

HASHICORP VAULT / KMES SERIES 3 PKCS #12 PASSWORD EXPORT

Integration Guide

Applicable Devices: KMES Series 3



THIS DOCUMENT CONTAINS CONFIDENTIAL INFORMATION PROPRIETARY TO FUTUREX, LP. ANY UNAUTHORIZED USE, DISCLOSURE, OR DUPLICATION OF THIS DOCUMENT OR ANY OF ITS CONTENTS IS EXPRESSLY PROHIBITED.



TABLE OF CONTENTS

[1] OVERVIEW OF THE HASHICORP VAULT / KMES SERIES 3 SECRET EXPORT INTEGRATION	3
[1.1] About HashiCorp Vault	3
[1.2] Purpose of the Integration	3
[2] PREREQUISITES	4
[3] VAULT SETUP AND CONFIGURATION	5
[3.1] Download and install Vault	5
[3.2] CONFIGURE VAULT	5
[3.3] Start the Dev Server	7
[3.4] Accessing the Vault UI	9
[3.5] Modify the default ACL Policy	10
[4] SETTING UP AUTHENTICATION BETWEEN THE KMES SERIES 3 AND VAULT	11
[4.1] Userpass Auth Method	11
[4.2] TLS Certificates Auth Method	17
[5] OFFLOADING RANDOMLY GENERATED PKCS #12 PASSPHRASES TO VAULT	27
[5.1] Required Setup on the KMES Series 3	27
[5.2] Example - Excrypt Command RAUP	27
[5.3] Example - POST request to the KMES RESTFUL API using Postman	30
APPENDIX A: XCEPTIONAL SUPPORT	38

[1] OVERVIEW OF THE HASHICORP VAULT / KMES SERIES 3 SECRET EXPORT INTEGRATION

[1.1] ABOUT HASHICORP VAULT

From HashiCorp's online documentation (<u>https://www.vaultproject.io/docs/what-is-vault</u>): "Vault is a tool for securely accessing secrets. A secret is anything that you want to tightly control access to, such as API keys, passwords, or certificates. Vault provides a unified interface to any secret, while providing tight access control and recording a detailed audit log.

A modern system requires access to a multitude of secrets: database credentials, API keys for external services, credentials for service-oriented architecture communication, etc. Understanding who is accessing what secrets is already very difficult and platform-specific. Adding on key rolling, secure storage, and detailed audit logs is almost impossible without a custom solution. This is where Vault steps in."

[1.2] PURPOSE OF THE INTEGRATION

This integration gives users the ability to store PKCS #12 passphrases in HashiCorp Vault automatically after they are generated on the KMES Series 3. The intention is to rid DevOps and developers from the hassles of creating secrets and populating them into Vault in a secure manner when requesting X.509 certificates and key pairs.



[2] PREREQUISITES

Supported Hardware:

• KMES Series 3, version 6.1.3.11 and above, with the External Secret Storage license enabled

Other:

- HashiCorp Vault application
- OpenSSL



[3] VAULT SETUP AND CONFIGURATION

[3.1] DOWNLOAD AND INSTALL VAULT

Please refer to HashiCorp's Vault documentation at the following link for instructions on how to download and install the Vault application: https://www.vaultproject.io/docs/install

NOTE: The second installation option at the link above is using a precompiled binary. These binaries can be downloaded at the following url: https://www.vaultproject.io/downloads

To verify Vault is properly installed, run **vault -h** on your system. You should see help output. If you are executing it from the command line, make sure it is on your PATH or you may get an error about Vault not being found.

\$ vault -h

[3.2] CONFIGURE VAULT

Vault uses documented sane defaults so only non-default values must be set in the configuration file.

Create /etc/vault.d directory.

\$ sudo mkdir --parents /etc/vault.d

Create a Vault configuration file, vault.hcl.

\$ sudo touch /etc/vault.d/vault.hcl

Create a unique, non-privileged system user to run Vault.

\$ sudo useradd --system --home /etc/vault.d --shell /bin/false vault

Set the ownership of the */etc/vault.d* directory.

\$ sudo chown --recursive vault:vault /etc/vault.d

Set the file permissions.

\$ sudo chmod 640 /etc/vault.d/vault.hcl

Configure tcp Listeners in the Vault configuration file

The TCP listener configures Vault to listen on a TCP address/port, as shown in the example below.

```
listener "tcp" {
   address = "127.0.0.1:8200"
}
```

The **listener** stanza may be specified more than once to make Vault listen on multiple interfaces. If you configure multiple listeners you also need to specify **api_addr** and **cluster_addr** so Vault will advertise the correct address to other nodes.

The *vault.hcl* configuration file used for demonstration in this guide is shown below. It shows Vault listening on a private interface, as well as localhost.



NOTE: The values defined in the *vault.hcl* file need to be customized for each specific use case (i.e., IPs and ports; file paths to certificates).

NOTE: In this integration guide, the Vault server will be run in "Dev" mode. When deploying Vault in a production setting there are more things to consider (i.e., the storage backend), but the concept of configuring tcp listeners, as described below, still applies in that case. Please refer to Vault's documentation for specifics on how to deploy Vault in a production environment (<u>https://learn.hashicorp.com/tutorials/vault/getting-started-deploy</u>).

```
# Configure the storage backend for Vault
storage "file" {
    path = "/tmp/vault"
}
# Address and port on which Vault will respond to requests from the KMES Series 3
listener "tcp" {
    address = "10.0.5.118:8210"
    tls_disable = false
    tls_cert_file = "/home/bbarrows/Documents/Vault/client-cert.pem"
    tls_key_file = "/home/bbarrows/Documents/Vault/client-privatekey.pem"
}
# Advertise the non-loopback interface
api_addr = "https://10.0.5.118:8210"
# Enable the Vault web UI
ui = true
# Lock process memory pages, preventing them from being swapped to disk
disable mlock = true
```

Please reference the comments before each block. They explain what each of the defines is doing.

The most critical information to note is that **10.0.5.118** is the IP of the machine that Vault is installed on, and **8210** is the port on which Vault will listen for requests from the KMES Series 3.

An in-depth explanation of how to set up the client TLS certificates is beyond this course's scope, but **there is one crucial thing to note concerning this: The client certificate's common name must match the IP address set in the address define**. Otherwise, the KMES Series 3 will not verify the certificates presented by Vault to the KMES Series 3.

NOTE: cluster_address is not defined in the *vault.hcl* file above because only a single Vault server is being utilized for this demo.

