



Configure NetApp hardware-based encryption

ONTAP 9

NetApp
May 24, 2023

This PDF was generated from <https://docs.netapp.com/us-en/ontap/encryption-at-rest/support-storage-encryption-concept.html> on May 24, 2023. Always check docs.netapp.com for the latest.

Table of Contents

- Configure NetApp hardware-based encryption 1
 - Configure NetApp hardware-based encryption overview..... 1
 - Hardware-based encryption workflow 2
 - Configure external key management..... 3
 - Configure onboard key management..... 15
 - Assign a FIPS 140-2 authentication key to a FIPS drive 22
 - Enable cluster-wide FIPS-compliant mode for KMIP server connections 23

Configure NetApp hardware-based encryption

Configure NetApp hardware-based encryption overview

NetApp hardware-based encryption supports full-disk encryption (FDE) of data as it is written. The data cannot be read without an encryption key stored on the firmware. The encryption key, in turn, is accessible only to an authenticated node.

Understanding NetApp hardware-based encryption

A node authenticates itself to a self-encrypting drive using an authentication key retrieved from an external key management server or Onboard Key Manager:

- The external key management server is a third-party system in your storage environment that serves keys to nodes using the Key Management Interoperability Protocol (KMIP). It is a best practice to configure external key management servers on a different storage system from your data.
- The Onboard Key Manager is a built-in tool that serves authentication keys to nodes from the same storage system as your data.

You can use NetApp Volume Encryption with hardware-based encryption to “double encrypt” data on self-encrypting drives.



If an HA pair is using encrypting SAS or NVMe drives (SED, NSE, FIPS), you must follow the instructions in the topic [Returning a FIPS drive or SED to unprotected mode](#) for all drives within the HA pair prior to initializing the system (boot options 4 or 9). Failure to do this may result in future data loss if the drives are repurposed.

Supported self-encrypting drive types

Two types of self-encrypting drives are supported:

- Self-encrypting FIPS-certified SAS or NVMe drives are supported on all FAS and AFF systems. These drives, called *FIPS drives*, conform to the requirements of Federal Information Processing Standard Publication 140-2, level 2. The certified capabilities enable protections in addition to encryption, such as preventing denial-of-service attacks on the drive. FIPS drives cannot be mixed with other types of drives on the same node or HA pair.
- Beginning with ONTAP 9.6, self-encrypting NVMe drives that have not undergone FIPS testing are supported on AFF A800, A320, and later systems. These drives, called *SEDs*, offer the same encryption capabilities as FIPS drives, but can be mixed with non-encrypting drives on the same node or HA pair.
- All FIPS validated drives use a firmware cryptographic module that has been through FIPS validation. The FIPS drive cryptographic module does not use any keys that are generated outside of the drive (the authentication passphrase that is input to the drive is used by the drive’s firmware cryptographic module to obtain a key encryption key).



Non-encrypting drives are drives that are not SEDs or FIPS drives.

When to use external key management

Although it is less expensive and typically more convenient to use the onboard key manager, you should use

external key management if any of the following are true:

- Your organization's policy requires a key management solution that uses a FIPS 140-2 Level 2 (or higher) cryptographic module.
- You need a multi-cluster solution, with centralized management of encryption keys.
- Your business requires the added security of storing authentication keys on a system or in a location different from the data.

Support details

The following table shows important hardware encryption support details. See the Interoperability Matrix for the latest information about supported KMIP servers, storage systems, and disk shelves.

Resource or feature	Support details
Non-homogeneous disk sets	<ul style="list-style-type: none">• FIPS drives cannot be mixed with other types of drives on the same node or HA pair. Conforming HA pairs can coexist with non-conforming HA pairs in the same cluster.• SEDs can be mixed with non-encrypting drives on the same node or HA pair.
Drive type	<ul style="list-style-type: none">• FIPS drives can be SAS or NVMe drives.• SEDs must be NVMe drives.
10 Gb network interfaces	Beginning with ONTAP 9.3, KMIP key management configurations support 10 Gb network interfaces for communications with external key management servers.
Ports for communication with the key management server	Beginning with ONTAP 9.3, you can use any storage controller port for communication with the key management server. Otherwise, you should use port e0M for communication with key management servers. Depending on the storage controller model, certain network interfaces might not be available during the boot process for communication with key management servers.
MetroCluster (MCC)	<ul style="list-style-type: none">• NVMe drives support MCC.• SAS drives do not support MCC.

