[How-to] configure secure LDAP for Azure AD Domain Services

Pre-requisites 1

Step 1: Creating a certificate for secure LDAP 1

Step 2: Exporting a certificate for Azure AD DS 2

Step 3: Exporting a certificate for client computers 4

Step 4: Enabling secure LDAP for Azure AD DS 6

Step 5: Whitelisting IPs on Azure for secure LDAP access over the internet 7

Step 6: Configuring DNS zone for external access 8

Step 7: Testing queries to the managed domain 9

Step 8: Binding users to the managed domain 9

What's next? 12

Have any questions? 12

Pre-requisites

To configure secure LDAP, you need the following resources and privileges:

- **An active Azure subscription**. If you don't have an Azure subscription, create an account.
- **An Azure Active Directory tenant associated with your subscription.** It should be either synchronized with an on-premises directory or a cloud-only directory. Create an Azure Active Directory tenant or associate an Azure subscription with your account.
- **An Azure Active Directory Domain Services managed domain enabled and configured in your Azure AD tenant.** If needed, create and configure an Azure Active Directory Domain Services instance.
- **The LDP.exe tool installed on your computer.** Install the Remote Server Administration Tools (RSAT) for AD Domain Services and LDAP.

To better understand Azure AD and its documentation, we recommend reviewing the terms mentioned here.

Below we'll explain the configuration in a few steps.

Step 1: Creating a certificate for secure LDAP

The first step involves creating a digital certificate.

What do we use it for?

- You apply this digital certificate to your Azure AD DS managed domain.

- It lets tools like LDP.exe use secure encrypted communication when querying data.

Ways to create a digital certificate

You can use a digital certificate created in two ways to secure LDAP access to the managed domain.

a. A certificate from a public certificate authority (CA) or an enterprise CA.
b. A self-signed certificate that you create yourself.

Below we'll explain the steps using a self-signed certificate. Skip Step 1-8 if you are using a public certificate authority (CA).

Creating a certificate requires to run a set of commands on Windows Powershell.

1. Open a PowerShell window as **Administrator** and run the following commands one by one.
2. Replace the **$dnsName** variable with the *DNS name* used by your own managed domain, such as *onmicrosoft.com.*

PowerShell Copy

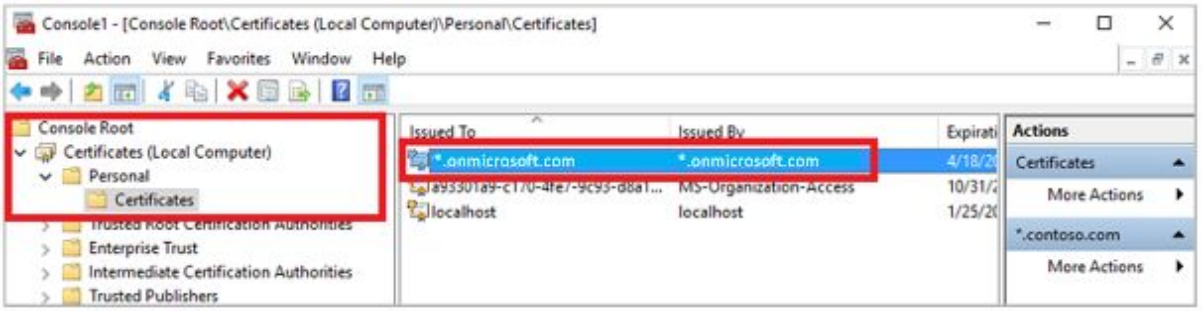
```
# Define your own DNS name used by your Azure AD DS managed domain
$dnsName="onmicrosoft.com"

# Get the current date to set a one-year expiration
$lifetime=Get-Date

# Create a self-signed certificate for use with Azure AD DS
New-SelfSignedCertificate -Subject *.$dnsName `
  -NotAfter $lifetime.AddDays(365) -KeyUsage DigitalSignature, KeyEncipherment `
  -Type SSLServerAuthentication -DnsName *.$dnsName, $dnsName
```