For more information about configuring the Vault configuration file, please refer to Vault's documentation at the following url: <u>https://learn.hashicorp.com/tutorials/vault/configure-vault</u>



[3.3] START THE DEV SERVER

To start the Vault dev server, run:

```
$ vault server -dev -config=/etc/vault.d/vault.hcl
==> Vault server configuration:
             Api Address: https://10.0.5.118:8210
                     Cgo: disabled
         Cluster Address: https://10.0.5.118:8211
             Go Version: gol.14.7
              Listener 1: tcp (addr: "127.0.0.1:8200", cluster address: "127.0.0.1:8201", max
request_duration: "1m30s", max_request_size: "33554432", tls: "disabled")
              Listener 2: tcp (addr: "10.0.5.118:8210", cluster address: "10.0.5.118:8211", max
request_duration: "1m30s", max_request_size: "33554432", tls: "enabled")
              Listener 3: tcp (addr: "127.0.0.1:8210", cluster address: "127.0.0.1:8211", max
request duration: "1m30s", max request size: "33554432", tls: "enabled")
              Log Level: info
                  Mlock: supported: true, enabled: false
           Recovery Mode: false
                 Storage: file
                 Version: Vault v1.5.4+ent
             Version Sha: 1d81c1e64854fb0dcb3323468d95ad5590460a40
WARNING! dev mode is enabled! In this mode, Vault runs entirely in-memory
and starts unsealed with a single unseal key. The root token is already
authenticated to the CLI, so you can immediately begin using Vault.
You may need to set the following environment variable:
    $ export VAULT ADDR='http://127.0.0.1:8200'
The unseal key and root token are displayed below in case you want to
seal/unseal the Vault or re-authenticate.
Unseal Key: I29KTEqQVcl2Pa3xKgXffcwP9ae0ow157NFuG7Pj14A=
Root Token: s.XtzYp0lIJtaW3fMAtgWHdXxo
Development mode should NOT be used in production installations!
==> Vault server started! Log data will stream in below:
```

You should see output similar to that above. Notice that Unseal Key and Root Token values are displayed.

NOTE: The dev server stores all its data in-memory (but still encrypted), listens on **localhost** without TLS, and automatically unseals and shows you the unseal key and root access key.

With the dev server started, perform the following:

- 1. Launch a new terminal session.
- 2. Copy and run the export VAULT_ADDR ... command from the terminal output. This will configure the Vault client to talk to the dev server.

\$ export VAULT_ADDR='http://127.0.0.1:8200'

Vault CLI determines which Vault servers to send requests using the VAULT_ADDR environment variable.



- 3. Save the unseal key somewhere. Don't worry about *how* to save this securely. For now, just save it anywhere.
- 4. Set the VAULT_TOKEN environment variable value to the generated Root Token value displayed in the terminal output.

```
$ export VAULT_TOKEN="s.akT1I498dqOy4Z2C5ZimAS1R"
```

To interact with Vault, you must provide a valid token. Setting this environment variable is a way to provide the token to Vault via CLI.

Verify the Server is Running

Verify the server is running by running the vault status command. If it ran successfully, the output should look like the following:

\$ vault status	
Кеу	Value
Seal Type	shamir
Initialized	true
Sealed	false
Total Shares	1
Threshold	1
Version	1.5.4+ent
Cluster Name	vault-cluster-8667d21d
Cluster ID	b3977a72-9be9-d900-c0ec-c6012b1902da
HA Enabled	false

IMPORTANT: If using the Enterprise version of Vault, the dev server will seal itself 30 minutes after it is started. This means that it will be necessary to perform the following actions every time that the dev server times out, if using the Enterprise version of Vault:

- 1. Trigger a Vault shutdown with CTRL+C in the terminal window where the Vault server was running.
- 2. Run the following command in a terminal:

\$ rm -r /tmp/vault/

- 3. Re-perform steps 3.2 thru 3.5.
- 4. Re-configure either the userpass or TLS authentication auth method in Vault, as described in section 4.

For more information about how the dev server works, please refer to <u>https://www.vaultproject.io/docs/concepts/dev-server</u>.

As stated previously, the information provided in this integration guide can be applied to a production implementation of Vault. For specifics on how to deploy Vault in a production environment please refer to HashiCorp's Vault documentation (<u>https://learn.hashicorp.com/tutorials/vault/getting-started-deploy</u>).



[3.4] ACCESSING THE VAULT UI

Go to http://localhost:8200 in a web browser.

Vault × +		_ = ×
← → C () localhost:8200/ui/vault/auth?with	token	🖈 🕒 Incognito 🕴
Vault	Sign in to Vault	● Status →
	Namespace /(Root) Method Token C Token	
	Contact your administrator for login credentials	
1	U 2020 HashiCorp Vault 1.5.0+ent.hsm Documentation	

Copy and paste in the Root Token that was output from the **vault server** command into the "Token" field, then click "Sign In".





[3.5] MODIFY THE DEFAULT ACL POLICY

Navigate to the Policies menu, then select the "default" ACL policy.

Vault × +		_ = ×
← → C () localhost:8200/ui/vault/p	oolicy/acl/default	⊶ ☆ <mark>©</mark> 🏛 🗯 🕒 :
Secrets Access Policies	Tools	. Status v ⊡ v 🛛 v
POLICIES	« ACL policies default	Dausland palloy > Edit palloy >
ACL Policies Role Governing Policies Endpoint Governing Policies	<pre>Policy (HCLformat)</pre>	Download policy > Edit policy >
	Documentation	

Click "Edit Policy", then scroll to the bottom of the policy and paste in the following starting at line 89:

```
path "secret/data/*" {
   capabilities = ["create", "read", "update", "delete", "list"]
}
path "secret/*" {
   capabilities = ["create", "read", "update", "delete", "list"]
}
path "sys/*" {
   capabilities = [ "create", "read", "update", "delete", "list" ]
}
path "sys/mounts/*" {
   capabilities = [ "create", "read", "update", "delete", "list" ]
}
# List enabled secrets engine
path "sys/mounts" {
   capabilities = [ "create", "read", "update", "delete", "list" ]
}
# Work with pki secrets engine
path "pki*" {
   capabilities = [ "create", "read", "update", "delete", "list" ]
}
```

Click "Save", then a message should appear in the bottom left-hand side of the page confirming that ACL policy "default" was successfully saved.



[4] SETTING UP AUTHENTICATION BETWEEN THE KMES SERIES 3 AND VAULT

Two different methods can be used to authenticate the KMES Series 3 with Vault, which is the **Userpass Auth Method** or the **TLS Certificates Auth Method**. Instructions for both methods are provided in the following sections.

[4.1] USERPASS AUTH METHOD

The **userpass** auth method allows the KMES Series 3 to authenticate with Vault using a username and password combination.

Configuring userpass authentication in Vault

Method 1 - Using the Vault UI

Navigate to the Access page in the Vault UI, then select "Enable new method".

