Understand On-Demand Resource Reservation for AD on ISE 3.3 Patch 4

Contents

Introduction

This document describes about On-Demand Resource Reservation for Active Directory on ISE 3.3 Patch 4

Pre-Requisites

Knowledge on Cisco Identity Services Engine (ISE)

Knowledge on Active Directory (AD)

Knowledge on ISE and AD Integration

Components Required

The information in this document is based on these software and hardware versions

- Cisco Identity Services Engine 3.3 Patch 4
- Microsoft Windows Active Directory 2016 or latest

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, ensure that you understand the potential impact of any command.

Background Information

AD Authentications are sometimes slow and eventually fail. Possible reasons could be ADID Queue starting to pile-up or All ADID Pool Threads getting exhausted.

More details on ADID:

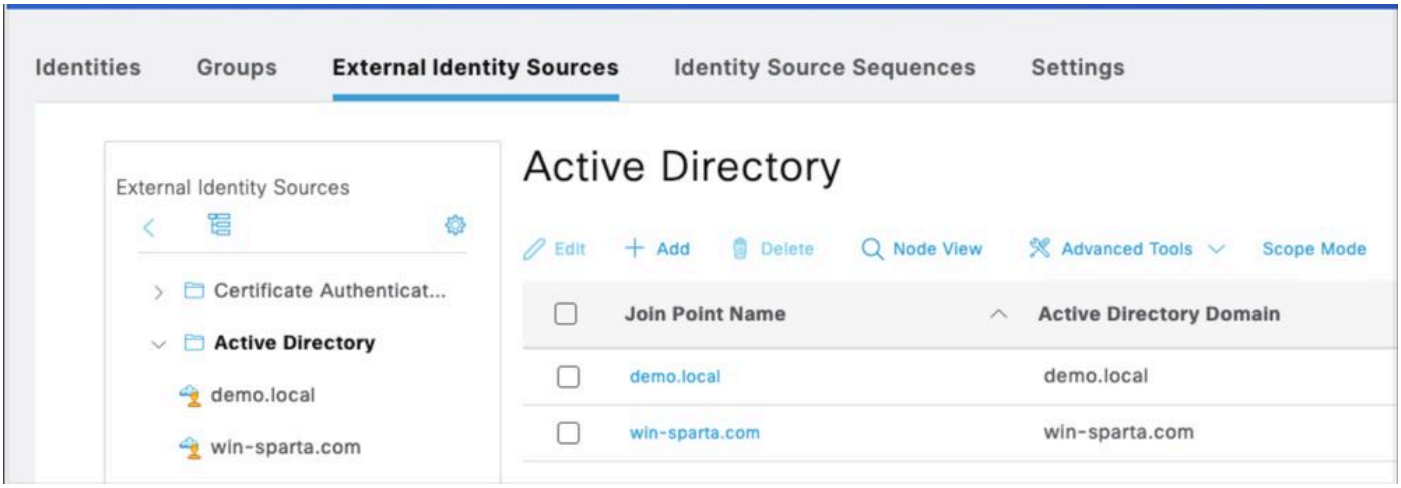
An ADID, also known as a distinguished name (DN), is a string that uniquely identifies an object within the Active Directory directory. They are used to locate and manage objects within the Active Directory domain. ADIDs are crucial for managing user accounts, permissions, and other resources within an Active Directory environment.

A typical ADID must look like this: CN=John Doe,OU=Sales,DC=example,DC=com; where,

CN=John Doe: Represents the common name of the user, John Doe.
OU=Sales: Represents the organizational unit (OU) where the user belongs, in this case, the Sales department.
DC=example,DC=com: Represents the domain components, which is example.com.

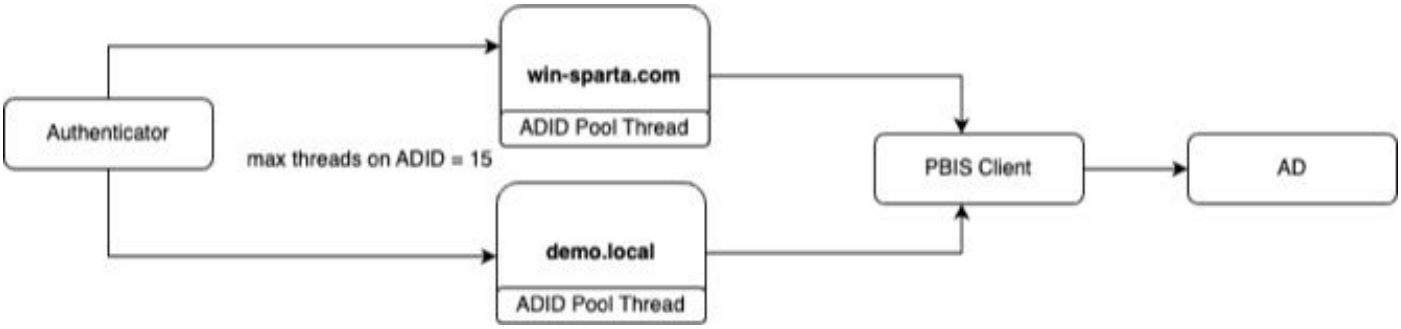
For Example:

Refer to Picture 1: A typical AD Join Point Configuration



Picture 1: AD Join Points

Refer to Picture 2: A typical AD flow diagram with 2 Join Points



Picture 2: A Typical AD Flow Diagram

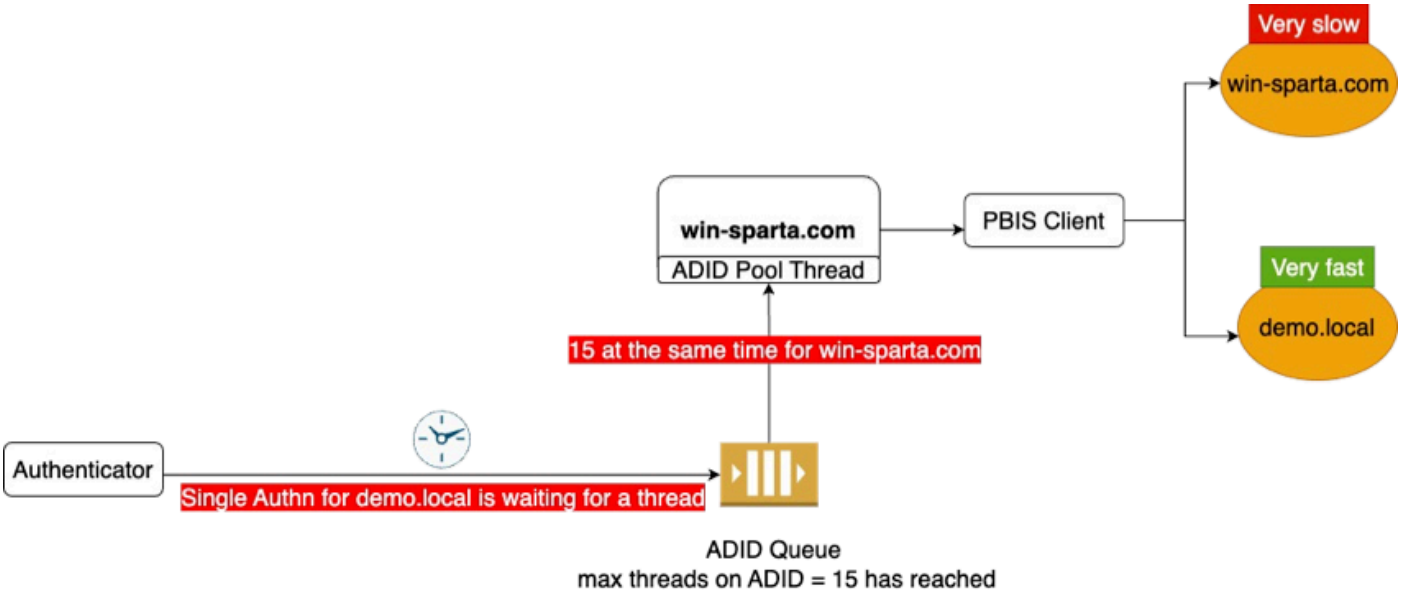
Symptom

Slow Join Point under same ADID Thread Pool

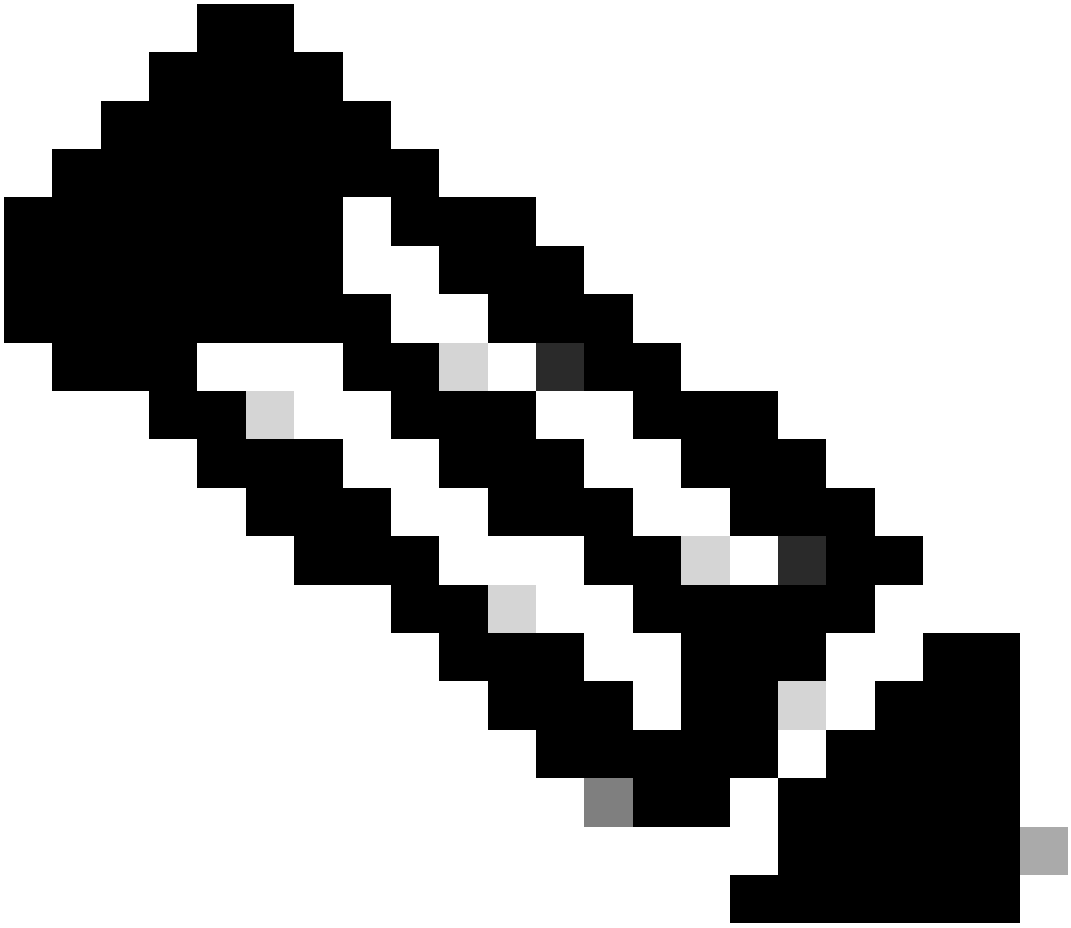
Problem

1. What would be the Consequences of One of the Join Points being very slow? For instance, if 15 authentications are sent to ISE at the same time for "demo.local" and "demo.local" is unusually slow, we would need to wait for the response from "demo.local" before handling the subsequent win-sparta authentication.
2. What if both the Join Points share the same ADID Thread Pool under One Join Point?