Related information

- [NetApp Hardware Universe](#)
- [NetApp Volume Encryption and NetApp Aggregate Encryption](#)

Hardware-based encryption workflow

You must configure key management services before the cluster can authenticate itself to the self-encrypting drive. You can use an external key management server or an onboard key manager.



Configure external key management

Configure external key management overview

You can use one or more external key management servers to secure the keys that the cluster uses to access encrypted data. An external key management server is a third-party system in your storage environment that serves keys to nodes using the Key Management Interoperability Protocol (KMIP).

For ONTAP 9.1 and earlier versions, node management LIFs must be assigned to ports that are configured with the node management role before you can use the external key manager.

NetApp Volume Encryption (NVE) can be implemented with Onboard Key Manager in ONTAP 9.1 and later. In ONTAP 9.3 and later, NVE can be implemented with external key management (KMIP) and Onboard Key Manager. Beginning in ONTAP 9.11.1, you can configure multiple external key managers in a cluster. See [Configure clustered key servers](#).

Collect network information in ONTAP 9.2 and earlier

If you are using ONTAP 9.2 or earlier, you should fill out the network configuration worksheet before enabling external key management.



Beginning with ONTAP 9.3, the system discovers all needed network information automatically.

Item	Notes	Value
Key management network interface name		
Key management network interface IP address	IP address of node management LIF, in IPv4 or IPv6 format	
Key management network interface IPv6 network prefix length	If you are using IPv6, the IPv6 network prefix length	
Key management network interface subnet mask		
Key management network interface gateway IP address		
IPv6 address for the cluster network interface	Required only if you are using IPv6 for the key management network interface	
Port number for each KMIP server	Optional. The port number must be the same for all KMIP servers. If you do not provide a port number, it defaults to port 5696, which is the Internet Assigned Numbers Authority (IANA) assigned port for KMIP.	
Key tag name	Optional. The key tag name is used to identify all keys belonging to a node. The default key tag name is the node name.	

Related information

[NetApp Technical Report 3954: NetApp Storage Encryption Preinstallation Requirements and Procedures for IBM Tivoli Lifetime Key Manager](#)

[NetApp Technical Report 4074: NetApp Storage Encryption Preinstallation Requirements and Procedures for SafeNet KeySecure](#)

Install SSL certificates on the cluster

The cluster and KMIP server use KMIP SSL certificates to verify each other's identity and establish an SSL connection. Before configuring the SSL connection with the KMIP server, you must install the KMIP client SSL certificates for the cluster, and the SSL public certificate for the root certificate authority (CA) of the KMIP server.

What you'll need

- The time must be synchronized on the server creating the certificates, the KMIP server, and the cluster.

- You must have obtained the public SSL KMIP client certificate for the cluster.
- You must have obtained the private key associated with the SSL KMIP client certificate for the cluster.

The SSL KMIP client certificate must not be password-protected.

- You must have obtained the SSL public certificate for the root certificate authority (CA) of the KMIP server.
- In a MetroCluster environment, you must install the KMIP SSL certificate on both clusters.



You can install the client and server certificates on the KMIP server before or after installing the certificates on the cluster.

About this task

In an HA pair, both nodes must use the same public and private KMIP SSL certificates. If you connect multiple HA pairs to the same KMIP server, all nodes in the HA pairs must use the same public and private KMIP SSL certificates.

Steps

1. Install the SSL KMIP client certificates for the cluster:

```
security certificate install -vserver admin_svm_name -type client
```

You are prompted to enter the SSL KMIP public and private certificates.

```
cluster1::> security certificate install -vserver cluster1 -type client
```

2. Install the SSL public certificate for the root certificate authority (CA) of the KMIP server:

```
security certificate install -vserver admin_svm_name -type server-ca
```

```
cluster1::> security certificate install -vserver cluster1 -type server-ca
```

Enable external key management in ONTAP 9.6 and later (HW-based)

You can use one or more KMIP servers to secure the keys the cluster uses to access encrypted data. You can connect up to four KMIP servers to a node. A minimum of two servers is recommended for redundancy and disaster recovery.

