

Secure Storage for Third Party Applications

Table 1: Feature History Table

Feature Name	Release	Description
Secure Storage for Third Party Applications	Release 7.4.1	This release introduces *secure vault*, a secure storage for keys, user credentials, and other security-related information for third-party applications. With this functionality, non-native and third-party applications running on the router will now be able to boot securely. This functionality provides significant benefit for Datacenters that run a mix of native and non-native applications.

As we move away from static infrastructure defined by data centers that consisted of high trust networks, towards multiple cloud and private data centers that have blurred perimeters, the definition and requirement of security has changed.

Network applications need to securely store and provide access control to secrets that include, but are not limited to, API keys, tokens, SSH/TLS credentials, routing protocol authentication keys, user credentials, certificates like X.509 certificates, and so on.

IOS-XR applications and modules, such as Type6, CEPKI, and Attestation use Trusted Anchor Module (TAm) based secure storage through the XR-TAm services layer. This layer provides entropy sources, key management, and SUDI handling in addition to the secure storage interfaces.

Non-native IOS-XR applications (also called Third-party applications or TPAs) that typically run as a docker applications, often require secrets, such as private keys or API tokens to be stored on the devices.

The TPAs have the following basic requirements for secure storage:

- Secure Storage: Stores the Key-Value pair type data of the TPA. These values must be stored in encrypted form in persistent storage.
- Access Control: Manages the access of the secrets through one or more authentication methods.
- Authenticated access: Supports authenticated access to the stored secrets, such as via a token-based authentication method.

To cater to these requirements, Cisco IOS XR Release 7.4.1 supports the Secure Vault feature.

Effective Cisco IOS XR Release 7.4.1, key-value pair based secure storage for TPAs is supported through the Vault Open Source Software (Vault OSS) front-end and middleware, with Cisco TAm services forming the storage backend. Vault OSS provides a multi-paradigm, API-driven approach to secrets management.

- Configuring Secure Vault, on page 2
- Configuring Secure Vault Address and Port, on page 2
- Profiles, on page 3
- Onboarding the TPAs As a Docker Instance, on page 3
- TPA Operations Using Secure Vault, on page 4
- Verifying Secure Vault Configuration, on page 4

Configuring Secure Vault

The Secure Vault process (svault) runs by default, but the feature and the service are **not** enabled by default. You must enable the secure vault server and service to allow the TPA to perform create, read, update, or delete (CRUD) operations with secrets.

Note You must have **admin** privileges to configure the secure vault feature.

- Use the **svault server enable** command to enable the service.
- Use the **svault server disable** command to stop the secure vault service. When the feature is disabled after enabling the same, the relevant keys are kept safe in Cisco TAm secure storage to restart the service without further user intervention.

If you re-enable the feature or restart the secure vault process, the process fetches the relevant keys from the Cisco TAm and the service is re-started and all the policies and path related configurations are re-played and re-configured automatically.

Configuring Secure Vault Address and Port

```
RP/0/RP0/CPU0:ios# configure terminal
RP/0/RP0/CPU0:ios(config)# svault server address ipv4 192.168.23.1
RP/0/RP0/CPU0:ios(config)# svault server port 8200
RP/0/RP0/CPU0:ios(config)# svault server enable
```

Note The secure vault (svault) server IP address must be a Class C IP address.

Note It is assumed that the client address is the next higher address than the server IP address.

Note you must make any changes to the svault server address or port while the svault server is in disabled state.

Profiles

Profiles are used to specify a virtual path for storing secrets, rules, and privileges to access the path. Profiles can be attached to TPAs.

Setting Up a TPA Profile

```
RP/0/RP0/CPU0:ios(config)#svault profile-name PROF1 auth-method token ttl 20
RP/0/RP0/CPU0:ios (config)#svault profile-name PROF1 global env-addr SVAULT_ADDR
RP/0/RP0/CPU0:ios (config)#svault profile-name PROF1 global env-path SVAULT_PATH
RP/0/RP0/CPU0:ios(config)#svault profile-name PROF1 global env-token SVAULT_TOKEN
RP/0/RP0/CPU0:ios(config)#svault profile-name PROF1 policy PROF1_POLICY preserve-data
RP/0/RP0/CPU0:ios(config)#svault profile-name PROF1 policy PROF1_POLICY path tpa1 delete
read update write
RP/0/RP0/CPU0:ios(config)#commit
```

