Hvrwalletconfig

Since v5.6.5/5

Contents

Name

hvrwalletconfig - Configure HVR hub wallet.

Synopsis

hvrwalletconfig -*options hubdb* [*properties*]

Description

Command **hvrwalletconfig** configures the hub encryption wallet. For more information about hub wallet, see Hub Wallet and Encryption. For steps to configure hub wallet, see Configuring and Managing Hub Wallet.

This command is used to enable/disable the hub wallet, set wallet password, auto open hub wallet, rotate the hub wallet encryption key, change wallet password, and delete hub wallet.

The first argument *hubdb* specifies the connection to the hub database. For more information about supported hub databases and the syntax for using this argument, see Calling HVR on the Command Line.

The second argument *properties* specifies the properties that define the hub wallet type and configuration. For more information, see section Properties.

Options

This section describes the options available for command **hvrwalletconfig**.

Parameter	Description

-d*arg*	Delete wallet. Valid values for *arg* are:
	
	• **a**: Delete wallet but retain the artifacts (encryption key sequence and key history). This requires option **-p** and **Encryption** to be set to **NONE**.
	• **A**: Delete wallet and artifacts. This requires option **-p** and **Encryption** to be set to **NONE**.
	• **f**: Force wallet deletion even if the wallet is in use. This option can only be used in combination with the above options :
	  • **af** : Delete wallet but retain the artifacts such as historical keys (the wallet will be removed even if the encryption is not disabled). This can be used for example if wallet password is lost. Keeping artifacts requires access to the wallet (must be open and accessible). Historical keys will be lost. If **Encryption=SECRETS_ONLY** is not set before, encrypted passwords in a hub database will remain if wallet is not open or accessible. These passwords need to be manually fixed by a user by re-entering the passwords in the **HVR GUI** and saving them.
	  • **Af** : Delete wallet, remove artifacts such as historical keys (the wallet will be removed even if the encryption is not disabled). This can be used if the wallet password is lost. If **Encryption=SECRETS_ONLY** is not set before, encrypted passwords in a hub database will remain if wallet is not open or accessible. These passwords need to be manually fixed by a user by re-entering the passwords in the **HVR GUI** and saving them.
	
	Retaining artifacts is good to handle the transition, so that service passwords, jobs mehtioning encrypted password, etc continue to work as normal. However, when the wallet is deleted, those artifacts are not protected anymore (they were protected with wallet), so the historical keys become unprotected. This might compromise your previously encrypted values.

-h*class*	Location *class* of the hub database. Valid values for *class* are **db2**, **db2i**, **ingres**, **mysql**, **oracle**, **postgresql**, **sqlserver**, or **teradata**. For more information, see Calling HVR on the Command Line.
-m	Migrate a hub wallet to different storage instead of modifying its configuration in place. Wallet migration moves the encryption key from one wallet configuration file to another. The encryption key does not change, but its encrypted storage is first decrypted by the old wallet and then encrypted by a new wallet. For more information, see section Hub Wallet Migration in Hub Wallet and Encryption. In software wallet, this option is used to get a new password to change a wallet password to a new password. This option is mandatory when changing the wallet password (e.g. it protects against unintended password changes when setting up auto-open password option). A new password must be provided using option **-p**. The old password must be available either via auto-open password feature, or wallet must be opened using **hvrwalletopen** (through a running **HVR Scheduler**). In KMS wallet, this option is used to migrate a hub wallet from a previous KMS account/settings to new KMS account/settings or a user switches to a non-KMS wallet . This option is mandatory when migrating to another KMS wallet.
-p	Ask for a password of the hub wallet after command **hvrwalletconfig** is run. The following operations require providing the existing or a new password: • Operations that can lock the user out (such as removing **Wallet_Auto_Open_Password**) require the existing password. • Operations that install a new wallet, migrate a wallet to another device (to a different **Wallet_Type** or to the same **Wallet_Type** with a different account) require a new password.

-P	Enable automatic wallet open feature. This option saves the provided password into the **Wallet_Auto_Open_Password** property. This requires option **-p**. For more information about wallet auto-open, see section Methods to Supply Wallet Password in Hub Wallet and Encryption.
-r	Rotate (retire and regenerate) the encryption key. This option creates a new encryption key, encrypts it, and stores it in the wallet. The previous encryption key is moved to the history (encrypted with the new key) for the cases when HVR needs it to decrypt data encrypted with it. Then HVR decrypts the hub catalogs with the old key and re-encrypts them with the new key. During this key rotation process, both the old and new keys are available in the history. Historical keys are kept in the wallet configuration file each encrypted with the latest key. TX/Log files do not undergo key rotation. Instead, the old key is left in the history protected by the latest key. Existing password (**-p**) of the hub wallet is required if the wallet is not already open by the **HVR Scheduler** and if the **Wallet_Auto_Open_Password** property is not set. This option can be used alone or with other options that change the **Wallet_*** properties. It cannot be combined with the other options such as getting wallet configuration or removing historical keys.
-S*sequence*	Delete historical keys older than sequence number *sequence*. This option cannot be combined with others.

-T*tstamp*	Delete historical keys rotated before timestamp *tstamp*.
	
	This option cannot be combined with others.
	
