BeyondTrust

AD Bridge
Delegation of Domain Join Permissions

TC:7/27/2020

Table of Contents

AD Bridge Delegation of Domain Join Permissions

AD Bridge agents, like Windows systems, need to be joined into an Active Directory domain to participate in authentication, security, and configuration. In a typical Windows enterprise environment, a Domain Administrator grants the permissions to join computers to specific accounts for separation of duties or automation tasks. While the basic steps for delegating permissions to join AD Bridge systems is based on the standard Windows approach, AD Bridge requires additional permissions to provide the following abilities:

- Write additional AD Bridge specific information to the computer object.
- Support the ability to target a specific OU on join (--ou).
- Support the ability for long or duplicate computer names.

The remainder of this document describes how to delegate the permissions required for the various domain join methods.

Delegation of Control Overview

In a traditional Windows environment, all AD users can join up to ten systems to the default Computers container at a time (using the **ms-DS-MachineAccountQuota** attribute). After reaching this quota, additional attempts will be denied. To allow non-Administrator users to join Windows systems beyond their quota, the **Delegation of Control Wizard** in Active Directory Users and Computers can be used to provide basic join rights.

The basic procedure that most AD administrators are familiar with is as follows:

1. In Active Directory Users and Computers, right-click the root of the domain you want to add computers to, and then click **Delegate Control**.
2. In the **Delegation of Control Wizard**, click **Next**.
3. Click **Add** to add the specific security principal to the **Selected users and groups** list, and then click **Next**. We strongly recommend using a group, even if that group only contains a single user.
4. In the **Tasks to Delegate** page, click **Join a computer to the domain** and click **Next**, and then click **Finish**.

Note: *The delegated task,* ***Join a computer to the domain****, grants the* ***Create computers object*** *permission at the domain root to the selected security principals.*

The above method is outlined for completeness; however it is not sufficient for joining AD Bridge systems to the domain in all circumstances. If only the above procedure is followed the actual results may vary; sometimes joining without error, and failing in other instances.

Joining a computer to the domain is not always a straightforward operation from a permissions perspective in AD. A computer object may already exist, or exist in another OU within the directory. It may have been pre-staged, or created previously by another account. It may have been moved from a different OU, bringing its previous permissions with it. Because of all the variations in how a system may be joined, the above procedure is not sufficient in all circumstances, even for Windows systems.

The remainder of this document discusses the various intricacies and scenarios that differ from a standard Windows domain join and why additional consideration is required when granting join rights. There are three main reasons why AD Bridge requires more rights. AD Bridge provides additional functionality that may not be found in a typical AD deployment:

- Ability or need to join two or more systems with the same host name (but unique FQDNs)
- Ability or need to join with a disjointed DNS name space
- Ability or need to set additional computer properties for functional or reporting purposes

AD Bridge Domain Join Process Overview

When a domain join process is initiated on the AD Bridge agent, it first must determine what name to join Active Directory with. By default, this name will be the FQDN of the system. If the system does not have a FQDN, the domain join process uses the host name of the system and update the FQDN to match the Active Directory domain being joined.

For example:

System 1:

Host Name: **server01**

Domain Name: **contoso.com**

FQDN: **server01.contoso.com**

System 2:

Host Name: **server02**

Domain Name: none

FQDN: none

System 3:

Host Name: **server03**

Domain Name: **widgets.com**

FQDN: **widgets.com**

When joining the above systems to the **contoso.com** Active Directory domain, all three will be updated (if not already) to **servername.contoso.com** in their local configuration files and then created in AD with that updated information. The new computer object will be created with a **sAMaccountName** equal to the host name of the system and a **dNSHostName** equal to the FQDN.

If preserving the existing FQDN of a system is required, the domain join process can use an optional **--disable hostname** parameter. When used, the system will keep its FQDN and attempt to create the computer object in AD with a matching **dNSHostName**. This scenario is known as a *disjointed namespace*.

