



Microsoft Defender for Identity

CyberArk – MDI Integration

Name of Company: Microsoft

Website: <https://www.microsoft.com>

Name of Product: Microsoft Defender for Identity (MDI)

Date: May 2026

Table of Contents

<i>Solution Overview</i>	<i>3</i>
<i>1 Integration Description and Diagram</i>	<i>4</i>
1.1 Accounts Match.....	4
<i>2 Requirements</i>	<i>5</i>
<i>3 CyberArk Configuration.....</i>	<i>5</i>
3.1 Define the application ID (app ID) and authentication details	5
3.2 Set permissions for application access	7
3.3 Create a REST API user for CyberArk REST API Calls.....	9
<i>4 Microsoft Defender for Identity (MDI) Configuration.....</i>	<i>9</i>
4.1 Allow MDI IP range	9
4.2 Create an Entra application to access the MDI without a user	9
4.3 Setup Authentication for MDI API.....	12
<i>5 CyberArk Connector Configuration & Execution</i>	<i>13</i>
<i>6 Troubleshooting.....</i>	<i>14</i>
6.1 Common Errors handling.....	14
<i>7 Partner contact info.....</i>	<i>17</i>

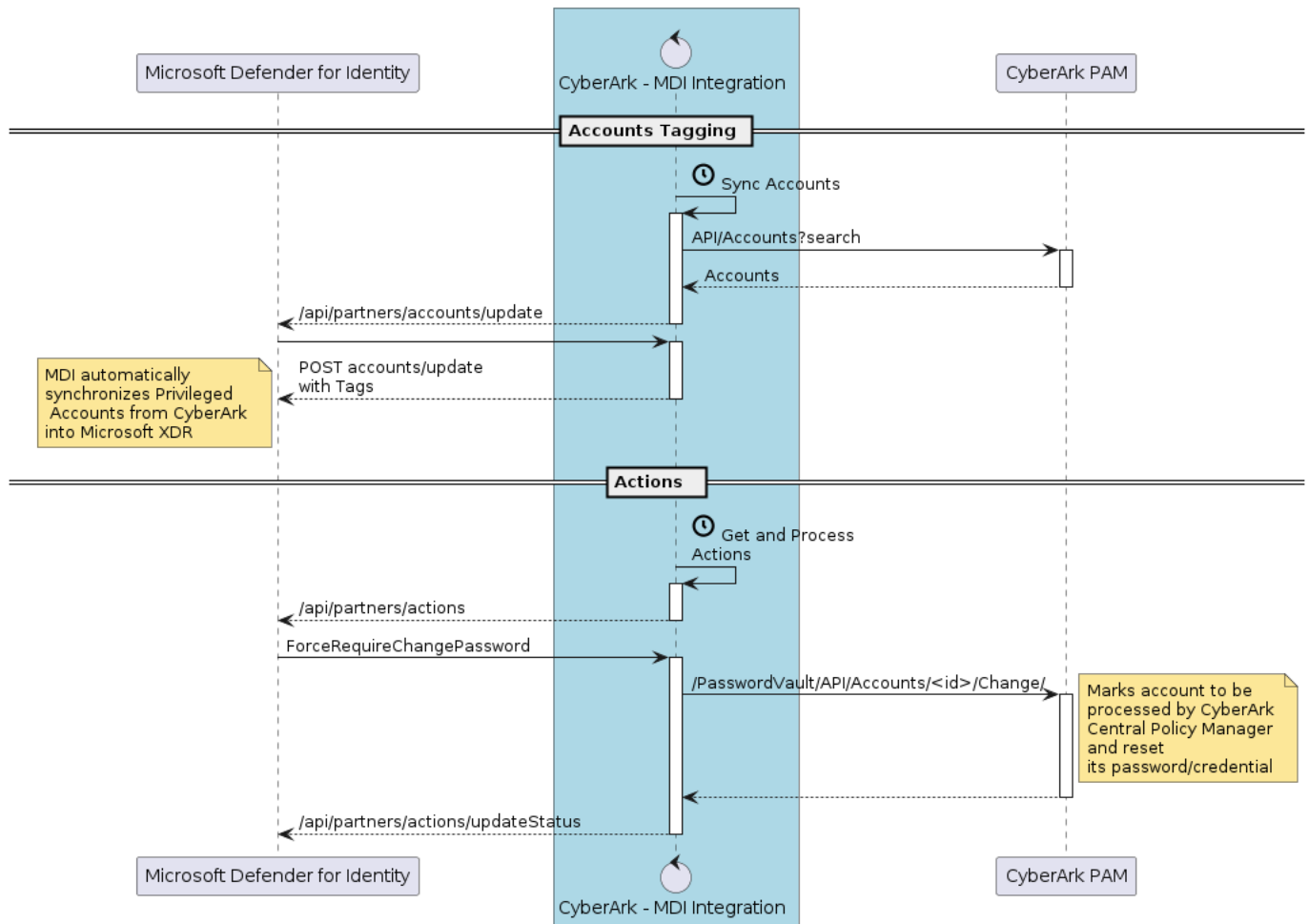
SOLUTION OVERVIEW

Microsoft Defender for Identity (MDI) is a cloud-based security solution that helps secure customers' identities monitoring across their organization. This modern identity threat detection (ITDR) is fully integrated with Microsoft Defender XDR, and leverages signals across hybrid environments, from both on-premises Active Directory and cloud identities, to help SecOp teams better identify, detect, and investigate advanced threats directed at their organization: breach prevention, threat detection, investigation of suspicious activities, and response to attacks.

The MDI team facilitates the security operations by enabling all identity security capabilities from a single pane of glass, onboarding sensors on all identity infrastructure-related servers in the customer's environment and in the cloud, including Microsoft first-party capabilities alongside other third-party solutions. By seamlessly integrating CyberArk Identity Security with Microsoft Defender for Identity, organizations can attain a unified, comprehensive view of the entire identity security landscape. This integration extends CyberArk's privilege controls into Microsoft's Extended Detection and Response (XDR) ecosystem, effectively bridging the existing gaps in data correlation. By connecting data from diverse systems, this integration empowers Security Operations Center (SOC) and Identity teams to enhance their investigative capabilities, streamline threat hunting efforts, facilitate more precise attack path identification, and, in turn, fortify the overall security posture.

