Fsas Technologies Inc.

P3AG-6042-03ENZ0

ETERNUS AB series All-Flash Arrays, ETERNUS HB series Hybrid Arrays

Security Hardening Guide for SANtricity

Guidelines for Secure Deployment of SANtricity

Table of Contents

List of Figures

List of Tables

Preface

This security hardening guide provides guidance to help organizations deploy SANtricity 11.70.4 and later to meet prescribed security objectives for information system confidentiality, integrity, and availability.

Third Edition
March 2025

Trademarks

Third-party trademark information related to this product is available at:
https://www.fujitsu.com/global/products/computing/storage/eternus/trademarks.html

Trademark symbols such as ™ and ® are omitted in this document.

About This Manual

Intended Audience

This manual is intended for system administrators who configure and manage operations of the ETERNUS AB/HB, or field engineers who perform maintenance. Refer to this manual as required.

Related Information and Documents

The latest information for the ETERNUS AB/HB is available at:
https://www.fujitsu.com/global/support/products/computing/storage/manuals-list.html

Document Conventions

Notice Symbols

The following notice symbols are used in this manual:

Caution	Indicates information that you need to observe when using the ETERNUS AB/HB. Make sure to read the information.
Note	Indicates information and suggestions that supplement the descriptions included in this manual.

1. Introduction

The evolution of the current threat landscape presents an organization with unique challenges for protecting its most valuable assets: data and information. The advanced and dynamic threats and vulnerabilities that organizations face are ever increasing in sophistication. Coupled with an increase in the effectiveness of obfuscation and reconnaissance techniques on the part of potential intruders, system managers must address the security of data and information in a proactive manner. This guide seeks to assist operators and administrators in that task by leveraging the confidentiality, integrity, and availability integral to our solution.

2. Local Storage Administrator Accounts

Roles

With role-based access control (RBAC), local users have access to only the systems and options that are required for their job roles and functions. The RBAC solution in SANtricity limits users' administrative access to the level granted for their defined role, which allows administrators to manage local users by assigned role. SANtricity provides several predefined roles. The local user accounts are static, and the assigned roles cannot be modified. Table 1 lists the predefined roles in SANtricity.

Table 1 Predefined roles for local users

Role	Brief Description
Admin	Top-level administrative account. This is the only role that allows the user to change the passwords of any local users and run any command supported by the storage system.
Security	This role allows the user to modify the security configuration on the storage system, including the ability to view audit logs, configure a secure syslog server, set LDAP/LDAPS server connections, and manage certificates. This role does not provide write access to storage system properties like pool and volume creation/deletion, but it does have read access. It also has privileges to enable/disable SYMbol access to the storage system.
Storage	This role has full read/write access to the storage system properties, but with no access to perform any security configuration functions.
Support	This role has access to all hardware resources on the storage system, failure data, MEL/audit log, and CFW upgrades.
Monitor	This role gives read-only access to all storage system properties. This user cannot view the security configuration.

Table 2 lists the predefined users and the mapped roles in SANtricity

Table 2 Local user-to-roles mapping

Local User	Mapped Roles
Admin	Root admin, security admin, storage admin, support admin, monitor
Security	Security admin, monitor
Storage	Storage admin, support admin, monitor
Support	Support admin, monitor
Monitor	Monitor

SANtricity also supports the Lightweight Directory Application Protocol (LDAP/LDAPS) and Active Directory. With LDAP/Active Directory user accounts, administrators can create, modify, or delete custom access control roles, and they can specify account restrictions for specific users.

Caution

It is recommended to configure Secure LDAP with TLS (LDAPS) for added security.

Log-In and Password Parameters

An effective security posture adheres to established organizational policies, guidelines, and any governance or standards that apply to the organization. Examples of these requirements include password-length requirements, handling failed login attempts, and automatic inactive logout of such accounts. The SANtricity solution provides features and functions to address these security constructs.

Configuring Password Policies to Enforce a Minimum Number of Digits (1–30)

It is recommended to increase the minimum number of digits for the password length. The default is eight. This setting can be changed by using the REST API or System Manager.

REST API

To change the minimum password length using, use the following REST API:

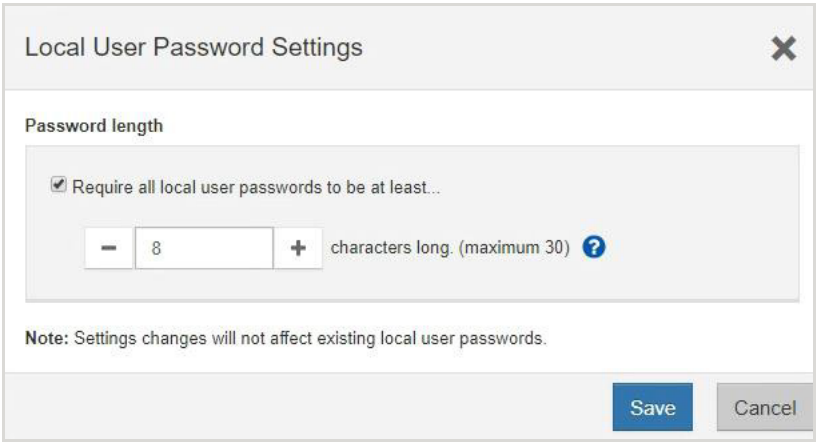
- API
  `Administration > POST /storage-systems/{system-id}/local-users/password-length`
- URL parameter
  `system-id`
- POST body

```
{
 "minimumPasswordLength": "length-in-characters" // numeric length
}
```

System Manager

To change the minimum password length using System Manager, go to Settings > Access Management > View/Edit Settings window, as shown in Figure 1.

Figure 1 System Manager window for changing the minimum password length



Configuring Lockout Settings Due to Failed Login Attempts

Lockout Mode (IP Address Versus User Name) for Failed Login Attempt

By default, SANtricity tracks the IP address of the web clients that fail to log in to the storage system. With this setting, an attacker who has exceeded the maximum number of attempts allowed and is locked out, can go on a different host that has a different IP address, and attempt again. When changed to User Lockout mode, SANtricity locks out the user name after the maximum number of login attempts have been reached.

Caution

It is recommended to change the lockout mode to user-based instead of IP-address-based (default). This setting can be changed by using the REST API, as shown below.

Lockout Time (In Minutes) After the Maximum Login Attempts Have Been Reached

The `lockoutTime` refers to the number of minutes the user is not able to log in to his/her account. The default value is 10 minutes.

Caution

It is recommended to leave the lockout time as 10 minutes (default).

Maximum Log-in Attempts Before the Account is Locked Out

The `maximumLoginAttempts` refers to the maximum number of attempts that the user is allowed to try before his/her account gets locked out. The default value is six. For example, if the value is set to six, after having entered six failed attempts, the account is locked out. The seventh log-in attempt might contain correct credentials, but SANtricity won't allow access until the user logs in again with the correct credentials after the `lockoutTime` period expires.

