

About the connector

CyberArk helps you to manage all the privileged accounts within your organization with automatic password management, access control, dual control, video recordings and numerous features.

This document provides information about the CyberArk connector, which facilitates automated interactions with CyberArk using FortiSOAR™ playbooks. Add the CyberArk connector as a step in FortiSOAR™ playbooks and perform automated operations, such as getting all the account groups in a specific safe from CyberArk, or adding a new user to a Vault in CyberArk.

This connector can also be used to configure other connectors using credentials that are stored in the CyberArk vault.

FortiSOAR™ integrates with CyberArk vault to allow users to securely store their sensitive data and credentials. Configure the Password Vault Manager in FortiSOAR™ to allow users to use the credentials stored in CyberArk in the connector configurations. For more information, see the FortiSOAR™ product documentation, i.e., the Security Management chapter in the "Administration Guide."

Version information

Connector Version: 2.0.0

FortiSOAR™ Version Tested on: 5.1.0-464

Authored By: Fortinet

Certified: Yes

Release Notes for version 2.0.0

Following enhancements have been made to the *CyberArk* connector in version 2.0.0:

- Added the Application ID and Safe Name configuration parameters.
- Updated the input parameters for the following operations:
 - Add Account Group
 - Activate User
 - Delete User
 - Add Safe
 - Update Safe
 - Add Safe Member

- Update Safe Member
- Removed the following operations and playbooks:
 - Add Policy/ACL
 - List Policy/ACL
 - Delete Policy/ACL

Installing the connector

From FortiSOAR™ 5.0.0 onwards, use the **Connector Store** to install the connector.

You can also use the yum command to install connectors. Connectors provided by FortiSOAR™ are delivered using a FortiSOAR™ repository. Therefore, you must set up your FortiSOAR™ repository and run the yum command as a root user to install connectors:

```
yum install FortiSOAR-connector-cyberark
```

Prerequisites to configuring the connector

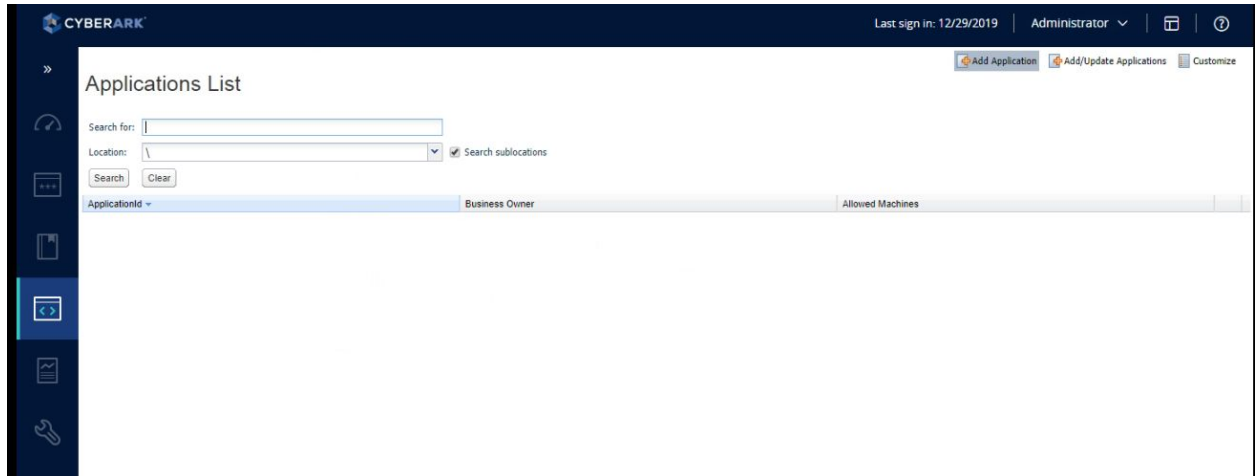
- You must have the URL of CyberArk server to which you will connect and perform automated operations and credentials (username-password pair) to access that server.
- You must also have the application ID that has been issued to you by CyberArk that is used for the password retrieval process and the name of the Safe that stores the credentials, including passwords. See the "Creating an application in CyberArk" section for the procedure on how to create an application in CyberArk.
- To access the FortiSOAR™ UI, ensure that port 443 is open through the firewall for the FortiSOAR™ instance.

