**Connect Wiz Connector (New) to Wiz Outpost (Azure)**

This document describes the steps should be taken to connect the new version of Wiz Connector to Wiz Outpost for Azure.

**Note:** this is currently not available in Wiz portal, and currently require the operation by Wiz API. The UI version should be available in few days.

**Configuration**  
Deployment of the connector is accomplished with a per-tenant Wiz app. It is a multi tenant app, hence**it doesnt require you to create any applications in your tenant**. The only requirement is to create a service principal for the app (this requires only *directory.serviceprincal.create*by the user doing the deployment).

**Azure permissions**  
The connector itself assigns Azure roles to the app, to allow for Wiz scanning process. No AAD permissions required for this step.

**AAD permissions**  
For the AAD part, Wiz performs an additional application consent step which assigns the AAD read permissions to allow for AAD identity analysis. Note that the script has a special flag  **--with-aad-consent-only**so this step can be performed separately. Running the script initially without the AAD consent flag, will allow Wiz to start scanning without the additional identity analysis.

Wiz can scan the cloud and workloads without any AAD permissions. The AAD access allows Wiz to perform identity analysis and part of the CIEM functionality, analyze users & groups with access to sensitive resources, lateral movement paths, excessive privileges etc.  
See [**here**](https://docs.wiz.io/wiz-docs/docs/azure-req-perm#wiz-azure-active-directory-enterprise-application-aka-wiz-for-azure-ad)the details for each permission in AAD.  
The most important permissions is Directory.ReadAll which allows Wiz to do the user permissions analysis.

Wiz Enterprise App fetches cloud metadata via Azure API calls, for the required permissions, please see [**here**](https://docs.wiz.io/wiz-docs/docs/azure-outpost-req-perm#wiz-enterprise).

**Procedure**

1. Get your “wiz-managed-id”.

* Wiz Portal | [API Explorer](https://app.wiz.io/api-explorer)
* Run the following query to get the “wiz-managed-id” and save it for the next steps.

**Query:**

query TenantManagedIdentity($cloudProvider: CloudProvider!) {

tenantIdentityClient(cloudProvider: $cloudProvider) {

metadata

}

}

**Variables:**

{

"cloudProvider": "Azure"

}

**Results:**

{

"data": {

"tenantIdentityClient": {

"metadata": {

"appId": "<APP-MANAGE-ID>",

"name": "wiz-<WIZTENANTID>"

}

}

}

}

Graphical user interface, application

Description automatically generated with medium confidence

1. Run the connector deployment script.

**Before** running the script, please note about the following values.

* **--with-aad-consent** Will create the consent application. Using the new connector, we do not have any more 2 consent applications (“**Wiz for Azure**”, and “**Wiz for Azure AD**“) that were used by the Old connector (New connector from March 2023). The new connector will use only one consent application that will contain a unique name based on the customer tenant that will be received in section 1 above (<**name**>) + roles + all the relevant resources for the Wiz Connector deployment. **Required AAD Admin permissions.**
* **--with-aad-consent-only** Creates ONLY the consent APP. Using the new connector, we do not have any more 2 consent applications (“**Wiz for Azure**”, and “**Wiz for Azure AD**“) that were used by the Old connector (New connector from March 2023). The new connector will use only one consent application that will contain a unique name based on the customer tenant that will be received in section 1 above (<**name**>) value received from the API. This is mostly required for users that are not allowed to create the consent applications and need to run it by another teams. In this case, you could run the script below with this flag (by the team who could create the consent applications), and then, run the script without the “**consent**” flags to complete the deployment. **Required AAD Admin permissions.**
* **Without the flag**, Will run all the relevant resources except the consent applications. **Required Owner or User Access Administrator roles of the subscription or management group.**
* **The new consent application detail:**

**Name:** “0ac12345-1234-1234-1234-e01234567890” should be equal to the “**name**” result of step 1 above.

**Consent Application permissions:**

* Directory.Read.All
* Policy.Read.All
* RoleManagement.Read.All

**Management Group Level:**

curl -fsSL <https://wizio-public.s3.amazonaws.com/deployment-v2/azure/cli/wiz-azure.sh> | bash /dev/stdin outpost **management-group-deployment** <**management-group-id**> --scanner-app-id="<**scanner-client-id**>" --with-data-scanning --wiz-managed-id=<**managed-id**> --with-aad-consent

**Subscription Level:**

curl -fsSL <https://wizio-public.s3.amazonaws.com/deployment-v2/azure/cli/wiz-azure.sh> | bash /dev/stdin outpost **subscription-deployment** <**subscription-id**> --scanner-app-id="<**scanner-client-id**>" --with-data-scanning --wiz-managed-id=<**managed-id**> --with-aad-consent

**Note:** Save the “**Outpost Secret Name**” from the script output for future use.

1. Add Wiz Connector in Wiz Portal

* Wiz Portal | [API Explorer](https://app.wiz.io/api-explorer)
* Run the following query to get the create a new connector.