There can be up to two controllers in a storage array; the settings above apply to the entire storage array. However, each controller manages the lockout separately, and the accounting is not shared among the two controllers. Because the user can try to gain access to the storage array through either controller A or controller B (by providing either controller's IP address), technically, the user is allowed to have up to two times the value set in `maximumLoginAttempts`.

Caution

It is recommended to reduce the maximum number of log-in attempts from six (default) to five.
This means on the sixth failed login attempt, the account is locked out.

To change the setting for lockout mode, lockout time, and maximum log-in attempts, use the following REST API:

- API
  `Administration > POST /storage-systems/{system-id}/settings/lockout`
- URL parameter
  `system-id`
- POST body

```
{
"lockoutMode": "mode-to-use", // either "user" or "ip"
"lockoutTime": "lockout-minutes",
"maximumLoginAttempts": "attempts-before-lockout"
}
```

Defining the Account Inactive Limit

It is recommended to reduce the session inactivity period from 30 minutes (default) to 15 minutes. This setting can be changed by using the REST API or System Manager.

REST API

To change the inactive period setting, use the following REST API:

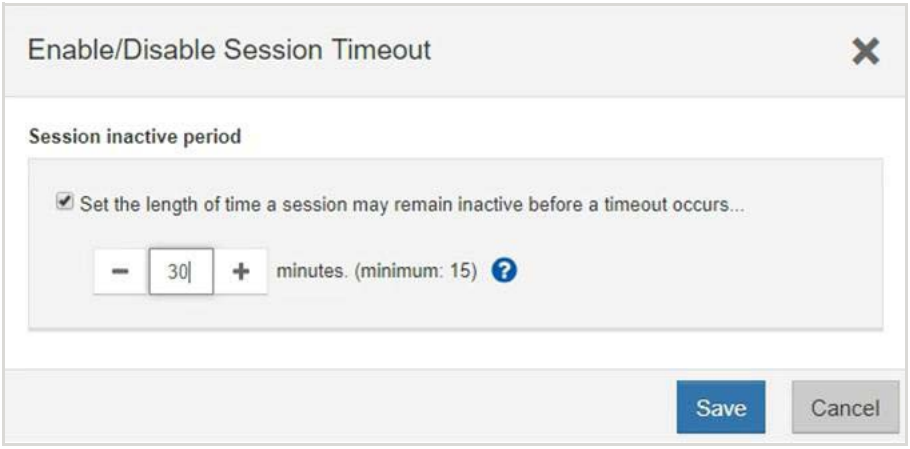
- API
  `Administration > POST /storage-systems/{system-id}/settings/sessions`
- URL parameter
  `system-id`
- POST body

```
{
"sessionInactivePeriod": "value-in-seconds"
}
```

System Manager

To change the inactive period setting using System Manager, go to Settings > System > Enable/Disable Session Timeout as shown in Figure 2.

Figure 2 System Manager window for changing the inactive period



SHA-512 Support

To enhance password security, SANtricity supports the SHA-2 password hash function and defaults to using SHA-512 for hashing newly created or changed passwords. Operators and administrators can also expire or lock accounts as needed.

3. System Administrator Methods

Console Access

Establishing secure access to the storage system console is a critical part for secure troubleshooting. The most common console access options are SSH, Telnet, and RSH. Of these, SSH is the most secure, industry-standard best practice for remote console access.

Console access is reserved for troubleshooting by our support engineers—it is available through:

- Serial USB cable
  After it is connected, the console is protected with a user login and password. The password is the same as the storage system password set by the user.
- SSH
  For security reasons, SSH is disabled by default. If there is a requirement or unique need for a secure remote access, SSH must be manually enabled. The user can manually enable or disable SSH through the secure CLI or through the SANtricity System Manager.

Caution

It is recommended to disable SSH at all times unless instructed by our support engineer. The following section describes how to disable and enable SSH through HTTP.

Disable and Enable SSH Through HTTP

To disable and enable SSH through HTTP, complete the following steps.

1. Allow remote login.

   Caution

   - Remote log-in users from outside the LAN should start an SSH session and change the settings on the controller.
   - For security reasons, enable the remote login for use only by technical support.

2. Select Hardware.
3. Select the Controllers & Components tab at the top. The graphic will change to show the controllers instead of the drives.
4. Click the controller for which you want to enable remote login.
   The controller's context menu appears.
5. Select [Enable remote login] checkbox.

Multifactor Authentication for SSH

SSH access can be hardened by forcing users to use an SSH key and the SSH password to access the system. Both a REST API and System Manager facility are in place to allow a system administrator to set up MFA.

To enable MFA for SSH, provide the following:

- An SSH public key
  This value is placed into the SSH server's `authorized_keys` file, which is the traditional means of providing SSH authentication for system users.
- A setting that forces SSH to require an SSH key and an SSH password
  After this setting is enabled, both the key and password are required to access the system.

Caution

Use of SSH keys is allowed independently of forcing a password to be provided. The only accounts that SSH keys are germane for are the `diag` and `eos` local accounts.

Enabling SSH key usage requires the entire content of the `authorized_keys` file to be presented each time there is a change to the file. When using the REST API, a GET call is made to get the content of the file, then a POST call is made to update the content of the file. This allows change, remove, update and delete operations to be performed on the file. If all SSH keys are removed, an empty `authorized_keys` file is written, and SSH will not allow authentication with a key.

For the user interface, a dialog is presented that provides an editor for the `authorized_keys` file and a switch to turn on MFA.

It is recommended to enable the most secure environment (SAML), which requires any system access to go through MFA at the UI layer. Changing the SSH settings would, therefore, require MFA through the UI to change the settings.

REST API

To enable MFA for SSH, use the following REST APIs:

- API
  Administration > GET/devmgr/v2/ssh/enable-ssh-mfa
- Get return data

```
{
 "mfaEnablement": [{
    "controllerRef": "controller-ref", // the controllerRef this setting belongs to
    "mfaEnabled": true // true or false
   }],
 "authorizedKeys": "authorized-keys-content"
}
```

- API
  `Administration > POST/devmgr/v2/ssh/enable-ssh-mfa`
- POST body

```
{
 "mfaEnablement": [{
    "controllerRef": "controller-ref", // the controllerRef this setting belongs to
    "mfaEnabled": true // true or false
   }],
 "authorizedKeys": "authorized-keys-content"
}
```

If the `mfaEnabled` parameter value is set to True, use an SSH key and a password to log into the system. Note that there is an array of settings, one per controller that should be specified in order to make the settings the same on both.