Step 2: Exporting a certificate for Azure AD DS

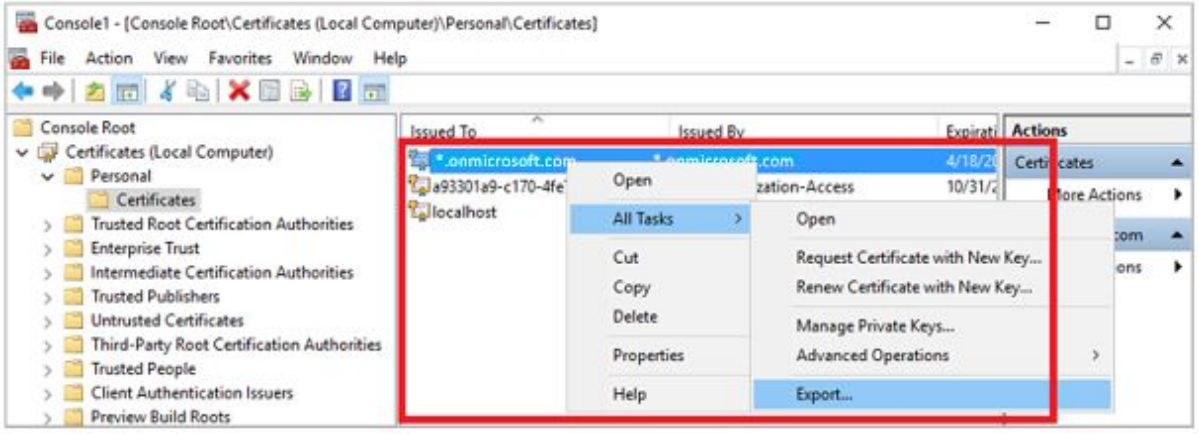
Now you need to export the certificate to a (***.PFX)*** certificate file that includes the private key. You must do this before you can use the digital certificate created with your Azure AD DS managed domain.

1. To open the *Run* dialog, select the **Windows** and **R** keys.
2. Open the Microsoft Management Console (MMC) by entering **mmc** in the *Run* dialog, then select **OK**.
3. On the **User Account Control** prompt, click **Yes** to *launch MMC* as administrator.

4. From the **File** menu, click **Add/Remove** Snap-in.
5. In the **Certificates** snap-in wizard, choose **Computer account**, then select **Next**.
6. On the **Select Computer** page, choose **Local computer: (the computer this console is running on)**. Then select **Finish**.
7. In the **Add or Remove** snap-in dialog, click **OK** to add the certificates snap-in to MMC.
8. In the *MMC window*, expand **Console Root**. Select **Certificates (Local Computer)**. Then expand the **Personal** node, followed by the Certificates node.



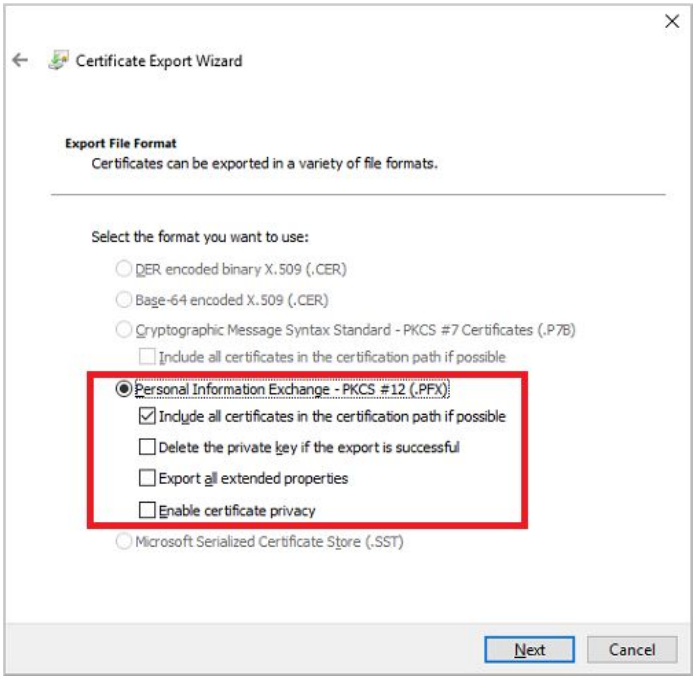
9. You will see the self-signed certificate created in the previous step, such as *onmicrosoft.com*. Right-select this certificate, then choose **All Tasks > Export**.



10. In the **Certificate Export** Wizard, select **Next**.
11. You must export the private key for the certificate. Without the private key in the exported certificate, the action to enable secure LDAP for your managed domain will fail.

    On the **Export Private Key** page, choose **Yes** > **export the private key**, then select **Next**.
12. Azure AD DS managed domains only support the (.PFX) certificate file format that includes the private key. Don't export the certificate as .CER certificate file format without the private key.

    On the **Export File Format** page, select **Personal Information Exchange - PKCS #12 (.PFX)** as the file format for the exported certificate. Check the box for *Include all certificates in the certification path if possible*.



13. You are using this certificate to decrypt data. You should be careful when controlling access. You can use a password to protect the use of the certificate. Without the correct password, you cannot apply the certificate to a service.

    On the **Security** page, choose the option for **Password** to protect the (.PFX) certificate file. Enter and confirm a password, then select Next. We'll use this password in the next section to enable secure LDAP for your Azure AD DS managed domain.

