BulkCPMManage Implementation Guide

Ver 1.1

31 Oct 2023

Timothy A. Sawyer, Global Support Operations, CyberArk Ltd.

Table of Contents

Introduction

About BulkCPMManage

This script is designed to 'bulk' manage the status of CPM management for multiple accounts in a vault. As opposed to the current UI which only allows the edit of this property sequentially, this allows the end user to perform this in bulk, based on searches in an active platform or a safe that is managed by a CPM.

Development and Version Compatibility

Development

BulkCPMManage was developed in Python 3.12. This has been tested on Mac (Ventura 13.5), Windows 11 and Windows Server 2019.

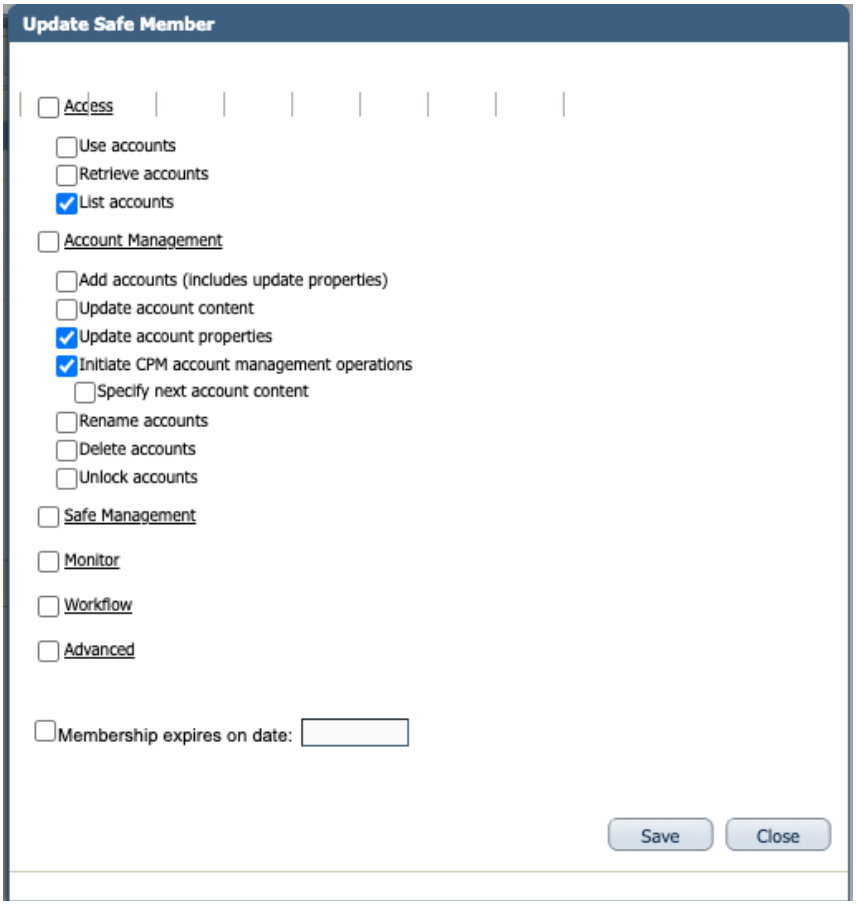
Version Compatibility

This has been tested with Privileged Authentication Manager (PAM) v13.2 (Self-Hosted). It has not been tested with Privilege Cloud.

Pre-Requisites and Installation

Pre-Requisites

- On the server that will be running BulkCPMManage, Python must be installed
- On the vault, a user must be created to make the appropriate API calls.
    - The user must be a vault user as the script uses CyberArk authentication
    - The vault user must have the following permissions on all safes where affected accounts reside:

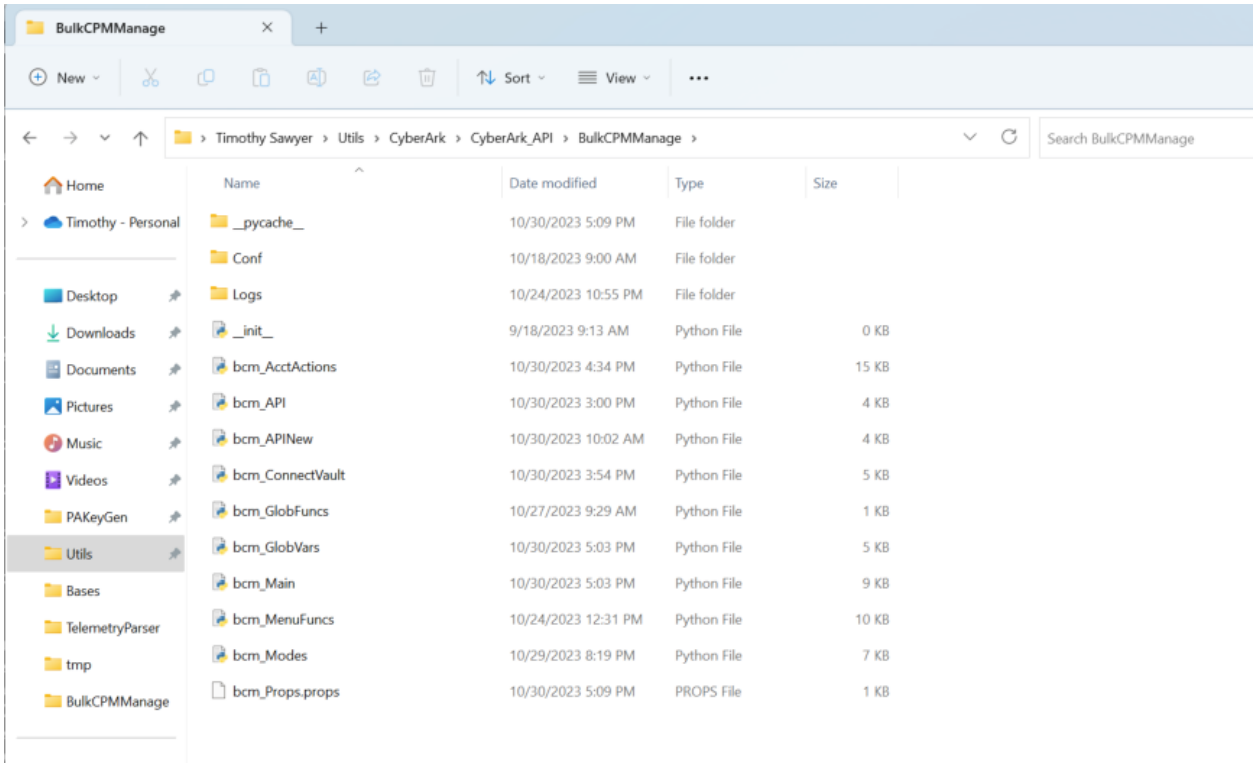


- Verify the following information for the Password Vault Web Access (PVWA) server
  - Hostname (FQDN)
  - PVWA URL
- Certificates
  - A PEM file containing the root CA as well as all intermediate CA must be configured and stored in a directory.
    1. Export the root CA and all intermediate CA into DER encoded certificates.
    2. For each certificate, convert it into a PEM format with the following command
       openssl x509 -inform der -in certificate.cer -out certificate.pem
       - certificate.cer is the path to the DER certificate
       - certificate.pem is the path to the new PEM certificate
       - Perform this step on all certificates in the CA chain
    3. Open each certificate with a text editor. In another new text file, copy the PEM from each certificate to the new text file. Start with the root CA and then all intermediate certificates
    4. Save the file with the PEM extension. It may be necessary to remove other extensions such as .txt

Installation

The script will be packaged as a compressed (.zip) file.

1. Create a top-level directory
2. In this directory, create a directory named “CyberArk_API”
3. Unzip the zip file to the folder created in step 2. This underlying folder should be named “BulkCPMManage”



4. Take the bcm_Props.props file and move it to the top-level directory created in step 1.

Properties File

The properties file (bcm_Props.props) contains necessary information to run BulkCPMManage. It is divided into four separate sections.

- General Properties
  - The main property is Debug mode. Unless there are some issues with a particular execution, this should usually be set to “False”.
  - Note: Check the size of the bcm_Main.py.log regularly, especially if debug is being used as this can fill up.
- Installation and Logs Folder
  - s_BaseDir is the location to where you uncompressed the zip file. Example:

```
###
# Installation and Logs folder
###
s_BaseDir = "/Volumes/Data/Programs/CyberArk/Marketplace/CyberArk_API/BulkCPMManage"
s_LogDir = "/Logs/"
```

  - s_LogDir is the location where logs are stored
  - Note: If this will be run on Windows, backslashes must be used. See the top of the bcm_Props.props file and below. Note that backslashes ('\') must be negated with an additional backslash:

```
###
# Installation and Logs folder
###
s_BaseDir = "C:\\Users\\tims1\\Utils\\CyberArk\\CyberArk_API\\BulkCPMManage"
s_LogDir = "\\Logs\\"
```

- PVWA Information
  - Change the host, context, protocol, and port as necessary to reflect the current environment. Other than the host, the context, protocol, and port is valid for a majority of PAM implementations
- Certificate Information
  - If the PVWA is configured to run over https, then certificate validation is required.
  - Make sure a directory is created to store certificates. The path to the CA PEM file (see Pre-Requisites) is noted in the s_CAPEM parameter in bcm_Props.props.