Creating an application in CyberArk

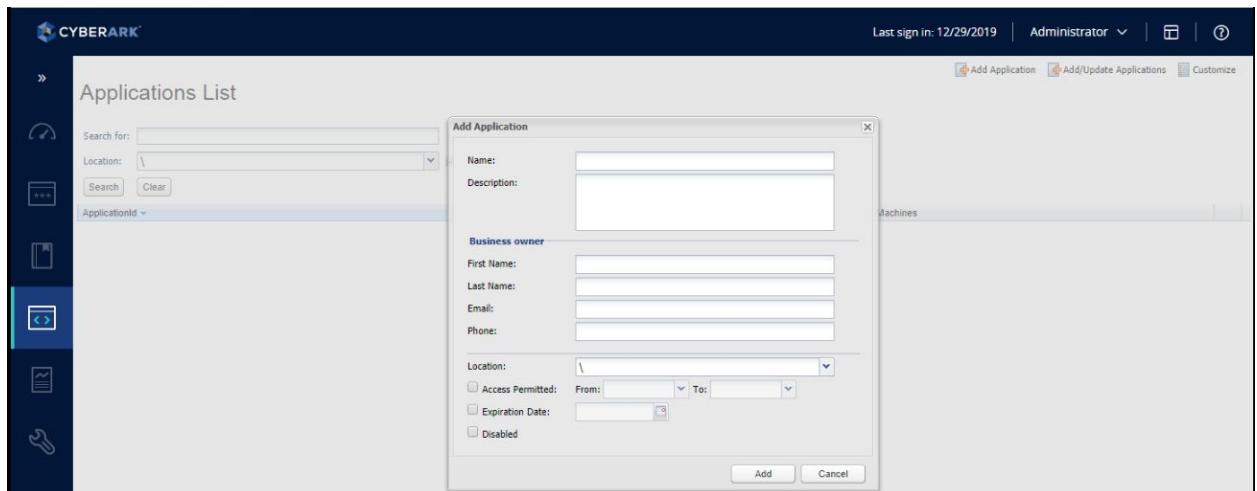
To configure your CyberArk Connector you must have the application ID that has been issued to you by CyberArk that is used for the password retrieval process and the name of the Safe that stores the credentials, including passwords. Use the following procedure to retrieve your application ID:

1. Log onto the CyberArk Portal using your credentials.
2. On the left menu, click **Applications**.

You can search for applications on the Applications List page and also retrieve the application ID for your application from this page. You will require the Application ID when you are configuring the CyberArk connector.

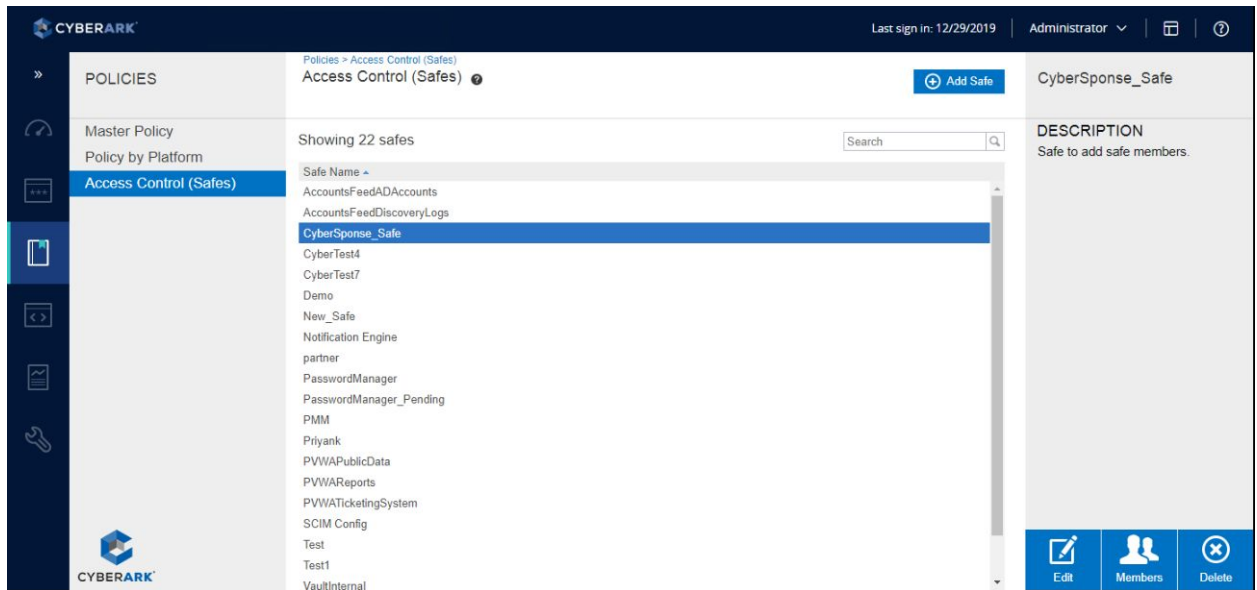


3. Click the **Add Application** button, which displays the Add Application form.



Enter the details required in the Add Application form and click **Add**.

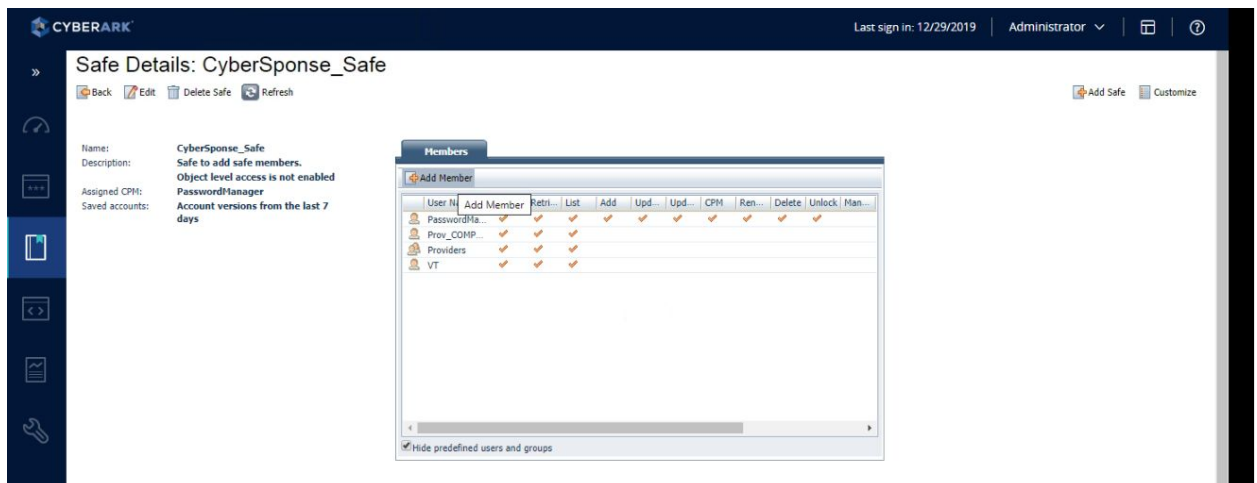
4. Click **Policies** in the left menu and in the Policies menu, click **Access Control (Safes)**.