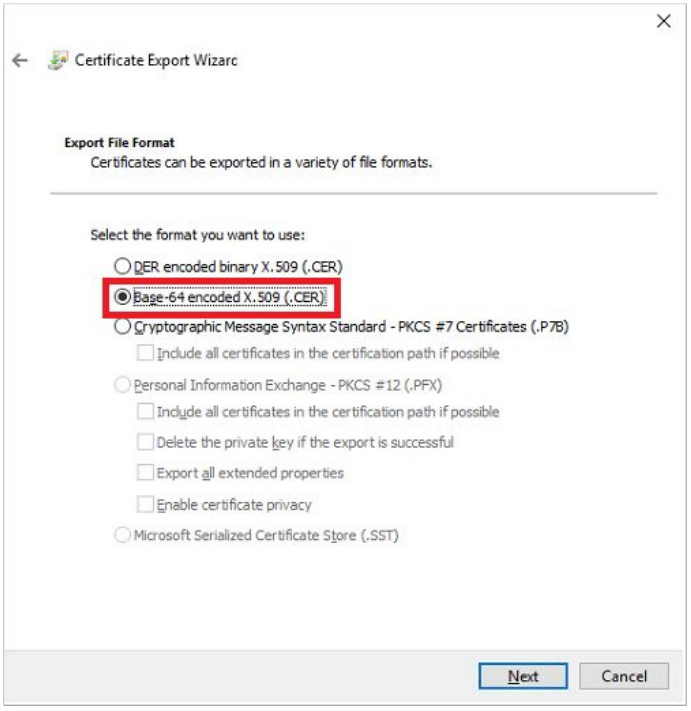
14. On the **File to Export** page, specify the file name and location. It's where you'd like to *export* the certificate, such as *C:\Users\accountname\azure-ad-ds.pfx*.

15. On the review page, select **Finish** to export the certificate to a (.PFX) certificate file. You will see a confirmation dialog when the certificate has been successfully exported.

Step 3: Exporting a certificate for client computers

Now we will follow the same steps for the client certificate with a little bit of alternation.

1. Go back to the MMC for *Certificates (Local Computer) > Personal > Certificates* store. You will see the self-signed certificate created in a previous step, such as *onmicrosoft.com*. Right-select this certificate, then choose **All Tasks > Export**.
2. In the Certificate Export Wizard, select Next.
3. You do not need the private key for clients. On the **Export Private Key** page, choose **No, do not export the private key**, then select **Next**.
4. On the **Export File Format** page, select **Base-64 encoded X.509 (.CER)** as the file format - for the exported certificate.

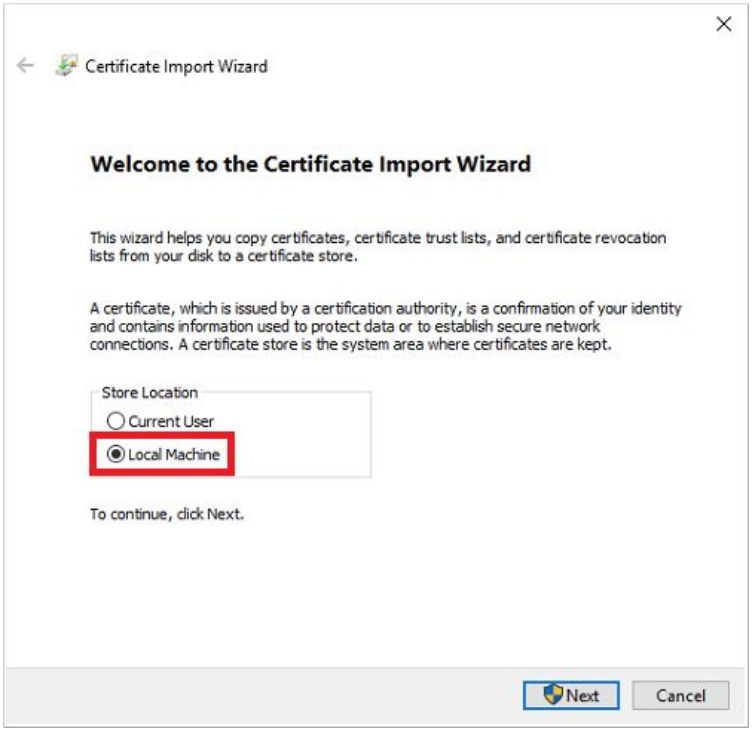


5. On the **File to Export** page, specify the file name and location. For example, *C:\Users\accountname\azure-ad-ds-client.cer*.
6. On the review page, select **Finish** to export the certificate to a (.CER) certificate file. You will see a confirmation dialog when the certificate exported is successful.

   You can now distribute the (.CER) certificate file to client computers. Computers that need to trust the secure LDAP connection to the Azure AD DS managed domains.

   Let's install the certificate on the local computer.

- Open File Explorer and browse to the location where you saved the (.CER) certificate file. Such as *C:\Users\accountname\azure-ad-ds-client.cer*.
- Right-select the (.CER) certificate file, then choose **Install Certificate**.
- In the **Certificate Import Wizard**, choose the certificate in the Local machine. Then select **Next**.