Beginning in ONTAP 9.11.1, you can add up to 3 secondary key servers per primary key server to create a clustered key server. For more information, see [Configure clustered external key servers](#).

Before you begin

- The KMIP SSL client and server certificates must have been installed.
- You must be a cluster administrator to perform this task.
- You must configure the MetroCluster environment before you configure an external key manager.
- In a MetroCluster environment, you must install the KMIP SSL certificate on both clusters.

Steps

1. Configure key manager connectivity for the cluster:

```
security key-manager external enable -vserver admin_SVM -key-servers  
host_name|IP_address:port,... -client-cert client_certificate -server-ca-cert  
server_CA_certificates
```



- The `security key-manager external enable` command replaces the `security key-manager setup` command. You can run the `security key-manager external modify` command to change the external key management configuration. For complete command syntax, see the man pages.
- In a MetroCluster environment, if you are configuring external key management for the admin SVM, you must repeat the `security key-manager external enable` command on the partner cluster.

The following command enables external key management for `cluster1` with three external key servers. The first key server is specified using its hostname and port, the second is specified using an IP address and the default port, and the third is specified using an IPv6 address and port:

```
cluster1::> security key-manager external enable -key-servers  
ks1.local:15696,10.0.0.10,[fd20:8b1e:b255:814e:32bd:f35c:832c:5a09]:1234  
-client-cert AdminVserverClientCert -server-ca-certs  
AdminVserverServerCaCert
```

2. Verify that all configured KMIP servers are connected:

```
security key-manager external show-status -node node_name -vserver SVM -key  
-server host_name|IP_address:port -key-server-status available|not-  
responding|unknown
```



The `security key-manager external show-status` command replaces the `security key-manager show -status` command. For complete command syntax, see the man page.


```
cluster1::> security key-manager external show-status
```

Node	Vserver	Key Server	Status

node1			
	cluster1		
		10.0.0.10:5696	available
		fd20:8b1e:b255:814e:32bd:f35c:832c:5a09:1234	available
		ks1.local:15696	available
node2			
	cluster1		
		10.0.0.10:5696	available
		fd20:8b1e:b255:814e:32bd:f35c:832c:5a09:1234	available
		ks1.local:15696	available

6 entries were displayed.

Enable external key management in ONTAP 9.5 and earlier

You can use one or more KMIP servers to secure the keys the cluster uses to access encrypted data. You can connect up to four KMIP servers to a node. A minimum of two servers is recommended for redundancy and disaster recovery.

What you'll need

- The KMIP SSL client and server certificates must have been installed.
- You must be a cluster administrator to perform this task.
- You must configure the MetroCluster environment before you configure an external key manager.
- In a MetroCluster environment, you must install the KMIP SSL certificate on both clusters.

About this task

ONTAP configures KMIP server connectivity for all nodes in the cluster.

Steps

1. Configure key manager connectivity for cluster nodes:

```
security key-manager setup
```

The key manager setup starts.



In a MetroCluster environment, you must run this command on both clusters.

2. Enter the appropriate response at each prompt.
3. Add a KMIP server:

```
security key-manager add -address key_management_server_ipaddress
```

```
cluster1::> security key-manager add -address 20.1.1.1
```



In a MetroCluster environment, you must run this command on both clusters.

4. Add an additional KMIP server for redundancy:

```
security key-manager add -address key_management_server_ipaddress
```

```
cluster1::> security key-manager add -address 20.1.1.2
```



In a MetroCluster environment, you must run this command on both clusters.

5. Verify that all configured KMIP servers are connected:

```
security key-manager show -status
```

For complete command syntax, see the man page.

```
cluster1::> security key-manager show -status
```

Node	Port	Registered Key Manager	Status
-----	----	-----	-----
cluster1-01	5696	20.1.1.1	available
cluster1-01	5696	20.1.1.2	available
cluster1-02	5696	20.1.1.1	available
cluster1-02	5696	20.1.1.2	available

Configure clustered external key servers

Beginning in ONTAP 9.11.1, you can configure connectivity to clustered external key management servers on an SVM. With clustered key servers, you can designate primary and secondary key servers on a SVM. When registering keys, ONTAP will first attempt to access a primary key server before sequentially attempting to access secondary servers until the operation completes successfully, preventing duplication of keys.