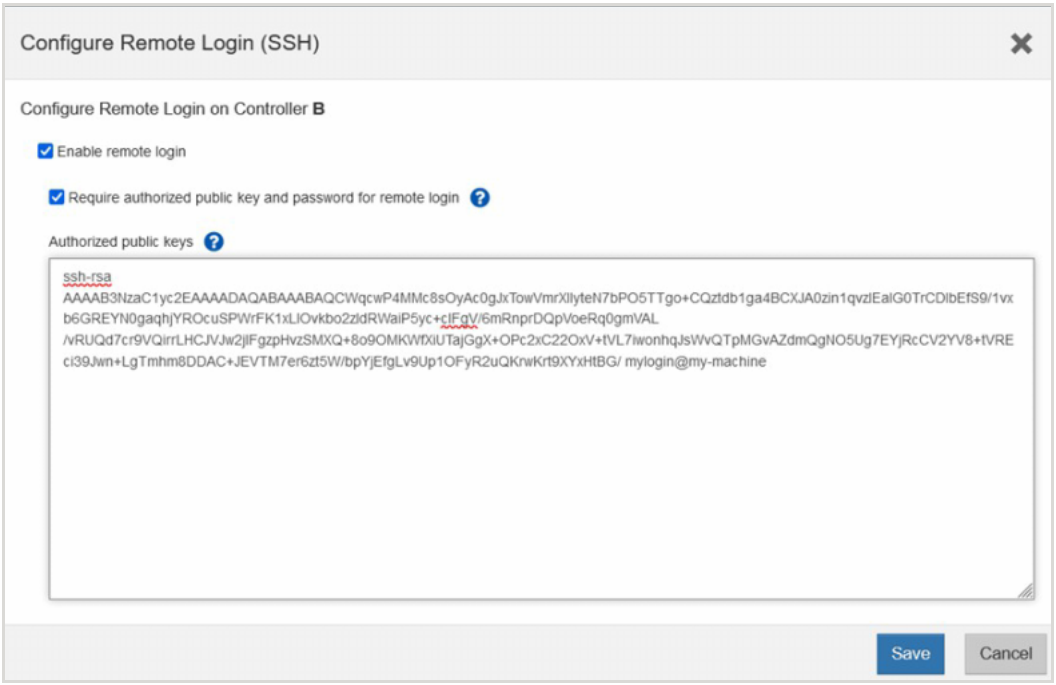
System Manager

System Manager contains a dialog box (see Figure 3) that allows a system administrator to manage SSH keys and manipulate the enable MFA setting.

Note

The Authorized Public Keys text box allows multiple SSH keys to be pasted in.

Figure 3 Enable SSH with password



Regenerating SSH Server Keys

Security conscious system administrators may want to regenerate the SSH server keys on a periodic basis. This facility is exposed through a REST API interface–no System Manager facility is exposed to perform this action.

After regenerating the SSH server keys, any clients that subsequently connect to the storage array are asked to reset their cached server keys.

To regenerate SSH server keys, use the following API:

- API
  `Administration > POST/devmgr/v2/ssh/regenerate-server-keys`

There is no POST body or API parameters.
The API will return a 200 response status for a successful key regeneration.

Caution

This API must be *called against both controllers* to regenerate both SSH server keys.

Command Line Access

The ETERNUS AB/HB series storage system has multiple secure command-line access points.

SANtricity Web Services REST API

To reach the REST API by using a web browser on the host where the proxy is installed, access the following URL.

```
https://localhost/devmgr/docs/#/
```

If this is the first time you are accessing the REST API, each type of browser displays the following:

- Chrome displays Your Connection is Not Private. Click Advanced to proceed to the website.
- Internet Explorer displays There is a Problem with This Website's Security Certificate. Click Continue to This Website (Not Recommended) to proceed to the website.
- Firefox displays Your Connection is Not Secure. Click the Advanced button and add an exception for the certificate to proceed to the website.

Caution

For security reasons, it is recommended to disable HTTP basic access authentication for REST API.

To disable basic authentication, run the following REST API:

- API
  `Administration > POST /storage-systems/{system-id}/settings/authentication`
- URL parameter
  `system-id`
- POST body

```
{
   "disableBasicAuthentication": true // true or false
}
```

Secure CLI

The secure SMcli allows an SMcli client to interact with a storage system through a secure HTTPS channel. It provides a thin HTTPS client that allows you to interoperate with storage systems by using traditional SANtricity SMcli grammar and command semantics, but with a secure protocol.

CLI Session Timeout

The default CLI session timeout is 30 minutes. The timeout is important to prevent stale sessions and session piggybacking.

Legacy Management Interface

A new REST API for storage management is introduced but the proprietary legacy management interface (SYMbol, port 2463) is kept enabled by default from the factory in order to prevent certain external management tools from not functioning.

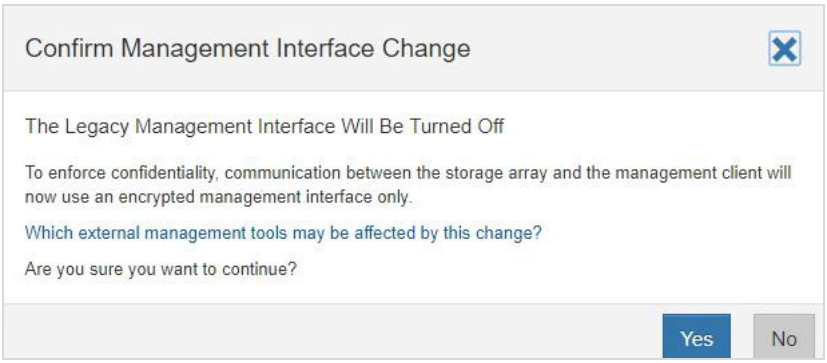
Tools that communicate directly with the legacy management interface, such as the SANtricity SMI-S Provider or OnCommand Insight, do not work unless the Legacy Management Interface setting is enabled. In addition, you cannot use legacy CLI commands or perform mirroring operations if this setting is disabled.

Future releases of SANtricity will remove the Legacy Management Interface access default enabled setting.

Caution

If you're not using the affected external management tools, it is recommended to change the array to a secure interface by disabling the legacy management interface. Prior to doing that, you must install the appropriate certificate authority (CA) root, intermediate, and signed server certificates on both storage array controllers. After those are installed, change the array management interface to the secure mode by using the System Manager (Settings > System > Additional Settings > Change Management Interface), as shown in Figure 4.

Figure 4 System Manager window for disabling the legacy management interface



JSON Web Token Access

A system that has SAML (MFA) enabled will by default not allow access to traditional automation tools. The REST API and secure CLI effectively become inoperable because the MFA workflow requires a redirect to an identity provider server for authentication. System Manager is the only entity that has been previously fully functional with SAML enabled.

Providing JSON Web Token (JWT) access while SAML is enabled provides MFA-like authentication. If SAML is enabled, token generation must be accomplished through System Manager which mandates a user is authenticated through MFA.

Tokens have properties that associate them to a specific user (including LDAP users), a set of permissions, and an expiration date. The security administrator can set the maximum duration that a token is issued for. It is not possible to generate a token with an expiration period longer than this value. After a token is generated and supplied to a user, it can be used to access the system both through the REST API and the secure CLI. Tokens do not have passwords, so they must be managed carefully. After an access token expires, authentication attempts will fail.

Note

It is not necessary to have SAML enabled to use web tokens, but SAML is recommended for the highest level of security.

Token Revocation

If a security administrator decides that a token or tokens are compromised, they can regenerate the keys used to sign the tokens. After the keys are reset, all issued tokens are immediately invalidated. It is recommended that new tokens have as short a time to live as possible to reduce the chance that they are compromised. Tokens are not tracked on the storage array, so revocation of individual tokens is not possible. You can manage tokens through the REST API or System manager.