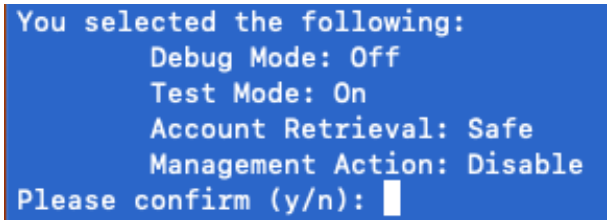
```
# Path to certificate
s_Cert = "C:\\Users\\tims1\\Utils\\CyberArk\\certs\\lab-comp.pem"
# Path to CA certificate (All CA must be bundled into a PEM format)
s_CAPEM = "C:\\Users\\tims1\\Utils\\CyberArk\\certs\\cachain3.pem"
```

- API Information
  - This section should not have to be modified

Running BulkCPMManage

1. At a command prompt, type the following command:
   '*python -m CyberArk_API.BulkCPMManage.bcm_Main.py*'
2. Accept the disclaimer terms by pressing 'y'. Note that if any key other than 'Y' or 'y' is pressed, the program will terminate.
3. Select 'Test Mode'. Test mode allows you to go through the process of selecting a safe or platform and selecting accounts to be processed without changing the database. Any changes (regardless of test mode) are logged to bcm_Main.py_Changes.log (located in the same directory as the log folder).

4. Select Platform or Safe. Accounts to be modified will be selected from either a platform or a safe.
    a. Note: Only active platforms and safes that are managed by a CPM will be selected. If a platform is inactive or a safe is not managed by a CPM, it will be excluded from any searches.
5. Select Disable or Enable CPM management.
    a. Disable will disable CPM management for that account
    b. Enable will enable CPM management for that account
    c. Note: Depending on the selected action:
        i. If “Disable” is selected, only those accounts where CPM management is enabled will be available
        ii. If “Enable” is selected, only those accounts where CPM management is disabled will be available
6. Confirm your choices



Note that pressing any key other than ‘Y’ or ‘y’ will end the program

7. PVWA FQDN: The value of the PVWA FQDN is based on the parameter s_PVWAFQDN in bcm_Props.props.

If this is correct, press ‘Enter’. If it is not correct and press ‘x’ to stop the program. At this point, re-check the PVWA FQDN in bcm_Props.props.

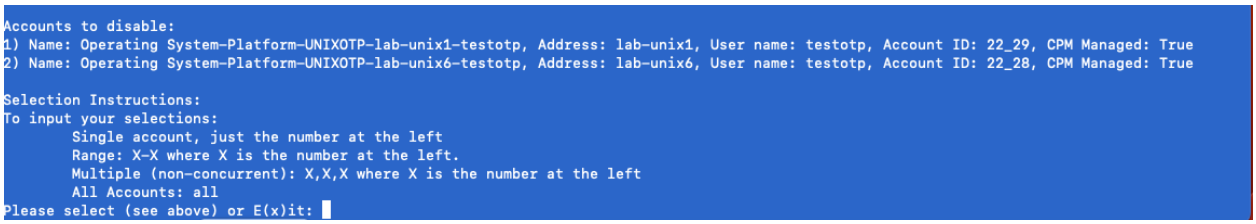
8. PVWA URL: This is asking for confirmation of the PVWA URL. If it is correct, type ‘y’. Type any other key if it is not correct, and check values in the PVWA Information section of bcm_Props.props.

9. Vault User credentials: Enter the username and then password for the vault user making the API calls. Ensure this user has the appropriate permissions as stated in Pre-Requisites

10. Depending on the selection of platforms or safes, a menu will be shown with the available choices:

Select one of the choices or ‘x’ can be selected to exit the program.

11. Another menu will be the list of eligible accounts



Note the selection instructions. One or multiple accounts can be selected.

