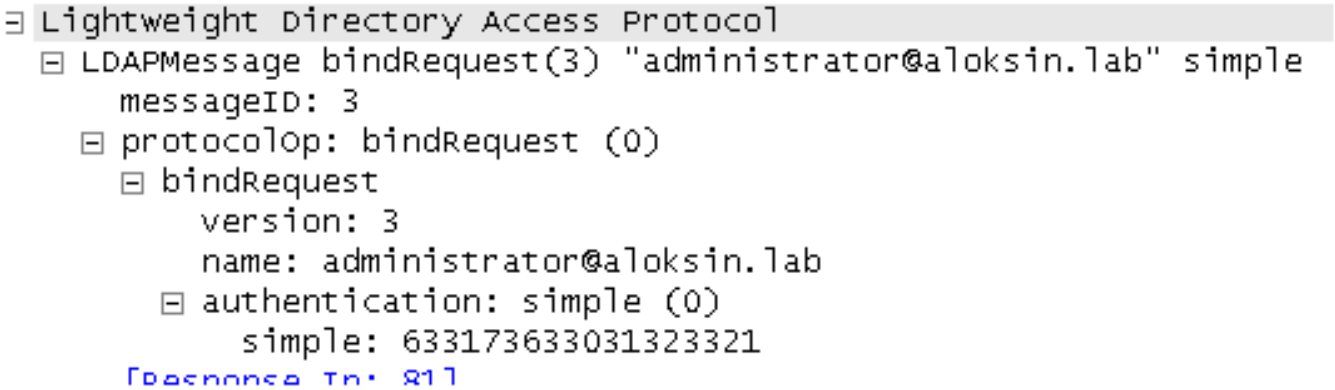



Essas mensagens aparecem nas capturas de pacote:

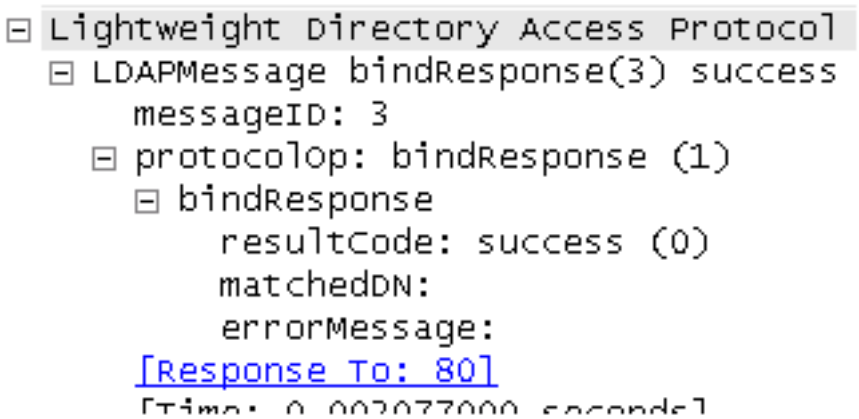
- Aqui está o handshake TCP, seguido pela solicitação BIND:




- Aqui está a expansão da solicitação BIND:




- Aqui está a expansão da resposta BIND, que indica uma autenticação bem sucedida do usuário (**administrador** neste exemplo):




Após uma associação bem-sucedida, o servidor envia uma solicitação de PESQUISA ao AD para importar usuários. Esta solicitação de PESQUISA contém o filtro e os atributos usados pelo AD. Em seguida, o AD procura usuários na base de pesquisa definida (conforme detalhado na mensagem de solicitação SEARCH), que atende aos critérios no filtro e na verificação de atributos.

Aqui está um exemplo da solicitação SEARCH que é enviada pelo CUCM:

```
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>
```

Quando o AD recebe esta solicitação do CUCM, ele procura usuários no **baseObject: dc=aloksin,dc=lab**, que satisfaz o filtro. Qualquer usuário que não atenda aos requisitos detalhados pelo filtro é deixado de fora. O AD responde ao CUCM com todos os usuários filtrados e envia os valores dos atributos solicitados.

**Note**: Não é possível importar objetos. Somente *os usuários* são importados. Isso ocorre

porque o filtro enviado na mensagem de solicitação SEARCH inclui **objetclass=user**. Portanto, o AD procura apenas usuários, não contatos. O CUCM tem todos esses mapeamentos e um filtro por padrão.