External key servers can be used for NSE, NVE, NAE, and SED keys. An SVM can support up to four primary external KMIP servers. Each primary server can support up to three secondary key servers.

Before you begin

- [KMIP key management is already enabled for the SVM.](#)
- This process only supports key servers that use KMIP. For a list of supported key servers, check the

- All nodes in the cluster must be running ONTAP 9.11.1 or later.
- The order of servers list arguments in the `-secondary-key-servers` parameter reflects the access order of the external key management (KMIP) servers.

Create a clustered key server

The configuration procedure depends on whether or not you have configured a primary key server.

Add primary and secondary key servers to an SVM

1. Confirm that no key management has been enabled for the cluster:
`security key-manager external show -vserver vservice_name`
If the SVM already has the maximum of four primary key servers enabled, you must remove one of the existing primary key servers before adding a new one.
2. Enable the primary key manager:
`security key-manager external enable -vserver vservice_name -key-servers server_ip -client-cert client_cert_name -server-ca-certs server_ca_cert_names`
3. Modify the primary key server to add secondary key servers. The `-secondary-key-servers` parameter accepts a comma-separated list of up to three key servers.
`security key-manager external modify-server -vserver vservice_name -key-servers primary_key_server -secondary-key-servers list_of_key_servers`

Add secondary key servers to an existing primary key server

1. Modify the primary key server to add secondary key servers. The `-secondary-key-servers` parameter accepts a comma-separated list of up to three key servers.
`security key-manager external modify-server -vserver vservice_name -key-servers primary_key_server -secondary-key-servers list_of_key_servers`
For more information about secondary key servers, see [Modifying secondary key servers](#).

Modify clustered key servers

You can modify external key servers clusters by changing the status (primary or secondary) of particular key servers, add and removing secondary key servers, or by changing the access order of secondary key servers.

Converting primary and secondary key servers

To convert a primary key server into a secondary key server, you must first remove it from the SVM with the `security key-manager external remove-servers` command.

To convert a secondary key server into a primary key server, you must first remove the secondary key server from its existing primary key server. See [Modifying secondary key servers](#). If you convert a secondary key server to a primary server while removing an existing key, attempting to add a new server before completing the removal and conversion can result in the the duplication of keys.

Modifying secondary key servers

Secondary key servers are managed with the `-secondary-key-servers` parameter of the `security key-manager external modify-server` command. The `-secondary-key-servers` parameter accepts a comma-separated list. The specified order of the secondary key servers in the list determines the

access sequence for the secondary key servers. The access order can be modified by running the command `security key-manager external modify-server` with the secondary key servers entered in a different sequence.

To remove a secondary key server, the `-secondary-key-servers` arguments should include the key servers you want to keep while omitting the one to be removed. To remove all secondary key servers, use the argument `-`, signifying none.

For additional information, refer to the `security key-manager external` page in the [ONTAP command reference](#).

Create authentication keys in ONTAP 9.6 and later

You can use the `security key-manager key create` command to create the authentication keys for a node and store them on the configured KMIP servers.

What you'll need

You must be a cluster administrator to perform this task.

About this task

If your security setup requires you to use different keys for data authentication and FIPS 140-2 authentication, you should create a separate key for each. If that is not the case, you can use the same authentication key for FIPS compliance that you use for data access.

ONTAP creates authentication keys for all nodes in the cluster.

- This command is not supported when Onboard Key Manager is enabled. However, two authentication keys are created automatically when Onboard Key Manager is enabled. The keys can be viewed with the following command:

```
security key-manager key query -key-type NSE-AK
```

- You receive a warning if the configured key management servers are already storing more than 128 authentication keys.
- You can use the `security key-manager key delete` command to delete any unused keys. The `security key-manager key delete` command fails if the given key is currently in use by ONTAP. (You must have privileges greater than “admin” to use this command.)