- Range: For a range of accounts, select the item numbers at left (Example: 1-2)
- For multiple accounts, select the item(s), separated by commas (Example: 1,2)
- For all items in the list, type ‘all’
- For one item, type its item number (Example: 2)
- Exit (‘x’) can be selected to end the program

12. There will now be a prompt to process the actions for the selected accounts

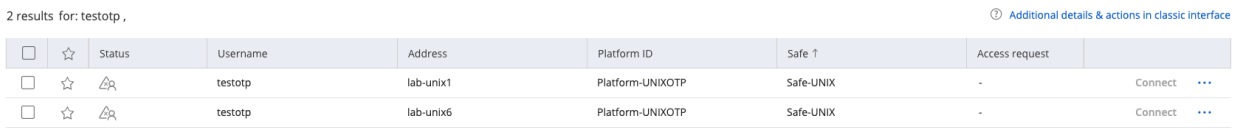
```
Accounts to be processed:
1) Name: Operating System-Platform-UNIXOTP-lab-unix1-testotp, Address: lab-unix1, ID: 22_29, Action: Disable
2) Name: Operating System-Platform-UNIXOTP-lab-unix6-testotp, Address: lab-unix6, ID: 22_28, Action: Disable

Disabling selected accounts:
1): Name: Operating System-Platform-UNIXOTP-lab-unix1-testotp, Address: lab-unix1, User name: testotp
2): Name: Operating System-Platform-UNIXOTP-lab-unix6-testotp, Address: lab-unix6, User name: testotp
Proceed? (y/n):
```

Type ‘y’ to process the accounts or any other key to end the program.

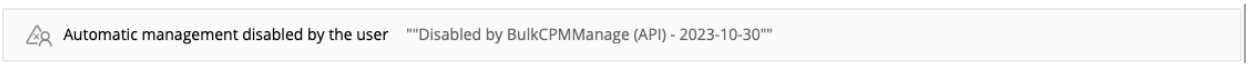
13. If test mode was not selected, check the accounts via the PVWA to see that the action was taken

```
Accounts to be processed:
1) Name: Operating System-Platform-UNIXOTP-lab-unix1-testotp, Address: lab-unix1, ID: 22_29, Action: Disable
2) Name: Operating System-Platform-UNIXOTP-lab-unix6-testotp, Address: lab-unix6, ID: 22_28, Action: Disable

Disabling selected accounts:
1): Name: Operating System-Platform-UNIXOTP-lab-unix1-testotp, Address: lab-unix1, User name: testotp
2): Name: Operating System-Platform-UNIXOTP-lab-unix6-testotp, Address: lab-unix6, User name: testotp
Proceed? (y/n): y
Account Operating System-Platform-UNIXOTP-lab-unix1-testotp processed successfully.
Account Operating System-Platform-UNIXOTP-lab-unix6-testotp processed successfully.
Disconnected from vault

Processing complete.
```

And checking the accounts in the PVWA



Note the accounts are disabled. It is also worth noting that accounts which are disabled by BulkCPMManage will have this notation in the disable reason



Logs and Troubleshooting

bcm_Main.py.log

The bcm_Main.py.log will show basic logging when debug is not enabled.

When debug is enabled, information such as safes, platforms and accounts will also be logged. This will help in questions as ‘Why didn’t platform X’ or ‘Why didn’t safe ‘X’’ show in the menu, or ‘Why didn’t account X’ show in the menu.

If in doubt, run in test mode with debug set to ‘True’ in bcm_Props.props and examine the output. Note that account, platform and safes listings are written in the JSON format which can be easily copied and pasted to a text editor.

bcm_Main.py_Changes.log

This shows the status of accounts that were selected for change.

If there was a failure processing an account, it may be due to the API call, of which the HTML error code will be displayed. Also, executions run in test mode will also display the test mode status.

```
2023-10-30 18:31:57
Account: Operating System-Platform-UNIXOTP-lab-unix1-testotp, Action: Disable, Result: Not processed, in test mode
Account: Operating System-Platform-UNIXOTP-lab-unix6-testotp, Action: Disable, Result: Not processed, in test mode
######
2023-10-30 18:35:04
Account: Operating System-Platform-UNIXOTP-lab-unix1-testotp, Action: Disable, Result: Processed successfully
Account: Operating System-Platform-UNIXOTP-lab-unix6-testotp, Action: Disable, Result: Processed successfully
######
```

Known Issues

- Do not run BulkCPMManage when other applications that analyze traffic are running (e.g., Wireshark, Fiddler) as this may interfere with SSL connections