For more information, please see "Disjointed Namespaces in AD Bridge" on page 12

Once a system updates its local information, it then attempts to find a computer object in Active Directory with a **dNSHostName** attribute that matches its local value. For example, server03 will query AD looking for any computer object with a **dNSHostName** of **server03.contoso.com** (remember the domain values are updated by default to the domain being joined).

Assuming that server03 does not find a computer object in the directory with its desired **dNShostName**, it will attempt to create one. If it does find a computer object with a matching **dNSHostName**, it will attempt to join using the existing object. This is true even when the **sAMAccountName** of the computer object does not match the host name of the system. This function allows AD Bridge to support pre-staged computer objects.

For more information, please see Avoid Generated (Hashed) Computer Names.

If the system decides to create a computer object, it must then determine the **sAMccountName** for the computer object. As discussed above, this will always default to the host name of the local machine. However, if the **sAMAccountName** is already present in the

directory (with a different **dNSHostName**) or if it is greater than 15 characters, the system will generate a hashed computer name to ensure uniqueness.

For example:

System 1 (already exists in AD):

sAMAccoutName: **server01**

dnsHostName: **server01.contoso.com**

System 2:

Host Name: **server01**

Domain Name: **widgets.com**

FQDN: **server01.widgets.com**

When joining, System 2 will attempt to use a **sAMAccountName** of server01, which is already in use by System 1. Since, by default, System 2 will also update its FQDN to the AD domain it is joining it will attempt to overwrite the existing object.

However, when joining with the **--disable hostname** switch, System 2 will keep its FQDN as **server01.widgets.com**. Since no other computer object exists with this FQDN, and because another computer object already exists with a **sAMAccountName** of **server01**, System 2 will generate a hashed value (for example, **server0-p37ym1j**) to use as the new name.

A successful join, either with a new computer object or using an existing one, is always dependent on the rights the joining user has to the existing OU or objects.

How to Delegate Control in Active Directory

Delegate Control to Join AD Bridge Computers to the Domain

Because of the complexities outlined in the Domain Join Process Overview, the basic delegation procedure described in the "Delegation of Control Overview" on page 3 is not sufficient. Additional modifications are required to ensure that a computer account can join the domain in all circumstances. The following procedure can be performed either at the root of the domain, the **Computers** OU, or one or more specific OUs.

We recommend designating a specific OU to hold all subordinate AD Bridge joined systems and that delegation is granted over this OU. This is the preferred method since scoping the location for an account to create computer objects in the domain is more secure. Additionally, joining systems directly to a targeted OU ensures that they will receive the appropriate security and configuration setting (for example, GPO) without delay.

> For more information about the basic rights required for joining a computer to a specific OU, please see the following knowledgebase article from Microsoft under the section “Users cannot join a computer to a domain”: https://support.microsoft.com/en-us/help/932455

Following the KB article grants the minimum required rights to limit any errors on domain join. However,AD Bridge requires additional rights not required natively by Windows systems. While domain join errors may not be immediately present when following the KB article only, we recommend you complete the procedure below to ensure optimal operation of AD Bridge.

> ***Note:*** *Granting a user or group Full Control to all computer objects in a subset of the directory (Container or OU) can be sufficient. This might conflict with the desired security policy of the organization. The following procedure outlines the minimal rights required by AD Bridge to work in all join scenarios.*

To delegate control, first identify a specific user or (preferably) group with the right to join. Then, using Active Directory Users and Computers, perform the following tasks:

1. Right-click the OU to add computers to, and then click **Delegate Control**.
2. In the **Delegation of Control Wizard**, click **Next**.
3. Click **Add** to add a user or group to the **Selected users and groups** list, and then click **Next**. We strongly recommend using a group, even if that group only contains one user.
4. On the **Tasks to Delegate** page, click **Create a custom task to delegate**, and then click **Next**.
5. Click **Only the following objects in the folder**,
    - From the list, select **Computer objects**.
    - Select the following options below the object list:
        - **Create selected objects in this folder**
        - **Delete selected objects in this folder**
6. Click **Next**.
7. In the **Permissions** list, select the **General** and **Property-Specific** check boxes.
8. Select the required permissions shown in the table below.
9. Click **Next**, and then click **Finish**.