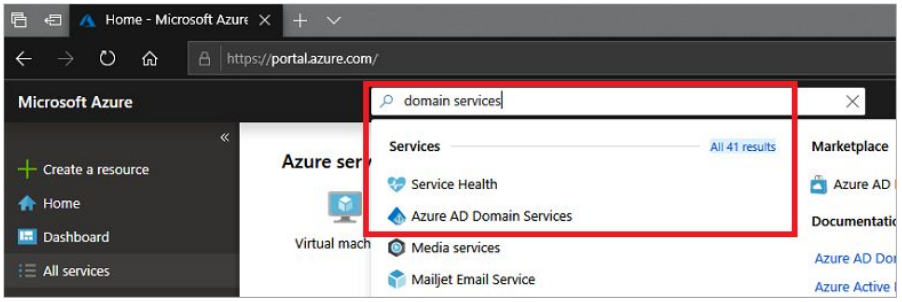


- When prompted, choose **Yes** to allow the computer to make changes.
- Choose to **Automatically select the certificate store based on the type of certificate**. Then, select **Next**.
- On the review page, select **Finish** to import the (.CER) certificate file. You will see a confirmation dialog when the certificate has been successfully imported.

Step 4: Enabling secure LDAP for Azure AD DS

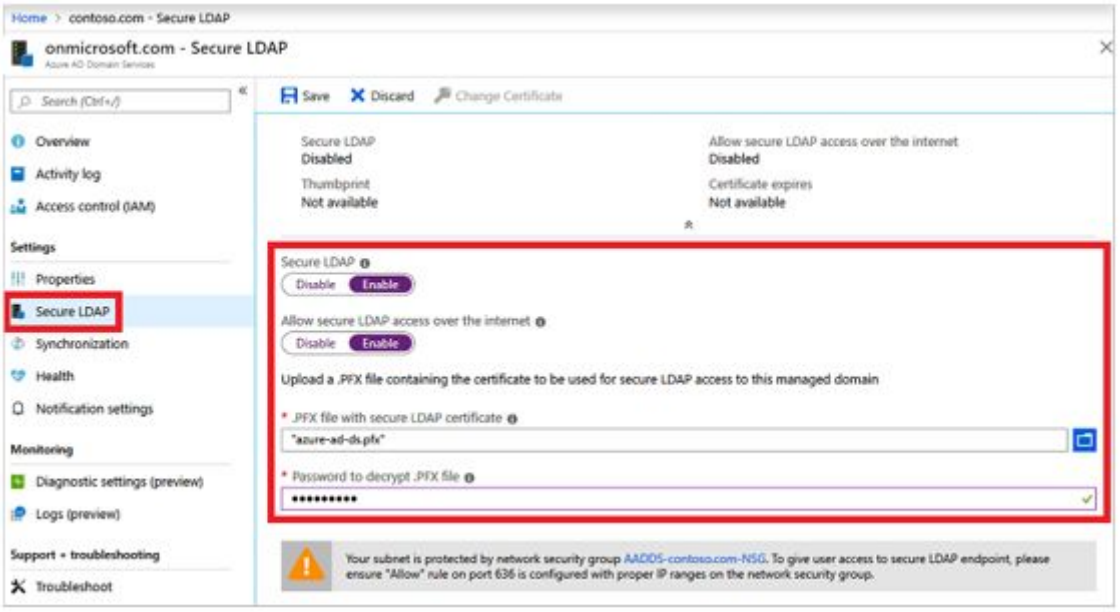
Now that you have created and exported a digital certificate - and a client computer, let's move to the next step. You now have to enable secure LDAP on your Azure AD DS managed domain.

1. In the Azure portal, search for domain services in the **Search resources** box. Select **Azure AD Domain Services** from the search result.



2. Choose your managed domain, such as *onmicrosoft.com*.
3. On the left-hand side of the Azure AD DS window, choose **Secure LDAP**.
4. By default, secure LDAP access to your managed domain is disabled. Toggle **Secure LDAP** to **Enable**.
   Toggle **Allow secure LDAP access over the internet** to **Enable**.
5. In the next step, you'll have to configure a network security group. This is to lock down access to only the required source IP address ranges.
   Toggle **Allow secure LDAP access over the internet** to **Enable**.

6. Select the folder icon next to **(.PFX) file with secure LDAP certificate**. You have to browse to the path of the (.PFX) file. Then, select the certificate created in a previous step that includes the private key.
7. Enter the **Password to decrypt (.PFX) file** set in a previous step when you exported the certificate to a (.PFX) file.
8. Select **Save** at the top to enable secure LDAP.



You will see a notification that secure LDAP is being configured for the managed domain. You cannot change other settings for the managed domain until the process completes.

