

CyberArk Privileged Access Security – Trident HSM Integration Guide

i4p Informatics Ltd.

Email: info@i4p.com

Website: <http://www.i4p.com>
<http://support.i4p.com>

[20200702-0001]

1 Introduction

The goal of this document is to describe how the Trident HSM can be used to store (and even generate) the server key for CyberArk Privileged Access Security (PAS), which is used to protect the sensitive information contained in the CyberArk Vault.

CyberArk Privileged Access Security is a solution to secure and control access to privileged credentials, like passwords and SSH keys, in a centralized manner. It is able to protect privileged user sessions (for example RDP, SSMS, SSH), enforce security policies, monitor and record user activity, and detect suspicious behavior in real-time which may indicate attacks in progress.

Trident HSM is a hardware security module developed by i4p Informatics, designed to perform sensitive cryptographic tasks and to securely manage cryptographic keys and data.

Please note that the installation and configuration steps presented in this document are valid for Windows Server 2012 R2, version 11.4.0 of the CyberArk Digital Vault, version 1.5.0 of the Trident PKCS#11 driver, and with respect to Trident HSM, for version 0.14.1 of the MPCM submodule. Later (or earlier) versions may need slightly different or even additional steps – please consult the homepage of i4p at <https://www.i4p.com> for up to date information or contact i4p support at <http://support.i4p.com>.

2 Requirements

This document assumes that the Trident HSM has been properly installed and set up (see the Trident HSM installation guide for details), is operational, and is accessible from the server on which CyberArk Digital Vault is running.

You will also need the `mpcm-pkcs11-x64-1.5.0.zip` file which you should have received from i4p as part of Trident SDK (the name of the file may refer to a slightly different version number if it contains a more recent driver release).

3 Preparing Trident HSM

There are a few steps you have to perform to prepare Trident HSM for the integration. First, you have to create a cryptoki user which will be utilized by CyberArk Digital Vault to connect to Trident HSM. Although you could utilize a cryptoki user already created for a different purpose, due to security reasons it is recommended to have a separate cryptoki user dedicated only to CyberArk Digital Vault. To create a cryptoki user, log in to the Trident HSM and issue the command:

```
mpcmd mpc_newusr
```

The command will ask a series of questions. You need to specify:

- the username
- the user role; enter “u” to create a key user
- the user type; enter “c” to create a cryptoki-enabled user
- the client application type; enter “e” to select External Client Application
- whether the user will be used by SAM; enter “n” to decline this option
- the authentication factor; enter “1” to select password-only authentication
- the initial password (twice)
- username of a CM administrator
- the password for the given CM administrator

In order to activate the user, its initial password should be modified by the user itself, so the command below has to be issued:

```
mpcmd mpc_passwd
```

This command will also ask a few questions, namely:

- the username; enter the same username you did for the `mpc_newuser` operation
- confirmation for the password change; enter “y”
- the new password (twice)
- username of the user on which behalf you want to perform the operation; enter again the same username you did for `mpc_newuser`
- password of the user on which behalf you want to perform the operation; enter the initial password you specified for `mpc_newuser`

A PIN code has to be assigned to the cryptoki user, which will be used later by CyberArk Digital Vault for authenticating itself. Trident SDK provides a Java-based tool to perform this operation, namely you have to issue the following command from any machine which has Java version 8 or later installed and can access Trident HSM:

```
java -jar cryptoki-init.jar <url> <username> <password> <pin>
```

where `<url>` points to the API endpoint of Trident HSM (for instance “https://192.168.1.10:2001”), `<username>` represent the username of the newly created cryptoki user, `<password>` is the password set by the `mpc_passwd` command, and finally `<pin>` is the PIN code.

Moreover, you have to set the timeout for user sessions, so that the session management of Trident HSM is completely aligned with how PKCS#11 operations are carried out by CyberArk Digital Vault. Log in to the Trident HSM and issue the command:

```
setconf -t mpcm
```

A text editor is launched with the MPCM configuration file opened in it. Change the values of the `mpcm_createsesstmo_u` and `mpcm_lastacc2sesstmo_u` parameters to a relatively large integer, e.g. 31,536,000 (they specify the number of seconds after a user session times out, measured from the session creation and last operation, respectively). To save the modified configuration, press Control-X and then “Y”. As the last step, restart the MPCM first by issuing the following command:

```
mpcmd mpc_stop
```

After MPCM is stopped, start it again using the command:

```
mpcmd daemon -t mpcm
```

4 Installing the Trident PKCS#11 driver

In order to integrate CyberArk Digital Vault with Trident HSM, you first have to install the Trident PKCS#11 driver on the server where CyberArk Digital Vault runs, and then configure CyberArk Digital Vault to use this driver.