The integrated solution delivers transformative benefits to organizations by enabling swift and proactive threat response. SOC analysts can efficiently investigate and contain potential threats, reducing the mean time to detection and response. Ultimately, the integration delivers a solution that enables businesses to proactively address security challenges, mitigate risks, and safeguard against potential financial losses, reputational damage, and legal ramifications.

1 INTEGRATION DESCRIPTION AND DIAGRAM



The integration runs as a Windows .Net 8 executable. Upon start, application will periodically perform accounts tagging by querying CyberArk and matching the information from Microsoft Defender for Identity (MDI) and the information found in CyberArk. It then posts an account update with tag “Privileged Account (CyberArk)”.

In addition, the integration periodically fetches actions from MDI and process those by executing the corresponding action (currently ForceRequiredChangePassword) on CyberArk to trigger CyberArk’s Central Policy Manager (CPM) to execute a password change on the target account.

1.1 Accounts Match

The match for the accounts to be updated in MDI is based on the following criteria:

1. The CyberArk Platform ID for the account in the configuration of the integration based on the parameter **CyberArkPlatformId** (single platform) or **CyberArkPlatformIds** (multiple platforms) on the **CyberArkClientOptions** section
2. The Account ID in MDI matches the value of “**UserName@Address**” where **UserName** and **Address** are the values stored for the account in CyberArk

2 REQUIREMENTS

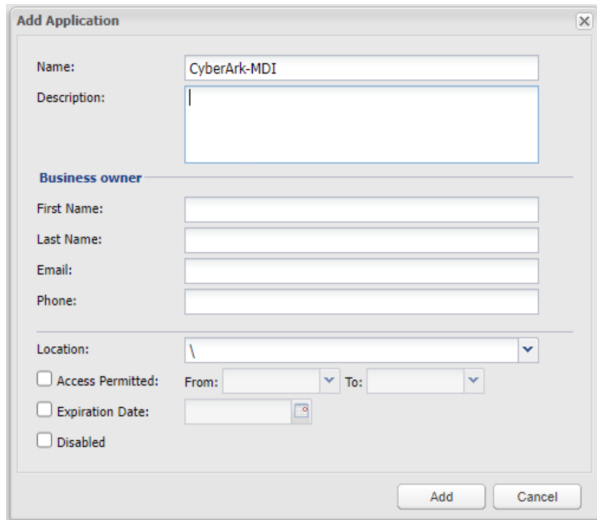
3. The integration is supported on CyberArk PAM Self-Hosted v12.0 to v15
- CyberArk Identity Security Platform's Privileged Access Management
 - Central Credential Provider (CCP)
 - PVWA REST API
 - Microsoft Defender for Identity (MDI)

3 CYBERARK CONFIGURATION

3.1 Define the application ID (app ID) and authentication details

The application definition is done manually through the CyberArk Password Vault Web Access (PVWA) / Privilege Cloud interface:

1. Log in with a user that has permissions to manage applications (it requires **Manage Users** authorization).
2. In the Applications tab, click **Add Application**. The Add Application page appears.