▼ Vault × +	_ 0	×	
\leftrightarrow \rightarrow C (i) localhost:8200/ui/vault/acc	$- \rightarrow \mathbb{C}$ (i) localhost:8200/ui/vault/access $\sim \pm$ 6 $\approx \pm$ 1 :		
Secrets Access Policies	Tools 💿 Status ~ 🛛 🐑 🗶 ~	^	
ACCESS	Authentication Methods		
Auth Methods	Enable new method +		
Entities	⊙ token/		
Groups	auth_token_6861fe07		
Leases Namespaces			
Control Groups			
	[] © 2020 HashiCorp Vault 1.5.4+ent Documentation		
	() a care normally years instrume and an instrumentation	~	



Select the "Username & Password" authentication method, then click "Next".

	×	+			
→ C () I	ocalhost:8200/ui/v	vault/settings/auth	ı/enable		여 ☆ 😋 🏛 🛸 🌘
Secret	s Access Po	olicies Tools			◉ Status 🗸 🕞 🖌 💄
Enable an	Authentic	ation Met	hod		
Generic					
:			<u>ē</u>	—	
AppRole	JWT	OIDC	TLS Certificates	Username & Password	
				۲	
Cloud					
C-)	aws	Δ	۵	0	
AliCloud	AWS	Azure	Google Cloud	GitHub	
nfra					
()		okta			
-	LDAP	Okta	RADIUS		
Kubernetes					

Leave the path as the default value, "userpass", then click "Enable Method".





Navigate back to the menu for the "userpass" auth method just created, then click "Create user".

Vault × +			×
C (i) localhost: 8200/ui/vault/access/userpass/item/user			
Secrets Access Policies	Tools	ම Status ∽ 🛛 ► 👻 Š	^
ACCESS Auth Methods Entitles Groups Leases Namespaces Control Groups	✓ methods Users Configuration No users yet Alist of users will be listed here. Create your first user to get started. Create user Create user	Create user +	
	Documentation		÷

Specify a username and password for the new user, then click "Save".

▼ vault × +		_ = ×
← → C ① localhost:8200/ui/vault/acc ▼ Secrets Access Policies	ess/userpass/item/user/create	☆ ⓒ ≗ ☆ ⓒ : Status ~ │
ACCESS Auth Methods Entitles Groups Leases Namespaces Control Groups	 ⟨ users Create user Username () userpass_authentication_demo Password () Futurex123 view Tokens Suve Cancel 	
	Documentation	~

A message should appear in the bottom left-hand side of the page confirming that the new "userpass_ authentication_demo" user was saved successfully.



Method 2 - Using the Vault CLI

Enable the **userpass** auth method:

\$ vault auth enable userpass Success! Enabled userpass auth method at: userpass/

Configure it with users that are allowed to authenticate:

```
$ vault write auth/userpass/users/userpass_authentication_demo \
    password=Futurex123 \
    policies=admins
```

Success! Data written to: auth/userpass/users/userpass_authentication_demo

This creates a new user "userpass_authentication_demo" with the password "Futurex123" that will be associated with the "admins" policy. This is the only configuration necessary.

Create a Vault Userpass Authentication Cloud Credential

Log in to the KMES Series 3 application interface with the default Admin users.

Navigate to the *Cloud Credentials* menu, then click the "Add Cloud Credential..." button.

Click the "Service" dropdown and select "Vault Userpass Authentication".





Any value can be specified in the "Name" field, but the "Access Name" value must match the name of the user that was created under the userpass auth method in Vault.

In the "Password" field, click the "Enter" button and set the same password that was set for the user created in Vault. Click "Save".

The Add Cloud Credential dialog should look similar to the image below, then click "OK".

•	Cloud Credential $ imes$
Name:	vault_userpass_authentication
Service:	Vault Userpass Authentication
Access Name:	userpass_authentication_demo
Password:	<loaded></loaded>
	OK Cancel

Testing userpass authentication

Navigate to the Configuration menu, then double-click on "Vault API Options".

<u>File H</u> elp			Futurex a L A
FLITUREX.COM	CONFIGURATION	MFK: 8071, PI	МК: 1Е26
3 cloud credential shown	Configuration	∕_ Value	Description
Certificate Authoritie 2 CAs, 0 generated keys	Major Keys	MFK: 8071, PMK: 1E26	Keys stored in the HSM used to encrypt other keys.
X.509 DN Profiles X.509 DN profiles total	Manage Duplicate Objects		View and delete duplicate objects.
■ X.509 Extension Profi © 8 X.509 v3 ext profiles tota	Manage Smart Card		Change the PIN or PUK stored on a smart card.
Signing Approval 8 certificate requests, 8 sig	Network Options		Set network configuration options.
PKI Keys 0 PKI Key Pairs shown, 7 to	OCSP Options		Set OCSP configuration options.
WCCE Templates 33 WCCE templates total	Options		Set various configuration options.
Remote LDAP Servers	Rassword Security		Adjust global password security settings.
Token Generation Pro 0 token generation profiles	Peer options		Set deletion time period for peering objects.
KMIP Objects	Permission Options		Set Permission configuration options.
0 KMIP objects shown, 0 to Hosts/Networks	Proxy Options		Set proxy options for certain HTTP client services.
O hosts total Encryption Devices	Registration Authority Options		Set registration authority configuration options.
Peers	🔮 Reset Database	Database initialized: 2020-10-01 21:28:05	Rebuild the database and reset the system.
0 peers	Restore	Previous restore: 2020-10-13 18:35:50	Restore database including keys and certificates.
Configuration MFK: 8071, PMK: 1E26	SCSA Options		Set SCSA configuration options.
Users 3 groups, 4 users	SNMP Options		Configure SNMP walk options.
Logs 418 logs total	System Key Options		Manage system keys.
Reports 3 reports	Update Firmware		Update the encryption card firmware.
Templates 0 templates	Upload Feature Request		Upload new application abilities
Databases 0 databases	Vault API Options		Set options for interacting with an external Vault service.
Hardware 0 disks, 0 RAIDs	Web Server Configuration		Set web server configuration options.
Jobs 0 Jobs			Edit
	·		Group: Admin Group Users: Admin1,Admin2
			Futurex 1:18



Check the "Enable Vault Service" box.

•	Vault Options $ imes$
Enable Vault Service	
Vault API URL:	https://10.0.5.118:8210/v1
Namespace:	
Certificate Mount Point:	cert
Userpass Mount Point:	userpass
Secret Mount Point:	secret
Secret Path:	
Require User Path:	
Timeout:	5
Credential:	vault_userpass_authentication Select
Test Configuration	
	Save Cancel

Set the Vault API URL to "https://10.0.5.118:8210/v1" and select the Vault Userpass Authentication Cloud Credential created in the previous step.