Running Configuration

```
svault
  auth-method
   token
    ttl 20                          the generated token will be in force for 20 hours
  global                           this command sets the global parameters for the profile
 tpa_prof1
   env-addr SVAULT_ADDR            the IP address and port of svault server is imported
as SVAULT_ADDR in the TPA
   env-path SVAULT_PATH            the profile's path is imported as SVAULT_PATH in the
TPA
   env-token SVAULT_TOKEN          the token generated for this profile is imported as
SVAULT_TOKEN in the TPA
  policy PROF1_POLICY              defines the privileges and capabilities for tpa_prof1
   preserve-data                  preserves TPA data even if the configuration is removed

   path tpa_path1                  lets tpa_prof1 know that its data is stored only in
this path
    delete read update write
```

Onboarding the TPAs As a Docker Instance

App Manager is used to onboard the TPA as docker instances.

First, install the TPA using App Manager

```
RP/0/RP0/CPU0:ios(config)# appmgr package install rpm
/misc/disk1/centos-0.1.0-XR_7.3.1.x86_64.rpm
```

To activate the TPA with Secure Vault profile

```
RP/0/RP0/CPU0:ios(config)# appmgr
RP/0/RP0/CPU0:ios(config-appmgr)# application centos_tpa1
RP/0/RP0/CPU0:ios(config-application)# application centos_tpa1 activate type docker source
```

```
 centos docker-run-opts "-v=/var/run/netns/global-vrf:/var/run/netns/global-vrf
-cap-add=SYS_ADMIN -net=host" svault "tpa_prof1" docker-run-cmd "sleep 86400"
```

where:

- **centos_tpa1** is the application to be onboarded
- **docker** is the type of the container for application centos_tpa1
- **sleep** is the amount of time (in seconds) that docker can use to keep the TPA running
- **svault "tpa_prof1"** is the svault profile linked with the TPA.

For more information on docker commands, see Docker Command Reference.

To verify that a TPA is onboarded:

```
RP/0/RP0/CPU0:Svault)# show appmgr application name centos_tpa1 info summary
```

For further information about configuration, see the *Application Hosting Configuration Guide for Cisco NCS 540 Series Routers*.

TPA Operations Using Secure Vault

```
Router:vty> run
Router:vty> docker exec -it centos_tpa1 /bin/sh opens the docker shell
Router:vty> env
Router:vty> curl -v  -X PUT -H "X-Vault-Request: true" -H "X-Vault-Token: echo $SVAULT_TOKEN1"
 -d '{"k":"v"}' 'echo $SVAULT_ADDR1' 'echo $SVAULT_PATH1' write and update operation
```

where:

- **curl** is a command line tool for sending and receiving data that allows you to interact with the vault
- **PUT** can write or update and replace as directed
- **k** and **v** are the key-value pair you can store or update

Some other curl operations;

- To read data:

  ```
  curl -H "X-Vault-Request: true" -H "X-Vault-Token: `echo $SVAULT_TOKEN`" `echo
  $SVAULT_ADDR``echo $SVAULT_PATH`
  ```

- To delete data:

  ```
  curl -v -X DELETE -H "X-Vault-Request: true" -H "X-Vault-Token: `echo $SVAULT_TOKEN'"
  `echo $SVAULT_ADDR' `echo $SVAULT_PATH'
  ```

Verifying Secure Vault Configuration

```
RP/0/RP0/CPU0:ios# show svault status

Mon May 10 10:26:56.365 IST
Secure Vault Status:
--------------------
Server:Enabled (Note: Disabled when svault server not enabled)
Server Redundancy:Disabled (Note: Always Disabled in 7.4.1)
```

```
Server IP Address:192.168.10.10(Note: service address)
Server port:3000 (Note: service port)
Secure Server:Disabled (Note: Always disabled in 7.4.1)
Server Auth Methods:Token (Note: Always 'Token' in 7.4.1)
```

```
RP/0/RP0/CPU0:ios# show run svault

Mon May 10 10:26:22.444 IST
svault
server
  address ipv4 192.168.10.10
  port 3000
  enable
!
profile-name tpa_prof1
  auth-method
  token
  ttl 24
 !
!
global
  env-addr SVAULT_ADDR1
  env-path SVAULT_PATH1
  env-token SVAULT_TOKEN1
!
policy tpa_pol1
  path tpa_path1
    read write update delete
   !
  !
!
!
```