You can either add new safe by clicking the **Add Safe** button or, you can edit the details of an existing safe.

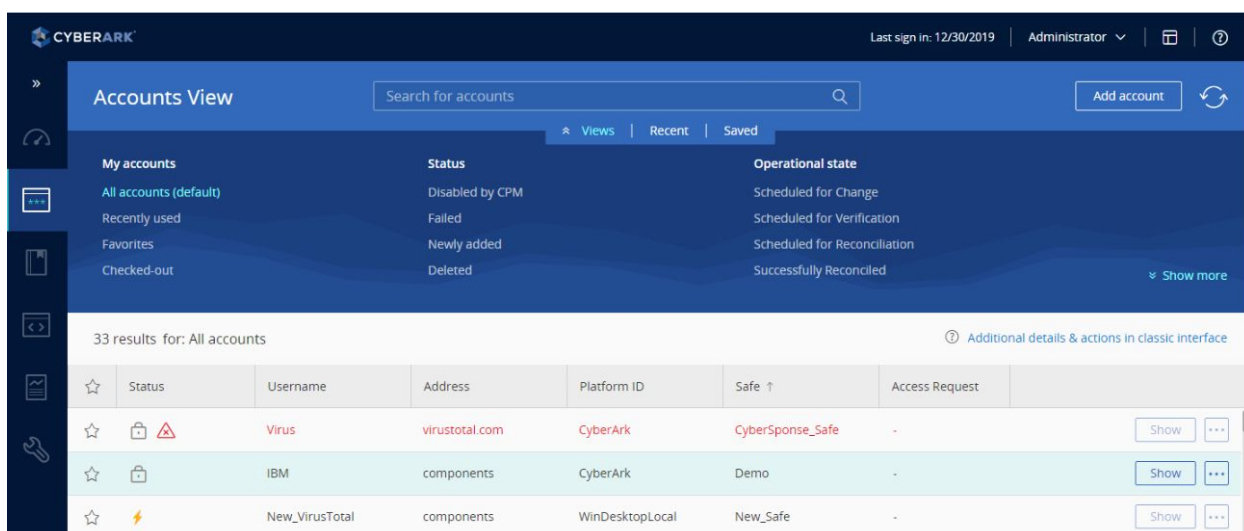
If you are adding a new safe then fill in the details of the new safe in the Add Safe form and then edit the members of the newly-created safe to add the application that you had created in *Step 3*.

If you want to edit the details of an existing safe, then select the safe that you want to edit as shown in the above image and then click **Members**.

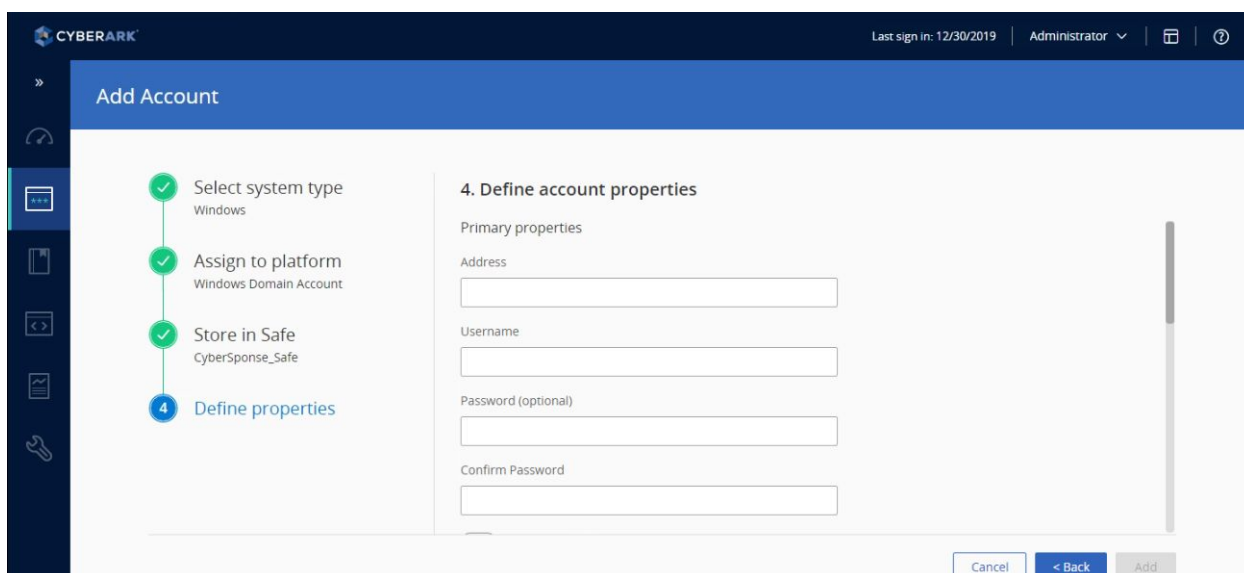


Click **Add Member** to add the application that you had created in *Step 3*.