Note: *It takes a few minutes to enable secure LDAP for your managed domain.*

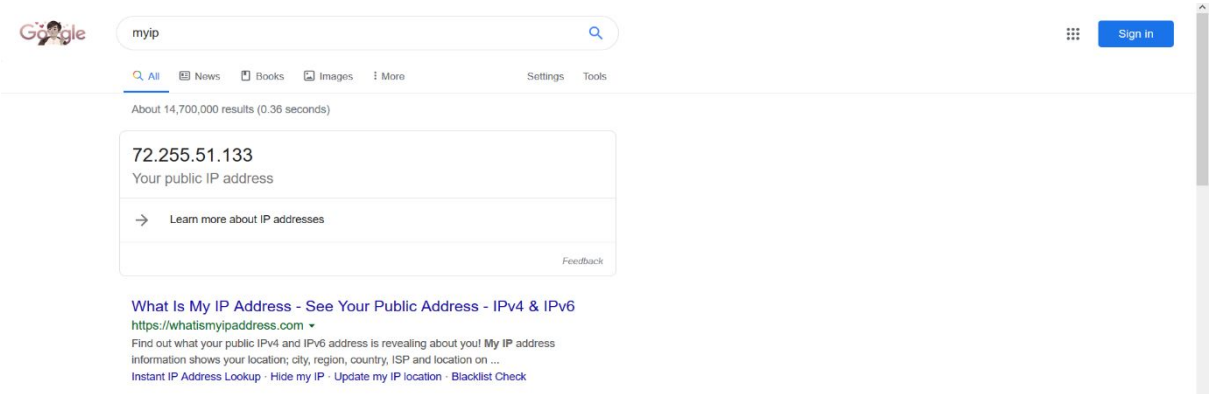
Step 5: Whitelisting IPs on Azure for secure LDAP access over the internet

The managed domain is reachable from the internet on TCP port 636. It's recommended to restrict access to the managed domain.

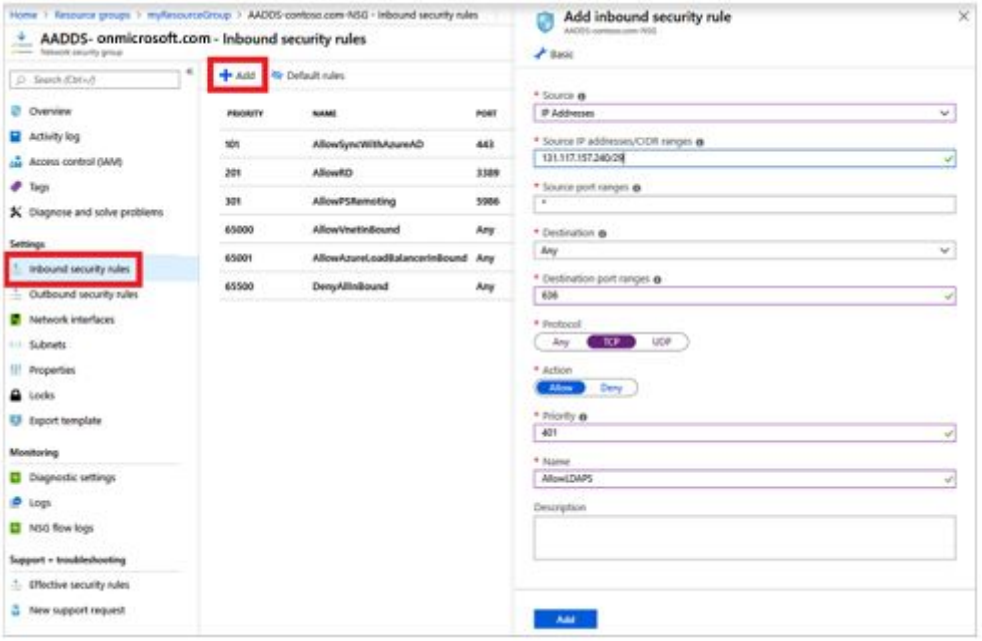
Let's create a rule to allow inbound secure LDAP access over TCP port 636 from a specified set of IP addresses.

1. In the Azure portal, select **Resource groups** on the left-hand side navigation.
2. Choose your resource group, such as *myResourceGroup*. Then select your **network security group**, such as *AADDS-onmicrosoft.com-NSG*.
3. You will see the list of existing inbound and outbound security rules. On the left-hand side of the network security group windows, choose **Security > Inbound security rules**.
4. Select **Add**, then create a rule to *allow TCP port 636*.
   For improved security, choose the source as **IP Addresses**. And then specify your own valid IP address or range for your organization.

   Tip: Type myip on Google.com if you don’t know your public IP Address.



5. Select **Add** at the bottom-right to save and apply the rule.



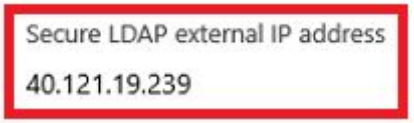
Step 6: Configuring DNS zone for external access

With secure LDAP access enabled over the internet, you now have to update the DNS zone. This will enable the client computers to find this managed domain.

Secure LDAP (LDAPS) is a connection protocol. It is used between the application and Network Directory/Domain Controller - within the infrastructure.

You will see the Secure LDAP external IP address listed on the Properties tab.

1. Go to the **properties** of your domain and copy the mentioned *Secure LDAP external IP*. For example:



2. **Add** the IP in your host file located at "*C:\Windows\System32\drivers\etc*".