REST API

If SAML is not enabled, it is possible to generate and manage tokens through the REST API. The following API calls are available.

To generate a new token, run the following REST API:

- API
  `Authentication > POST/devmgr/v2/access-token`
- POST body

```
{
  "duration": num-seconds-token-valid
}
```

- POST return

```
{
  "accessToken": "string-token-value",
  "duration": num-seconds-token-valid
}
```

To revoke access to all generated tokens, run the following REST API:

- API
  `Authentication > DELETE/devmgr/v2/access-token`

There is no body or return data. After calling this API, the keys are reset to new values. All the tokens previously issued are immediately invalidated.

To set the token generation parameters, run the following REST API:

- API
  `Authentication > POST/devmgr/v2/access-token/settings`
- POST body:

```
{
  "maxDuration": max-token-duration-in-seconds
}
```

There is no response body.

Token Usage Guidance

- REST API usage
  To use a token in a REST API request, add an HTTP header to your requests as follows:
  `Authorization: Bearer <access-token-value>`
- Secure CLI usage
  The following two arguments are available to use web tokens.
  - `-t access-token-value` –the token value on the command line
  - `-T access-token-file` –path to a file containing the token value

These parameters are mutually exclusive, so only one should be specified. Use of access tokens and user name and password should also not be used together.

Note

If no user name, password, or token is specified, the CLI will prompt you for an access token value on the command line.

- Through System Manager
  New tokens can be generated through the System Manager UI (see Figure 5). Users are asked to choose the duration of the token in days (see Figure 6). A new token is generated for the currently logged-in user. The issued token will have the user's permissions encoded in it and will be valid for the duration selected. After token generation, the new token is presented in a dialog box. You must copy the token text (see Figure 7) and store it in a secure place. There is not another opportunity to see the token's value after the dialog box is closed. Access tokens can also be revoked (see Figure 8).