Add Application

Name: CyberArk-MDI

Description:

Business owner

First Name:

Last Name:

Email:

Phone:

Location: \

☐ Access Permitted: From: To:

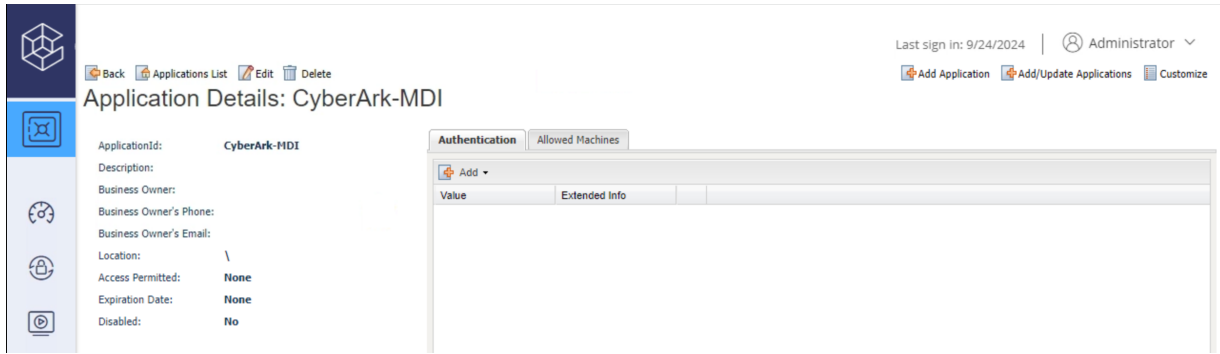
☐ Expiration Date:

☐ Disabled

Add Cancel

3. Specify the following information:
 - In the Name box, provide the unique name (ID) of the application. The suggested application ID for this integration is: CyberArk-MDI
 - In the **Description** box, provide a short description of the application that will help you identify it.
 - In the **Business owner** section, provide contact information about the application's business owner.
 - In the **Location** box, specify the location of the application in the Vault / Privilege Cloud hierarchy. If a location is not selected, the application will be added in the same location as the user who is creating this application.

4. Click **Add**. The application is added and will be displayed on the Application Details page.



Application Details: CyberArk-MDI

ApplicationId: CyberArk-MDI

Description:

Business Owner:

Business Owner's Phone:

Business Owner's Email:

Location:

Access Permitted: None

Expiration Date: None

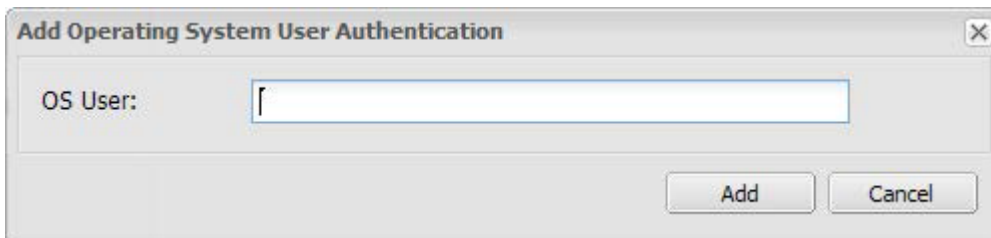
Disabled: No

Authentication Allowed Machines

Add

Value	Extended Info
-------	---------------

5. Select the **Allowing extended authentication restrictions** option. This enables you to specify an unlimited number of machines and Windows domain OS users for a single application.
6. Specify the application's **Authentication** details. This information enables the Credential Provider to check certain application characteristics before retrieving the application password.
 - a. In the Authentication tab, click **Add**. A drop-down list of authentication characteristics is displayed.
 - b. Select the authentication characteristic to specify.
 - c. Select **OS user**. The Add Operating System User Authentication window appears.

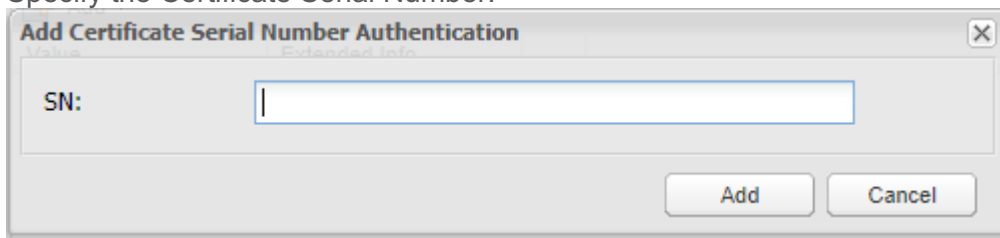


Add Operating System User Authentication

OS User:

Add Cancel

- d. Specify the name of the OS user who will run the application, then click **Add**. The OS user is listed in the Authentication tab.
7. Specify the Certificate Serial Number:

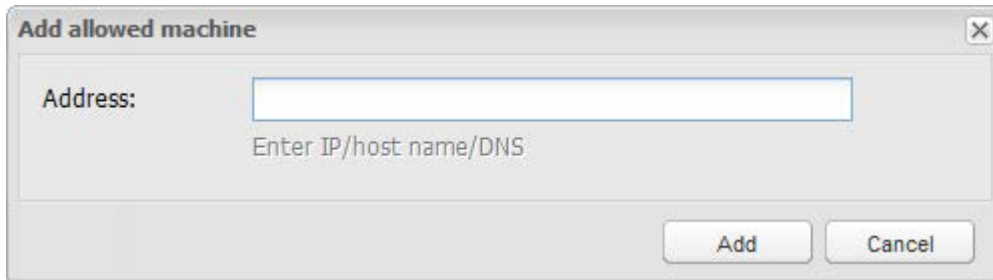


Add Certificate Serial Number Authentication

SN:

Add Cancel

8. Specify the application's Allowed Machines. This information enables Credential Provider to make sure that only applications that run from specified machines can access their passwords.
 - a. In the Allowed Machines tab, click **Add**. The **Add allowed machine** window is displayed.



- b. In the **Address** box, specify the IP/hostname/DNS of the machine where the application will run and will request passwords, then click **Add**. The IP address is listed in the Allowed Machines tab.

3.2 Set permissions for application access

To perform its tasks, the integration application requires access to either existing accounts or new accounts that need to be provisioned in CyberArk Vault / Privilege Cloud.

Provision, if needed, the privileged accounts that will be required by the application. You can do this in one of the following ways:

- **Manually** – Add accounts manually, one at a time, and specify all the account details
- **Automatically** – Add multiple accounts automatically using the Password Upload feature

For this step, you need **Add accounts** permissions in the safe(s) that they will be added to.

For more information about adding and managing privileged accounts, see the documentation for [PAM solution](#).

Once the accounts are managed by CyberArk, make sure to set up access to both the application and CyberArk Application Credential Providers serving the Application.

Add the CCP user and application users as members of the safes storing application passwords. Do this manually in the Safes tab or specify the Safe names in a CSV file to add multiple applications.

1. Add the CCP user as a Safe member with the following access:

- **List accounts**
- **Retrieve accounts**
- **View Safe Members**

Note: When installing multiple providers for this integration, create a group and add it to the Safe once with the appropriate authorization.

Add Safe Member

Search: Search In:

Selected Search: Vault

Name	Business Email	Full Name

☐ Access

☐ Use accounts

☒ Retrieve accounts

☒ List accounts

☐ Account Management

☐ Safe Management

☐ Monitor

☐ View Audit log

☒ View Safe Members

2. Add the application (the app ID) as a Safe member with the following access:
 - **Retrieve accounts**

Add Safe Member

Search:

Search In:

Vault

Search

Selected Search: Vault

Name	Business Email	Full Name
------	----------------	-----------

☐ Access

☐ Use accounts

☒ Retrieve accounts

☐ List accounts

☐ Account Management

☐ Safe Management

☐ Monitor

☐ View Audit log

☐ View Safe Members

Add

Close

3. If Safe is configured for object level access, make sure that both the CP user and the application have access to the passwords to retrieve.

4. For more information about configuring Safe members, see the documentation for [PAM solution](#).

3.3 Create a REST API user for CyberArk REST API Calls

The integration requires a user to authenticate to CyberArk and perform the different actions (get accounts and reset account credentials) via REST API calls. The Application (AppID) defined in step 5.1 needs to have access (as explained in step 5.2) to the safe where this account is stored; this account will be retrieved by the integration, to be used by the REST API calls.

1. Create the user in CyberArk,
2. Grant user permissions to list the accounts to be considered by the integration with the ability to perform CPM (Central Policy Manager) actions.