- Click **Accounts** in the left menu and click the **Add Account** button.



Complete the **Add Account** wizard that includes selecting the system type and platform for the account. In the **Store in Safe** step, ensure that you select the same safe that you had selected or created in *Step 4*, and then define the properties of the account and click **Add**.



Configuring the connector

You need to configure the CyberArk connector using the Password Vault Manager. For more information, see the FortiSOAR™ product documentation, i.e., the *Security Management* chapter in the "Administration Guide."

Configuration parameters

You need to configure the CyberArk connector using the "Password Vault Manager" if you have appropriate permissions. You can open the "Password Vault Manager" by clicking the **Settings** icon on the top-right corner in FortiSOAR™, and then click **Password Vault** in the Security Management section. You can also open the Password Vault from the Connectors page, by selecting the **CyberArk** connector row (if you are in the **Grid** view on the Connectors page), and clicking the **Password Vault Manager** link in the **Configurations** tab. This opens the Password Vault page, where you can select **CyberArk** from the **Select Vault Manager** drop-down list and enter the required configuration details.

Parameter	Description
Server URL	URL of the CyberArk server to which you will connect and perform automated operations.
Username	Username used to access the CyberArk server to which you will connect and perform the automated operations.
Password	Password used to access the CyberArk server to which you will connect and perform the automated operations.
Application ID	Application ID that has been issued to you by CyberArk, which is used for the password retrieval process. See the "Creating an application in CyberArk" section for the procedure on how to create an application in CyberArk.
Safe Name	Name of the Safe that stores the credentials, including passwords.
Verify SSL	Specifies whether the SSL certificate for the server is to be verified or not. By default, this option is set as True.

Actions supported by the connector

The following automated operations can be included in playbooks, and you can also use the annotations to access operations from FortiSOAR™ release 4.10.0 and onwards:

Function	Description	Annotation and Category
----------	-------------	-------------------------

Add Account Group	Adds a new account group to the vault based on the account ID and group ID you have specified.	add_account_group Miscellaneous
Get Account Group by Safe	Retrieves all account groups associated with a specific safe from CyberArk based on the safe ID you have specified.	get_account_group_info Investigation
Get Account Group Members	Retrieves all the members of an existing account group from CyberArk based on the group ID you have specified.	get_account_group_info Investigation
Delete Member from Account Group	Removes a member from an account group in CyberArk based on the account ID and group ID you have specified.	delete_account_group_members Investigation
Add User	Adds a new user to the vault based on the username, initial password, email address. and other input parameters you have specified.	add_user Miscellaneous
Add User to Group	Adds a specific user to an existing user group in the vault based on the username and group name you have specified.	update_group Miscellaneous
Logged on User Details	Retrieves the user information of the user who is logged on to CyberArk.	user_details Investigation
Get User Details	Retrieves information for a specific user in the vault based on the user ID you have specified.	user_details Investigation
Update User	Updates an existing user in the vault based on the username, initial password, email address. and other input parameters you have specified.	update_user Miscellaneous

Activate User	Activates a specified vault user based on the user ID you have specified.	activate_user Investigation
Delete User	Deletes a specified vault user based on the user ID you have specified.	delete_user Miscellaneous
Add Safe	Adds new safe to the vault based on the safe name and other input parameters you have specified.	add_safe Miscellaneous
List Safes	Retrieves information for all of the user's safes in the Vault.	list_safes Investigation
Get Safe Details	Retrieves details about a specific safe in the vault based on the safe name you have specified.	safe_details Investigation
Search Safe	Retrieves information about the safes in the vault based on the criteria mentioned in the search query you have specified.	safe_details Investigation
Get Safe Account Groups	Retrieves all the existing account groups that are associated with a specific safe in the vault based on the safe name you have specified.	get_safe_account_groups Investigation
Update Safe	Updates an existing safe in the vault based on the safe name and other input parameters you have specified.	update_safe Miscellaneous
Delete Safe	Deletes a specified safe from the vault based on the safe name you have specified.	delete_safe Miscellaneous
Add Safe Member	Adds an existing user as a safe member in the vault based on the safe name, member name, and other input parameters you have specified.	add_safe_member Miscellaneous
List Safe Members	Retrieves a list of members of the specified safe from the vault, based on the safe name you have specified.	list_safe_members Investigation

Update Safe Member	Updates an existing safe member in the vault based on the safe name, member name, and other input parameters you have specified.	update_safe_member Investigation
Delete Safe Member	Removes a specific member from a specific safe based on the safe name and member name you have specified.	delete_safe_member Investigation

operation: Add Account Group

Input parameters

Parameter	Description
Account ID	ID of the account that you want to add to the specified group in the vault.
Group ID	Group ID in which you want to add the specified account in the vault.

Output

The output contains the following populated JSON schema:

```
{
  "AccountId": ""
}
```

operation: Get Account Group by Safe

Input parameters

Parameter	Description
Safe	Name of the safe whose associated account groups you want to retrieve from CyberArk.