Figure 5 Create access token

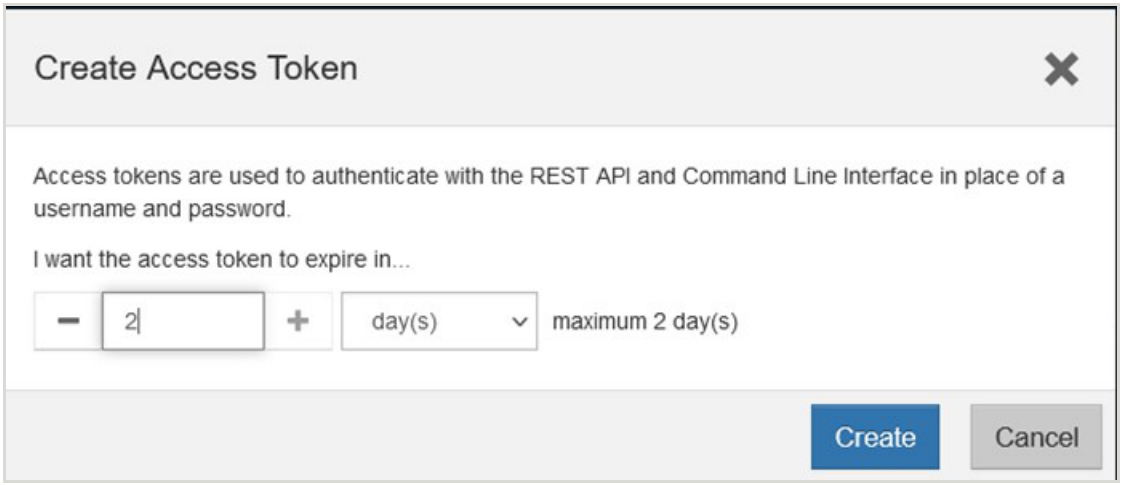


Figure 6 Change maximum token duration

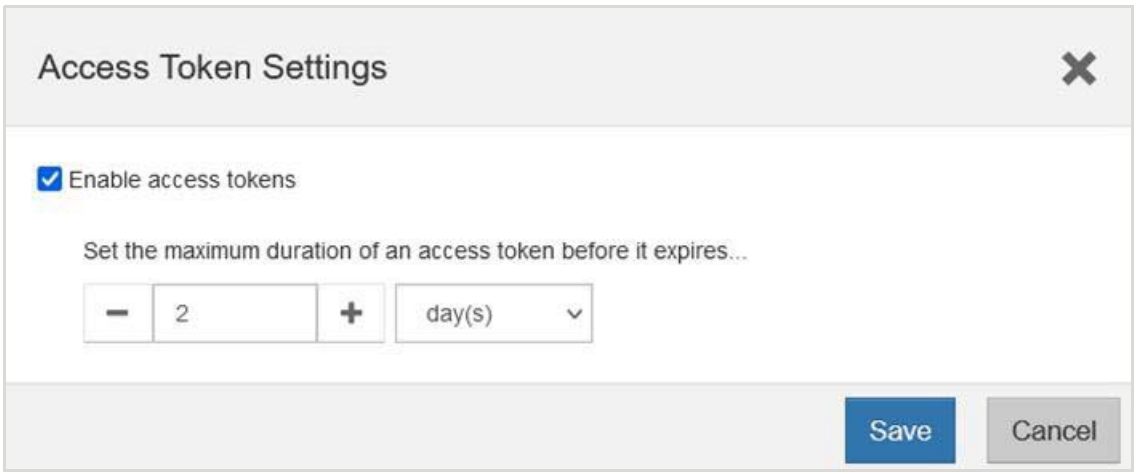


Figure 7 Copy access token

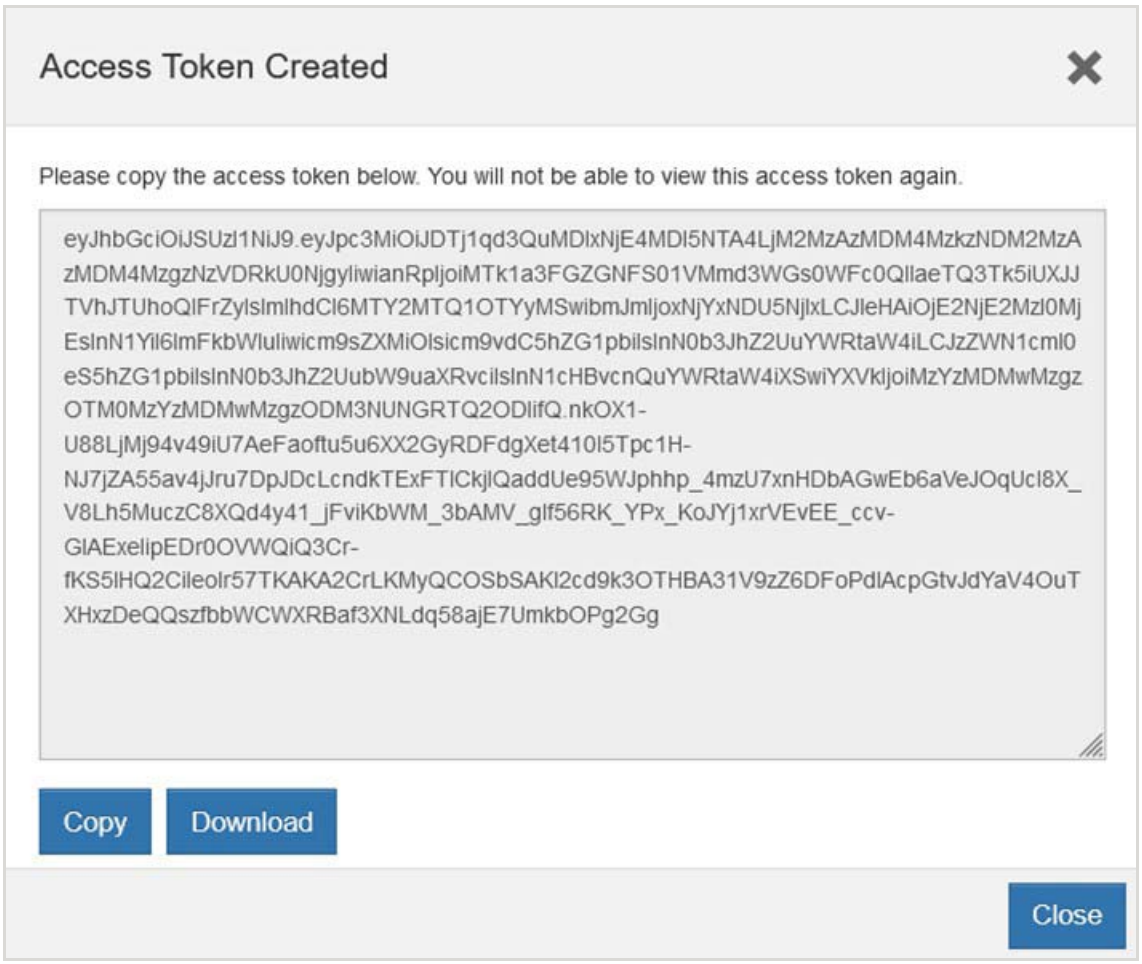
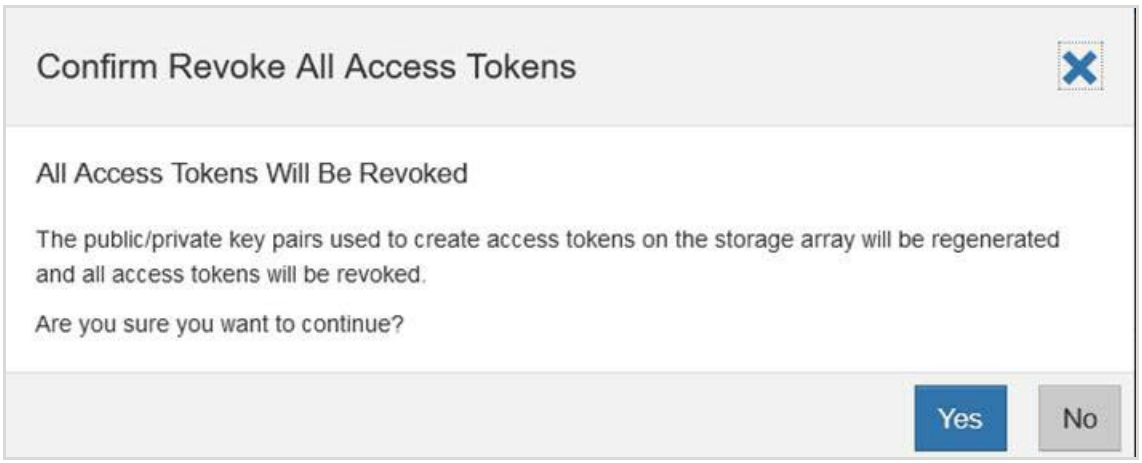


Figure 8 Revoke access tokens



Web Access

SANtricity System Manager

If the SANtricity administrator prefers to use a graphical interface instead of the CLI for accessing and managing the storage system, he/she must use SANtricity System Manager, which is included with SANtricity as a web service, enabled by default, and accessible by using a browser. Point the browser (using https://) to the host name (if using DNS) or to the IPv4 or IPv6 address.

If the controllers use a self-signed digital certificate, the browser might display a warning indicating that the certificate is not trusted. You can either acknowledge the risk to continue the access or install a CA-signed digital certificate.

Caution

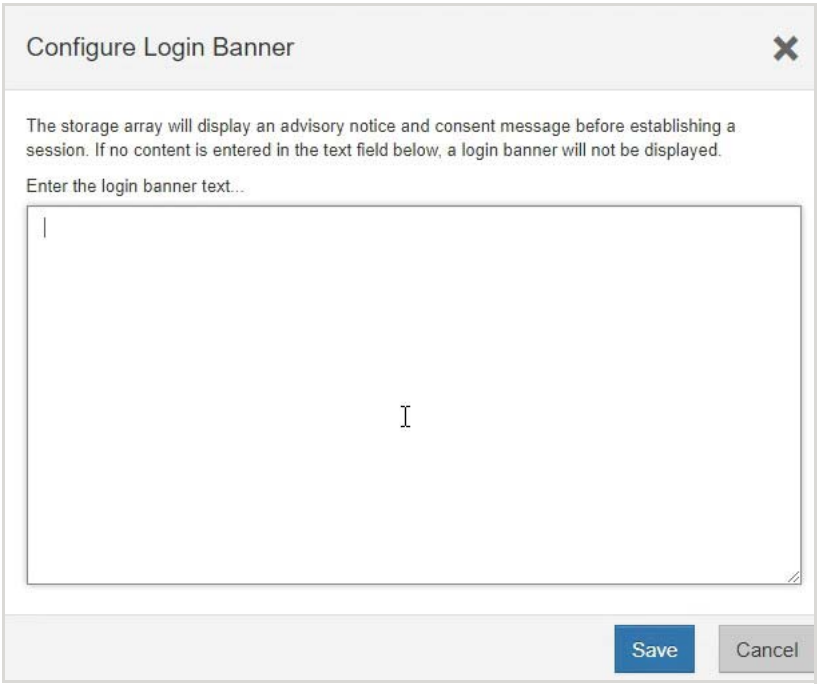
For security purposes, it is recommended to install a CA-signed digital certificate on the storage system for server authentication.

Security Assertion Markup Language (SAML) authentication is available as an option for SANtricity System Manager.

Log-in Banners

Log-in banners allow an organization to present any operators, administrators, and even miscreants with terms and conditions of acceptable use, and they indicate who is permitted access to the system. This approach is helpful for establishing expectations for access and use of the system. Figure 9 illustrates the System Manager (Settings > System > Configure Login Banner) window on how to configure the log-in banner.

Figure 9 System Manager window on how to configure the log-in banner



SAML Authentication for SANtricity System Manager

SAML 2.0 is an industry standard for sending authentication requests and user data securely between multiple systems. This standard allows many applications to use a single service to manage all user authentication and session management.