The rest of the fields can be left as their default values. Click "Test Configuration...".



If all of the configuration steps were completed properly the test should result in a success.



[4.2] TLS CERTIFICATES AUTH METHOD

The **cert** auth method allows the KMES Series 3 to authenticate with Vault using SSL/TLS client certificates which are either signed by a CA or self-signed.

Configure the Vault Client connection pair on the KMES Series 3

For the TLS Certificates auth method, it is necessary to first configure the **Vault Client** connection pair on the KMES Series 3 before configuring the **cert** authentication method in Vault.

Log in to the KMES Series 3 application interface with the default Admin users.

Navigate to the *Configuration* menu, then double-click on *Network Options*. Go to the *TLS/SSL Settings* tab, then click the dropdown and switch to the **Vault Client** connection pair.

UTUREX.COM	CONFIGURATION	MFK: 8071, PMK: 1E26
		Network Options X
Keys 0 keys, 0 configuration set	Ethernet Settings Network Settings TCP Settings TLS/SSL	L Settingsption device security mode(s)
Application Keys 0 application key groups, 0	Connection:	Vault Client
Cloud Credentials 0 cloud credential total	Settings	Allow Anonymous Connections
Certificate Authoritie 0 CAs, 0 generated keys	_ TLS Enabled	Tokens.
X.509 DN Profiles 2 X.509 DN profiles total	Ciphers: 10 selected	v ion options.
X.509 Extension Profi	Protocols: 3 selected	▼ I ranges from one device serial to and
8 X.509 v3 ext profiles tota Signing Approval	Cert Type: RSA	thentication configuration options.
0 certificate requests, 0 sig PKI Keys	User Certificates	n5.
0 PKI Key Pairs shown, 0 to WCCE Templates	PKI keys Not loaded	Edit
33 WCCE templates total Remote LDAP Servers	Certificates Not loaded	Edit ISM used to encrypt other keys.
0 remote LDAP directories	Use Futurex certificates	plicate objects.
0 token generation profiles		VK stored on a smart card.
KMIP Objects 0 KMIP objects shown, 0 to Hosts /Notworks	Anonymous connection key size: 2	2048 viration options.
Hosts/Networks 0 hosts total		tion options.
Encryption Devices 1 group, 1 device, 0 clients		ration options.
Peers 0 peers		ord security settings.
Configuration MFK: 8071, PMK: 1E26		eriod for peering objects.
Users 1 group, 2 users		iguration options.
Logs 48 logs total		OK X Cancel or certain HTTP client services.
Reports 3 reports	Registration Authority Options	Set registration authority configuration options.
Templates	Reset Database Database ini	nitialized: 2020-10-14 01:47:57 Rebuild the database and reset the system.
Databases		Ec
		Group: Admin Group Users: Admin1,

Uncheck the "Use System/Host API SSL Parameters" and "Use Futurex Certificates" boxes, then click the "Edit..." button next to "PKI keys" in the *User Certificates* section.



No PKI Key Pair Generate.	Clear
Click "Generate" to create. Clear	
clear or	Request
Request	

Generate a new PKI key pair, then request a CSR.

Next, the CSR just generated needs to be signed by a Certificate Authority (CA). For this demonstration, the CSR will be signed by the same CA that signed the client certificate that was configured in the Vault configuration file in section 3.2. Upload the CA certificate and the signed CSR to the storage medium configured on the KMES Series 3, then click the "Edit..." button next to "Certificates" in the User Certificates section, as shown below.

thernet Settings	-				
Connection:			Vault Client		-
Settings	Host API SSL Paramet	ers*	Allow Anonymous	Connections	
Ciphers:	10 selected				-
Protocols:	3 selected				•
Cert Type:	RSA				-
	Loaded s Not loaded rex certificates				Edit
Anonymous conr	ection key size:		2048		•



Right-click on the "Vault Client SSL CA" X.509 Certificate Container, then click "Import...".

ame	/ Notes	Status
Vault Client SSL CA	Import Certificate Co	ontainer
Vault Client Truster	d Export Certificate Co	ontainer
Vault Client Truste	d CA 2 X.509 Certificate Co	ontainer
Vault Client Truste	d CA 3 X.509 Certificate Co	ontainer
Vault Client Truste	d CA 4 X.509 Certificate Co	ontainer
Vault Client Truste	d CA 5 X.509 Certificate Co	ontainer

Click the "Add" button in the lower-left area of the *Import Certificates* dialog, then find, select, and open the CA and signed CSR, and click "OK".

			Import Certificate
erified: 2 certificates			
Subject	∇	File	Key Usage
		ssl-ca-cert.pem	Sign/Verify
Vault Connection Pair		ssl-server-cert.pem	Sign/Verify
Wault Connection Pair		ssl-server-cert.pem	Sign/Verify
Wault Connection Pair		ssl-server-cert.pem	Sign/Verify
		551 561 761 661 6. penn	2.3.1 (2.1.)
nverified: 0 certificates			
	~	51-	F
	7	File	Error
	7	File	Error
	7	File	Error
	$\overline{\nabla}$	File	Error
	7	File	Error
	~	File	Error
	~	File	Error
Subject	∇ d Folder: □	File	Error
Inverified: 0 certificates		File	Error OK Cancel



The certificate tree for the Vault Client connection pair is loaded now, as shown below.

lame 🛛 🛆	Notes	Status
Vault Client SSL CA	X.509 Certificate Container	
– 🌉 са	Self-signed	Valid
Vault Connection Pair	SSL Certificate	Valid
Vault Client Trusted CA 1	X.509 Certificate Container	
Vault Client Trusted CA 2	X.509 Certificate Container	
Vault Client Trusted CA 3	X.509 Certificate Container	
Vault Client Trusted CA 4	X.509 Certificate Container	
Vault Client Trusted CA 5	X.509 Certificate Container	
(****	•
Use SSL CA as Trusted CA	ОК	Cancel

Click the "OK" button to save.

NOTE: As previously noted, for this demonstration, the same CSR was used to sign both the Vault Client connection pair CSR and the client certificate set in the Vault configuration file. If this were not the case, then the CA that signed the Vault Client connection pair CSR would need to be loaded to the "Vault Client SSL CA" X.509 Certificate Container with the accompanying signed certificate, and the CA that signed the client certificate set in the Vault need to be loaded to any of the "Vault Client Trusted CA" X.509 Certificate Containers, along with the accompanying signed certificate.

Click the "OK" button to save and exit out of the *Network Options* menu.