Refer to Picture 3: Flow diagram of Slow Joint Point



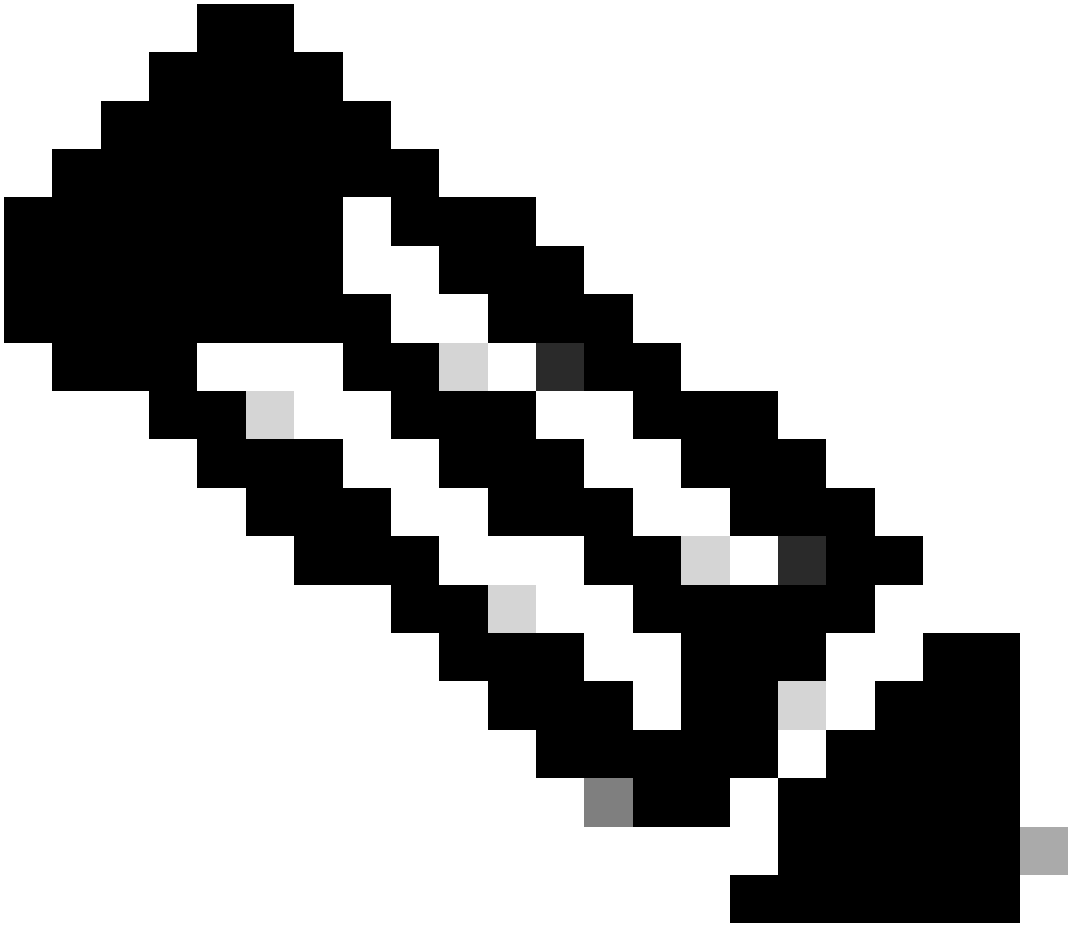
Picture 3: Problematic Flow



Note: Here, all 15 Threads are occupied by win-sparta.com at the same time leaving no thread for demo.local

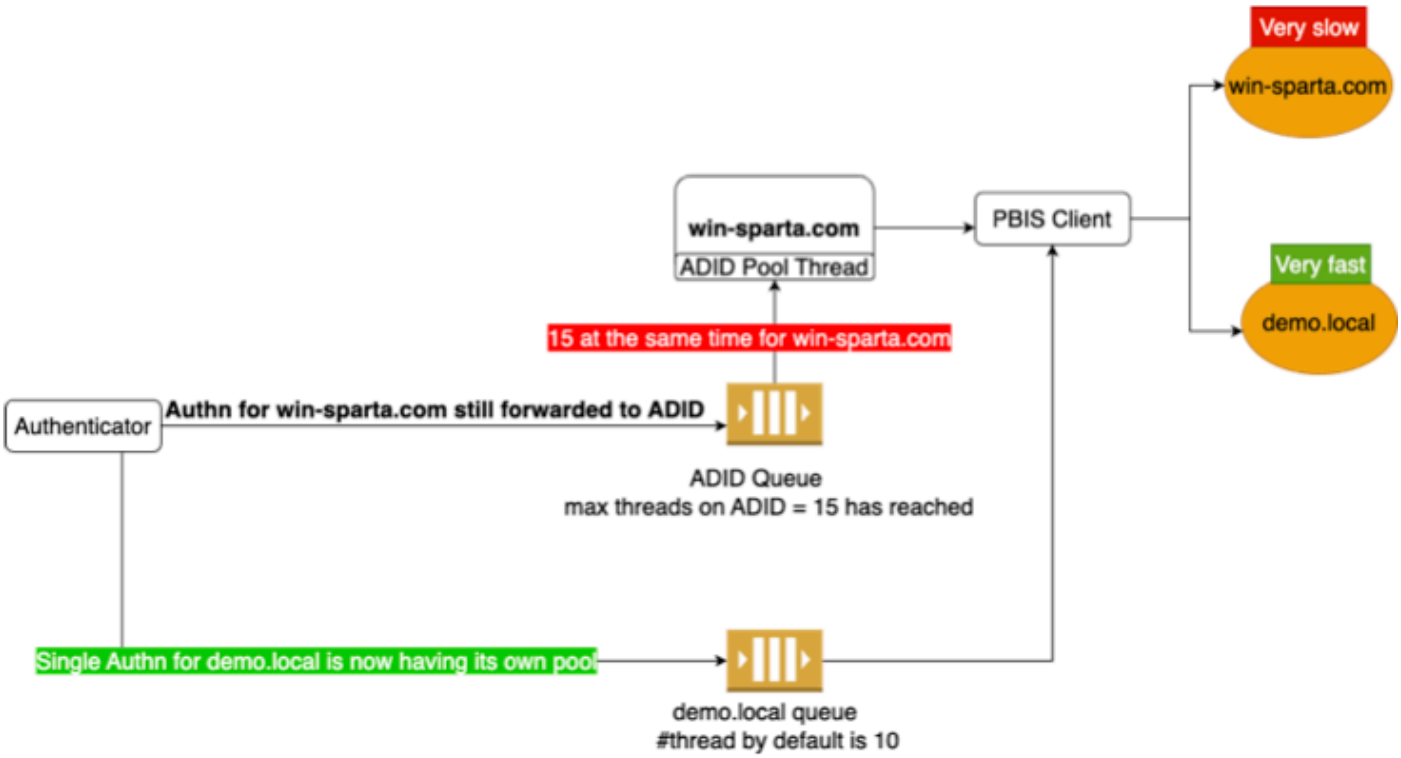
Solution

- Default behavior is a Common Thread Pool for all AD join points
- However, Admins can segment each join point to have its own resources.



Note: When AD Prioritisation is applied, the default is 10 Threads per Thread Pool.