Permissions

- Read permissions are not absolutely required, but preferred since Write permissions are granted.
- Using a Write permission allows any value to be placed in the attribute without validation. Using only a Validated Write permission might be more secure. However, this might limit AD Bridge's ability to create hashed names when conflicts occur.

For more information on the Microsoft requirements, please see https://support.microsoft.com/kb/932455.

Permission	Microsoft Requirement	AD Bridge Requirement
Reset Password	X	
Read and write Account Restrictions	X	
Validated write to DNS host name	X	
Validated write to service principal name	X	
Read Description		X
Write Description		X
Read dNSHostName		X
Write dNSHostName		X
Read msDS-SupportedEncryptionTypes		X
Write msDS-SupportedEncryptionTypes		X
Read Operating System		X
Write Operating System		X
Read Operating System Version		X
Write Operating System Version		X
Read operatingSystemServicePack		X
Write operatingSystemServicePack		X
Read operatingSystemHotFix		X
Write operatingSystemHotFix		X
Read servicePrincipalName		X
Write servicePrincipalName		X

Delegate Control to Move Computer Objects on Rejoin

AD Bridge supports the ability to target a computer to a specific OU at join time. If the delegation procedure specified in the previous section has been performed, users will be able to join new computer objects in all scenarios, including a targeted OU. However, when attempting to re-join a computer with an existing object already in AD (including pre-staged computer objects), additional complications can arise when requesting a targeted OU.

When rejoining the domain and targeting a specific OU (using the **domainjoin-cli --ou** parameter), LDAP requests a move on the computer object in the AD hierarchy (even if specifying the same OU the object already resides). The modification of the object requires the ability to write to specific attributes of the object which will need to be properly delegated.

To allow rejoins when using the targeted **--ou** parameter the appropriate modDNRequest LDAP operations need to be performed on the existing object. The following permissions must be delegated:

- DELETE_CHILD on the source container or DELETE on the object being moved
- CREATE_CHILD on the destination container.
- WRITE_PROP on the object being moved for two properties: name/Name and cn (or whatever happens to be the cn: RDN attribute for the class. For example, ou for organizational units).

> ***Note:*** *The DELETE_CHILD and CREATE_CHILD are standard permissions granted to an OU if the steps in "Delegate Control to Join AD Bridge Computers to the Domain" are followed (specifically Step #5). Ensure these permissions are granted on any additional OUs the computer objects will be moved between.*

The WRITE_PROP permissions need to be assigned using ADSIEdit as the necessary permissions are not exposed using Active Directory Users and Computers.

To use ADSIEdit to set the appropriate WRITE_PROP permissions, perform the following on each required OU:

1. Launch **adsiedit.msc**.
2. Connect to the Default Naming Context for the domain.
3. Right-click the OU and choose **Properties**.
4. Click the **Security** tab.
5. Click **Advanced**.
6. Click **Add** to add the security principal.
7. Enter the group name to delegate and click **OK**.
8. Select the **Properties** tab.
9. From the menu, select **Descendent Computer Objects**.
10. Select the following Allow permissions:
    - **Read and Write canonicalName**
    - **Read and Write name**
    - **Read and Write Name**
11. Click **OK** on all open dialog boxes.

Join the Active Directory Domain

Default Container

Once the necessary permissions are granted to the appropriate security principals, the **domainjoin-cli** command can be used on the AD Bridge agent to join the computer to the domain's default container:

```
domainjoin-cli join domain.com user password
```

For example, to join to the domain contoso.com with the delegated user jsmith and a password of AlphaOne1, use:

```
domainjoin-cli join contoso.com jsmith AlphaOne1
```

Targeted OU

Once the necessary permissions have been granted to the appropriate security principals, the **domainjoin-cli** command can be used on the AD Bridge agent to join the computer to the domain while a targeting a specific OU:

```
domainjoin-cli join --ou OUName domain.com user password
```

For example, to join to the domain contoso.com under the UnixServers OU, with the delegated user jsmith and a password of AlphaOne1, use:

```
domainjoin-cli join --ou UnixServers contoso.com jsmith AlphaOne1
```

For more information, please see the AD Bridge Installation Guide at www.beyondtrust.com/docs/ad-bridge/getting-started/installation; run the **domainjoin-cli** command from the console; or review the **domainjoin-cli** man page.