4 MICROSOFT DEFENDER FOR IDENTITY (MDI) CONFIGURATION

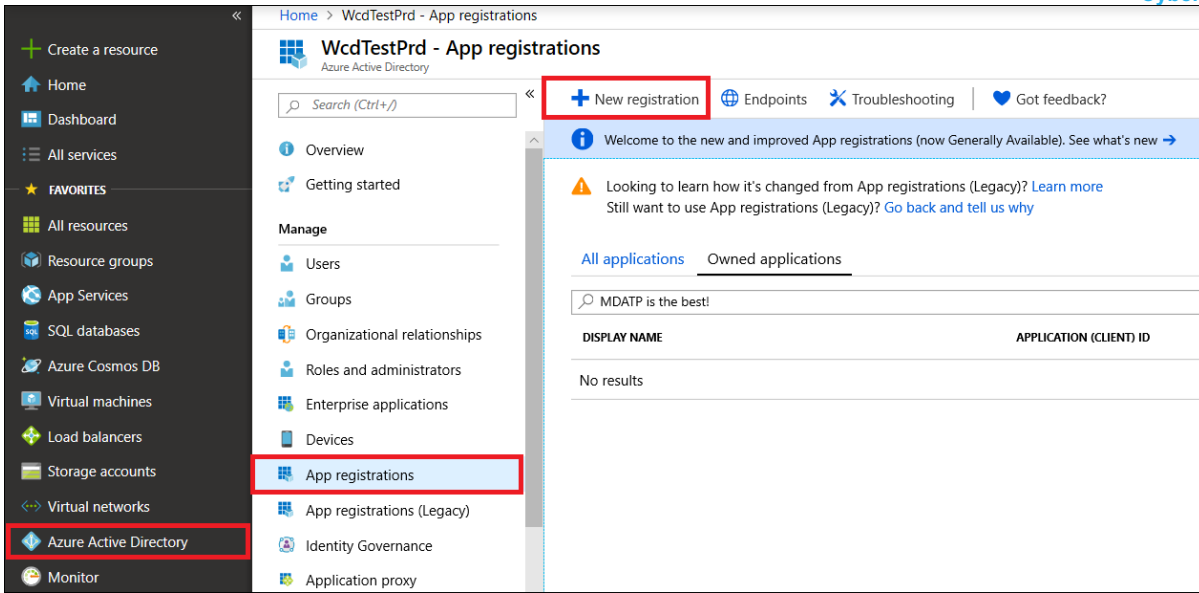
4.1 Allow MDI IP range

First, ensure that both MDI is deployed in your environment. Then, you will need a computer that can access both MDI externally and CyberArk internally. The following MDI IPs need to be allowed:

20.22.79.132
20.227.22.55
20.1.250.89
20.103.179.92
20.93.233.6
51.132.42.238
20.250.208.9
20.199.194.194
20.84.192.71
20.221.42.147
20.54.20.183
51.140.253.253
20.166.140.38
4.200.134.43

4.2 Create an Entra application to access the MDI without a user

Sign in to the [Azure portal and](#) navigate to **Microsoft Entra ID > App registrations > New registration**



Home > WcdTestPrd - App registrations

WcdTestPrd - App registrations
Azure Active Directory

Search (Ctrl+/)

+ New registration Endpoints Troubleshooting Got feedback?

Welcome to the new and improved App registrations (now Generally Available). See what's new →

Looking to learn how it's changed from App registrations (Legacy)? [Learn more](#)
Still want to use App registrations (Legacy)? [Go back and tell us why](#)