Start the installation by creating a directory which will serve as the home directory of the Trident PKCS#11 driver, e.g. `C:\trident`. Open the `mpcm-pkcs11-x64-1.5.0.zip` archive and unpack the following two files to this directory:

- `bin/cmapi.dll`
- `bin/mpcm-pkcs11.dll`

Also create a configuration file in the same directory, for instance named `mpcm-pkcs11.conf`; it can contain entries in “name=value” format, each in a separate line, similar to “INI” files used throughout Microsoft Windows. An example configuration can be found in the ZIP archive as `conf/mpcm-pkcs11.conf`.

There are three mandatory parameters:

- `url` contains the URL of the API endpoint of Trident HSM, for example “`https://192.168.1.10:2001`”.
- `slot<n>` specifies the name of the cryptoki user to be used for CyberArk Digital Vault, where `<n>` is a positive integer identifying the PKCS#11 slot in Trident HSM assigned to the cryptoki user.
- `keytec` determines which method to use for generating Trident HSM keys. The value of this parameter must be 5, because CyberArk Digital Vault uses an AES key, and therefore multiparty key generation is not supported. (See the “MPCM Development Guide” for the list of possible values of `keytec` and their meaning.)

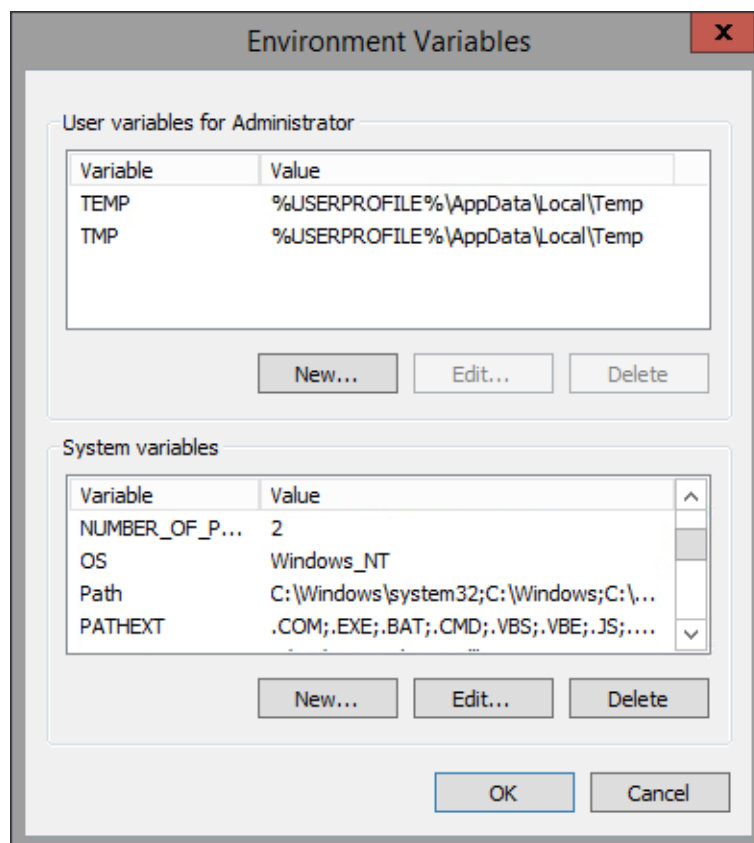
The optional parameters control the logging:

- `log_to_std_output`: set to “1” if you want log messages to be written to standard output, and to “0” if you do not (which is the default).

- **log_to_file**: absolute path of the file where log messages should be written; if value of this parameter is empty (the default), log messages will not be written to any file.
- **log_level**: an integer between 1 and 4 determining how detailed logging should be, corresponding to the widely used ERROR, WARNING, INFO and DEBUG levels. “1” means that only error messages are logged; “2” has the same effect as “1”, since currently there are no warning messages; “3” also logs inputs and outputs of invocations of PKCS#11 functions; and “4” will cause the Trident HSM API calls and their return status to be logged, too.

It is recommended to forward log messages to a file, so that potential issues with the integration between CyberArk Digital Vault and Trident HSM can be more easily debugged. Please note that the log file is not rotated automatically, it is the responsibility of the system operator to ensure its regular archival or removal.

Now that the driver home directory is properly set up and populated, you have to set two environment variables. As administrator, open the Control Panel, select **System** item, click on the **Advanced system settings** link, and in the opened dialog window click the **Environment Variables** button. Another dialog pops up showing the currently defined environment variables.



In the **System variables** section find the PATH environment variable, select it, and click the **Edit...** button. Add the absolute path of the driver home directory to the end of the **Variable value** field, ensuring that it is separated by a semicolon from the existing value. Click **OK** to apply the change and return to the dialog showing the environment variables.

Still in the **System variables** section, click the **New...** button. In the opened editor window, enter “MPCM_PKCS11_CONFIG_PATH” in the **Variable name** field, and the absolute path of the driver configuration file you just created in the driver home directory (e.g. “C:\trident\mpcm-pkcs11.conf”) in the **Variable value** field. Click **OK** to save the new environment variable. Close the dialogs and then the Control Panel itself.

