



Device Authority Keyscaler IOT Devices Plug-in for the Central Policy Manager

The Cyber-Ark Password Vault
Version 12.x

Copyright © 2024 Cyber-Ark Software Ltd. All rights reserved.

This document contains information and ideas, which are proprietary to Cyber-Ark Software Ltd. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, without the prior written permission of Cyber-Ark Software Ltd.

EPVEMC07-2-0-1



Changing Passwords on Device Authority Keyscaler IOT Devices

The Central Password Manager enables organizations to change passwords automatically on remote machines and store the new passwords in the Enterprise Password Vault, with no human intervention, according to the organizational policy. This plug-in supports the changing of passwords for users on the Device Authority Keyscaler IOT Devices using the REST API protocol.

Installation

You will receive the Device Authority Keyscaler IOT Devices zip archive from your CyberArk representative or downloaded via the CyberArk Marketplace. The following instructions show you how to install it.

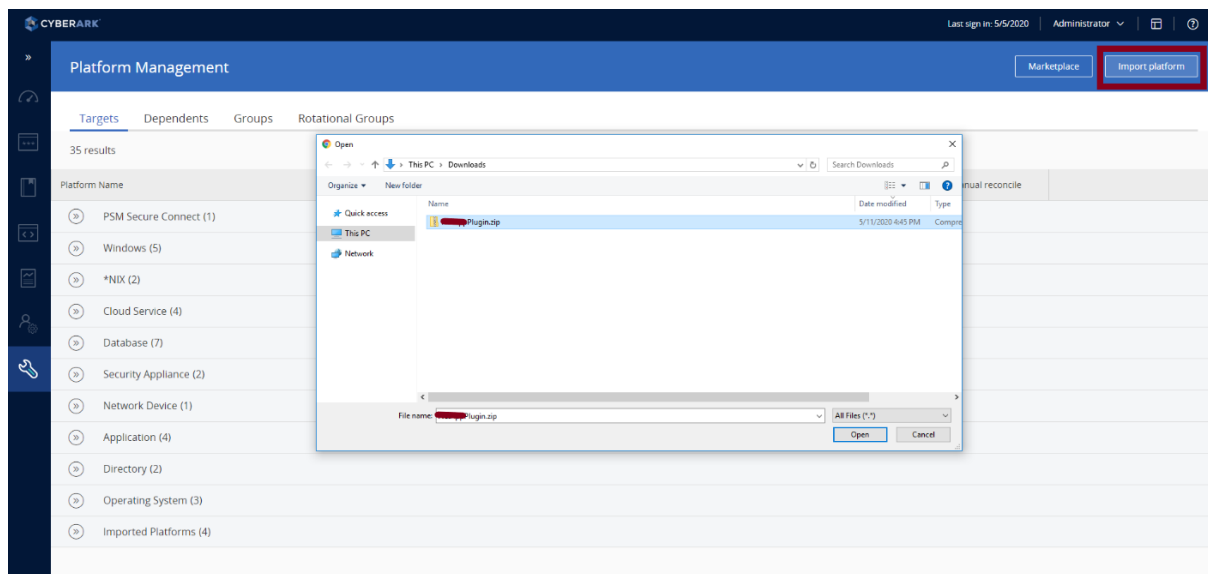
Prerequisites

- The plug-in uses KeyScaler's API and requires software provided by Device Authority.
- Download the SAT API Application Version 1.0.2.4.1 (Signed) installer: <https://support.deviceauthority.com/hc/en-us/categories/29389205189005-CyberArk-Integration>
- Follow the installation instructions provided by Device Authority: <https://support.deviceauthority.com/hc/en-us/categories/29389205189005-CyberArk-Integration>
- The SAT API Application Version 1.0.2.4.1 can run as a stand-alone application, and should be installed and tested independently before attempting installation of the CyberArk CPM Plug-in

Installing the Device Authority Keyscaler IOT Devices Plug-in

1. Log in to the PVWA with an administrative account.
2. Go to Administration > Platform Management.
3. Click Import Platform. Upload the DAKeyscalerIOTDevicesPlugin.zip file.





Platform Configuration

- Please check that the following default platform settings match your environment.
- They can be modified by navigating to Administration → Platform Management → DAKeyScalerIoTDevices → Edit → Automatic Password Management → Additional Policy Settings → Parameters

Parameter Name	ksConfigFilePath
Description	Path to KeyScaler Secure Asset Transfer configuration file.
Acceptable Values	String
Default Value	C:\Program Files\DeviceAuthority\KeyScaler Secure Asset Transfer\config.json

Parameter Name	KsSecureAssetTransferPath
Description	Path to KeyScaler Secure Asset Transfer executable
Acceptable Values	String
Default Value	C:\Program Files\DeviceAuthority\KeyScaler Secure Asset Transfer\dist\KeyScaler_SecureAssetTransfer.exe

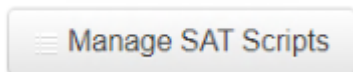
Parameter Name	changepassScriptName
Description	KeyScaler Secure Asset Transfer script name used during changepass action
Acceptable Values	String
Default Value	<Change Me>

Parameter Name	verifypassScriptName
Description	KeyScaler Secure Asset Transfer script name used during verifypass action
Acceptable Values	String
Default Value	<Change Me>

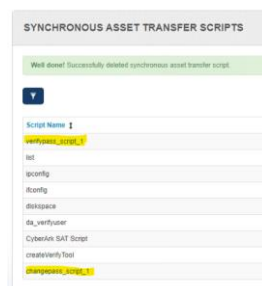


SAT Script names can be located by logging into the KeyScaler Web UI and navigating to:

Manage Policies > SYNCHRONOUS
ASSET TRANSFER POLICIES >
Manage SAT Scripts



A list of SAT script names will be displayed:



Parameter Name	KsPlaceholderNames
Description	List of KeyScaler Secure Asset Transfer script placeholder names Note: The list of placeholder names used by the changepass and verifypass scripts will always contain the following placeholder names and should not be included in the KsPlaceholderNames list parameter: “user” “password” <i>verificationuser</i> is a placeholder used by the example verifypass script
Acceptable Values	List of placeholder names, space delimited
Default Value	verificationuser

Parameter Name	KsPlaceholderValues
Description	List of KeyScaler Secure Asset Transfer script placeholder values
Acceptable Values	List of placeholder values, space delimited
Default Value	<Change Me>



The list of KeyScaler Secure Asset Transfer script placeholder names can be determined by viewing or editing the verifypass and changepass scripts.

Script Name ↑	Script Description	Actions
verifypass_script_1	verifypass_script_1	View Edit Delete
list	basic director listing and echo user name	View Edit Delete
ipconfig	Get IP info	View Edit Delete
ifconfig	get the IP	View Edit Delete
diskspace	diskspace	View Edit Delete
da_verifyuser	verify user on linux boxes - in dev	View Edit Delete
CyberArk SAT Script	CyberArk Echo test	View Edit Delete
createVerifyTool	createPasswordVerifyTool	View Edit Delete
changepass_script_1	changepass_script_1	View Edit Delete

[New Script](#)

[Manage SAT Policies](#)

View SAT Script

Script Name changepass_script_1

Script Description changepass_script_1

Script Platform linux

Script Placeholders ['password', 'user']

Script

```
#!/bin/bash echo "{\"message\": \"verifying user existence ${user}\"}" # Check that the user
exists if ! id -u ${user} > /dev/null 2>&1 then echo "{\"message\": \"user ${user} not found\"}"
exit fi # Change password echo "${password}\n${password}" | passwd ${user} echo "
{Password change successful}" exit 0
```

View SAT Script

Script Name verifypass_script_1

Script Description verifypass_script_1

Script Platform linux

Script Placeholders ['user', 'password', 'verificationuser']

Script

```
#!/bin/bash set -eu # Used for testing #user=test1 #verificationuser=test2 #password=test2
echo "{\"message\": \"verifying user ${user}\"}" # Adding a check to see if expect is in place.
if ! type "expect" > /dev/null then echo "{\"message\": \"expect not found\"}" echo "
{\"success\": false}" exit fi # Check that the user exists if ! id -u ${user} > /dev/null 2>&1 then
echo "{\"message\": \"user ${user} not found\"}" echo "{\"success\": false}" exit fi # Check
that the verification user exists if ! id -u ${verificationuser} > /dev/null 2>&1 then echo "
{\"message\": \"user ${verificationuser} not found\"}" echo "{\"success\": false}" exit fi su
${verificationuser} -c "expect -c \"log_user 0; spawn su - ${user} -c true expect { Password: {
send -- \"${password}\n\" exp_continue } failure
(https://tenant.mykeyscaler.com/cpladmin/scripts/sat/edit.cancel/1 log_user 1 send_user \"
{\"message\": \"password not correct\"}\n\" send_user \"{\"success\": false}\n\" exit }
timeout { log_user 1 send_user \"{\"message\": \"timeout\"}\n\" send_user \"{\"success\":
false}\n\" exit } passwd { log_user 1 send_user \"{\"message\": \"password not set\"}\n\"
send_user \"{\"success\": false}\n\" exit } eof { log_user 1 send_user \"{\"success\":
true}\n\" send_user \"{\"Password is correct\"}\n\" exit } }"
```

From the above examples, the only Script Placeholder used within both the changepass and verifypass SAT scripts that is not “user” or “password” is *verificationuser*



CYBERARK®

Device Authority KeyScaler IOT Devices via REST API Plug-in

© Cyber-Ark Software Ltd. | <http://www.CyberArk.com>

Adding DAKeyscalerIoTDevices Accounts

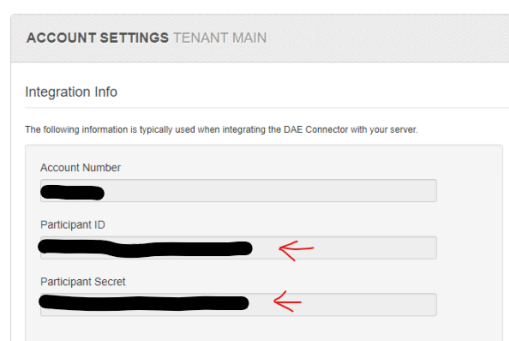
Adding Accounts through the Password Vault Web Access

- Create a new DAKeyscalerIoTDevices account or use an existing one.
- Specify the required account properties:

Property Name	Description
Username	Target IOT device username to manage
IOT Device Name (Address)	Target IOT device name

- Associate a reconcile account for the new password object.

The reconcile account object will store the API Participant ID (username) and Participant Secret (password).



ACCOUNT SETTINGS TENANT MAIN

Integration Info

The following information is typically used when integrating the DAE Connector with your server.

Account Number

Participant ID

Participant Secret

Note: The Participant ID and Participant Secret cannot be managed by CyberArk. It is recommended to use the ["Dummy Plugin"](#) available on the CyberArk Marketplace as the platform for this account object.

For more information about adding password objects in the Password Vault Web Access, refer to the online documentation [here](#).

General Notes

DAKeyscalerIoTDevices plug-in supports the following:

- Password change (as reconciliation)
- Password verification
- Password reconciliation