All applications Owned applications

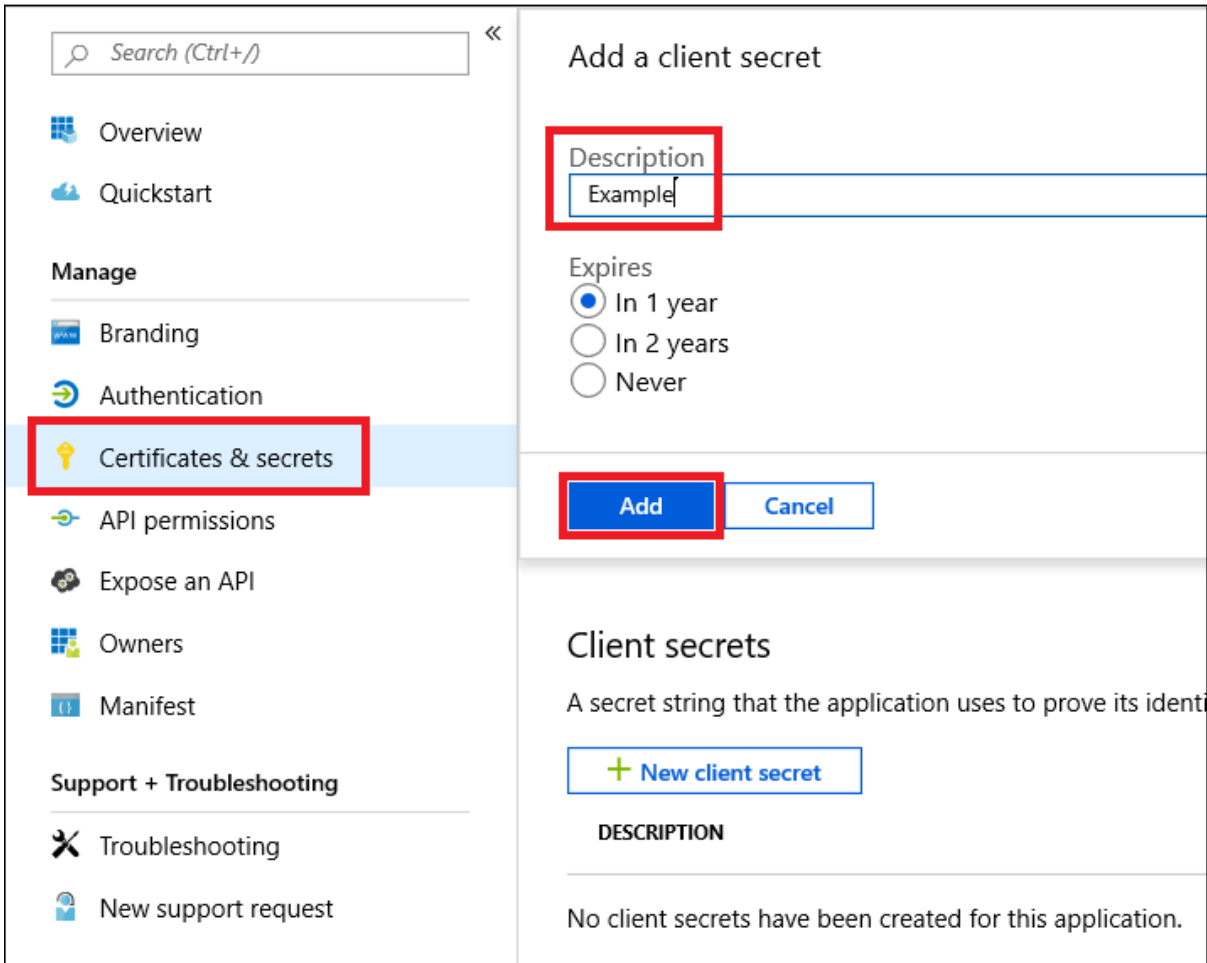
MDATP is the best!

DISPLAY NAME	APPLICATION (CLIENT) ID
No results	

Overview
Getting started
Manage
Users
Groups
Organizational relationships
Roles and administrators
Enterprise applications
Devices
App registrations
App registrations (Legacy)
Identity Governance
Application proxy

Create a resource
Home
Dashboard
All services
FAVORITES
All resources
Resource groups
App Services
SQL databases
Azure Cosmos DB
Virtual machines
Load balancers
Storage accounts
Virtual networks
Azure Active Directory
Monitor

In the registration form, choose a name for your application, and then select **Register**. No special API permissions are required currently. To add a secret to the application, select **Certificates & secrets**, add a description to the secret, and then select **Add**:



Search (Ctrl+/)

Overview
Quickstart

Manage
Branding
Authentication
Certificates & secrets
API permissions
Expose an API
Owners
Manifest

Support + Troubleshooting
Troubleshooting
New support request

Add a client secret

Description
Example

Expires
☒ In 1 year
☐ In 2 years
☐ Never

Add Cancel

Client secrets

A secret string that the application uses to prove its identity

+ New client secret

DESCRIPTION

No client secrets have been created for this application.

Then, select **API Permissions**, click **Add a permission**. In the **Request API Permissions** section, search for **Azure Advanced Threat Protection**, then **Application Permissions**. Select both **identity.PrivilegeAccountActions** and **identity.PrivilegeAccountTagging**.

Request API permissions

[← All APIs](#)

 Azure Advanced Threat Protection
https://atp.azure.com

What type of permissions does your application require?

Delegated permissions

Your application needs to access the API as the signed-in user.

Application permissions

Your application runs as a background service or daemon without a signed-in user.

Select permissions

[expand all](#)

Start typing a permission to filter these results	
Permission	Admin consent required
Identity (2)	
☒ Identity.PrivilegeAccountActions ⓘ get a list of actions and update on the status of said actions	Yes
☒ Identity.PrivilegeAccountTagging ⓘ Send which accounts are managed by the PAM solution	Yes

Write down your application ID and your tenant ID. On your application page, go to **Overview** and copy the following for configuration:

Home > WcdTestPrd - App registrations > YourAppName

YourAppName

Search (Ctrl+/)

Overview

Quickstart

Manage

- Branding
- Authentication
- Certificates & secrets
- API permissions

Delete Endpoints

Welcome to the new and improved App registrations. Looking to learn how it's

Display name : YourAppName

Application (client) ID : 94fbf312-ae3f-47f4-8eeb-d375d1299e8f

Directory (tenant) ID : f839b112-d9d7-4d27-9bf6-94542403f21c

Object ID : b687654e-064c-4514-b48c-3fd5aa009a1f

4.3 Setup Authentication for MDI API

The integration should be authenticated with AAD token with admin consent from tenant.