Output

The output contains the following populated JSON schema:

```
{
  "Safe": "",
  "GroupName": "",
  "GroupPlatformID": "",
  "GroupID": ""
}
```

operation: Get Account Group Members

Input parameters

Parameter	Description
Group ID	ID of the group whose members you want to retrieve from CyberArk.

Output

The output contains the following populated JSON schema:

```
{
  "Address": "",
  "SafeName": "",
  "PlatformID": "",
  "UserName": "",
  "AccountID": ""
}
```

operation: Delete Member from Account Group

Input parameters

Parameter	Description
Group ID	ID of the Group in CyberArk from which you want to delete a specific member.
Account	ID of the account that you want to delete from the specific group

ID	in CyberArk.
----	--------------

Output

The output contains the following populated JSON schema:

```
{
  "message": ""
}
```

operation: Add User

Input parameters

Parameter	Description
Username	Username that you want to add to the vault.
First Name	First name of the user that you want to add to the vault.
Last Name	Last name of the user that you want to add to the vault.
Initial Password	Initial password of the user that you want to add to the vault.
Email	Email address of the user that you want to add to the vault.
User Type Name	Name of the user type of the user that you want to add to the vault. For example, EPVUser.
Expiry Date	Date on which the user account that you want to add to the vault will expire.
Disabled	Select this option, i.e., set it to true (default) if you want the ability to disable the newly created user.
Change Password On The Next Logon	Select this option, i.e., set it to true, if you want the newly added user to change the initial password on their next logon. By default, this is set as false.

Output

The output contains the following populated JSON schema:

```
{
  "changePassOnNextLogon": "",
  "authenticationMethod": [],
  "componentUser": "",
  "suspended": "",
  "id": "",
  "vaultAuthorization": [],
  "personalDetails": {
    "profession": "",
    "organization": "",
    "state": "",
    "title": "",
    "country": "",
    "lastName": "",
    "department": "",
    "middleName": "",
    "city": "",
    "firstName": "",
    "street": "",
    "zip": ""
  },
  "enableUser": "",
  "source": "",
  "internet": {
    "homePage": "",
    "otherEmail": "",
    "businessEmail": "",
    "homeEmail": ""
  },
  "expiryDate": "",
  "distinguishedName": "",
  "username": "",
```

```
"businessAddress": {
  "workZip": "",
  "workCountry": "",
  "workCity": "",
  "workStreet": "",
  "workState": ""
},
"passwordNeverExpires": "",
"location": "",
"userType": "",
"phones": {
  "pagerNumber": "",
  "homeNumber": "",
  "cellularNumber": "",
  "businessNumber": "",
  "faxNumber": ""
},
"description": "",
"unAuthorizedInterfaces": []
}
```

operation: Add User to Group

Input parameters

Parameter	Description
Username	Username that you want to add to a specified existing group in the vault.
Group Name	Name of the group to which you want to add the specified user.

Output

The output contains a non-dictionary value.

operation: Logged on User Details

Input parameters

None.

Output

The output contains the following populated JSON schema:

```
{
  "Users": [
    {
      "vaultAuthorization": [],
      "componentUser": "",
      "username": "",
      "id": "",
      "userType": "",
      "personalDetails": {
        "firstName": "",
        "lastName": "",
        "middleName": ""
      },
      "location": "",
      "source": ""
    }
  ]
}
```

operation: Get User Details

Input parameters

Parameter	Description
User ID	ID of the user whose details you want to retrieve from CyberArk.

Output

The output contains the following populated JSON schema:

```
{
  "changePassOnNextLogon": "",
  "authenticationMethod": [],
  "componentUser": "",
  "suspended": "",
  "id": "",
  "vaultAuthorization": [],
  "personalDetails": {
    "profession": "",
    "organization": "",
    "state": "",
    "title": "",
    "country": "",
    "lastName": "",
    "department": "",
    "middleName": "",
    "city": "",
    "firstName": "",
    "street": "",
    "zip": ""
  },
  "enableUser": "",
  "source": "",
  "internet": {
    "homePage": "",
    "otherEmail": "",
    "businessEmail": "",
    "homeEmail": ""
  },
}
```

```
"expiryDate": "",
"distinguishedName": "",
"username": "",
"businessAddress": {
  "workZip": "",
  "workCountry": "",
  "workCity": "",
  "workStreet": "",
  "workState": ""
},
"passwordNeverExpires": "",
"location": "",
"userType": "",
"phones": {
  "pagerNumber": "",
  "homeNumber": "",
  "cellularNumber": "",
  "businessNumber": "",
  "faxNumber": ""
},
"description": "",
"unAuthorizedInterfaces": []
}
```

operation: Update User

Input parameters

Parameter	Description
Username	Username of the user that you want to update in the vault.
First Name	First name of the user that you want to update in the vault.

Last Name	Last name of the user that you want to update in the vault.
New Password	New password that you want to set for the user that you want to update in the vault.
Email	Email address of the user that you want to update in the vault.
User Type Name	Name of the user type of the user that you want to update in the vault. For example, EPVUser.
Expiry Date	Date on which the user account that you want to update in the vault will expire.
Disabled	Select this option, i.e., set it to true (default) if you want the ability to disable the updated user.
Change Password On The Next Logon	Select this option, i.e., set it to true, if you want the updated user to change the password on their next logon. By default, this is set as false.

Output

The output contains a non-dictionary value.

operation: Activate User

Input parameters

Parameter	Description
User ID	ID of the user who you want to activate in the vault.