Configuring cert authentication in Vault

Method 1 - Using the Vault UI

Navigate to the Access page in the Vault UI, then select "Enable new method".

▼ Vault × +	_ □ ×
$\leftrightarrow \rightarrow \mathbb{C}$ (i) localhost:8200/ui/vault/acco	ss 🗣 🖈 🕒 🗄
Secrets Access Policies	Tools 💿 Status 🗸 🕞 🗸 🖺 🗸
ACCESS Auth Methods Entities Groups Leases Namespaces Control Groups	Authentication Methods
	🕑 © 2020 HashiCorp Vault 1.5.4+ent Documentation



ault	×	+		
→ C () la	calhost:8200/ui/v	ault/settings/auth	/enable	
Secrets	Access Po	licies Tools		
Faabla aa	Authoptic	ation Met	had	
Enable an	Authentic	ación Mec	noa	
Generic				
:			<u>5</u> -	0=
AppRole	JWT	OIDC	TLS Certificates	Username & Password
			۲	
Cloud				
C-J	aws	Δ	<u>۵</u>	0
AliCloud	AWS	Azure	Google Cloud	GitHub
Infra				
۲		okta		
Kubernetes	LDAP	Okta	RADIUS	

Select the "TLS Certificates" authentication method, then click "Next".

Leave the path as the default value, "cert", then click "Enable Method".





Navigate back to the menu for the "cert" auth method just created, then click "Create certificate".

Vault × +		_	
$\leftrightarrow \rightarrow \mathbb{C}$ (i) localhost:8200/ui/vault/access/cert/item/certificate \Rightarrow (i) \approx (i) \approx			
Secrets Access Policies	Tools	ම Status → 🛛 ⊨ → 🛋 →	
ACCESS Access Auth Methods Entitles Groups Leases Namespaces Control Groups	Image: continue of the second seco	Create certificate +	
	D 2020 HashiCorp Vault 1.5.4+ent Documentation		

Specify a name for the certificate and upload a single *.pem* file that contains the certificate chain configured for the Vault Client connection pair on the KMES Series 3. Then click "Save".

▼ Vault × +		_			
← → C O localhost.8200/ui/vault/access/cert/item/certificate/create ☆ G					
Secrets Access Policies	Tools (*) Status ~				
ACCESS	<pre>< certificates Create certificate</pre>				
Auth Methods Entities	Name ① certificate_authentication_demo				
Groups Leases Namespaces	Certificate ①				
Control Groups	theorem a file chain.pem Select a file from your computer	٢			
	Display name () certificate_authentication_demo				
	Required extensions ③				
	 ✓ Constraints ✓ Tokens 	Add			
	Save Cancel				
	(1) © 2020 HashiCorp Vault 1.5.4+ent Documentation				

A message should appear in the bottom left-hand side of the page confirming that the new "certificate_ authentication_demo" certificate was saved successfully.



Method 2 - Using the Vault CLI

Enable the **cert** auth method:

```
$ vault auth enable cert
Success! Enabled cert auth method at: cert/
```

Configure it with trusted certificates that are allowed to authenticate:

```
$ vault write auth/cert/certs/certificate_authentication_demo \
    display_name=certificate_authentication_demo \
    policies=web,prod \
    certificate=@chain.pem \
    ttl=3600
Success! Data written to: auth/cert/certs/certificate_authentication_demo
```

This creates a new trusted certificate "certificate_authentication_demo" with same display name and the "web" and "prod" policies. The certificate (public key) used to verify clients is given by the "chain.pem" file. Lastly, an optional ttl value can be provided in seconds to limit the lease duration.

Create a Vault Certificate Authentication Cloud Credential

In the KMES Series 3 application interface, navigate to the *Cloud Credentials* menu, then click the "Add Cloud Credential..." button.

Click the "Service" dropdown and select "Vault Certificate Authentication".





Any value can be specified in the "Name" field, but the "Access Name" value must match the name of the certificate that was created under the cert auth method in Vault.

The value for the "TLS Config" field defaults to "Vault Client". This configures the Cloud Credential to use the "Vault Client" connection pair for authenticating with Vault.

The Add Cloud Credential dialog should look similar to the image below, then click "OK".

•	Cloud Credential $ imes$
Name:	vault_certificate_authentication
Service:	Vault Certificate Authentication
Access Name:	certificate_authentication_demo
TLS Config:	Vault Client 💌
	OK Cancel

Testing cert authentication

Navigate to the Configuration menu, then double-click on "Vault API Options".

<u>File H</u> elp				
FLITUREXE		CONFIGURATION		MFK: 8071, PMK: 1E26
3 cloud credential sh	own 🔺	Configuration	Value	Description
Certificate Author 2 CAs, 0 generated k		Major Keys	MFK: 8071, PMK: 1E26	Keys stored in the HSM used to encrypt other keys.
■ X.509 DN Profiles • 2 X.509 DN profiles to	a btal	Manage Duplicate Object	cts	View and delete duplicate objects.
S X.509 Extension 8 X.509 v3 ext profile		Manage Smart Card		Change the PIN or PUK stored on a smart card.
Signing Approva	8 sig	Network Options		Set network configuration options.
PKI Keys 0 PKI Key Pairs shown	, 7 to	OCSP Options		Set OCSP configuration options.
WCCE Templates	otal	Options		Set various configuration options.
Remote LDAP Se	vers	Password Security		Adjust global password security settings.
Token Generation	Pro	Peer options		Set deletion time period for peering objects.
KMIP Objects		Permission Options		Set Permission configuration options.
Hosts/Networks	, 0 to	Proxy Options		Set proxy options for certain HTTP client services.
Encryption Devic	es	Registration Authority (Options	Set registration authority configuration options.
1 group, 1 device, 0 d	lients	🔮 Reset Database	Database initialized: 2020-10-	01 21:28:05 Rebuild the database and reset the system.
0 peers		Restore	Previous restore: 2020-10-13	18:35:50 Restore database including keys and certificates.
Configuration MFK: 8071, PMK: 1E2	6	SCSA Options		Set SCSA configuration options.
Users 3 groups, 4 users		SNMP Options		Configure SNMP walk options.
Logs 418 logs total		System Key Options		Manage system keys.
Reports 3 reports		Update Firmware		Update the encryption card firmware.
Templates 0 templates		Upload Feature Reques	t	Upload new application abilities
Databases 0 databases		Vault API Options		Set options for interacting with an external Vault service.
Hardware 0 disks, 0 RAIDs		Web Server Configuration	on	Set web server configuration options.
Jobs 0 Jobs	•			Edit
		`		Group: Admin Group Users: Admin1,Admin2
				Futurex 1:18



Check the "Enable Vault Service" box.