Follow this wiki (MDI Partners integration will be very similar to that of MDE Partners):

[Create an app to access Microsoft Defender for Endpoint without a user | Microsoft Learn](#)

The AAD app has been allowed by Microsoft before the APIs can be called, please exchange this information (App ID) with the Microsoft team.

The consent should be for the application:

Name: Azure Advanced Threat Protection

AppId: 7b7531ad-5926-4f2d-8a1d-38495ad33e17

5 CYBERARK CONNECTOR CONFIGURATION & EXECUTION

1. Download distribution and unzip it in a folder (referred as CONNECTOR_FOLDER) on the server that will host the connector
2. Configure the following parameters in the appsettings.json file:

Name	Description
MdiPartnerClientOptions Section	
MdiPartnersDiscoveryUrl	Default: https://prd.mdi.securitycenter.microsoft.com/api/partners/serviceinfo
AuthenticationOptions	
o AuthorizationServer	https://login.microsoftonline.com/
o ResourceName	7b7531ad-5926-4f2d-8a1d-38495ad33e17
o ClientId	Application (Client) Id associated to the application defined following the steps outlined in previous section
o TenantId	Directory (Tenant) Id associated to the application defined following the steps outlined in previous section
o CertificateSubjectName	This parameter is used to search for the SSL Certificate to be used for server and client authentication against CyberArk CCP. The value will be matched in the following order: • Certificate Distinguished Name • Subject Name • Thumbprint
o StoreName	Certificate Store Name where the SSL certificate is stored (in case of Certificate Authentication). Possible values are: AddressBook AuthRoot CertificateAuthority Disallowed My Root TrustedPeople TrustedPublisher
o StoreLocation	Location for SSL Certificate. Possible values are: CurrentUser or LocalMachine
CyberArkClientOptions Section	
CyberArkApiBaseUrl	The base URL for PVWA/CCP server, For example: https://comp01.cybr.com
CcpApplicationId	The Application ID defined in step 5.1 above. For example: CyberArk-MDI
CcpQueryForCyberArkApiCredential	CCP query to retrieve the credentials to be used by CyberArk REST API calls For example: safe=SAFENAME;object=INTERNALOBJECTNAME Note: if this parameter is not defined or is empty, the connector/integration will prompt for username and password to be used by CyberArk REST API
CcpQueryForMdiApplicationKey	CCP query to retrieve the MDI application key (defined in step 6.1 above) For example: safe=SAFENAME;object=Mdi_key Note: if this parameter is not defined or is empty, the connector/integration will prompt for the MDI application key

CyberArkPlatformId	The single CyberArk platform assigned to the accounts to be matched by the integration when correlating with the MDI accounts.
CyberArkPlatformIds	A set of CyberArk platforms assigned to the accounts to be matched by the integration when correlating with the MDI accounts. Please note that CyberArkPlatformIds overwrites any value stated in the CyberArkPlatformId (single platform.)
CertificateSubjectName	This parameter is used to search for the SSL Certificate to be used for server and client authentication against CyberArk CCP. The value will be matched in the following order: • Certificate Distinguished Name • Subject Name • Thumbprint
StoreName	Certificate Store Name where the SSL certificate is stored. Possible values are: AddressBook AuthRoot CertificateAuthority Disallowed My Root TrustedPeople TrustedPublisher
StoreLocation	Location for SSL Certificate. Possible values are: CurrentUser or LocalMachine
Serilog Section	
MinimumLevel	
• Default	Minimum level of logging, possible values: • Information • Debug

Note: The appsettings.json file is found in the distribution folder. You can configure an environment variable called “Hosting:Environment” to use a specific appsettings file named appsettings<VALUE of Hosting:Environment>.json.

3. Install the CyberArk MDI Sync Service from Windows PowerShell with the following command:

```
New-Service -Name "CyberArkMDISync" -BinaryPathName "CONNECTOR_FOLDER\CyberArkConnector.exe" -
DisplayName "CyberArk MDI Sync Service" -Description "Syncs CyberArk accounts to MDI." -StartupType
Automatic
```
4. Start the Service “CyberArk MDI Sync Service”.