Step 7: Testing queries to the managed domain

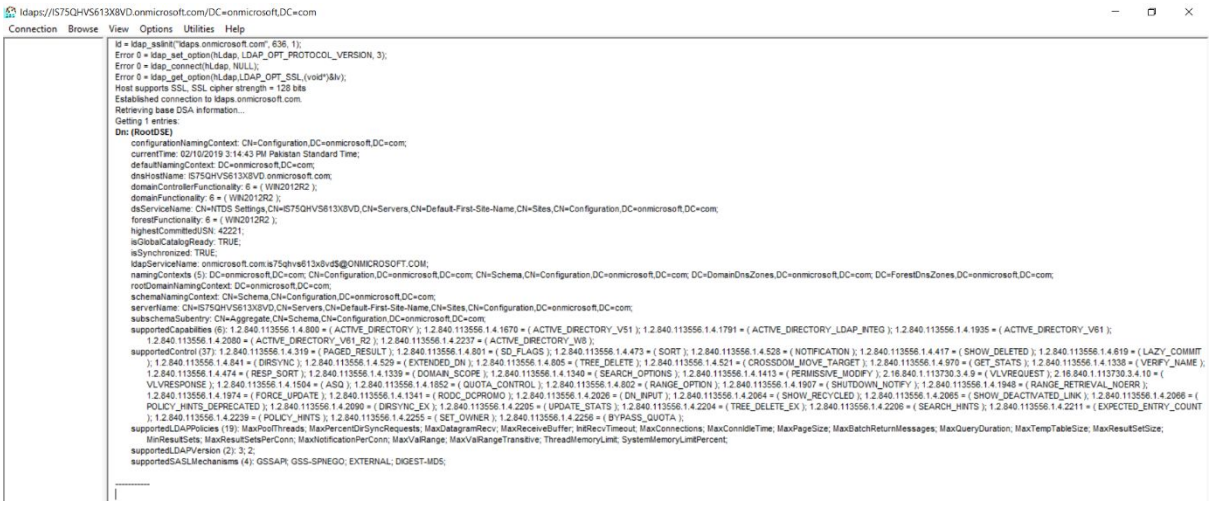
To connect your Azure AD DS managed domain and search over LDAP, you need to use the LDP.exe too.

If you don't have LDP.exe installed, [install it from here](#).

1. Open **LDP.exe** and connect to the managed domain. Select **Connection**, then choose **Connect**.

2. Enter the *secure LDAP DNS domain name* of your managed domain - created in the previous step. Such as *ldaps.onmicrosoft.com*.
   To use secure LDAP, set **Port** to **636**, then check the **box** for **SSL**.

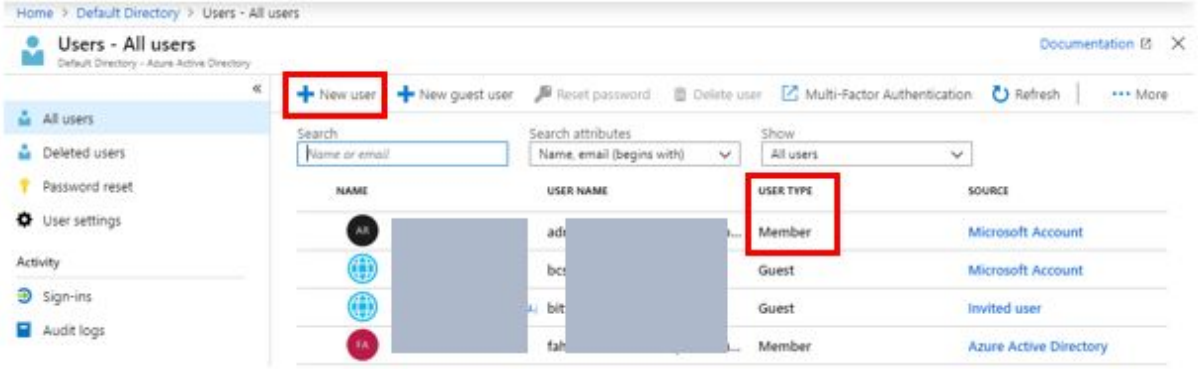
3. Select **OK** to connect to the managed domain. The following output will occur.