5 Configuring CyberArk Digital Vault to use Trident HSM

After successfully installing the Trident PKCS#11 driver, you have to configure CyberArk Digital Vault to use it for storing the server key. In the first step, open the `dbparm.ini` configuration file, located under the CyberArk Digital Vault installation folder (which is typically “C:\Program Files (x86)\PrivateArk”), inside the `Server\Conf` subdirectory. Add – or, if for some reason it already exists, replace – the “[HSM]” section with the following lines:

```
[HSM]
PKCS11ProviderPath=C:\trident\mpcm-pkcs11.dll
```

Of course, if your driver home directory is not at `C:\trident`, you need to modify the path accordingly.

You also have to tell CyberArk to allow bidirectional traffic with the Trident HSM API endpoint in its firewall, by adding the line below to the “[MAIN]” section of the configuration file:

```
AllowNonStandardFWAddresses=[<host>],Yes,<port>:inbound/tcp, ➡
<port>:outbound/tcp
```

where `<host>` represents the domain name or IP address of the Trident HSM API endpoint, while `<port>` is the port number on which it listens, usually 2000. If the configuration file already contains a value for `AllowNonStandardFWAddresses`, consult the relevant CyberArk manuals for how to merge the two settings.

Finally, the PIN code of the `cryptoki` user designated for the CyberArk Vault Server should be added in encrypted form to the configuration file. Open a PowerShell window and issue the following command:

```
CAVaultManager SecureSecretFiles /SecretType HSM /Secret <pin>
```

where <pin> is the PIN code. (The command is located under the CyberArk Digital Vault installation folder, inside the `Server` subdirectory.) After the command completed successfully, check that a `HSMPinCode` parameter was indeed added to the “[HSM]” section of the configuration.

Restart the CyberArk Vault Server for the new configuration settings to take effect.

6 Importing an existing server key or generating a new one

You have two choices for utilizing Trident HSM for the storage of the server key of CyberArk Digital Vault: you can either import the existing key into Trident HSM or, which is the recommended approach, generate a new key inside Trident HSM. In both cases the CyberArk Digital Vault must be stopped before you issue the commands presented in subsequent paragraphs.

If you want to import the existing server key into Trident HSM, issue the following command in a PowerShell window:

```
CAVaultManager LoadServerKeyToHSM
```

Of course, you also need to tell CyberArk Digital Vault to use the server key stored in Trident HSM by changing the value of the `ServerKey` configuration parameter in the “[HSM]” section of `dbparm.ini` to:

```
ServerKey=HSM
```

After you restart CyberArk Digital Vault, it should use the key stored inside Trident HSM instead of the one stored in the Microsoft Windows filesystem.

On the other hand, if you want to generate a new server key and instruct CyberArk Digital Vault to switch to it (a more secure approach as in this case the key never leaves the Trident HSM), issue the command:

```
CAVaultManager GenerateKeyOnHSM /ServerKey
```

If the command succeeded, it will display on console a message like:

```
CAVLT187I Server Key was successfully generated on HSM device➔  
(KeyID=HSM#<n>).
```

where <n> is the generation number of the Trident HSM key. For the very first generated key it is 1; it will be increased by one at each key generation, until it reaches 255, after it is reset to 1.

To activate the new server key, you have to change the value of the `ServerKey` parameter in the “[MAIN]” section of `dbparm.ini` to:

```
ServerKey=HSM#<n>
```

where <n> is the key generation number returned by the previous command. The following command also has to be issued to re-encrypt the content of the vault using the new server key:

```
ChangeServerKeys <master> <emergency> HSM#<n>
```

where <master> represents the path of the current master CD, <emergency> is the path of the encrypted emergency password. (The command is located under the CyberArk Digital Vault installation folder, inside the Server subdirectory.) The command will first ask for confirmation to start the re-encrypt operation, and once again to ensure that the current master key is indeed in the <master> folder. There is a third confirmation question about the new server key, but you can safely ignore it and answer “y”, as it is not relevant in case of a new key stored in an HSM.

It is highly recommended to back up critical files and folders which are listed in the CyberArk Privileged Access Security Administration manual, as part of the detailed description of the ChangeServerKeys command.

After CyberArk Digital Vault is restarted, the new server key should be in effect.

Please note that after you imported an existing server key to Trident HSM, you cannot immediately generate a new key, because the two keys would be in conflict. The imported key should be deleted, for example with the Pkcs11Admin tool, or you have to temporarily revert to the original key stored in the Microsoft Windows filesystem by changing the ServerKey configuration parameter in the “[MAIN]” section of dbparm.ini to the path pointing to the key and issuing the command:

```
ChangeServerKeys <master> <emergency>
```

where the meaning of <master> and <emergency> is the same as before.

7 Contacting i4p

If you have any questions about Trident HSM in general, encountered issues with integrating Windows applications or services with Trident HSM, or would like to share your ideas about how to improve this document, feel free to contact us

- via email: info@i4p.com
- via our homepage: <https://www.i4p.com>

In case you want to provide feedback regarding the document, please include the document title and version number (as shown in the footer).

Please use our support request form available at:

<http://support.i4p.com>