Integración CUAC con Microsoft AD

Contenido

Introducción

Este documento describe la manera de las cuales el Lightweight Directory Access Protocol (LDAP) trabaja entre la Consola de Attendant unificada Cisco (CUAC) y el Microsoft Active Directory (AD) y los procedimientos que se utilicen para integrar los dos sistemas.

Prerrequisitos

Requisitos

Cisco recomienda que tenga conocimiento sobre estos temas:

- CUCM
- CUAC
- LDAP
- AD

Componentes Utilizados

La información en este documento se basa en la versión 10.x CUAC.

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si la red está funcionando, asegúrese de haber comprendido el impacto que puede tener cualquier comando.

Antecedentes

En versiones anteriores CUAC, el servidor obtiene a los usuarios directamente del administrador de las Comunicaciones unificadas de Cisco (CUCM) vía las interrogaciones predefinidas y los filtros. Con la edición superior CUAC (CUACPE), se permite a los administradores integrar e importar a los usuarios directamente del AD. Esto concede la flexibilidad a los administradores para la implementación de los atributos y los filtros de su propia opción y requisitos.

> Nota: El CUACPE ahora se ha substituido por la edición avanzada CUAC para las versiones 10 y posterior.

Integre el AD con CUAC e importe a los usuarios del AD

Complete estos pasos para integrar el CUAC con el AD e importar a los usuarios del AD:

1. Habilite la Sincronización de directorio para el AD en el CUAC.

2. Seleccione el **Microsoft Active Directory** y marque la casilla de verificación de la **sincronización del permiso**:

3. Entre a los detalles de la configuración para el servidor Active Directory:

   Por este ejemplo, **administrator@aloksin.lab** se utiliza para la autenticación:

4. En la propiedad las configuraciones seccionan, ingresan a los detalles de la configuración para la propiedad única, que aparece usted ingresa los otros detalles y hace clic una vez la **salvaguardia**.

   Nota: Esto es un valor único para cada entrada en el AD. Si hay valores duplicados, el CUAC tira de solamente una entrada.
5. En la sección del envase, ingrese a los detalles de la configuración para la base DN, que es el alcance de la búsqueda de usuario en el AD.

   El campo de *clase de objeto* es utilizado por el AD para determinar el alcance pedido de la búsqueda. Por abandono, se fija *para entrar en contacto*, así que significa que el AD busca

los *contactos* (no usuarios) en la base pedida de la búsqueda. Para importar a los *usuarios* en el CUAC, cambie la clase de objeto que fija al **usuario**:

6. Salve las configuraciones, haga clic las **asignaciones del campo del directorio**, y configure todos los atributos que usted quisiera importar para cualquier usuario. Aquí está la configuración que se utiliza en este ejemplo:

7. Navegue a la página de la fuente del directorio y haga clic las **reglas del directorio**:

8. El tecleo **agrega nuevo** y crea una regla. Cuando usted agrega una regla del directorio, un filtro de la regla aparece por abandono.

   Nota: No hay necesidad de cambiar el filtro de la regla. Importa a todos los usuarios que hagan un número de teléfono configurar.
9. Para configurar auto-sincronice con el AD, hacen clic la lengueta de la **Sincronización de directorio**.

10. La configuración es completa ahora. Navegue a **dirigir > Administración del servicio** y recomience el LDAP plug-in para comenzar el sincronizar manualmente.

Funciones LDAP entre CUAC y el AD

Resumen del proceso LDAP

Aquí está un resumen del proceso LDAP entre el CUAC y el AD:

1. Establecen a una sesión TCP entre los dos servidores (CUAC y AD).

2. El CUAC envía una petición del LAZO al AD y la autentica vía el usuario que se configura en las configuraciones de la autenticación.

3. Una vez que el AD autentica con éxito al usuario, envía una notificación del éxito del LAZO al CUACPE.

4. El CUAC envía una petición de búsqueda al AD, que tiene la información de alcance de la búsqueda, los filtros para la búsqueda, y la atribuye para cualquier usuario filtrado.

5. El AD analiza para el objeto solicitado (configurado en las configuraciones de la clase de objeto) en la base de la búsqueda. Filtra hacia fuera los objetos que hacen juego los criterios (filtro) detallados en el mensaje de la petición de búsqueda.

6. El AD responde al CUAC con los resultados de la búsqueda.

Aquí está una captura del sniffer que ilustra estos pasos:

Detalles de proceso LDAP

La configuración en el CUAC se completa una vez y el LDAP plug-in se recomienza, el servidor CUAC configura a una sesión TCP con el AD.