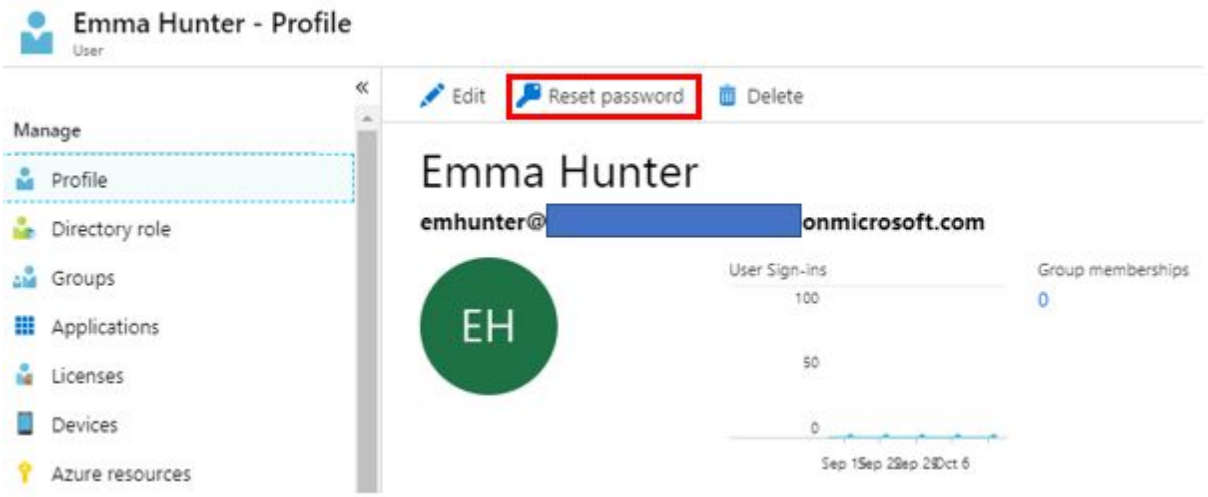
Step 8: Binding users to the managed domain

To bind a user with ldp.exe, you need to have at least one user exist in the Active Directory.

1. To create a user, go to **Azure Active Directory** in your account.
2. Select **Users** under **Manage** on the left panel. Then click **+New User > Create User** to create a user of *type Member*.
   Please note that the user type should not be a guest user.



3. Reset the user password once you have created a new user. To do that, *select the user* and then click **Reset Password** on the profile page.



4. With the new user created in the AD, the user will now have to log in to the account using *PowerShell*.
5. Run **Windows PowerShell**, as an administrator.
6. Enter the following command to log in to your Azure account.

   → **Login-AzAccount**

7. Then, enter your credentials. Once logged in, you will receive the following output:

8. Please make sure you have **enabled** your **NTLM password hash synchronization**. To enable it, enter the following commands:

   → **$DomainServicesResource = Get-AzResource -ResourceType "Microsoft.AAD/DomainServices"**

   → **$securitySettings = @{"DomainSecuritySettings"=@{"NtlmV1"="Enabled";"SyncNtlmPasswords"="Enabled";"TlsV1"="Enabled"}}**

   → **Set-AzResource -Id $DomainServicesResource.ResourceId -Properties $securitySettings -Verbose -Force**

   You should see the following output:

```
PS C:\Windows\system32> $DomainServicesResource = Get-AzResource -ResourceType "Microsoft.AAD/DomainServices"
PS C:\Windows\system32> $securitySettings = @{"DomainSecuritySettings"=@{"NtlmV1"="Enabled";"SyncNtlmPasswords"="Enabled";"TlsV1"="Enabled"}}
PS C:\Windows\system32> Set-AzResource -Id $DomainServicesResource.ResourceId -Properties $securitySettings -Verbose -Force
VERBOSE: Performing the operation "Updating the resource..." on target
"/subscriptions/26242db4-a8e8-4daf-afd2-1d8e79314db6/resourceGroups/myResourceGroup/providers/Microsoft.AAD/DomainServi
ces/p[redacted]k.onmicrosoft.com".


Name              : pucitedupk.onmicrosoft.com
ResourceId        : /subscriptions/26242db4-a8e8-4daf-afd2-1d8e79314db6/resourceGroups/myResourceGroup/providers/Micros
                    oft.AAD/domainServices/p[redacted]k.onmicrosoft.com
ResourceName      : p[redacted]k.onmicrosoft.com
ResourceType      : Microsoft.AAD/domainServices
ResourceGroupName : myResourceGroup
Location          : southeastasia
SubscriptionId    : 26242db4-a8e8-4daf-afd2-1d8e79314db6
Properties        : @{tenantId=952948a8-a52f-4d04-98ee-65ec68562909; domainName=p[redacted]k.onmicrosoft.com;
                    deploymentId=ef8554bc-7a2f-494f-a3f8-7c09f189c179; vnetSiteId=; subnetId=/subscriptions/26242db4-a8
                    e8-4daf-afd2-1d8e79314db6/resourceGroups/myResourceGroup/providers/Microsoft.Network/virtualNetwork
                    s/myResourceGroup/subnets/default; ldapsSettings=; resourceForestSettings=;
                    domainConfigurationType=FullySynced; domainControllerIpAddress=System.Object[];
                    serviceStatus=Running; notificationSettings=; domainSecuritySettings=; filteredSync=Disabled;
                    provisioningState=Succeeded}
ETag              : W/"datetime'2019-10-11T06%3A25%3A38.9427399Z'"
```