Active Directory Naming Limitations

The Active Directory database has certain constraints related to naming computer objects. Because these restrictions are often in conflict with the namespaces used within a Unix/Linux deployment, AD Bridge has ways to integrate the discrepancies between the two. AD Bridge accomplishes this by using the dNSHostName value of a computer object as the primary key to identify a computer in Active Directory. Because of this, the computer's name in AD (**sAMAccountName**) does not necessarily have to match the local host name of the system.

Computer Names Greater Than 15 Characters

Windows systems (and Active Directory) have a computer name (**sAMAccountName**) limit of 15 characters. This limit is honored and enforced throughout Windows. In UNIX environments, machine names can be greater than 15 characters, such as **prod-oracle-db12**. AD Bridge supports computer names greater than 15 characters by generating a new hashed computer name during the join process. The generated name consists of the first seven characters from the original name, a hyphen, and then a unique seven digit code.

For example, the Oracle machine name **prod-oracle-db12** might be joined as **PROD-OR-PC3LRX4**. This generated machine name represents the object in AD only. The name is not used as the host name on the local machine and is not used when communicating with the system from other hosts.

Duplicate Machine Names

A computer name must be unique throughout a particular Windows domain. When migrating UNIX system to AD with AD Bridge it is sometimes necessary to bring multiple machines with the same host name, but different FQDNs, into the same AD domain. For example, **oracle12.prod.domain.com** and **oracle12.dev.domain.com** both share the same host name of **oracle12**.

AD Bridge supports duplicate computer names by generating a new hashed computer name during the join process for subsequent conflicting computer names. The generated name will consist of the first seven characters from the original name, a hyphen, and then a unique seven digit code. The second Oracle machine **oracle12.dev.domain.com** might be joined as **ORACLE12-298GG**. This generated machine name represents the object in AD only. The name is not used as the host name on the local machine and is not used when communicating with the system from other hosts.

To preserve the FQDN of each machine, the **--disable hostname** parameter must be specified when performing the join operation.

For more information, please see "Disjointed Namespaces in AD Bridge" on page 12

SPN Uniqueness

While not directly related to any of the delegation issues listed, the following limitation is worth bearing in mind. Beginning with Windows 2012 R2, Microsoft started implementing SPN uniqueness across the entire Active Directory forest. Since each computer object registers a short SPN in the form of **HOST/COMPUTERNAME** even computers with different FQDNs will have problems joining across multiple domains with duplicate computer names.

In previous versions (prior to 2012 R2), two computer objects with the same **sAMAccountName** could exist in different domains in the same forest. For example, **DOMAINA\SERVERA** and **DOMAINB\SERVERA**. While this may not have been recommended it was allowed and some organizations may depend on this feature based on their process. In 2012 R2 and later, joining with the same **sAMAccountName** as another system in the forest will fail.

This is a limitation of Windows and not AD Bridge.

For more information, please see SPN and UPN Uniqueness at https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/manage/component-updates/spn-and-upn-uniqueness.

Avoid Generated (Hashed) Computer Names

In some environments, it may be preferable to control the value on a created computer object instead of relying on the dynamically hashed values. This can be accomplished by pre-staging the computer object. When pre-staging the computer account it is still necessary to choose a unique name for each computer. The Oracle machines above, for instance, might be pre-staged as **ORACLE12-PROD** and **ORACLE12-DEV** prior to join.

To pre-stage a computer account to avoid generated/hashed names, follow the standard procedure for pre-staging computer accounts. Then, perform the following:

1. Using Active Directory Users and Computers, ADSI Edit, or another tool able to directly modify AD attributes, locate and view the properties of the pre-staged computer account.
2. Locate and modify the **dnsHostName** attribute to equal the FQDN of the computer that will be joined using this computer account.
3. Save all changes.