Multifactor authentication (MFA) requires the user to provide two or more items as proof of identity to be successfully authenticated. The separate pieces of evidence are typically at least two of the following types: knowledge (something the user knows, such as a password); possession (something the user has, such as a device that provides a changing code); or inherence (something the user is, such as biometrics, like a fingerprint). The specific type of evidence required is configured by the end-user organization's security team.

SAML is integrated into ETERNUS AB/HB series products, making it possible to communicate with an external system that can authenticate a user with multiple forms of authentication and then report the success or failure of the authentication to the SANtricity System Manager application. The external system can be configured to use single-factor, two-factor, or multifactor authentication. The external system also provides the ability to support single sign-on capabilities with other applications.

After SAML is configured on the storage system, logging into SANtricity System Manager is possible only through a configured Identity Provider (IdP). When users attempt to access SANtricity System Manager, they are sent to their IdP's log-in page instead of to the default SANtricity System Manager log-in page. After entering their credentials, users are sent back to SANtricity System Manager with an authenticated session and are authorized based on attributes associated with their identity.

When SAML is enabled, it is the only method used to authenticate users for access to SANtricity System Manager. Other forms of management no longer work because they cannot authenticate.

This includes the EMW, SMcli client, software developer kit client, in-band management using UTM, REST API clients using HTTP basic authentication, and REST API clients using the standard login endpoint.

Caution

It is recommended to configure MFA for added security.

SNMP Monitoring

SANtricity supports alert notifications to be sent through email, SNMP traps, and syslog. Alerts notify administrators about important events that occur on the storage array. Beginning with SANtricity OS 11.70.2, SNMPv3 is supported to provide authentication and encryption for alert notifications. It is recommended to configure SNMP USM users with authentication protocol SHA-256 or SHA-512, and privacy protocol AES-128.

In releases earlier than 11.70.2, SANtricity supports only SNMPv2c, which does not support authentication and encryption.

Caution

- For security reasons, configuring SNMPv2c is not recommended.
- If SNMPv2c is required, it is recommended to configure the SNMPv2c community string to secure it. An SNMP community string is like a user ID or password that allows access to a device's statistics.

4. Storage Administrative System Auditing

Sending Out Syslog

Log and audit information is invaluable to an organization from a support and availability standpoint. In addition, the information and details contained in logs (syslog), audit reports, and outputs are generally of a sensitive nature. To maintain security controls and posture, it is imperative that organizations manage log and audit data in a secure manner.

Offloading of syslog information is necessary for limiting the scope or footprint of a breach to a single system or solution.

Caution

It is recommended to securely offload the syslog information to a secure storage location or retention location.

Configuration of the secure audit log channel is accomplished either through the SANtricity System Manager UI or the REST API calls, as described in the following sections. For more information about how to configure syslog for audit logs, see the SANtricity System Manager online help topic “Configure Syslog Server for Audit Logs.”

Configure Syslog for Audit Logs Using the SANtricity System Manager UI

To configure syslog for audit logs by using SANtricity System Manager UI, complete the following steps:

1. Select Settings > Access Management.
2. From the Audit Log tab, select Configure Syslog Servers.
   The Configure Syslog Servers dialog box is displayed.
3. Click Add.
   The Add Syslog Server dialog box is displayed.
4. Enter the server information and click Add:
   - Server address
     Enter a fully qualified domain name, an IPv4 address, or an IPv6 address.
   - Protocol
     Select a protocol from the drop-down menu (for example, TLS, UDP, or TCP).
   - Upload certificate (optional)
     If you selected the TLS protocol and have not yet uploaded a signed CA certificate, click Browse to upload a certificate file. Audit logs are not archived to a syslog server without a trusted certificate.

   Caution

   If the certificate later becomes invalid, the TLS handshake will fail. As a result, an error message is posted to the audit log and messages are no longer sent to the syslog server.
   To resolve this issue, you must fix the certificate on the syslog server and then go to Settings > Audit Log > Configure Syslog Servers > Test All.

- Port
  Enter the port number for the syslog receiver.
  After you click Add, the Configure Syslog Servers dialog box opens and displays your configured syslog server on the page.

5 To test the server connection with the storage array, select Test All.

Configure Syslog for Audit Logs Using the REST API

To configure syslog for the audit logs by using the REST API, run the following commands:

- API
  - `POST /storage-systems/{system-id}/syslog`
  - `POST /storage-systems/{system-id}/syslog/{id} //` is used to update a specific syslog server
- URL parameter
  - system-id
  - `id`–specifies the ID of a specific syslog server to update
- POST body: contains a syslog server configuration that will be added or updated

```
{
  "serverAddress": "string", // fully qualified name or IP address
  "port": 6514,              // port number that syslog server listens on
  "protocol": "tls",         // udp, tcp, tls
  "components": [
      {
          "type": "auditLog"   // auditLog is the only choice
      }]
}
```

Caution

For the connection and communication channel to be secure, it is recommended to use the TLS protocol type. Therefore, the syslog server that is to receive the audit log messages must also be configured to support the TLS protocol (a minimum TLS version of 1.2).

5. Storage Encryption

Data-at-rest encryption is important to protect sensitive data in the event of a disk that is stolen, returned, or repurposed. The ETERNUS AB/HB series storage systems provide at-rest data encryption through self-encrypting drives. These drives encrypt data on write operations and decrypt data on read operations regardless of whether the full disk encryption feature is enabled. If the SANtricity feature is not enabled, the data is encrypted at rest on the media, but automatically decrypted on a read request.

When the full disk encryption feature is enabled on the storage system, the drives protect the data at rest by locking the drive from read or write operations unless the storage system provides the correct security or authentication key. This process prevents another storage system from accessing the data without first importing the appropriate security key file to unlock the drives. It also prevents any third-party utility or operating system from accessing the data.

The full disk encryption feature is further enhanced by enabling you to manage the full disk encryption security key through a centralized key management system, such as Gemalto SafeNet KeySecure Enterprise Encryption Key Management which adheres to the Key Management Interoperability Protocol (KMIP) standard. This function is available for the ETERNUS AB2100/AB3100/AB5100/AB6100 and the ETERNUS HB2x00/HB5x00.

The encryption and decryption operations performed by the hardware in the drive are invisible to the user and do not affect the performance or user workflow. Each drive has its own unique encryption key that cannot be transferred, copied, or read from the drive. The encryption key is a 256-bit key as specified in the National Institute of Standards and Technology (NIST) AES. The entire drive, not just a portion, is encrypted.

Caution

It is recommended to enable the drive security feature using a centralized key management system.

6. Security for Data-In-Flight

For some situations, there might be a requirement to protect all client data transported over the wire (or in flight) to an ETERNUS AB/HB series storage system. Doing so protects against man-in-the-middle attacks by encrypting sensitive data while it is in flight. Currently, ETERNUS AB/HB series storage system host interfaces do not natively support data-in-flight encryption. To comply with data-in-flight encryption requirements, it is recommended to implement data-in-flight encryption at a higher level. As a result, data is encrypted prior to write operations to the ETERNUS AB/HB series storage system and remains encrypted during read operations. Host-level encryption is advantageous because ETERNUS AB/HB series software cannot decode or read the encrypted data. Unlike wire-level encryption, which requires decryption for command processing and exposes data at the array, host-level encryption ensures data remains secure throughout. If an attacker intercepts the filesystem data while it is on the wire, the host side encryption safeguards it. A couple methods to achieve this are through a cryptographic filesystem or database-level encryption.