**Management Group Connector:**

**Query:**

mutation CreateOrgConnector($input: CreateConnectorInput!) {

createConnector(input: $input) {

connector {

id

authParams

type {

id

}

}

}

}

**Variables:**

{

"input": {

"name": "<CONNATOR-NAME>",

"type": "azure",

"authParams": {

"outpostId": "<OUTPOST-ID>",

"diskanalyzerSecretName": "<**Outpost-Secret-Name**>",

"isManagedIdentity": true,

"tenantId": <AAD\_TENANT\_ID>",

"groupId": "<AZURE\_MG\_ID>"

},

"extraConfig": {}

}

}

**Subscription Connector:**

**Query:**

mutation CreateSubscriptionConnector($input: CreateConnectorInput!) {

createConnector(input: $input) {

connector {

id

authParams

type {

id

}

}

}

}

**Variables:**

{

"input": {

"name": "<CONNECTOR-NAME>",

"type": "azure",

"authParams": {

"outpostId": "<OUTPOST-ID>",

"diskanalyzerSecretName": "<**Outpost-Secret-Name**>",

"isManagedIdentity": true,

"tenantId": <AAD\_TENANT\_ID>",

"subscriptionId": "<AZURE\_SUBSCRIPTION\_ID>"

},

"extraConfig": {}

}

}

**Results:**

{

"data": {

"createConnector": {

"connector": {

"id": "<CONNECTOR-ID>",

"authParams": {

"diskanalyzerSecretName": "<**Outpost-Secret-Name**>",

"isManagedIdentity": true,

"outpostId": "<OUTPOST-ID>",

"subscriptionId": "<SUBSCRIPTION-ID>",

"tenantId": "<TENANT-ID>"

},

"type": {

"id": "azure"

}

}

}

}

}

Graphical user interface, application

Description automatically generated

The new connector should be added in the [Connectors](https://app.wiz.io/settings/deployments#~(type~'CLOUD_CONNECTOR~filters~(cloudPlatform~(equals~(~'Azure))))) page.

**Appendix: detailed overview of permissions**

See here of full details on permissions - [Azure Outpost Required Permissions (wiz.io)](https://docs.wiz.io/wiz-docs/docs/azure-outpost-req-perm).

**Azure connector**

**Deployment permissions:**

See this for details: [Connect an Azure Environment (wiz.io)](https://docs.wiz.io/wiz-docs/docs/azure-outpost-connector#before-you-begin)

No need for elevated AAD permissions.

AAD Required permission for deployment is *directory.serviceprincal.create*  in order to create a service principal for the Wiz app.

Azure subscription: Microsoft.Authorization/roleAssignments/\* permission on the subscription(s) or management group(s) in scope

Note- Wiz multi-tenant app is created in Wiz, no need to create an app on the customer side. Only creation of a service principal.

**Required permissions for Wiz:**

Connector requires two sets of permissions, read only permissions for metadata scanning and disk scanner permission for the outpost (not for wiz)

Part 1: Read-only permissions:

Details here - [Azure Outpost Required Permissions (wiz.io)](https://docs.wiz.io/wiz-docs/docs/azure-outpost-req-perm#wiz-enterprise)

|  |  |  |
| --- | --- | --- |
| **Permission** | **Scope** | **Reason** |
| Azure Kubernetes Service Cluster User Role | Subscription or group level | List cluster user credential action in order to get identity configuration in AKS. |
| Azure Kubernetes Service RBAC Reader | Subscription or group level | Allows read-only access to see most objects in a namespace. It does not allow viewing roles or role bindings. This role does not allow viewing Secrets. Wiz needs this role to get identity configuration in AKS. |
| Reader | Subscription or group level | View all resources, without being able to make any changes (allows reading all resources, except secrets). Required for cloud scanning, network assessment, and configuration assessment. |
| Microsoft.Web/sites/config/list/Action | Subscription or group level | Access the Storage account of the main function |
| Microsoft.Web/sites/slots/config/list/Action | Subscription or group level | Access the Storage account of the deployment slot function |
| Microsoft.DataFactory/factories/ querydataflowdebugsessions/action | Subscription or group level | Query data flow debug sessions |

**Required vs optional:**

If there are specific permissions that will be removed, Wiz will still operate but some of the visibility will be impacted.

**Note**: all the permissions are read-only.

**Part 2: Disk scanner permissions**

These are only used by the outpost within the customer control. Not by Wiz.

See details here - [Azure Outpost Required Permissions (wiz.io)](https://docs.wiz.io/wiz-docs/docs/azure-outpost-req-perm#wiz-disk-analyzer---scanner)

* Disk scanning permissions – are required to enable disk scanning. [Azure Outpost Required Permissions (wiz.io)](https://docs.wiz.io/wiz-docs/docs/azure-outpost-req-perm#wiz-disk-analyzer-role)
* Data scanning permissions – are optional to enable DB and bucket scanning (for DSPM) [Azure Outpost Required Permissions (wiz.io)](https://docs.wiz.io/wiz-docs/docs/azure-outpost-req-perm#wiz-data-scanning-role-1)

**Azure AD connector**

Deployment permissions:

 Global Administrator role in AAD is required in order to assign AAD permissions.

**Required permissions for Wiz:**

To provide identity analysis (CIEM) for users & groups and their access into Azure resources, Wiz requires read-only AAD permissions.

See details here - [Azure Required Permissions (wiz.io)](https://docs.wiz.io/wiz-docs/docs/azure-req-perm#wiz-azure-active-directory-enterprise-application-aka-wiz-for-azure-ad)

|  |  |  |
| --- | --- | --- |
| **Permission** | **Scope** | **Reason** |
| AuditLog.Read.All | Active Directory | Wiz cloud events, for excessive privileges and CDR |
| Directory.Read.All | Active Directory | Allows Wiz to read information about users, groups, applications and roles for the Identity Analysis module to process privileges and entitlements |
| Policy.Read.All | Active Directory | Allows Wiz to read information about Active Directory policies such as Conditional Access Policies |
| RoleManagement.Read.All | Active Directory | Allows Wiz to read information about role-based access control (RBAC) settings for all supported RBAC providers |

**Required vs optional:**

Directory.Read.All – required for users & groups.

Other permissions – optional – Wiz can operate without them with reduced visibility.

Good Luck!

Created by Yossi.Marzuk@wiz.io