El CUAC entonces envía una petición del LAZO para autenticar con el servidor AD. Si la autenticación es acertada, el AD envía una respuesta del éxito del LAZO al CUAC. Con esto, ambos servidores intentan configurar una sesión sobre los usuarios de sincronización del puerto 389 para y su información.

Aquí está la configuración en el servidor que define el nombre distintivo, que se utiliza para la autenticación en la transacción del LAZO:

Estos mensajes aparecen en las capturas de paquetes:

- Aquí está la aceptación decontacto con TCP, seguida por la petición del LAZO:

- Aquí está la extensión de la petición del LAZO:

- Aquí está la extensión de la respuesta del LAZO, que indica la autenticación satisfactoria del usuario (**administrador** en este ejemplo):

Sobre un lazo acertado, el servidor envía una petición de búsqueda al AD para importar a los usuarios. Esta petición de búsqueda contiene el filtro y los atributos que son utilizados por el AD. El AD entonces busca para los usuarios dentro de la base definida de la búsqueda (como se detalla en el mensaje de la petición de búsqueda), que satisface los criterios en el filtro y la verificación de los atributos.

Aquí está un ejemplo de la petición de búsqueda que es enviada por el 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>
```

Cuando el AD recibe esta petición del CUCM, busca para los usuarios en el **baseObject: dc=aloksin, dc=lab**, que satisface el filtro. Dejan cualquier usuario que no satisfaga los requisitos que son detallados por el filtro hacia fuera. El AD responde al CUCM con todos los usuarios filtrados y envía los valores para los atributos pedidos.

> Nota: Los objetos no pueden ser importados. Solamente importan a los *usuarios*. Esto es porque el filtro que se envía en el mensaje de la petición de búsqueda incluye el

objectclass=user. Por lo tanto, el AD busca solamente para los usuarios, no los contactos. El CUCM tiene todo el estas asignaciones y filtro por abandono.

El CUAC no se configura por abandono; no hay asignación detalla configurado para importar los atributos para los usuarios, así que usted debe entrar estos detalles manualmente. Para crear estas asignaciones, navegue a la **configuración del sistema > a la administración de fuentes del directorio > a la asignación del campo del Active Directory > del directorio**.

Se permite a los administradores asociar los campos por sus propios requisitos. Aquí tiene un ejemplo:

La información del campo de fuente se envía al AD en el mensaje de la petición de búsqueda. Cuando el AD envía el mensaje de respuesta de la BÚSQUEDA, estos valores se salvan en los Campos Destination en el CUACPE.

Observe que el CUAC por abandono tiene la clase de objeto fijada a los *contactos*. Si se utiliza esta configuración predeterminada, el filtro que se envía al AD aparece como se muestra aquí:

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

Con este filtro, el AD nunca vuelve a cualquier usuario al CUACPE, puesto que busca para los *contactos* en la base de la búsqueda, no los *usuarios*. Por este motivo, usted debe cambiar la clase de objeto al **usuario**:

Hasta esta punta, estas configuraciones se han configurado en el CUAC:

- Detalles de las conexiones
- Autenticación (usuario distinguido para atar)
- Configuraciones del envase
- Asignación del directorio

En este ejemplo, la propiedad única se configura como **sAMAccountName**. Si usted recomienza el LDAP plug-in en el CUAC y marca el mensaje de la petición de búsqueda, no contiene ningunos atributos o filtro excepto el **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 la regla del directorio falta aquí. Para sincronizar los contactos con el AD, usted debe crear una regla. Por abandono, no hay regla del directorio configurada. Tan pronto como se cree uno, un filtro está ya presente. No hay necesidad de cambiar el filtro, como usted debe importar a todos los usuarios que tengan un número de teléfono.

Recomience el LDAP plug-in para iniciar un sincronizar con el AD e importar a los usuarios. Aquí está la petición de búsqueda del 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>
```

Si el AD encuentra a los usuarios que hacen juego los criterios detallados en el mensaje de la petición de búsqueda, después envía un mensaje de *SearchResEntry* que contenga la información del usuario.

Aquí está el mensaje de 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}.........
       .....
```

Nota: No hay CORREO en la respuesta, aunque se pide este atributo. Esto es porque el CORREO ID no fue configurado para los usuarios en el AD.

Una vez que estos valores son recibidos por el CUAC, lo salva en la tabla del Lenguaje de consulta estructurado (SQL). Usted puede entonces registrar en la consola, y la consola trae la lista de usuarios de esta tabla SQL en el servidor CUACPE.