Output

The output contains the following populated JSON schema:

```
{
  "message": ""
}
```

operation: Delete User

Input parameters

Parameter	Description
Username	ID of the user that you want to delete from the vault.

Output

The output contains a non-dictionary value.

operation: Add Safe

Input parameters

Parameter	Description
Safe Name	Name of the safe that you want to add to the vault.
Description	(Optional) Description of the safe that you want to add to the Vault.
Managing CPM	(Optional) Name of the CPM that will manage the safe that you want to add to Vault. For example, PasswordManager.
Number of Versions Retention	(Optional) Number of versions of the safe you want to retain in the vault.
Number of Days Retention	(Optional) Number of days you want to retain the safe in the vault.
OLAC Enabled	Select this option, i.e., set it to true to enable Object Level Access Control (OLAC).

Output

The output contains the following populated JSON schema:

```
{
```

```
"Location": "",
"SafeName": "",
"OLACEnabled": "",
"NumberOfDaysRetention": "",
"Description": "",
"AutoPurgeEnabled": "",
"NumberOfVersionsRetention": "",
"ManagingCPM": ""
}
```

operation: List Safes

Input parameters

None.

Output

The output contains the following populated JSON schema:

```
{
  "Safes": [
    {
      "SafeName": "",
      "Location": "",
      "SafeUrlId": "",
      "Description": ""
    }
  ]
}
```

operation: Get Safe Details

Input parameters

Parameter	Description
-----------	-------------

Safe Name	Name of the safe for which you want to retrieve the details from the vault.
-----------	---

Output

The output contains the following populated JSON schema:

```
{
  "Location": "",
  "SafeName": "",
  "OLACEnabled": "",
  "NumberOfDaysRetention": "",
  "Description": "",
  "AutoPurgeEnabled": "",
  "NumberOfVersionsRetention": "",
  "ManagingCPM": ""
}
```

operation: Search Safe

Input parameters

Parameter	Description
Search Query	Query using which you want to retrieve details about safes from the vault.

Output

The output contains the following populated JSON schema:

```
{
  "Safes": [
    {
      "SafeName": "",
      "Location": "",
      "SafeUrlId": "",
      "Description": ""
    }
  ]
}
```

```
]
}
```

operation: Get Safe Account Groups

Input parameters

Parameter	Description
Safe Name	Name of the safe whose associated account group details you want to retrieve from the vault.

Output

The output contains the following populated JSON schema:

```
{
  "Safe": "",
  "GroupName": "",
  "GroupPlatformID": "",
  "GroupID": ""
}
```

operation: Update Safe

Input parameters

Parameter	Description
Safe Name	Name of the safe that you want to update in the vault.
Description	Description of the safe that you want to update in the vault.
OLAC Enabled	Select this option, i.e., set it to true to enable Object Level Access Control (OLAC).
Number of Versions Retention	Number of versions of the safe you want to retain in the vault.

Managing CPM	Name of the CPM that will manage the safe that you want to update in the vault. For example, PasswordManager.
Location	Location of the safe that you want to update in the vault.
Number of Days Retention	Number of days you want to retain the safe in the vault.

Output

The output contains the following populated JSON schema:

```
{
  "Location": "",
  "SafeName": "",
  "OLACEnabled": "",
  "NumberOfDaysRetention": "",
  "Description": "",
  "AutoPurgeEnabled": "",
  "NumberOfVersionsRetention": "",
  "ManagingCPM": ""
}
```

operation: Delete Safe

Input parameters

Parameter	Description
Safe Name	Name of the safe that you want to delete from the vault.

Output

The output contains the following populated JSON schema:

```
{
  "message": ""
}
```

}

operation: Add Safe Member

Input parameters

Parameter	Description
Safe Name	Name of the safe in which you want to add a specific member as a safe member.
Member Name	Name of the member who you want to add as a safe member to the specific safe.
IsExpired Membership Enable	Select this option to assign permission to the safe member that you are adding to enable the expiration of safe members' membership. Note: This and the following parameters define the type of permission that the administrator wants to assign to the user that you want to add as a safe member to the specific safe in the vault.
Use Accounts	Select this option to assign permission to the safe member that you are adding to add the safe members who can use this account but not able to view the passwords. This is applicable to the safe member.
Retrieve Accounts	Select this option to assign permission to the safe member that you are adding to have the ability to view and retrieve accounts associated with the user that you are adding as a safe member in the vault.
List Accounts	Select this option to assign permission to the safe member that you are adding to allow the safe member to view account lists.
Add Accounts	Select this option to assign permission to the safe member that you are adding to add accounts in the safe. Users who are given Add Accounts authorization receive Update Account Properties as well. Users who are assigned this permission are automatically assigned the permission to Update Account Properties. Therefore, when Add Accounts=True then Update Account Properties should also be True.
Update Account Content	Select this option to assign permission to the safe member that you are adding to update the account content of a safe member.