Refer to Picture 4: Flow diagram of On-Demand Reserved Joint Point



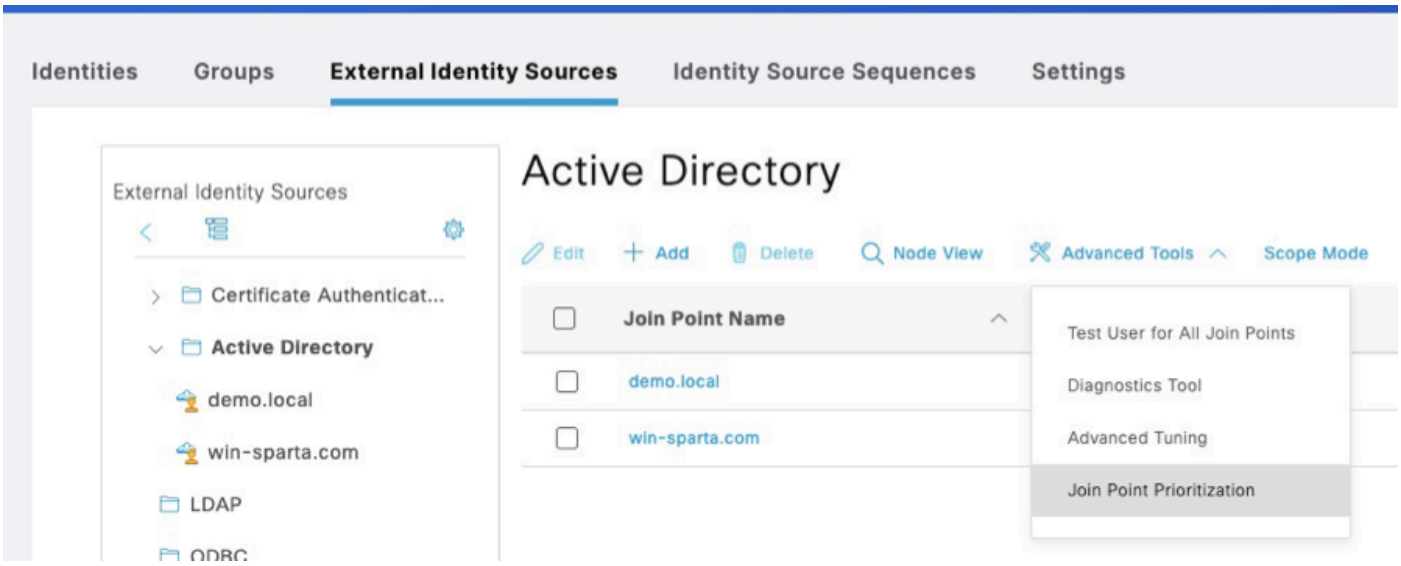
Picture 4: Solution Flow

Step by Step Configuration

Step1: Create 2 separate AD Join Points. Here for example we have: demo.local and win-sparta.com

Step2: Create Join Point prioritization after AD Join Point creation.

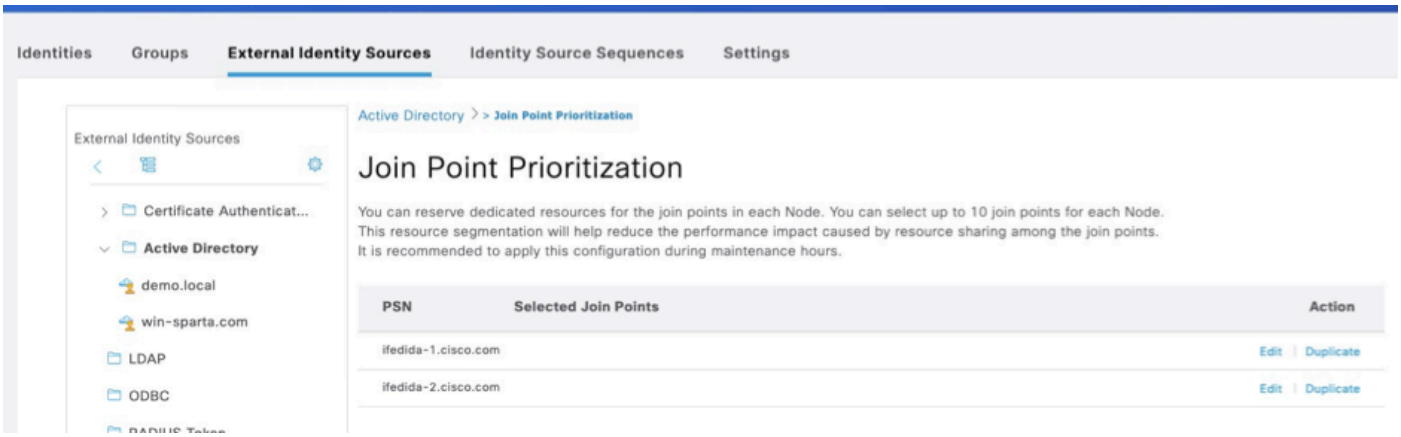
Refer to Picture 5:



Picture 5: Join Point Prioritisation

Step3: Under Join Point Prioritisation, select the PSN that you prefer to reserve dedicated AD resources. Click Edit.