•	Vault Options $ imes$
Enable Vault Service	
Vault API URL:	https://10.0.5.118:8210/v1
Namespace:	
Certificate Mount Point:	cert
Userpass Mount Point:	userpass
Secret Mount Point:	secret
Secret Path:	
Require User Path:	
Timeout:	5
Credential:	vault_certificate_authentication
Test Configuration	
	Save Cancel

Set the Vault API URL to "https://10.0.5.118:8210/v1" and select the Vault Certificate Authentication Cloud Credential created in the previous step.

The rest of the fields can be left as their default values. Click "Test Configuration...".



If all of the configuration steps were completed properly the test should result in a success.



[5] OFFLOADING RANDOMLY GENERATED PKCS #12 PASSPHRASES TO VAULT

In this section, two examples will be covered in which Futurex APIs are invoked to request an X.509 certificate and associated key pair, which is issued as a PKCS #12 file. Then, the passphrase that was generated for decrypting the PKCS #12 file is automatically offloaded to Vault by the KMES Series 3.

The first example will involve running the RAUP Excrypt command while connected to the System/Host API port on the KMES via OpenSSL.

The second example will involve sending a POST request to the KMES RESTful API using the Postman application.

[5.1] REQUIRED SETUP ON THE KMES SERIES 3

Before attempting either of the examples, the following must be set up on the KMES Series 3:

- 1. Create a Signing Approval Group.
- 2. Create a CA tree (**IMPORTANT:** Ensure that PMK is chosen as the major key for all certificates created in this CA tree)
- 3. Add an Issuance Policy to the leaf certificate in the CA tree (**IMPORTANT:** In the *X.509* tab of the *Issuance Policy* dialog, ensure that all of the permission boxes are checked and that the Signing Approval Group created in step 1 is selected)
- 4. Create a new User Group with all of the "Manage certificates", "Manage keys", and "Perform cryptographic operations" permissions, and set the number of users required to log in to "1".
- 5. Add one user to the User Group that was created in the previous step.
- 6. Give the created User Group "Use" permissions on all of the following:
 - The Signing Approval Group created in step 1
 - The entire CA tree created in step 2
 - The Cloud Credential that is being used in the Vault API Options menu

NOTE: Please refer to the KMES Series 3 user guide for information about completing the actions above.

[5.2] EXAMPLE - EXCRYPT COMMAND RAUP

First, ensure that the RAUP command is enabled in the *Host API Options* menu.

Then, connect to the System/Host API port on the KMES via OpenSSL.

```
$ openssl s_client -connect 10.0.8.28:2001 -cert signed-client-cert.pem -key private-key.pem -
CApath . -CAfile chain.pem
```

NOTE: The System/Host API connection pair on the KMES Series 3 must be configured so that this OpenSSL connection will work. Instructions for setting this up are outside the scope of this course. Please refer to the KMES Series 3 user guide for more information.



Once connected successfully, run the RKLO Excrypt command twice to login in with the default Admin users.

```
[AORKLO; DAAdmin1; CHsafest; ]
[AORKLO; ANY; UL1; UT2; RL1; ]
[AORKLO; DAAdmin2; CHsafest; ]
```

[AORKLO; ANY; UL2; UT2; RL0;]

Now that we're logged in, we can run the RAUP command to upload a new X.509 PKI request.

```
[AORAUP;CADemo CA;RTDemo Sub Cert;NATestUpload;HASHA256;GNDemo Approval Group;ENExample TLS Cer-
tificate;SN{2.5.4.3,12,436F6D6D6F6E4E616D6548657265}, {1.3.3.7,19,30313233};KTRSA 2048;MP1;]
[AORAUP;ANY;AP1;ID34E2CCE30BC2F336;PW567E463B516F3120246C265446366A585D63636A794342215F344B7D43266F-
765D454C3B5741307068614F657A487A2234692E51754B5C46342477282856253C;]
```

If the command succeeds, as it does above, a new X.509 PKI, issued as a PKCS #12 file, will be generated on the KMES Series 3, and the passphrase for the PKCS #12 file will be stored in Vault.

NOTE: The RAUP command above uses the CA tree that was created in section 5.1. For more information about the RAUP command tags, please refer to API documentation in the Futurex Portal.

Now, if we log in to the Vault UI and navigate into the "secret/" folder we'll see the passphrase secret was added successfully.

▼ Vault × +			_ D	×
← → C () localhost:8200/ui/vault/secrets/sec	cret/list	☆)	G 🙇 🗰 🖪	:
Secrets Access Policies Tools		● Status ~). · I ·	
<pre> secrets < secret </pre>				
Secret Version 2				
Secrets Configuration				
Q Filter secrets			Create secret +	
CommonNameHere-34E2CCE30BC2F336				
	🙌 © 2020 HashiCorp Vault 1.5.4+ent	Documentation		-

Click on the "CommonNameHere-34E2CCE30BC2F336" secret.



Vault	× +	_
\leftrightarrow > C	① localhost:8200/ui/vault/secrets/secret/show/CommonNameHere-34E2CCE30BC2F336	🖈 G 🏛 🌲 🖪 :
V Se	ecrets Access Policies Tools 💿 St	atus 🗸 📄 🗸 🔳 🗸
/ secret / Con	mmonNameHere-34E2CCE30BC2F336	
	NameHere-34E2CCE30BC2F336	
		ecret 🗸 Create new version +
Key	Value	
passphrase	Ē &	
	() © 2020 HashiCorp <u>Vault 1.5.4+ent</u> Documentation	

We see that the *Key* is "passphrase" and the *Value* is the passphrase for the PKCS #12 file that was generated on the KMES Series 3.



[5.3] EXAMPLE - POST REQUEST TO THE KMES RESTFUL API USING POSTMAN

This example requires an additional configuration to be made on the KMES Series 3.

Configure JWT Options

Navigate to Configuration -> JWT Options.