Update Account Properties	Select this option to assign permission to the safe member that you are adding to update the existing account properties of safe members. When you enable the Add Accounts permission, the Update Account Properties permission is automatically enabled.
Initiate CPM Account Management Operations	Select this option to assign permission to the safe member that you are adding to initiate password management operations through CPM, such as changing passwords, verifying and reconciling passwords. When this parameter is cleared, i.e., set to false, then the Specify Next Account Content parameter is automatically set to false.
Specify Next Account Content	Select this option to assign permission to the safe member that you are adding to specify the password that will be used when the CPM changes the password value. This parameter can only be specified when Initiate CPM Account Management Operations is set to true. If you clear the Initiate CPM Account Management Operations parameter, i.e., set to false, then this parameter is automatically set to false.
Rename Accounts	Select this option to assign permission to the safe member that you are adding to rename existing accounts in the safe.
Delete Accounts	Select this option to assign permission to the safe member that you are adding to delete existing accounts from the safe.
Unlock Accounts	Select this option to assign permission to the safe member that you are adding to unlock accounts that are locked by other users.
Manage Safe	Select this option to assign permission to the safe member that you are adding to perform administrative tasks of the safe, such as, updating the safe properties, recovering the safe, deleting the safe, etc.
Manage Safe Members	Select this option to assign permission to the safe member that you are adding to add and remove safe members, and update their authorizations in the safe.
Backup Safe	Select this option to assign permission to the safe member that you are adding to create a backup of a safe and its contents, and store the contents in another location of the safe.
View Audit Log	Select this option to assign permission to the safe member that you are adding to view account and user activity in the safe.

View Safe Members	Select this option to assign permission to the safe member that you are adding to view account permissions of the safe members.
Access Without Confirmation	Select this option to assign permission to the safe member that you are adding to access the safe without confirmation from other authorized users. This overrides the "Safe Properties" that specify that safe members require confirmation to access the safe.
Create Folders	Select this option to assign permission to the safe member that you are adding to create folders in the safe.
Delete Folders	Select this option to assign permission to the safe member that you are adding to delete folders from the safe.
Move Accounts And Folders	Select this option to assign permission to the safe member that you are adding to move accounts and folders of the safe to different folders and subfolders.
Requests Authorization Level1	Select this option to assign permission to the safe member that you are adding to access the safe without confirmation from other authorized users. This overrides the "Safe Properties" that specify that safe members require confirmation to access the safe.
Requests Authorization Level2	Select this option to assign permission to the safe member that you are adding to access the safe without confirmation from other authorized users. This overrides the "Safe Properties" that specify that safe members require confirmation to access the safe.

Output

The output contains the following populated JSON schema:

```
{
  "MemberType": "",
  "Permissions": {
    "AccessWithoutConfirmation": "",
    "UnlockAccounts": "",
    "RequestsAuthorizationLevel2": "",
    "CreateFolders": "",
    "DeleteAccounts": "",
    "UpdateAccountProperties": "",
```

```

    "MoveAccountsAndFolders": "",
    "RequestsAuthorizationLevel": "",
    "RenameAccounts": "",
    "RetrieveAccounts": "",
    "DeleteFolders": "",
    "SpecifyNextAccountContent": "",
    "InitiateCPMAccountManagementOperations": "",
    "ManageSafeMembers": "",
    "ListAccounts": "",
    "ManageSafe": "",
    "UseAccounts": "",
    "UpdateAccountContent": "",
    "BackupSafe": "",
    "ViewAuditLog": "",
    "AddAccounts": "",
    "ViewSafeMembers": ""
  },
  "MemberName": "",
  "IsPredefinedUser": "",
  "IsExpiredMembershipEnable": "",
  "MembershipExpirationDate": ""
}

```

operation: List Safe Members

Input parameters

Parameter	Description
Safe Name	Name of the safe whose safe members you want to retrieve from the vault.

Output

The output contains the following populated JSON schema:

```
{
  "SafeMembers": [
    {
      "MemberType": "",
      "Permissions": {
        "AccessWithoutConfirmation": "",
        "UnlockAccounts": "",
        "RequestsAuthorizationLevel2": "",
        "CreateFolders": "",
        "DeleteAccounts": "",
        "UpdateAccountProperties": "",
        "MoveAccountsAndFolders": "",
        "RequestsAuthorizationLevel1": "",
        "RenameAccounts": "",
        "RetrieveAccounts": "",
        "DeleteFolders": "",
        "SpecifyNextAccountContent": "",
        "InitiateCPMAccountManagementOperations": "",
        "ManageSafeMembers": "",
        "ListAccounts": "",
        "ManageSafe": "",
        "UseAccounts": "",
        "UpdateAccountContent": "",
        "BackupSafe": "",
        "ViewAuditLog": "",
        "AddAccounts": "",
        "ViewSafeMembers": ""
      },
      "MemberName": "",
      "IsPredefinedUser": "",
      "IsExpiredMembershipEnable": "",
      "MembershipExpirationDate": ""
    }
  ]
}
```

```
}  
]  
}
```

operation: Update Safe Member

Input parameters

Parameter	Description
Safe Name	Name of the safe in which you want to update a specific member as a safe member.
Member Name	Name of the member who you want to update as a safe member to the specific safe.
IsExpired Membership Enable	Select this option to assign permission to the safe member that you are updating to enable the expiration of safe members' membership. Note: This and the following parameters define the type of permission that the administrator wants to assign to the user that you want to add as a safe member to the specific safe in the vault.
Use Accounts	Select this option to assign permission to the safe member that you are updating to add the safe members to use this account but not able to view the passwords. This is applicable to the safe member.
Retrieve Accounts	Select this option to assign permission to the safe member that you are updating to have the ability to view and retrieve accounts associated with the user that you are adding as a safe member in the vault.
List Accounts	Select this option to assign permission to the safe member that you are updating to allow the safe member to view account lists.
Add Accounts	Select this option to assign permission to the safe member that you are updating to add accounts in the safe. Users who are given Add Accounts authorization receive Update Account Properties as well. Users who are assigned this permission are automatically assigned the permission to Update Account Properties. Therefore, when Add Accounts=True then Update Account Properties should also be True.