Disjointed Namespaces in AD Bridge

In some environments, AD Bridge agents must be joined with FQDNs that differ from the Active Directory domain name. For example, the computer **oracle12.prod.domain.com** might need to join the **company.com** domain. This is a disjointed namespace scenario since **prod.domain.com** and **company.com** are different DNS domains.

By default, the **domainjoin-cli** command updates the FQDN on the AD Bridge agent to match the AD domain being joined. To prevent this behavior, and allow the host to retain its original domain name, use the **--disable hostname** parameter. For example:

```
domainjoin-cli join --disable hostname --ou UnixServers contoso.com jsmith AlphaOne1
```

Only accounts given the authority to modify the **dnsHostName** attribute can join a computer with a domain name that differs from the Active Directory name. Accounts without this authority may see an error similar to the following when attempting to join a computer with a disjointed namespace:

```
Error: LW_ERROR_LDAP_CONSTRAINT_VIOLATION [code 0x00009d7b]
```

There are two ways to grant the rights necessary.

For more information, please see Delegate Control to Join AD Bridge Computers to the Domain.

Option #1: Grant Write permission to the dnsHostName attribute of the computer object:

Any account with **Write** permission can modify this attribute directly. This is the quickest and most direct way to grant the ability to join with a disjoint namespace and should already be defined if choosing to use the **Write** instead of **Validated Write** permission on **dNSHostName** attribute.

Option #2: Grant Validated Write permission and Modify the domain's AllowedDNSSuffixes:

This method grants a more restricted **Validated Write** permission to the computer object's **dnsHostName** value. Any attempt to modify this value is validated against a list of allowed domains listed in the domain's Naming Context (NC).

In addition to granting the **Validated Write** option to the computer object, the Domain NC must be updated. To modify this behavior, register additional namespaces in the **msDS-AllowedDNSSuffixes** attribute.

For more information, please see Create a Disjoint Namespace at https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc755926(v=ws.10).

Common Error Messages when Delegating Permissions

The following error message may be seen if proper permissions have not been delegated. The error messages are an interpretation of the direct error received from AD for the LDAP operation. To completely ensure the cause of the error, a packet capture of the join is usually required. A full diagnostic of the join, including the capture, can be created by running the AD Bridge support tool with the domain join parameter:

```
/opt/pbis/libexec/pbis-support.pl -dj
```

Follow the prompts to attempt the join and provide the compiled tarball to BeyondTrust Technical Support.

These error messages usually indicate insufficient rights have been given to join.

```
Error: LW_ERROR_LDAP_CONSTRAINT_VIOLATION [code 0x00009d7b]
```

```
Error: ERROR_ACCESS_DENIED [code 0x00000005]
```

For more information, please see "How to Delegate Control in Active Directory" on page 6.

The following error message usually indicates insufficient rights have been given to move a pre-existing computer object.

```
Error: LW_ERROR_LDAP_INSUFFICIENT_ACCESS [code 0x00009d8b]
```

For more information, please see Delegate Control to Move Computer Objects on Rejoin.

The above error may also occur when re-joining to the same OU when using the **--ou** parameter. It is not necessary to specify the **--ou** parameter to rejoin a computer to the same OU.

Additional References

Please also see the following references:

- Default limit to number of workstations a user can join to the domain at https://support.microsoft.com/en-us/help/243327/default-limit-to-number-of-workstations-a-user-can-join-to-the-domain
- 3.1.1.5.3.1.1.2 dNSHostName at https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-adts/5c578b15-d619-408d-ba17-380714b89fd1
- 3.1.1.5.3.1.1.4 servicePrincipalName at https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-adts/28ca4eca-0e0b-4666-9175-a37ccb8edada
- 2.1 Attribute name at https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-ada3/7cda9531-17ee-4b9a-a942-c3bb69c21754
- 2.110 Attribute cn at https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-ada1/ffc29c00-e8d1-4111-9562-8e6ae308c43a