In a MetroCluster environment, before you delete a key, you must make sure that the key is not in use on the partner cluster. You can use the following commands on the partner cluster to check that the key is not in use:

- `storage encryption disk show -data-key-id key-id`
- `storage encryption disk show -fips-key-id key-id`

Steps

1. Create the authentication keys for cluster nodes:

```
security key-manager key create -key-tag passphrase_label -prompt-for-key  
true|false
```



```
cluster1::> security key-manager key query
      Vserver: cluster1
      Key Manager: external
      Node: node1
```

Key Tag	Key Type	Restored
node1	NSE-AK	yes
Key ID: 000000000000000000002000000000001000c11b3863f78c2273343d7ec5a67762e00000000 00000000		
node1	NSE-AK	yes
Key ID: 000000000000000000002000000000001006f4e2513353a674305872a4c9f3bf79700000000 00000000		

```
      Vserver: cluster1
      Key Manager: external
      Node: node2
```

Key Tag	Key Type	Restored
node2	NSE-AK	yes
Key ID: 000000000000000000002000000000001000c11b3863f78c2273343d7ec5a67762e00000000 00000000		
node2	NSE-AK	yes
Key ID: 000000000000000000002000000000001006f4e2513353a674305872a4c9f3bf79700000000 00000000		

Create authentication keys in ONTAP 9.5 and earlier

You can use the `security key-manager create-key` command to create the authentication keys for a node and store them on the configured KMIP servers.

What you'll need

You must be a cluster administrator to perform this task.

About this task

If your security setup requires you to use different keys for data authentication and FIPS 140-2 authentication, you should create a separate key for each. If that is not the case, you can use the same authentication key for FIPS compliance that you use for data access.

ONTAP creates authentication keys for all nodes in the cluster.

- This command is not supported when onboard key management is enabled.
- You receive a warning if the configured key management servers are already storing more than 128 authentication keys.

You can use the key management server software to delete any unused keys, then run the command again.

Steps

1. Create the authentication keys for cluster nodes:

```
security key-manager create-key
```

For complete command syntax, see the man page for the command.



The key ID displayed in the output is an identifier used to refer to the authentication key. It is not the actual authentication key or the data encryption key.

The following example creates the authentication keys for `cluster1`:

```
cluster1::> security key-manager create-key
(security key-manager create-key)
Verifying requirements...

Node: cluster1-01
Creating authentication key...
Authentication key creation successful.
Key ID: F1CB30AFF1CB30B0010100000000000A68B167F92DD54196297159B5968923C

Node: cluster1-01
Key manager restore operation initialized.
Successfully restored key information.

Node: cluster1-02
Key manager restore operation initialized.
Successfully restored key information.
```

2. Verify that the authentication keys have been created:

```
security key-manager query
```

For complete command syntax, see the man page.

The following example verifies that authentication keys have been created for `cluster1`:

```
cluster1::> security key-manager query

(security key-manager query)

      Node: cluster1-01
    Key Manager: 20.1.1.1
  Server Status: available

Key Tag          Key Type  Restored
-----
cluster1-01      NSE-AK    yes
    Key ID:
F1CB30AFF1CB30B0010100000000000A68B167F92DD54196297159B5968923C

      Node: cluster1-02
    Key Manager: 20.1.1.1
  Server Status: available

Key Tag          Key Type  Restored
-----
cluster1-02      NSE-AK    yes
    Key ID:
F1CB30AFF1CB30B0010100000000000A68B167F92DD54196297159B5968923C
```

Assign a data authentication key to a FIPS drive or SED (external key management)

You can use the `storage encryption disk modify` command to assign a data authentication key to a FIPS drive or SED. Cluster nodes use this key to lock or unlock encrypted data on the drive.

About this task

A self-encrypting drive is protected from unauthorized access only if its authentication key ID is set to a non-default value. The manufacturer secure ID (MSID), which has key ID 0x0, is the standard default value for SAS drives. For NVMe drives, the standard default value is a null key, represented as a blank key ID. When you assign the key ID to a self-encrypting drive, the system changes its authentication key ID to a non-default value.

This procedure is not disruptive.

Before you begin

You must be a cluster administrator to perform this task.

Steps

1. Assign a data authentication key to a FIPS drive or SED:

```
storage encryption disk modify -disk disk_ID -data-key-id key_ID
```


For complete command syntax, see the man page for the command.