Update Account Content	Select this option to assign permission to the safe member that you are updating to update the account content of a safe member.
Update Account Properties	Select this option to assign permission to the safe member that you are updating to update the existing account properties of safe members. When you enable the Add Accounts permission, the Update Account Properties permission is automatically enabled.
Initiate CPM Account Management Operations	Select this option to assign permission to the safe member that you are updating to initiate password management operations through CPM, such as changing passwords, verifying and reconciling passwords. When this parameter is cleared, i.e., set to false, then the Specify Next Account Content parameter is automatically set to false.
Specify Next Account Content	Select this option to assign permission to the safe member that you are updating to specify the password that will be used when the CPM changes the password value. This parameter can only be specified when Initiate CPM Account Management Operations is set to true. If you clear the Initiate CPM Account Management Operations parameter, i.e., set to false, then this parameter is automatically set to false.
Rename Accounts	Select this option to assign permission to the safe member that you are updating to rename existing accounts in the safe.
Delete Accounts	Select this option to assign permission to the safe member that you are updating to delete existing accounts from the safe.
Unlock Accounts	Select this option to assign permission to the safe member that you are updating to unlock accounts that are locked by other users.
Manage Safe	Select this option to assign permission to the safe member that you are updating to perform administrative tasks of the safe, such as, updating the safe properties, recovering the safe, deleting the safe, etc.
Manage Safe Members	Select this option to assign permission to the safe member that you are updating to add and remove safe members, and update their authorizations in the safe.

Backup Safe	Select this option to assign permission to the safe member that you are updating to create a backup of a safe and its contents, and store the contents in another location of the safe.
View Audit Log	Select this option to assign permission to the safe member that you are updating to view account and user activity in the safe.
View Safe Members	Select this option to assign permission to the safe member that you are updating to view account permissions of the safe members.
Access Without Confirmation	Select this option to assign permission to the safe member that you are updating to access the safe without confirmation from other authorized users. This overrides the "Safe Properties" that specify that safe members require confirmation to access the safe.
Create Folders	Select this option to assign permission to the safe member that you are updating to create folders in the safe.
Delete Folders	Select this option to assign permission to the safe member that you are updating to delete folders from the safe.
Move Accounts And Folders	Select this option to assign permission to the safe member that you are updating to move accounts and folders of the safe to different folders and subfolders.
Requests Authorization Level1	Select this option to assign permission to the safe member that you are updating to access the safe without confirmation from other authorized users. This overrides the "Safe Properties" that specify that safe members require confirmation to access the safe.
Requests Authorization Level2	Select this option to assign permission to the safe member to access the safe without confirmation from other authorized users. This overrides the "Safe Properties" that specify that safe members require confirmation to access the safe.

Output

The output contains the following populated JSON schema:

```
{
  "MemberType": "",
```

```

"Permissions": {
  "AccessWithoutConfirmation": "",
  "UnlockAccounts": "",
  "RequestsAuthorizationLevel2": "",
  "CreateFolders": "",
  "DeleteAccounts": "",
  "UpdateAccountProperties": "",
  "MoveAccountsAndFolders": "",
  "RequestsAuthorizationLevel1": "",
  "RenameAccounts": "",
  "RetrieveAccounts": "",
  "DeleteFolders": "",
  "SpecifyNextAccountContent": "",
  "InitiateCPMAccountManagementOperations": "",
  "ManageSafeMembers": "",
  "ListAccounts": "",
  "ManageSafe": "",
  "UseAccounts": "",
  "UpdateAccountContent": "",
  "BackupSafe": "",
  "ViewAuditLog": "",
  "AddAccounts": "",
  "ViewSafeMembers": ""
},
"MemberName": "",
"IsPredefinedUser": "",
"IsExpiredMembershipEnable": "",
"MembershipExpirationDate": ""
}

```

operation: Delete Safe Member

Input parameters

Parameter	Description
Safe Name	Name of the safe from which you want to delete the specified safe member.
Member Name	Name of the member that you want to delete from the specified safe.

Output

The output contains the following populated JSON schema:

```
{  
  "message": ""  
}
```

Included playbooks

The *Sample - CyberArk - 2.0.0* playbook collection comes bundled with the CyberArk connector. These playbooks contain steps using which you can perform all supported actions. You can see bundled playbooks in the **Automation > Playbooks** section in FortiSOAR™ after importing the CyberArk connector.

- Activate User
- Add Account Group
- Add Safe
- Add Safe Member
- Add User
- Add User to Group
- Delete Member from Account Group
- Delete Safe
- Delete Safe Member
- Delete User
- Get Account Group by Safe
- Get Account Group Members
- Get Safe Account Groups
- Get Safe Details
- Get User Details
- List Safe Members

- List Safes
- Logged on User Details
- Search Safe
- Update Safe
- Update Safe Member
- Update User

Note: If you are planning to use any of the sample playbooks in your environment, ensure that you clone those playbooks and move them to a different collection since the sample playbook collection gets deleted during connector upgrade and delete.