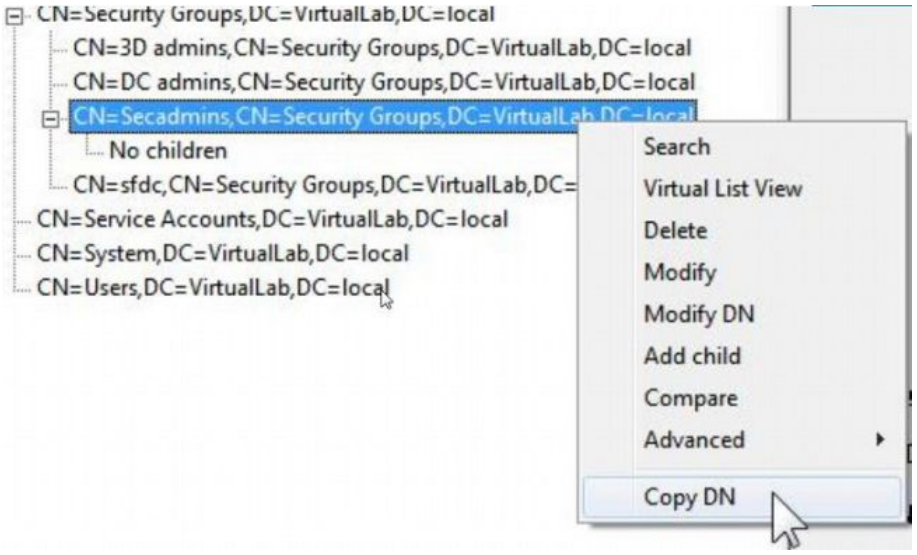
We recommend that you wait for a couple of minutes and then **run** *ldp.exe*.

9. Enter the *username*, only the part before **@**.
10. Enter the *password* and the *domain name*.
11. Choose the **Bind type** as **Binding with credentials** and click *Enter*. On successful binding, you should see the following output on screen:

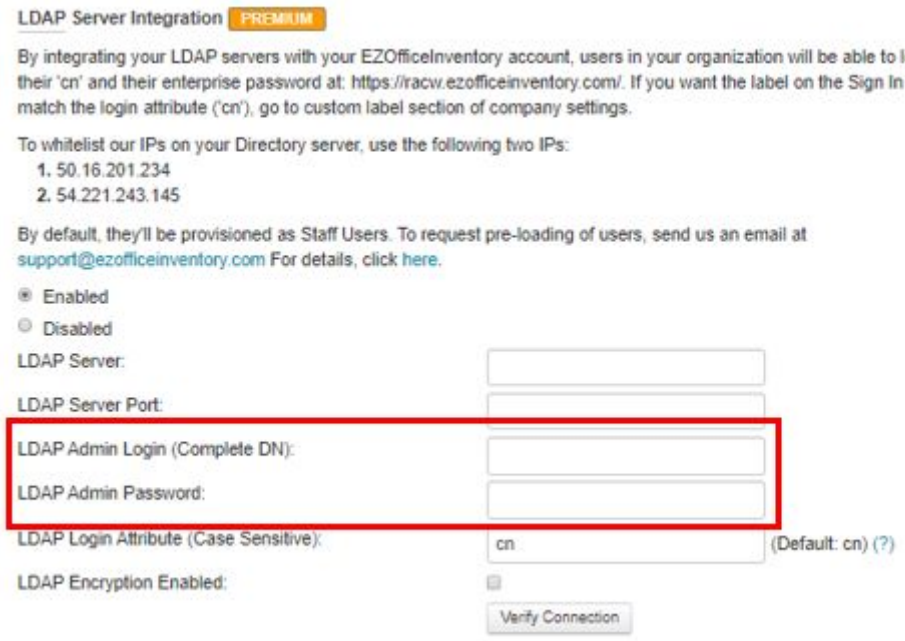
```
----------
53 = ldap_set_option(ld, LDAP_OPT_ENCRYPT, 1)
res = ldap_bind_s(ld, NULL, &NtAuthIdentity, NEGOTIATE (1158)); // v.3
        {NtAuthIdentity: User='sunny'; Pwd=<unavailable>; domain = 'p[redacted]k.onmicrosoft.com'}
Authenticated as: 'P[redacted]K\sunny'.
----------
```

Once you are able to bind the connection, you can find your **DN** from the **View tab > Tree**. Follow the steps below:

- Let the **base Dn** be empty.
- **Expand** the **explorer view** on the left-side and *select* **OU=AADDC Users**.
- Here, you will find the list of all **DN(s)** of your users. Below is what you should see:



- **Copy** the required DN and **paste** it into the EZOfficeInventory settings - for LDAP Admin Login (Complete DN).



- Use **password** of the same Microsoft account for which you acquired the DN details.

Read more: Troubleshooting Issues related to Azure AD

What's next?

The step forward is to integrate your Azure AD with EZOfficeInventory. This integration will enable you to 'sync' your staff database with EZOfficeInventory. It will help you avoid replicating LDAP staff members in EZOfficeInventory.

Enable LDAP Server Integration from *Settings → Add Ons*.

Learn more about integrating LDAP Server with EZOfficeInventory

Have any questions?

For more assistance, drop us an email at support@ezofficeinventory.com.