O CUAC não está configurado por padrão; não há detalhes de mapeamento configurados para importar atributos para usuários, portanto, você deve inserir esses detalhes manualmente. Para criar esses mapeamentos, navegue para **Configuração do sistema > Gerenciamento de origem de diretório > Ative Diretory > Mapeamento de campos de diretório**.

Os administradores podem mapear campos de acordo com seus próprios requisitos. Aqui está um exemplo:

**Directory Source**

Microsoft Active Directory

**Field Mappings**

| | | Source Fields | Destination Fields | Default Value |
|---|---|---|---|---|
| ☐ | Select | telephoneNumber | Extension | |
| ☐ | Select | mail | Email | |
| ☐ | Select | givenName | First Name | |
| ☐ | Select | sn | Last Name | |

As informações do campo de origem são enviadas ao AD na mensagem de solicitação de PESQUISA. Quando o AD envia a mensagem de resposta SEARCH, esses valores são armazenados nos campos de destino no CUACPE.

Observe que o CUAC por padrão tem a classe Object definida como *contatos*. Se esta configuração padrão for usada, o filtro que é enviado para o AD será exibido como mostrado aqui:

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

Com esse filtro, o AD nunca retorna nenhum usuário ao CUACPE, pois procura *contatos* na base de pesquisa, não *usuários*. Por esta razão, tem de alterar a Classe de Objeto para **utilizador**:

**Container**

Base DN:* dc=aloksin,dc=lab

Object class:* user (Case Sensitive)

Scope: Sub Tree Level

Até esse ponto, essas configurações foram configuradas no CUAC:

- Detalhes das conexões
- Autenticação (usuário distinto para associação)
- Configurações do contêiner
- Mapeamento de diretório

Neste exemplo, a propriedade Exclusiva está configurada como **sAMAccountName**. Se você reiniciar o plug-in LDAP no CUAC e verificar a mensagem de solicitação SEARCH, ele não conterá nenhum atributo ou filtro, exceto o **ObjectClass=user**:

```
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]
```

Observe que a regra Diretory está ausente aqui. Para sincronizar os contatos com o AD, você deve criar uma regra. Por padrão, não há regra de diretório configurada. Assim que um é criado, um filtro já está presente. Não há necessidade de alterar o filtro, pois você deve importar todos os usuários que têm um número de telefone.

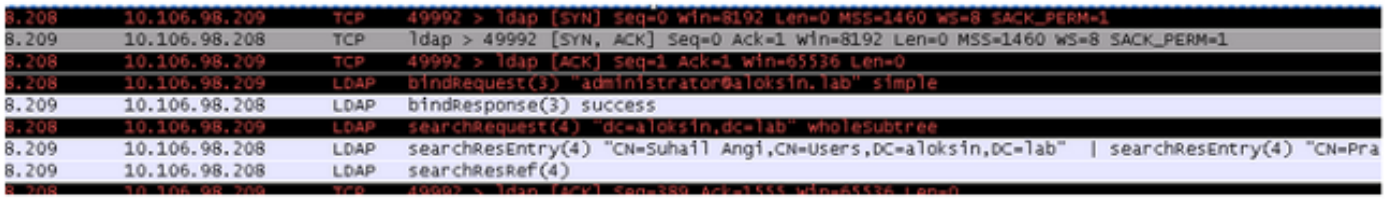
| Field | Operator | Val |
|---|---|---|
| telephoneNumber | = | * |

Reinicie o plug-in LDAP para iniciar uma sincronização com o AD e importar os usuários. Aqui está a solicitação de PESQUISA do CUAC:

```
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>
```

Se o AD encontrar usuários que correspondam aos critérios detalhados na mensagem de solicitação de PESQUISA, ele enviará uma mensagem *SearchResEntry* que contém as informações do usuário.




Aqui está a mensagem SearchResEntry:

```
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}..........
                           .....
```

**Note**: Não há MAIL na resposta, mesmo que este atributo seja solicitado. Isso ocorre porque a ID do CORREIO não foi configurada para usuários no AD.

Quando esses valores são recebidos pelo CUAC, ele os armazena na tabela Structured Query Language (SQL). Você pode então fazer login no console e o console busca a lista de usuários dessa tabela SQL no servidor CUACPE.