6 TROUBLESHOOTING

Once the CyberArk MDI Sync Service is installed and running, any error will be reported to a file under logs in the CONNECTOR_FOLDER. In the following section you can find common errors and actions suggested to resolve the errors/issues.

6.1 Common Errors handling

Error/Message in Log	Suggested Action
Logs show “Failure in CcpQuery System.Net.HttpRequestException: No such host is known (hostname:port)”	Fix the value for CyberArkApiBaseUrl To test if CyberArk CCP point is up and running, call its WSDL page like this: <pre>https://<CCP_SERVER>/AIMWebService/V1.1/AIM.asmx?WSDL</pre> This would still require client cert authentication.



Logs show “Failure in CcpQuery System.Net.Http.HttpRequestException: Response status code does not indicate success: 403 (Forbidden)” And Audit Log for CCP shows : “ITATS982E User XXX is not defined. Diagnostic Info: 6	Fix the value for CcpApplicationId and/or make sure the CCP Application is defined in CyberArk
Logs show “Failure in CcpQuery System.Net.Http.HttpRequestException: Response status code does not indicate success: 404 (Not Found).” And Audit Log for CCP shows: “APPAP004E Password Object matching query [QUERY] was not found (Diagnostic Info: 5)”	Make sure the Application (CcpApplicationId) has access to the account to be retrieved, verify the query is correct for both CyberArk API Credential (CCPQueryForCyberArkApiCredential) and MDI Application Key (CcpQueryForMdiApplicationKey) To confirm CyberArk CCP’s object (either CyberArk API Credential or MDI Application Key) is retrieved properly, use curl or postman to test: <pre>curl -v --cert cert.p12 --cert-type p12 --pass <CERT PASSWORD> https://<CYBERARK_CCP_URL>/AIMWebService/api/Accounts?appid=<CyberArk-MDI>&safe=<SAFENAME>&object=<OBJECTNAME></pre>
- Successfully got accounts from CyberArk - SyncCyberArkAccountsAsync - page index 1 - Current page count: 0, NextLink exists: False And no accounts are tagged.	Make sure the CyberArkPlatformId parameter corresponds to the CyberArk Platform for the accounts. Also verify the Application has access to the safes where such accounts exist. Refer to section 5.2 above, and make sure the CyberArk-MDI AppID user has access to the safe.
Certificate with subject name SUBJECTNAME not found.	Make sure the information for CertificateSubjectName, StoreName, and/or StoreLocation (CyberArkClientOptions section) is correct.
SyncCyberArkAccountsAsync Failed. System.Net.Http.HttpRequestException: No such host is known. (HOSTNAME:PORT)	Verify and correct the value for MdiPartnersDiscoveryUrl
SyncCyberArkAccountsAsync Failed. MSAL.NetCore.4.61.3.0.MsalServiceException: ErrorCode: invalid_instance Microsoft.Identity.Client.MsalServiceException: AADSTS50049: Unknown or invalid instance.	Verify and correct the value for AuthorizationServer under MdiPartnerClientOptions -> AuthenticationOptions
StatusCode: 400 ResponseBody: {"error": "invalid_resource", "error_description": "AADSTS500011: The resource principal named RESOURCENAME was not found in the tenant named TENANTNAME. This can happen if the application has not been installed by the administrator of the tenant or consented to by any user in the tenant. You might have sent your authentication request to the wrong tenant.	Verify and correct the value for ResourceName under MdiPartnerClientOptions -> AuthenticationOptions
StatusCode: 400 ResponseBody: {"error": "unauthorized_client", "error_description": "AADSTS700016: Application with identifier 'XXX' was not found in the directory 'DIRECTORY'. This can happen if the application has not been installed by the administrator of the tenant or consented to by any user in the tenant. You may have sent your authentication request to the wrong tenant.	Verify and correct the value for ClientId under MdiPartnerClientOptions -> AuthenticationOptions
StatusCode: 400 ResponseBody: {"error": "invalid_request", "error_description": "AADSTS90002: Tenant 'TENANTID' not found. Check to make sure you have the correct tenant ID and are signing into the correct cloud. Check with your subscription administrator, this may happen if there are no active subscriptions for the tenant.	Verify and correct the value for TenantId under MdiPartnerClientOptions -> AuthenticationOptions

7 PARTNER CONTACT INFO

Business Contact	Name	Caitlin Methe
	Email	Caitlin.Methe@cyberark.com
Technical Contact	Name	Business Development Tech Alliances
	Email	bizdevtech@cyberark.com
Support Contact	Name	Business Development Tech Alliances
	Email	bizdevtech@cyberark.com