CONFIGURATION MFK: 8071	, PMK: 1E26
VS Configuration / Value	Description
eys, 0 configuration set Encryption Device Security Mode	Change local encryption device security mode(s)
pplication Keys pplication key groups, 0 Host API Options	Set Host API configuration options.
loud credential shown Install	Install new versions of the application.
rtificate Authoritie (509s shown, 7 total JWT Options Json Web Token Option X	Configure Ison Web Tokens.
509 DN Profiles	Set KMIP configuration options.
EOO Evtension Brafi	
.509 v3 ext profiles tota	Mass reallocate KSN ranges from one device serial to an
certificate requests, 8 s	Set remote LDAP authentication configuration options.
KI Key S Log Options	Configure log options.
CCE Templates W2CE templates total	Manage continuous symmetric key load sessions for ma
mote LDAP Servers Major Keys	Keys stored in the HSM used to encrypt other keys.
emote LDAP directories Annage Duplicate Object:	View and delete duplicate objects.
oken generation profiles IIP Objects Manage Smart Card Base64 encoded Import	Change the PIN or PUK stored on a smart card.
MIP objects shown, 0 to	Set network configuration options.
Insts/Networks Insts total OCSP Options Leeway: 30 Seconds Validity: 10	Set OCSP configuration options.
cryption Devices roup, 1 device, 0 clients Save Cancel	Set various configuration options.
eers Password Security	Adjust global password security settings.
nfiguration K: 8071, PMK: 1E26 Peer options	Set deletion time period for peering objects.
iers Demoising Cations	Set Permission configuration options.
roups, 4 users	Set proxy options for certain HTTP client services.
eports Registration Authority Options	Set registration authority configuration options.
mplates emplates	· · · · · · ·
itabases v	

In the *Json Web Token Option* dialog, set the *Issuer Name* to "futurex" and set "safest123" as an HMAC Key password. The remaining fields can be left as their default values. Then, click "Save".

Sending a POST request to the KMES RESTful API using Postman

NOTE: To perform the steps that follow, you must have the Postman application installed on your local computer. The same concepts would apply if you wanted to use cURL rather than Postman. However, to use cURL, you would need to generate the JWT token elsewhere (i.e., a website such as https://jwt.io/), whereas, with Postman, we'll be able to generate a JWT token "on-the-fly" when the POST request is sent.



Start the Postman application, then click the orange "New" button in the top left area of the page.

+ New Import Runner □ ~	器 My Workspace ~	ቶ Invite	\$ & \$	ß ậ ♡ Sign In
Q Filter			No Environment	
History Collections APIs				
Save Responses				
You haven't sent any requests				
Any request you send in this workspace will appear here.				
🎋 Show me how				
		Open Launchpad		
🔍 Find and Replace 🛛 🕞 Console			단 Bootcamp	

Select "Collection".





Set any name for the Collection, then navigate to the *Variables* tab and set "jwt_token" in the VARIABLE field, then click "Create" (nothing else needs to be set in this dialog).

CREATE A NEW COLLECTION				×
Name				
HashiCorp Vault / KMES Series 3 Int	egration Demo			
Description Authorization	Pre-request Scripts Test	s Variables •		
		n more about collection variables.		
VARIABLE		CURRENT VALUE		
ywt_token				
O Use variables to reuse values in different places. Work with the current value of a variable to prevent sharing sensitive values with your team. Learn more about variable values				
				Create

Right-click on the 3 dots in the new Collection and select "Add Request".

+ New Import	Runner 📮 🗸		器 My Workspace ~	°₊ Invite	<i>₹</i> & €	ğ û ♡ Sign In
Q Filter					No Environment	
History	Collections	APIs				
+ New Collection						
> 🗋 HashiCorp Vault	t / KMES Series 3 Integr					
	🗟 Share Collection					
	A∏ Rename					
	🖉 Edit					
	ြို့ Create a fork					
	🔤 Add Request					
	🛱 🛛 Add Folder					
	🗋 Duplicate			Open Launchpad		
	🕁 Export					
	Monitor Collectio					
	G Mock Collection					
	🛱 Publish Docs					
	🔟 Delete					
Q Find and Replace	5. Console					• • • •



Set any name for the request. The most important part is that the request is associated with the newly created Collection. Save the request.

SAVE REQUEST
Requests in Postman are saved in collections (a group of requests). Learn more about creating collections Request name
Demo KMES RESTful API POST Request
Request description (Optional)
Select a collection or folder to save to:
Q Search for a collection or folder
HashiCorp Vault / KMES Series 3 Integr + Create Folder
Cancel Save to HashiCorp Vault / KMES S

Expand the Collection folder and select the request that was just created. Set the request URL to "https://10.0.5.125:8081/kmes/v6/certificates/signing-requests" and change the request type to "POST" in the drop-down.

+ New Import Runner 🛱 🗸	器 My Workspace ~	الم الم الم	ණ ශ [™] ෯ \$\$ ♡ Sign In
Q Filter	POST Demo KMES RESTful API POST • + •••	No	o Environment 🔹 💿 🛬
History Collections APIs	Demo KMES RESTful API POST Request		Examples 0 🔻 🛛 BUILD 🥖 🗉
+ New Collection Trash	POST v https://10.0.5.125:8081/km	es/v6/certificates/signing-requests	Send 🔻 Save 🔻
HashiCorp Vault / KMES Series 3 Integration	POSI + Https://10.0.5.125.8081/kHt	es/vo/certificates/signing-requests	Send Save Save
1 request	Params Authorization Headers (7)	Body Pre-request Script Tests	Settings Cookies Code
GET Demo KMES RESTful API POST Request			
	KEY	VALUE	DESCRIPTION *** Bulk Edit
			Description
		Hit Send to get a response	
🔍 Find and Replace 🛛 🕞 Console			😌 Bootcamp 🚺 🖬 🖗 🕐



Switch to the Authorization tab, set the TYPE to "Bearer Token", then set the Token value to "{{jwt_token}}".



Navigate to the *Body* tab, select the "raw" bullet, then paste in the JSON shown below.

+ New Import Runner □ ✓	品 My Workspace ~	😵 🕉 竣 🗘 ♡ Sign In
Q Filter	POST Demo KMES RESTful API POST	No Environment 🔹 💿 🛬
History Collections APIs	Demo KMES RESTful API POST Request	Examples 0 🔻 🛛 BUILD 🧷 🗐
+ New Collection Trash	POST + https://10.0.5.125:8081/kmes/v6/certificates/signing-requests	Send 🔻 Save 👻
HashiCorp Vault / KMES Series 3 Integration 1 request	Params Authorization Headers (8) Body Pre-request Script	Tests Settings Cookies Code
OET Demo KMES RESTful API POST Request	none form-data x-www-form-urlencoded 🍥 raw 🔍 binary 🌗	GraphQL Text 🔻
	<pre>1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1</pre>	T
Q、Find and Replace 反 Console		& Bootcamp 🚺 🖬 🖕 ?



The request's body is where we specify that we want the KMES Series 3 to create a new X.509 PKI. The CA tree that we built on the KMES is defined, along with several other parameters. One of the most important parameters to notice is the "randomPassphrase" value set to "true". The "randomPassphrase" parameter ensures that the PKI request is issued as PKCS #12. Then, the passphrase for the PKCS #12 file will be stored in Vault.

Navigate to the *Pre-request Script* tab and paste in the Javascript below. This code is responsible for generating the JWT token used for authentication to the KMES, on-the-fly. You'll notice that the value for "hmac_key" in line 1 matches what was set in the JWT Options menu on the KMES.