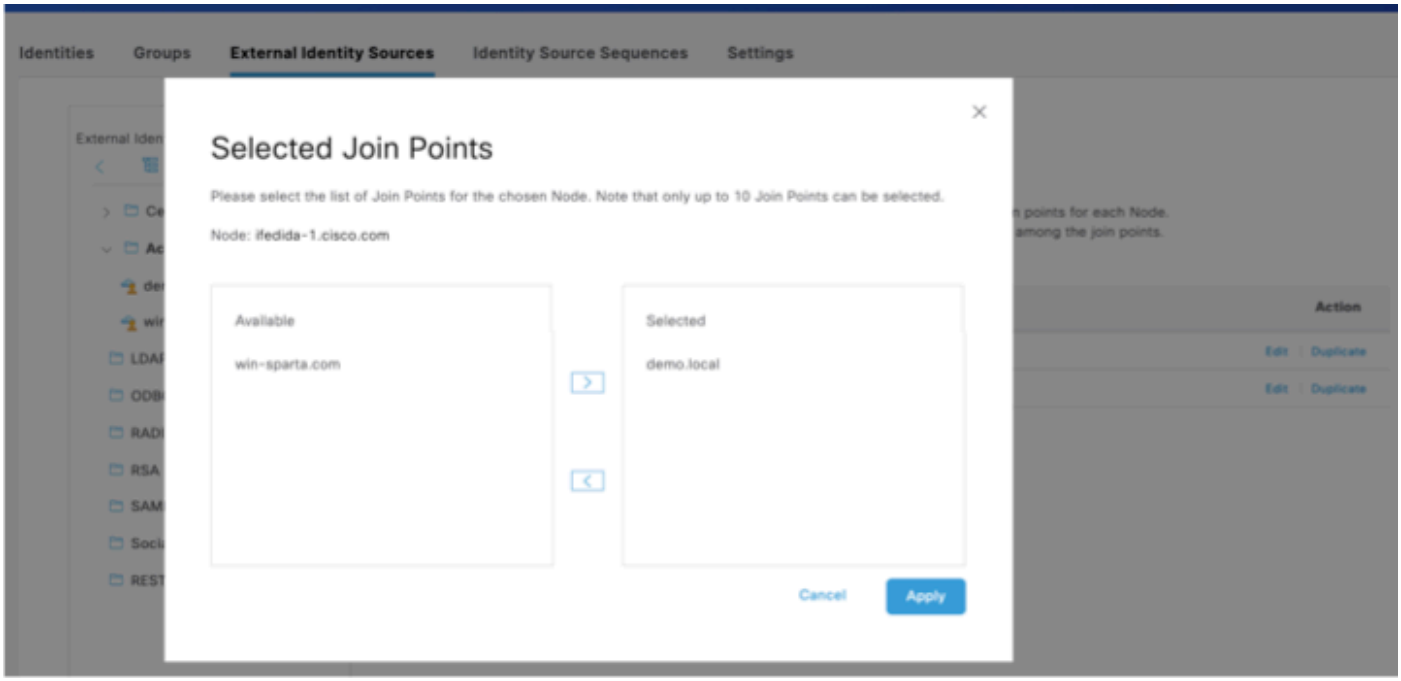
Refer to Picture 6:



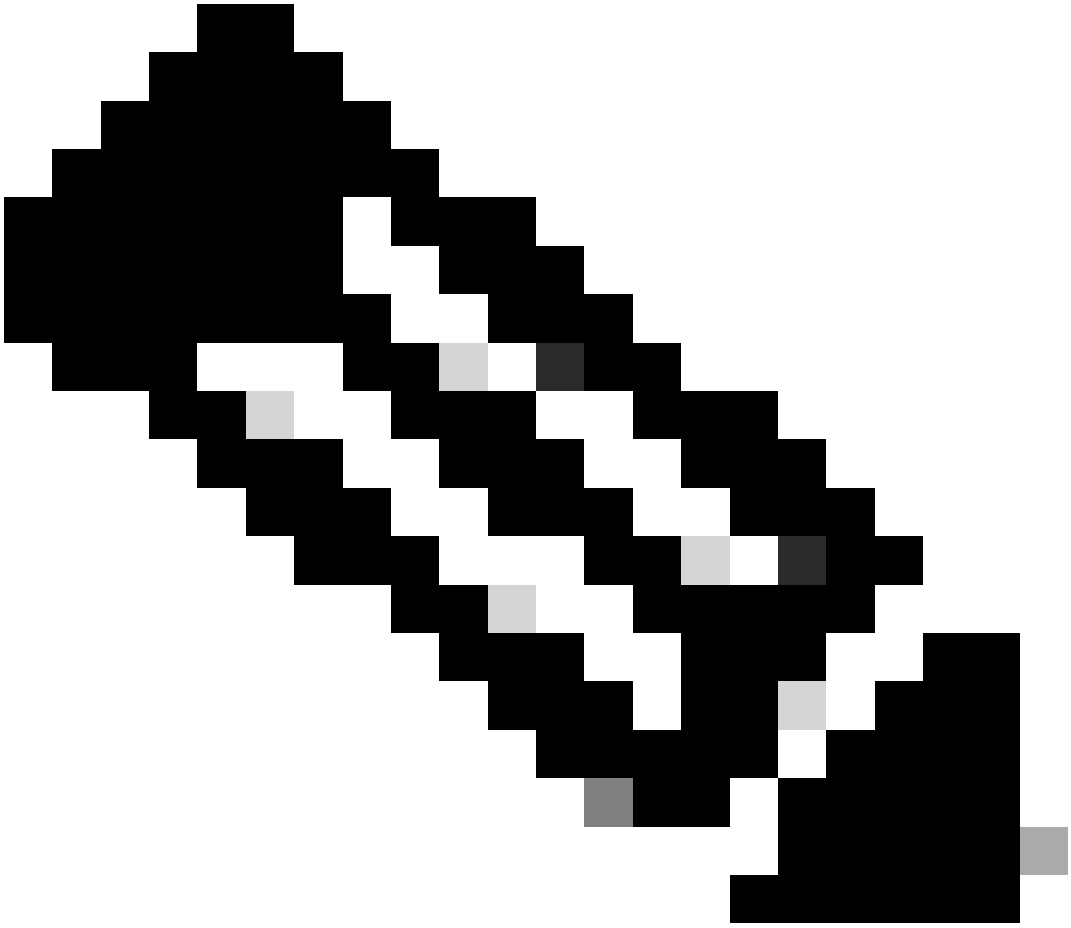
Picture 6: Edit PSN

Step4: Select the Preferred Join Point for the Preferred PSN.

Refer to Picture 7:



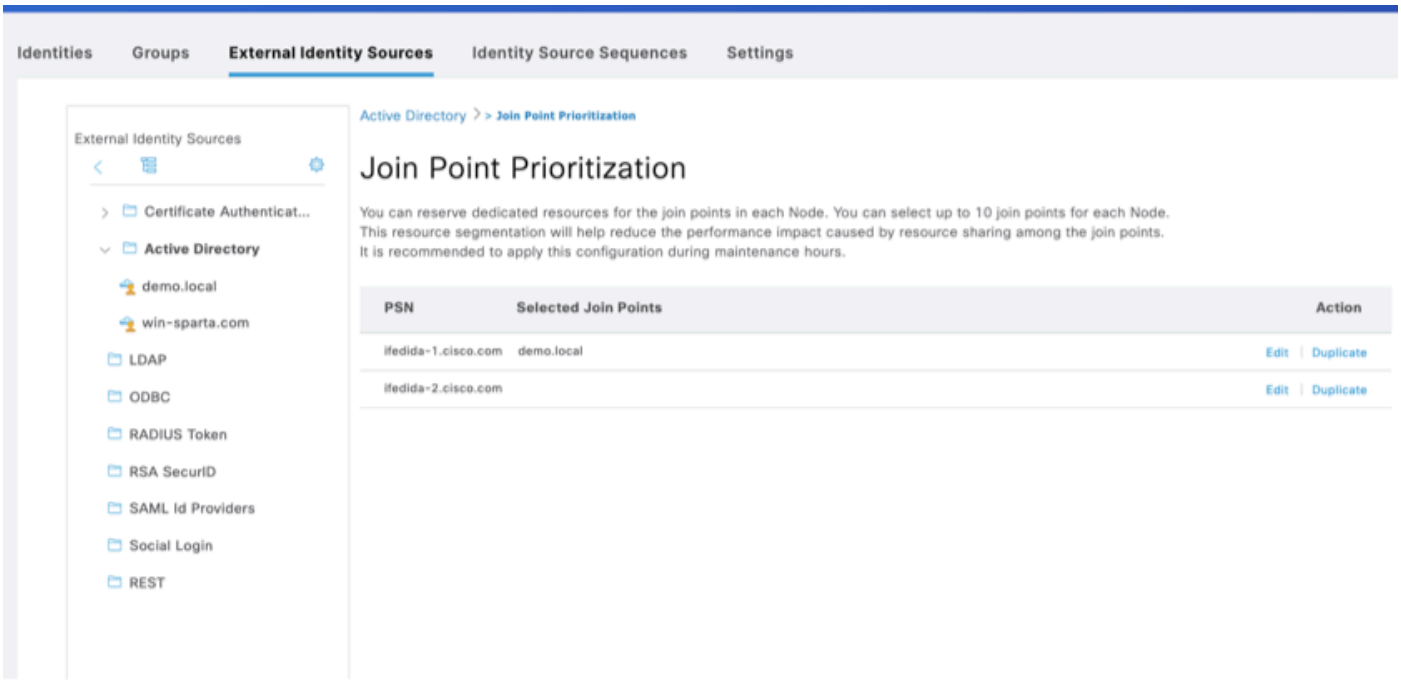
Picture 7: Selected Join Point



Note: Any Join Points not included in the Prioritisation utilize the Common Thread pool, which has a maximum limit of 15 threads.

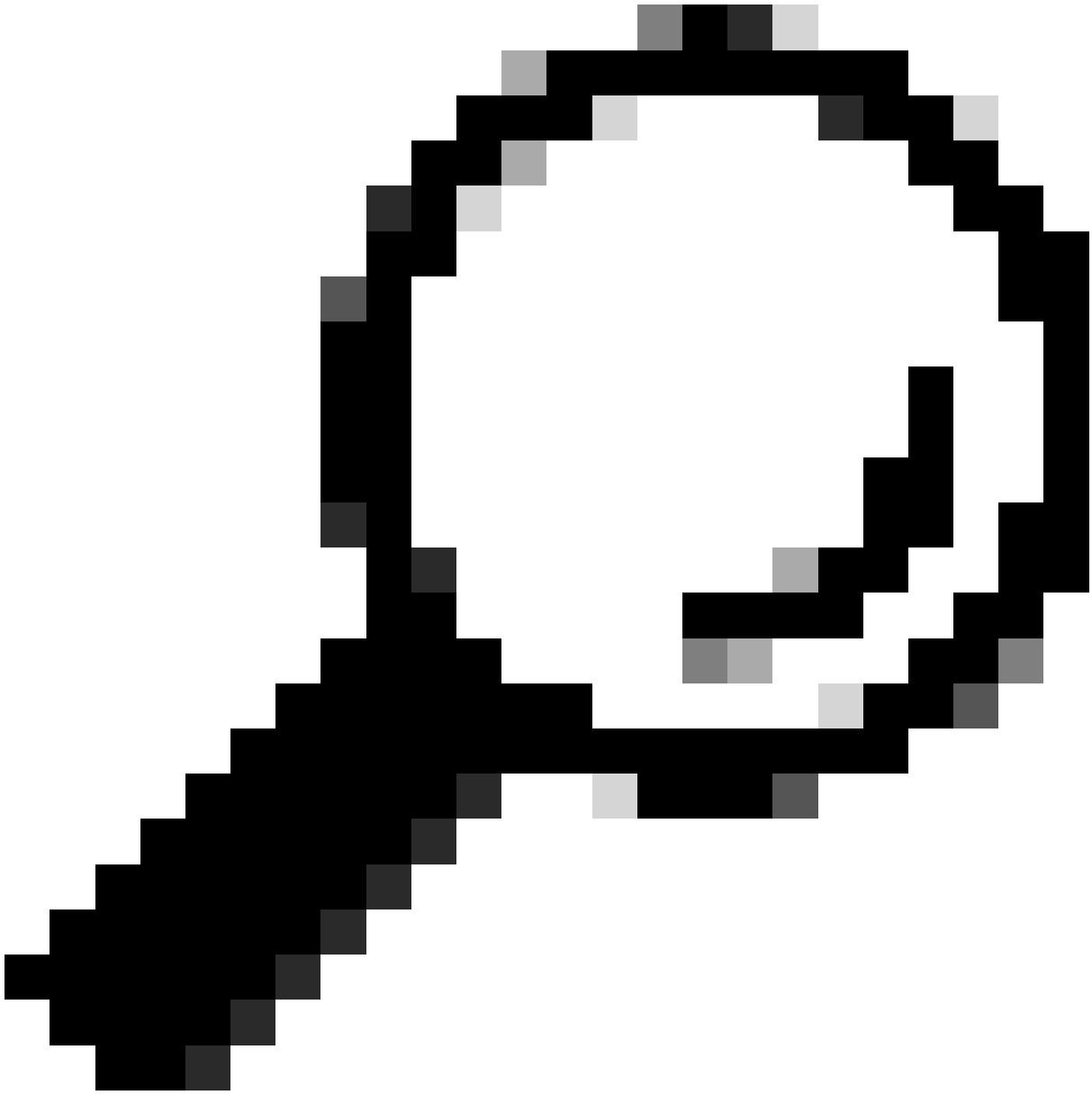
Step5: Prioritisation is Finalised

Refer to Picture 8:



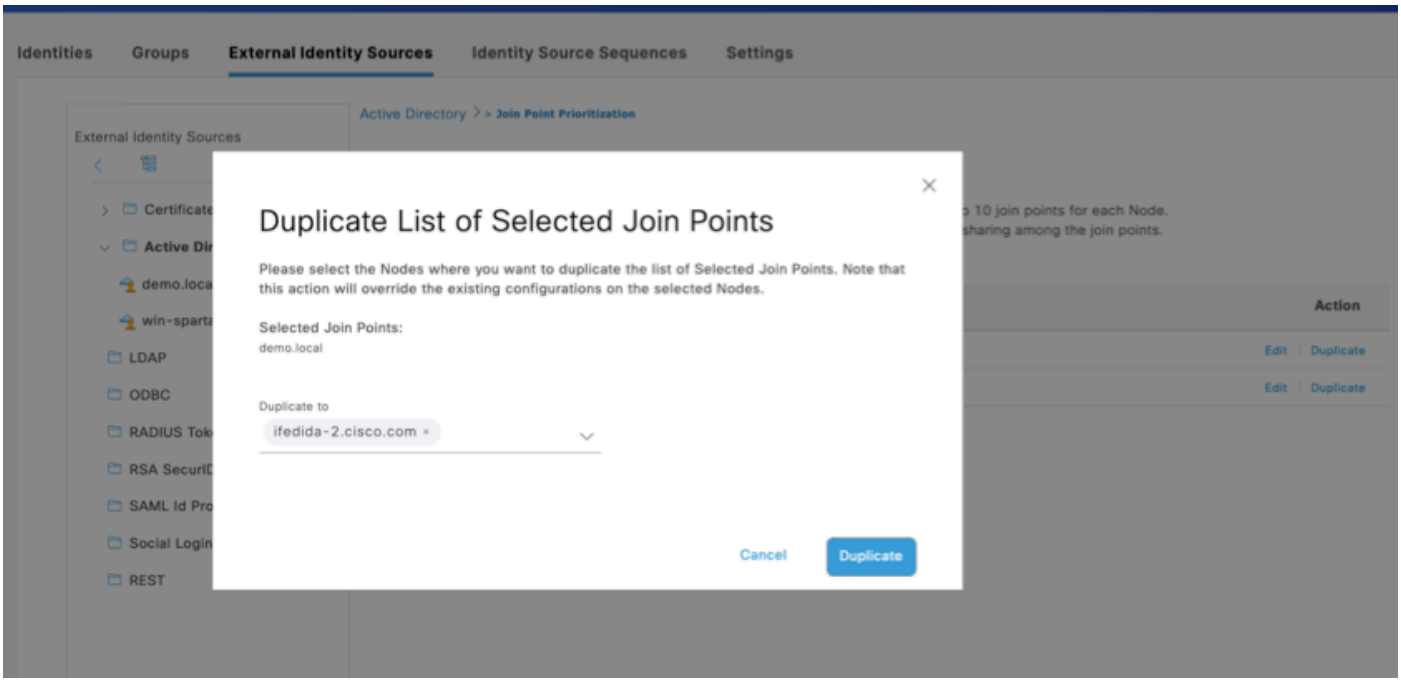
Picture 8: Prioritization config

Additional Details



Tip: If you would like to replicate the same settings to other PSNs, you can use the Duplicate option. Select the desired PSN, and choose the Join Point to be duplicated along with the original Prioritisation.

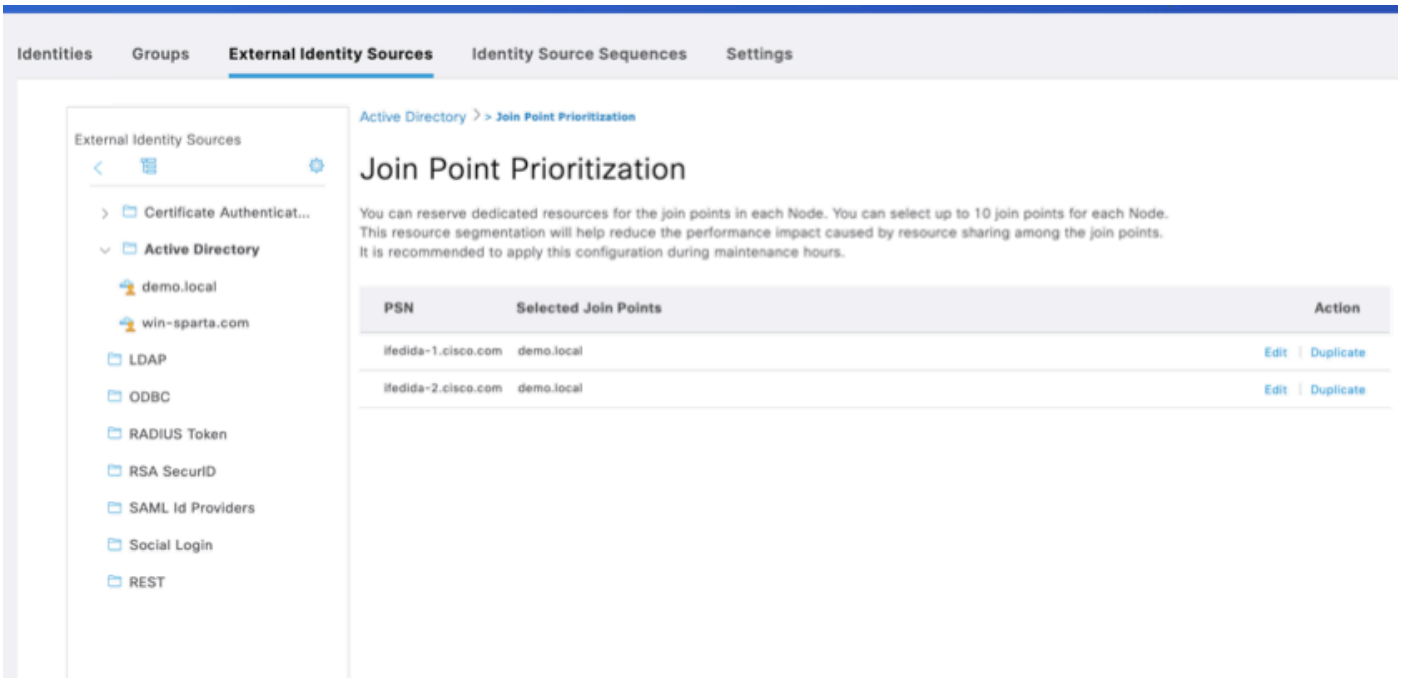
Refer to Picture 9: Configuration Tip:



Picture 9: Duplicate Prioritization Config

Step6: Final List after Duplication

Refer to Picture 10:



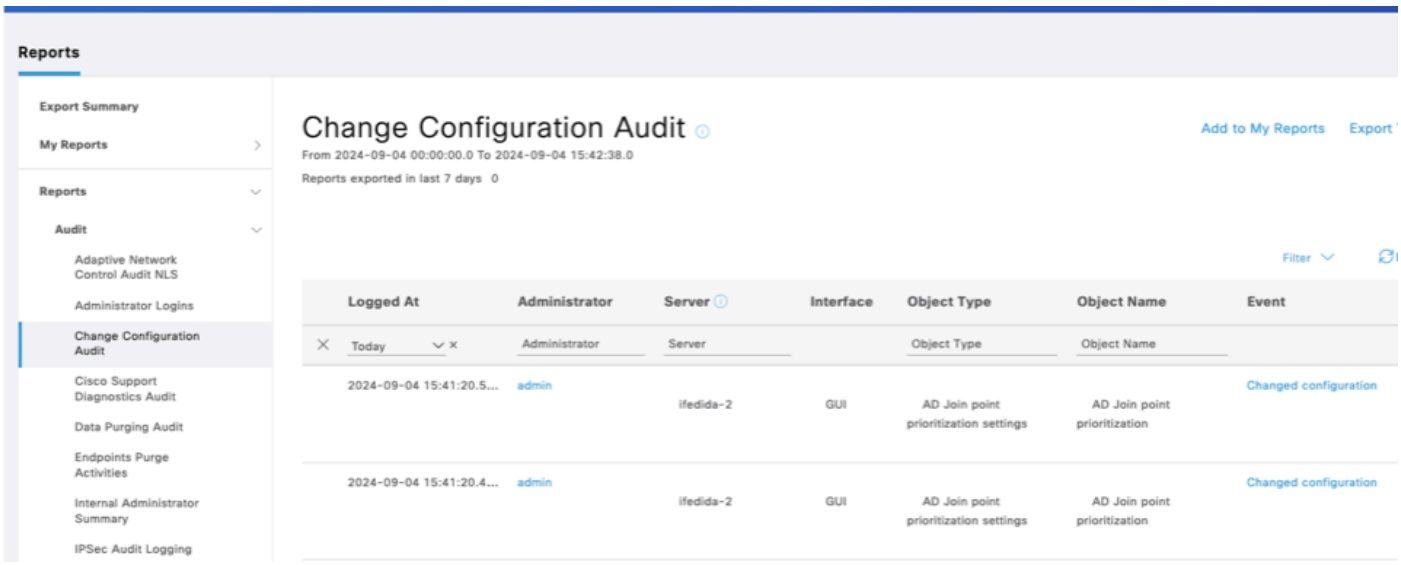
Picture 10: Final List after Prioritization

Troubleshooting

Verification

Verify the Configuration Changes. Navigate to: Operations > Reports > Audits > Change Configuration Audit

Refer to Picture 11:



Picture 11: Config Audit Report

Logging

- Enable debug level for runtime-AAA logs.
- Analyse prrt-server.log
  Refer to Picture 12:

Picture 12: Debug Log Config

Log Snippets

prrt-server.log [DEBUG]: **Default** Log:

EventHandler,2024-08-23 07:16:48,135,DEBUG,0x7fecd2ccc700,Allocated **default** thread pool : ADIDStore to IDP : win-sparta.com_wxETlH16Pk_106

prrt-server.log [INFO]: When we set **Dedicated Resources**:

- ActiveDirectoryIDStore,2024-09-08 16:52:01,048,INFO ,0x7f2452ccf700,Allocated thread pool : ADThreadPool0 to IDP : **win-sparta.com**_wxETlH16Pk_106
- ActiveDirectoryIDStore,2024-09-08 16:57:11,258,INFO ,0x7f2452ccf700,Allocated thread pool : ADThreadPool1 to IDP : **demo.local**_6EcNs6UzwX_89

prrt-server.log [INFO]:

- **Before** we have set dedicated resources:
  - EventHandler, 2024-09-02 08:45:54,673,INFO,0x7fafb793c700,Passed event to the next thread pool name=**ADIDStore**, queue size=1,EventDispatcher.cpp:757

- **After** we have set dedicated resources:
  - EventHandler,2024-09-02 08:45:54,673,INFO ,0x7f4867ff9700,Passed event to the next thread pool name=**ADThreadPool0**, queue size=1,EventDispatcher.cpp:841

To Track Thread Pool Usage of "ADThreadPool0":

1. **0x7f57792f7700**,Passed event to the next thread pool name=**ADThreadPool0** (few logs back StackID:0x7f57a4f761c0)

2. 0x7f57732c7700,Stack: 0x7f57a4f761c0 Calling ActiveDirectoryIDStore: Method MethodCaller<ActiveDirectoryIDStore, PlainAuthenticateAndQueryEvent>

3. *0x7f57732c7700,cntx=0000210117,sesn=ifedida-1/515863662/5273,CPMSessionID=C0A31430000000800018958,user=abcd,CallingStationID=[CAD] 956: CAD_PAPAuthenticate (abcd) called*

4. 0x7f57732c7700,cntx=*0000210117*,sesn=ifedida-*1/515863662/5273*,CPMSessionID=C0A31430000000800018958,user=*abcd*,CallingStationID=[CAD] 1026: CAD_PAPAuthenticate (abcd) succeeded

5. *0x7f57732c7700,Passed event to the next thread pool name=Main*

FAQ

Question: How many AD Join Points can ISE support?

Answer: You can configure up to 50 Active Directory join points on a single ISE deployment.

Question: If I have multiple AD Join Points, can I still use On-Demand Prioritisation?

Answer: Yes

Question: What is the default thread size without Prioritisation for a single domain?

Answer: 15 Threads

Question: If I configure Prioritisation, how is the calculation done? Consider, a 3 Join Point scenario - domain1.com, domain2.com and domain3.com with domain1.com is not configured for Prioritisation and domain2.com and domain3.com are configured for Prioritisation.

Answer: If domain1 is not configured for Prioritisation, domain1.com utilizes the common 15 threads available - all at the same time. However, since domain2.com and domain3.com are configured with Prioritisation, they use 10 Threads each by default and not follow/utilize the common 15 Threads Pool.