Cryptographic Filesystem

This strategy involves using a cryptographic filesystem on a disk mapped from an E-Series storage system. Cryptographic filesystems, such as BitLocker, eCryptfs, or gocryptfs, use NIST-approved key cipher algorithms like AES to encrypt files before writing them to disk and decrypt them upon reading. Note that filesystem-level encryption can introduce some performance overhead on the host side due to the additional CPU load required for the encryption and decryption processes.

Database Encryption

If your E-Series storage system is used to store database data, you can utilize the built-in encryption features of your database. Most modern databases, such as MySQL, PostgreSQL, Oracle, and SQL Server, support Transparent Data Encryption (TDE), which uses NIST-approved key cipher algorithms like AES.

Host Interface Mutual Authentication

Mutual authentication prevents attackers from establishing connections to an open interface by requiring both parties in communication to verify one another's identity before establishing a connection. For ETERNUS AB/HB series storage systems, mutual authentication can be implemented using Challenge Handshake Authentication Protocol (CHAP) for iSCSI or iSER over InfiniBand ports. CHAP is configured during the initial link setup and enables a host to validate itself to the storage array and for a storage array to validate itself to the host. Use the following instructions to configure CHAP on your storage array (target) and your host(s) (initiator).

System Manager

To configure CHAP authentication in System Manager, follow these steps:

Caution

Hosts being configured for CHAP authentication must first be defined on the storage array.

1. Go to Settings > System > Configure Authentication.
2. Select Two-way (mutual) authentication, then click Next.
3. Set the target CHAP secret and confirm it, then click Next.

Caution

Initiators attempting to access the target must provide the target's CHAP secret. The CHAP secret must be between 12 and 57 characters.

4. Enter the initiator secret for each host you want to configure with mutual CHAP authentication, then click Finish.

Caution

This provides the storage array the CHAP secrets you configured, or plan to configure, on each host.

Host iSCSI CHAP Configuration

To configure CHAP authentication on a standalone Linux host, follow these steps:

1. Edit your host's `/etc/iscsi/iscsid.conf` file's CHAP settings section with the information below, replacing all field values to match your configuration:

```
node.session.auth.authmethod = CHAP
node.session.auth.username = <host_iqn>
node.session.auth.password = <storage_target_secret>
node.session.auth.username_in = <host_iqn>
node.session.auth.password_in = <host_initiator_secret>
discovery.sendtargets.auth.authmethod = CHAP
discovery.sendtargets.auth.username = <host_iqn>
discovery.sendtargets.auth.password = <storage_target_secret>
discovery.sendtargets.auth.username_in = <host_iqn>
discovery.sendtargets.auth.password_in = <host_initiator_secret>
```

2. To apply the changes, restart the iSCSI services.

```
sudo systemctl restart iscsid
sudo systemctl restart iscsi
```

3. All discovered and initiated iSCSI sessions from this host will be mutually authenticated.

To configure CHAP authentication on a standalone Windows host, follow these steps:

1. Execute > `iscsicpl.exe` from a terminal window on the host to open the **iSCSI Initiator Properties** dialog box.
2. On the **Configuration** tab, select **CHAP**, and then set the CHAP secret for the host.
3. On the **Discovery** tab, select **Discover Portal**. Enter the IP address of one of the iSCSI target ports, then select **Advanced**.

4 From the **Advanced Settings**, configure the following:

4-1 Select the **Initiator IP** to be used to connect to the storage array.

4-2 Check the Enable CHAP log on box.

4-3 Enter the Target secret for the target storage array.

4-4 Check the Perform mutual authentication.

4-5 Click OK.

5 Click **OK** again to complete discovery.

6 Repeat Step 1 through Step 5 for all iSCSI connections. All connections initiated from these target portals will perform mutual authentication.

7. Managing SSL and TLS

Secure Sockets Layer (SSL) is the standard technology for keeping an internet connection secure. It safeguards any sensitive data that is being sent between two systems by using an encryption algorithm to scramble the data in transit, preventing attackers from reading it.

Transport Layer Security (TLS) is an updated and more secure version of SSL. These two terms (SSL and TLS) are used interchangeably in the industry.

SANtricity uses TLS to secure the following communication channels:

- SANtricity System Manager web client and the Web Server running on the storage system
- LDAP client running on the storage system and the LDAP/AD server
- Unified Manager/Web Server Proxy running on a host and the Web Server running on the storage system
- Storage system and the AutoSupport Server
- Storage system and the SAML Identity Provider
- Storage system and the audit-log syslog server
- Self-Encrypting Drive (SED/FDE) Lock Key Manager running on the storage system and the third-party external key manager

Caution

- SANtricity OSs from 11.60 support TLS 1.2 and TLS 1.3.
- It is recommended to enable strict certificate verification to check the expiration dates of the entire certificate chain. By default, it is disabled and only the server certificate date is checked for expiration.

TLS 1.3 Support

Starting with SANtricity OS 11.70.3, TLS 1.3 is supported as well because it is generally considered to be more secure and faster than TLS 1.2. Some security requirements, such as common criteria, do not allow TLS 1.3 to be enabled because 1.3 has not yet been evaluated. SANtricity allows for TLS 1.3 to be disabled to meet any security requirements. This setting can be changed by using the REST API.

To disable TLS 1.3, use the following REST API:

- API
  `Administration > POST /settings`
- POST body

```
{
  "serverSettings":
  {
    "tls13Disabled": true // true or false
  }
}
```

The setting change will be propagated to both controllers. After TLS 1.3 is disabled, the web server on both controllers will need to be restarted in order to update the TLS/SSL configuration. Duplex systems require the web server in both controllers to be restarted.

To restart the web server in a controller, use the following REST API:

Note

The request will reset only the controller the request is sent to and does not return a response.

- API
  `Administration > POST /restart`

After restarting the web servers, the currently supported TLS protocols and ciphers can be retrieved.

To retrieve the supported TLS and SSL protocols and ciphers for the web server, use the following REST API:

Caution

On a duplex system, the request would need to be sent to both controllers.

- API
  `Administration > GET /settings/tls-params`
- GET return data sample:

```
{
  "protocols": [
    "TLSv1.3",
    "TLSv1.2"
  ],
  "ciphers": [
    "TLS_AES_256_GCM_SHA384",
    "TLS_AES_128_GCM_SHA256",
    "TLS_CHACHA20_POLY1305_SHA256",
    "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384",
    "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256",
    "TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256",
    "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384",
    "TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256",
    "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
    "TLS_DHE_RSA_WITH_AES_256_GCM_SHA384",
    "TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256",
    "TLS_DHE_RSA_WITH_AES_128_GCM_SHA256"
  ]
}
```

8. External Key Management Server

Change the EKMS Default Key Size

Part of External Key Management Server (EKMS) setup is to generate a CSR for the client certificate (generated from the storage array). The key size for the certificate's key is 3072 bits (previously 2048). The key size is not selectable from the management interface (UI). If the user wishes to change the certificate key size, that change must be made via an NVSRAM modification. A user settable region of NVSRAM can be modified and the key size changed to a larger or smaller value.