You can use the `security key-manager query -key-type NSE-AK` command to view key IDs.

```
cluster1::> storage encryption disk modify -disk 0.10.* -data-key-id
F1CB30AFF1CB30B0010100000000000A68B167F92DD54196297159B5968923C
```

```
Info: Starting modify on 14 disks.
      View the status of the operation by using the
      storage encryption disk show-status command.
```

2. Verify that the authentication keys have been assigned:

```
storage encryption disk show
```

For complete command syntax, see the man page.

```
cluster1::> storage encryption disk show
Disk      Mode Data Key ID
-----
-----
0.0.0     data
F1CB30AFF1CB30B0010100000000000A68B167F92DD54196297159B5968923C
0.0.1     data
F1CB30AFF1CB30B0010100000000000A68B167F92DD54196297159B5968923C
[...]
```

Configure onboard key management

Enable onboard key management in ONTAP 9.6 and later

You can use the Onboard Key Manager to authenticate cluster nodes to a FIPS drive or SED. The Onboard Key Manager is a built-in tool that serves authentication keys to nodes from the same storage system as your data. The Onboard Key Manager is FIPS-140-2 level 1 compliant.

You can use the Onboard Key Manager to secure the keys that the cluster uses to access encrypted data. You must enable Onboard Key Manager on each cluster that accesses an encrypted volume or a self-encrypting disk.

What you'll need

- If you are using NSE with an external key management (KMIP) server, you must have deleted the external key manager database.

Transitioning to onboard key management from external key management

- You must be a cluster administrator to perform this task.
- You must configure the MetroCluster environment before the Onboard key manager is configured.

About this task

You must run the `security key-manager onboard enable` command each time you add a node to the cluster. In MetroCluster configurations, you must run `security key-manager onboard enable` on the local cluster first, then run `security key-manager onboard sync` on the remote cluster, using the same passphrase on each.

By default, you are not required to enter the key manager passphrase when a node is rebooted. Except in MetroCluster, you can use the `cc-mode-enabled=yes` option to require that users enter the passphrase after a reboot.

When the Onboard Key Manager is enabled in Common Criteria mode (`cc-mode-enabled=yes`), system behavior is changed in the following ways:

- The system monitors for consecutive failed cluster passphrase attempts when operating in Common Criteria mode.

If NetApp Storage Encryption (NSE) is enabled and you fail to enter the correct cluster passphrase at boot, the system cannot authenticate to its drives and automatically reboots. To correct this, you must enter the correct cluster passphrase at the boot prompt. Once booted, the system allows up to 5 consecutive attempts to correctly enter the cluster passphrase in a 24-hour period for any command that requires the cluster passphrase as a parameter. If the limit is reached (for example, you have failed to correctly enter the cluster passphrase 5 times in a row) then you must either wait for the 24-hour timeout period to elapse, or you must reboot the node, in order to reset the limit.

- System image updates use the NetApp RSA-3072 code signing certificate together with SHA-384 code signed digests to check the image integrity instead of the usual NetApp RSA-2048 code signing certificate and SHA-256 code signed digests.

The upgrade command verifies that the image contents have not been altered or corrupted by checking various digital signatures. The image update process proceeds to the next step if validation succeeds; otherwise, the image update fails. See the “cluster image” man page for information concerning system updates.

The Onboard Key Manager stores keys in volatile memory. Volatile memory contents are cleared when the system is rebooted or halted. Under normal operating conditions, volatile memory contents will be cleared within 30s when a system is halted.

Steps

1. Start the key manager setup command:

```
security key-manager onboard enable -cc-mode-enabled yes|no
```



Set `cc-mode-enabled=yes` to require that users enter the key manager passphrase after a reboot. The `- cc-mode-enabled` option is not supported in MetroCluster configurations. The `security key-manager onboard enable` command replaces the `security key-manager setup` command.

The following example starts the key manager setup command on `cluster1` without requiring that the passphrase be entered after every reboot:

```
cluster1::> security key-manager onboard enable
```

```
Enter the cluster-wide passphrase for onboard key management in Vserver  
"cluster1"::    <32..256 ASCII characters long text>  
Reenter the cluster-wide passphrase:    <32..256 ASCII characters long  
text>
```

2. At the passphrase prompt, enter a passphrase between 32 and 256 characters, or for “cc-mode”, a passphrase between 64 and 256 characters.



If the specified “cc-mode” passphrase is less than 64 characters, there is a five-second delay before the key manager setup operation displays the passphrase prompt again.