After pasting in the Javascript, click on the settings icon in the upper-right area of the page. Click the *Globals* button.

MANAGE ENVIRONMENTS	<
Globals Import Add	



Set two new variables exactly as they are shown below, then click "Save".

MANAGE ENVIRONMENTS X					×	
	Global variables for a workspace are a set of variables that are always available within the scope of that workspace. They can be viewed and edited by anyone in that workspace. Learn more about globals					
Global	s					
	VARIABLE	INITIAL VALUE	CURRENT VALUE			
~	jwt_issuer	futurex	futurex			
	jwt_user		jwt_user			
	Use variables to reuse valu sharing sensitive values wi		k with the current value of a about variable values	variable to prevent		

The setup is complete. Now click "Save" and then send the POST request.

Piter Port Demo KMES RESTful API POSX + **** No Environment * © © History Collections APIs + New Collection Trach Port - https://10.05.125.6081/kmes/k6/centificates/signing-requests Send * Save * Params Authorization Headers (P) Body © Pre-request Script • Tests Settings Cookes: Code 1 // Left header + (* staffset123*; 1 request * Pre-request Script • Tests Settings Cookes: Code 1 // Left header + (* staffset123*; 2 (left header + (* staffset123*; 4 (* styp*; *)W** * Pre-request Script • Tests Settings Cookes: Code 1 // Left header + (* staffset123*; 9 (// List*; head; heades (P) Body • Pre-request Script • Tests Settings Cookes: Code 1 // List*; head; heades (P) Body • Pre-request Script • Tests Settings Cookes: Code 1 // List*; head; headers (P) Body • Pre-request Script • Tests Settings Settings 3 // Left payled a = { * * Settings Settings Settings 1 // Left payled a = { * * Settings Setings Set ings Set ings <th>+ New Import Runner 🗔 🗸</th> <th>믬 My Workspace ~ 🗍 Invite 🧿 🖉</th> <th>竣 û ♡ 🕂 Upgrade ▼</th>	+ New Import Runner 🗔 🗸	믬 My Workspace ~ 🗍 Invite 🧿 🖉	竣 û ♡ 🕂 Upgrade ▼
How Collection Trash Post HashCorp Vault / KMES Series 3 Integr Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookles Cookles			ironment 🔹 💿 축
Post https://10.05.1258081/kmest/defertificates/signing-requests Send New Post https://10.05.1258081/kmest/defertificates/signing-requests Send <	History Collections APIs	Demo KMES RESTful API POST Request	kamples 0 🔻 🛛 BUILD 🥖 🗐
Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookles Code 1 let hanck key = "safest123"; 1 let hanck key = "safest123"; 2 let hack key = "safest123"; 2 let hack key = "safest123"; 3 alg: "H\$256", and ge: "h\$256", 4 git: "h\$256", and ge: "h\$256", 5 }; 6 let payload = { 6 let payload = { "sub": pn.globals.get("jwt_user"), 7 "sub": pn.globals.get("jwt_user"), "sub": pn.globals.get("jwt_user"), 9 j; 11 function base64url(source) { et a gobal variable 11 function base64url(source) { et a condedSource; et a gobal variable 14 } eturn encodedSource; gold variable 14 } feurn encodedSource; gold variable 14 ;	+ New Collection Trash	POCT - https://10.0.5.125/x0x1/kmac/v/6/certificates/signing-requests	Sand y Sava y
<pre>2 let header = { 3</pre>		Params Authorization Headers (9) Body Pre-request Script Tests	Settings Cookies Code
7 "status": "Success" 8 }		<pre>2 let header = { 3 "alg": "H5256", 4 "typ": "JWT" 5 }; 6 let payload = { 7 "iss": pm.globals.get('jwt_issuer'), 8 "sub": pm.globals.get('jwt_user"), 9 //'iat": Math.floor(Date.now() / 1000) 10 }; 11 function base64url(source) { 12 encodedSource = CryptoJS.enc.Base64.stringify(source).replace</pre>	and are run before the request is sent. Learn more about pre-request scripts SNIPPETS Get an environment variable Get a global variable Get a variable Set an environment variable Set a global variable Clear an environment variable ED 1056 ms 425 B Save Response •
🔾 Find and Replace 🕞 Console 👘 👘 👘 💾 👘 💾 🖓	्र Find and Replace जित्त Console		Browse

If the request is successful, the body of the response will contain the message "Successfully created new entry", as it does above.



Now if you log back in to the Vault UI and go to "secret/testpath/" you will see a new entry, with the ID values after "CommonNameHere" matching the "requestId" value in the body of the response in Postman.

Vault × +		-	• ×
$\leftarrow \rightarrow \mathbb{C}$ (i) localhost:8200/ui/vault/sec	rets/secret/list/testpath/	☆ 😋 🖴 🎓	B :
Secrets Access Policies	Tools	◉ Status v 🛛 🕞 v 🕻	•
<pre>« secrets < secret < testpath</pre>			
Secret Version 2			
Secrets Configuration			_
् <u>testpath</u> /	Tab to autocomplete	Create secret	: +
CommonNameHere-1CABD22001665A99	2		
(© 2020 HashiCorp Vault 1.5.4+ent Documentation		

This confirms that the PKCS #12 passphrase was successfully sent from the KMES Series 3 to Vault for storage.



APPENDIX A: XCEPTIONAL SUPPORT



In today's high-paced environment, we know you are looking for timely and effective resolutions for your mission-critical needs. That is why our Xceptional Support Team does whatever it takes to ensure you have the best experience and support possible. Every time. Guaranteed.

- 24x7x365 mission critical support
- Level 1 to level 3 support
- Extremely knowledgeable subject matter experts

At Futurex, we strive to supply you with the latest data encryption innovations as well as our best-in-class support services. Our Xceptional Support Team goes above and beyond to meet your needs and provide you with exclusive services that you cannot find anywhere else in the industry.

- Technical Services
- Onsite Training
- Virtual Training
- Customized Consulting
- Customized Software Solutions
- Secure Key Generation, Printing, and Mailing
- Remote Key Injection
- Certificate Authority Services

Toll-Free: 1-800-251-5112

E-mail: support@futurex.com



ENGINEERING CAMPUS

864 Old Boerne Road Bulverde, Texas, USA 78163 Phone: +1 830-980-9782 +1 830-438-8782 E-mail: info@futurex.com XCEPTIONAL SUPPORT 24x7x365 Toll-Free: 1-800-251-5112 E-mail: support@futurex.com SOLUTIONS ARCHITECT E-mail: solutions@futurex.com