	Valid values for *tstamp* can be an absolute timestamp or as a relative timestamp using seconds. Following are examples:
	
	`hvrwalletconfig -T 2019-11-26T10:54:59Z myhubuser /myhubpassword`
	
	The following example will remove keys rotated older than the last 86400 seconds (or 24 hours).
	
	`hvrwalletconfig -T now-86400 myhubuser/myhubpassword`

-u*user*[*/pwd*]	A hub database *user* name. For some databases (e.g. SQL Server) a password must also be supplied. For more information, see Calling HVR on the Command Line.

Properties

This section describes the properties that can be defined in the hub wallet configuration file.

Property	Description

Encryption	The category of data that should be encrypted using the hub wallet. Valid values are (case-sensitive): • **NONE** (**default**) - turns off the encryption. When setting up the hub wallet encryption without specifying **Encryption=SECRETS_ONLY** or **Encryption=ALL_CONFIDENTIAL**, then it remains as **Encrption=NONE**, and the previous behaviour remains. Also, to remove the hub wallet (without force), you need to set **Encryption=NONE** first. • **SECRETS_ONLY** - includes secret keys and passwords used for accessing /connecting to a database. For more information, refer to section Classification of Data on page Hub Wallet and Encryption. • **ALL_CONFIDENTIAL** - includes values in a user table and key-values exposed in the error message.
Wallet_Type	Type of the hub wallet. Valid values are (case-sensitive): • **SOFTWARE** is a file that stores the hub encryption key. • **KMS** is a network service (KMS) that encrypts the hub encryption key. For a detailed description on the wallet types, refer to section Hub Wallet Types on page Hub Wallet and Encryption.
Wallet_Auto_Open_Plugin	A user-supplied plugin that runs command **hvrwalletopen**. The **HVR Scheduler** can execute this plugin to obtain the wallet password. For example: */home/user/myplugin*.sh
Wallet_Auto_Open_Password	Remove a wallet auto-open password. This property is used only to disable the auto-open hub wallet feature. It does not accept any value. Just set it to blank for removing the auto-open password. For example: **Wallet_Auto_Open_Password=** For security reasons, "**Wallet_Auto_Open_Password=**" will work to unset the password, but "**Wallet_Auto_Open_Password**=*mypassword*" will not work. This is the only way to set it. For more information, refer to section Auto-Open Hub Wallet on page Configuring and Managing Hub Wallet.
Wallet_KMS_Region **KMS Wallet**	KMS region where the KMS server is located. For example: **Wallet_KMS_Region**=*eu-west-1* For more information, refer to section Creating and Enabling a KMS Wallet on page Configuring and Managing Hub Wallet.
Wallet_KMS_Access_Key_Id **KMS Wallet**	KMS access key ID of the AWS user to access KMS. The corresponding AWS Secret Access Key should be used as a password of the HVR hub wallet. For example: **Wallet_KMS_Access_Key_Id**=*AKIAJDRSJY123QWERTY* This property cannot be used with **Wallet_KMS_IAM_Role** For more information, refer to section Creating and Enabling a KMS Wallet on page Configuring and Managing Hub Wallet.

Wallet_KMS_Customer_Master_Key_Id **KMS Wallet**	Customer Master Key (CMK) ID that uniquely identifies CMK within your KMS region. CMK is used for encryption and decryption of the hub encryption key. For more information, refer to the AWS Documentation. For example: **Wallet_KMS_Customer_Master_Key_Id**=*1234abcd-12ab-1234590ab* For more information, refer to section Creating and Enabling a KMS Wallet on page Configuring and Managing Hub Wallet.
Wallet_KMS_IAM_Role **KMS Wallet**	KMS IAM role. This defines how to retrieve Access Key ID/Secret Access Key from an EC2 node. Using an IAM role does not require a wallet password. HVR fetches AWS credentials from the EC2 instance HVR hub is running on. This property cannot be used with **Wallet_KMS_Access_Key_Id**. For more information, refer to section Creating and Enabling a KMS Wallet on page Configuring and Managing Hub Wallet.
Encryption_Key_Filename **Software Wallet**	The name of the software wallet file (.p12) that stores the hub encryption key. The hub wallet file is a password-encrypted (using the PKCS#12 standard) file which is supplied by a user when creating the software wallet. For example: **hvrwallet-5e9f3869.p12** This property is automatically defined by HVR and cannot be manually configured by a user. For more information, refer to section Creating and Enabling a Software Wallet on page Configuring and Managing Hub Wallet.
Encryption_Key_Encrypted **KMS Wallet**	This defines the hub encryption key encrypted using the KMS wallet and stored encrypted in the HVR wallet configuration file. This property is automatically defined by HVR and cannot be manually configured by a user.
Encyption_Key_Sequence	Defines a unique sequence number of the hub encryption key. Every hub encryption key has a unique sequence number. At the same time, each encrypted secret contains its hub encryption key's sequence number. This sequence number is used to easily find the correct encryption key for the encrypted secret. This property is automatically defined by HVR and cannot be manually configured by a user.
Encyption_Key_History	Defines a history file that holds the historical record of old hub encryption keys (encrypted with the latest hub encryption key) in case they are needed for decrypting data encrypted with the old encryption keys. This property is automatically defined by HVR and cannot be manually configured by a user. For more information, refer to section History on page Hub Wallet and Encryption.