3. At the passphrase confirmation prompt, reenter the passphrase.
4. Verify that the authentication keys have been created:

```
security key-manager key query -node node
```



The `security key-manager key query` command replaces the `security key-manager query key` command. For complete command syntax, see the man page.

The following example verifies that authentication keys have been created for `cluster1`:

Node: node1

Restored

yes

```
000000000000000000002000000000001000c11b3863f78c2273343d7ec5a67762e00000000
00000000
```

yes

```
00000000000000000200000000001006f4e2513353a674305872a4c9f3bf79700000000
00000000
```

Node: node2

Restored

yes

```
000000000000000002000000000001000c11b3863f78c2273343d7ec5a67762e00000000
00000000
```

yes

```
00000000000000000200000000001006f4e2513353a674305872a4c9f3bf79700000000
00000000
```

disk.

What you'll need

- If you are using NSE with an external key management (KMIP) server, you must have deleted the external key manager database.

Transitioning to onboard key management from external key management

- You must be a cluster administrator to perform this task.
- You must configure the MetroCluster environment before the Onboard Key Manager is configured.

About this task

You must run the `security key-manager setup` command each time you add a node to the cluster.

If you have a MetroCluster configuration, review these guidelines:

- In ONTAP 9.5, you must run `security key-manager setup` on the local cluster and `security key-manager setup -sync-metrocluster-config yes` on the remote cluster, using the same passphrase on each.
- Prior to ONTAP 9.5, you must run `security key-manager setup` on the local cluster, wait approximately 20 seconds, and then run `security key-manager setup` on the remote cluster, using the same passphrase on each.

By default, you are not required to enter the key manager passphrase when a node is rebooted. Beginning with ONTAP 9.4, you can use the `-enable-cc-mode yes` option to require that users enter the passphrase after a reboot.

For NVE, if you set `-enable-cc-mode yes`, volumes you create with the `volume create` and `volume move start` commands are automatically encrypted. For `volume create`, you need not specify `-encrypt true`. For `volume move start`, you need not specify `-encrypt-destination true`.



After a failed passphrase attempt, you must reboot the node again.

Steps

1. Start the key manager setup:

```
security key-manager setup -enable-cc-mode yes|no
```



Beginning with ONTAP 9.4, you can use the `-enable-cc-mode yes` option to require that users enter the key manager passphrase after a reboot. For NVE, if you set `-enable-cc-mode yes`, volumes you create with the `volume create` and `volume move start` commands are automatically encrypted.

The following example starts setting up the key manager on cluster1 without requiring that the passphrase be entered after every reboot:

• • •

- 



- Verify that keys are configured for all nodes:

```
security key-manager key show
```

for the complete command syntax, see the man page.

Key ID	Used By
--------	---------

After you finish

All key management information is automatically backed up to the replicated database (RDB) for the cluster.

Whenever you configure the Onboard Key Manager passphrase, you should also back up the information manually to a secure location outside the storage system for use in case of a disaster. See [Back up onboard key management information manually](#).

Assign a data authentication key to a FIPS drive or SED (onboard key management)

You can use the `storage encryption disk modify` command to assign a data authentication key to a FIPS drive or SED. Cluster nodes use this key to access data on the drive.

What you'll need

You must be a cluster administrator to perform this task.

About this task

A self-encrypting drive is protected from unauthorized access only if its authentication key ID is set to a non-default value. The manufacturer secure ID (MSID), which has key ID 0x0, is the standard default value for SAS drives. For NVMe drives, the standard default value is a null key, represented as a blank key ID. When you assign the key ID to a self-encrypting drive, the system changes its authentication key ID to a non-default value.

Steps

1. Assign a data authentication key to a FIPS drive or SED:

```
storage encryption disk modify -disk disk_ID -data-key-id key_ID
```

For complete command syntax, see the man page for the command.



You can use the `security key-manager key query -key-type NSE-AK` command to view key IDs.

```
cluster1::> storage encryption disk modify -disk 0.10.* -data-key-id  
0000000000000000000020000000000010019215b9738bc7b43d4698c80246db1f4
```

```
Info: Starting modify on 14 disks.  
View the status of the operation by using the  
storage encryption disk show-status command.
```

2. Verify that the authentication keys have been assigned:

```
storage encryption disk show
```

For complete command syntax, see the man page.

```
cluster1::> storage encryption disk show
Disk      Mode Data Key ID
-----
-----
0.0.0     data
00000000000000000000200000000000010019215b9738bc7b43d4698c80246db1f4
0.0.1     data
00000000000000000000200000000000010059851742AF2703FC91369B7DB47C4722
[...]
```

Assign a FIPS 140-2 authentication key to a FIPS drive

You can use the `storage encryption disk modify` command with the `-fips-key-id` option to assign a FIPS 140-2 authentication key to a FIPS drive. Cluster nodes use this key for drive operations other than data access, such as preventing denial-of-service attacks on the drive.

About this task

Your security setup may require you to use different keys for data authentication and FIPS 140-2 authentication. If that is not the case, you can use the same authentication key for FIPS compliance that you use for data access.

This procedure is not disruptive.

Before you begin

The drive firmware must support FIPS 140-2 compliance. The [NetApp Interoperability Matrix Tool](#) contains information about supported drive firmware versions.

Steps

1. You must first ensure you have assigned a data authentication key. This can be done with using an [external key manager](#) or an [onboard key manager](#). Verify the key is assigned with the command `storage encryption disk show`.
2. Assign a FIPS 140-2 authentication key to SEDs:

```
storage encryption disk modify -disk disk_id -fips-key-id
fips_authentication_key_id
```

You can use the `security key-manager query` command to view key IDs.

```
cluster1::> storage encryption disk modify -disk 2.10.* -fips-key-id
6A1E21D8000000000100000000000005A1FB4EE8F62FD6D8AE6754C9019F35A
```

```
Info: Starting modify on 14 disks.
      View the status of the operation by using the
      storage encryption disk show-status command.
```


3. Verify that the authentication key has been assigned:

```
storage encryption disk show -fips
```

For complete command syntax, see the man page.

```
cluster1::> storage encryption disk show -fips
Disk      Mode FIPS-Compliance Key ID
-----
2.10.0    full
6A1E21D8000000000100000000000005A1FB4EE8F62FD6D8AE6754C9019F35A
2.10.1    full
6A1E21D8000000000100000000000005A1FB4EE8F62FD6D8AE6754C9019F35A
[...]
```

Enable cluster-wide FIPS-compliant mode for KMIP server connections

You can use the `security config modify` command with the `-is-fips-enabled` option to enable cluster-wide FIPS-compliant mode for data in flight. Doing so forces the cluster to use OpenSSL in FIPS mode when connecting to KMIP servers.

Before you begin

- The storage controller must be configured in FIPS-compliant mode.
- All KMIP servers must support TLSv1.2. The system requires TLSv1.2 to complete the connection to the KMIP server when cluster-wide FIPS-compliant mode is enabled.

About this task

When you enable cluster-wide FIPS-compliant mode, the cluster will automatically use only TLS1.2 and FIPS-validated cipher suites. Cluster-wide FIPS-compliant mode is disabled by default.

You must reboot cluster nodes manually after modifying the cluster-wide security configuration.

Steps

1. Set the privilege level to advanced:

```
set -privilege advanced
```

2. Verify that TLSv1.2 is supported:

```
security config show -supported-protocols
```

For complete command syntax, see the man page.

```
cluster1::> security config show
```

	Cluster		Cluster
Security			
Interface	FIPS Mode	Supported Protocols	Supported Ciphers Config
Ready			
-----	-----	-----	-----

SSL	false	TLSv1.2, TLSv1.1, TLSv1	ALL:!LOW: !aNULL:!EXP: !eNULL
			yes

3. Enable cluster-wide FIPS-compliant mode:

```
security config modify -is-fips-enabled true -interface SSL
```

For complete command syntax, see the man page.

4. Reboot cluster nodes manually.

5. Verify that cluster-wide FIPS-compliant mode is enabled:

```
security config show
```

```
cluster1::> security config show
```

	Cluster		Cluster
Security			
Interface	FIPS Mode	Supported Protocols	Supported Ciphers Config
Ready			
-----	-----	-----	-----

SSL	true	TLSv1.2, TLSv1.1	ALL:!LOW: !aNULL:!EXP: !eNULL:!RC4
			yes

Copyright information

Copyright © 2023 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.