Use the SMcli to Set the EKMS Key Size

Key size constants are shown in Table 3:

Table 3 -EKMS Key Sizes

Value	Key Size
0 (Default)	3027
1	2048
2	3027
3	4096

To use SMcli to set the EKMS key size, run the following command.

```
set controller[a] globalNVSRAMByte[0xc0]=3 // sets the EKMS key size to 4096
set controller[b] globalNVSRAMByte[0xc0]=3 // sets the EKMS key size to 4096
show allControllers globalNVSRAMByte[0xc0] // dumps the new value
```

Use the REST API to Set EKMS Key Size

- API
  POST /storage-systems/1/symbol/setControllerNVSRAM
- Post Body

```
{
  "regionId": "userConfig2Data",
  "offset": 32,
  "regionData": "3" // set key size to 4096
}
```

External Certificate Signing Request Workflow

As of SANtricity OS 11.90, the External Key Management Server certificate facility supports generating all artifacts externally, then importing all certificates in one step, see Figure 10. This allows a more sophisticated key rotation strategy to be employed such that private keys and signed certificates can be changed frequently to adhere to security policies.

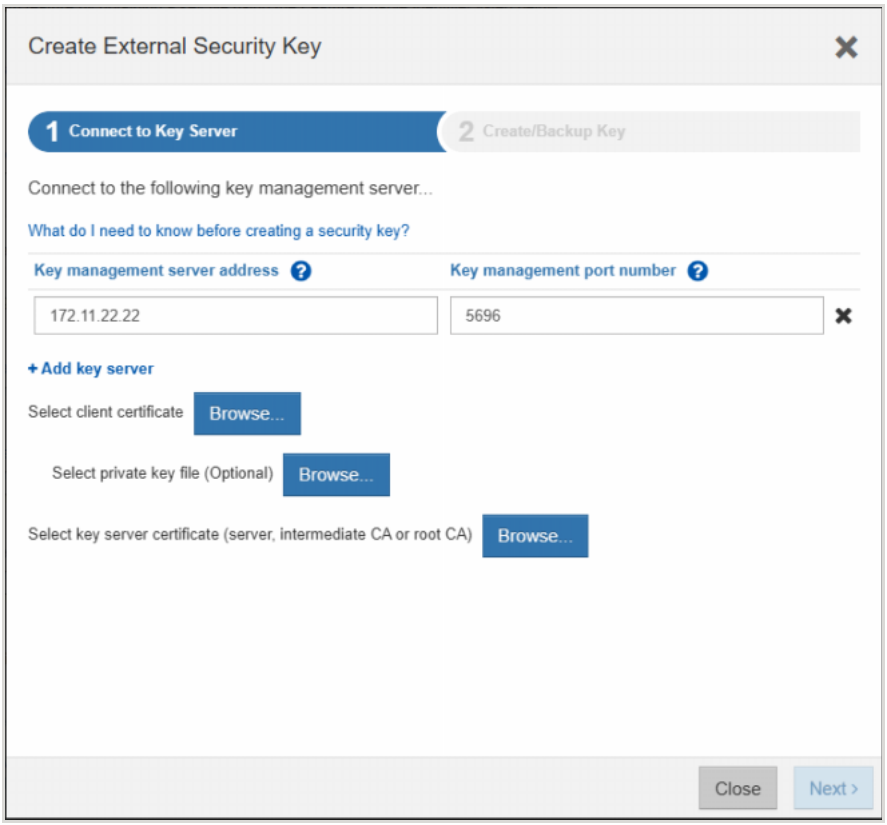
Note

The traditional internal CSR workflow is still applicable. The CSR generated may be generated and then resubmitted to the EKMS to retrieve new signed certificates. These can be reimported at will with no risk of communication loss with the KMS since the private key has not changed.

Steps to execute this workflow include:

1. Generating a leaf certificate public and private key.
2. Generating a leaf certificate signing request signed by the leaf private key.
3. Retrieving a signed certificate from the KMS server based on the generated CSR. The certificate will be signed by the EKMS's private key.
4. Retrieving the KMS server's trusted certificate.
5. Importing into the array the KMS trusted certificate, new signed certificate and leaf certificate private key.

Figure 10 Importing KMS Certificates, Including Private Key



Importing KMS certificates is also supported via the CLI grammar. Note the optional `privateKey` syntax.

```
download storageArray keyManagementClientCertificate certificateType=(client|server)
file="filename" [privateKeyFile = "keyFileName"]
```

9. Online Certificate Status Protocol

Online Certificate Status Protocol (OCSP) enables SANtricity applications that use TLS communications, such as LDAP over TLS, to receive digital certificate status when OCSP is enabled. The application receives a signed response signifying that the certificate requested is good, revoked, or unknown.

OCSP enables determination of the current status of a digital certificate without requiring certificate revocation lists (CRLs). By default, OCSP certificate status checking is disabled. It can be turned on with the REST API.

Caution

If your environment has an OCSP server, it is recommended to configure OCSP.

To enable the OCSP certificate status checking, use the following REST API:

- API
  `Administration > POST /certificates/settings`
- POST body:

```
[
  "revocationChecking": true,
  "ocspResponderAddress": "string", // name or IP of ocsp responder
  "strictCertVerificationEnabled": true
]
```

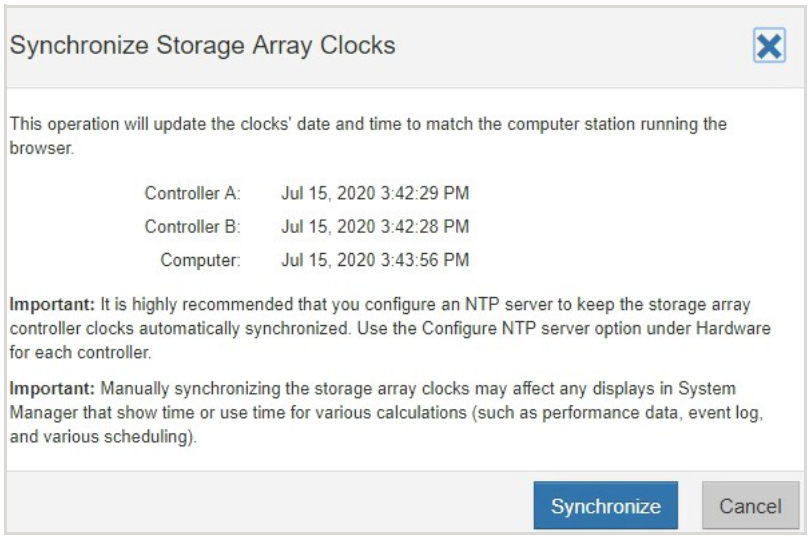
10. Network Time Protocol

SANtricity System Manager (Settings > System > Synchronize Storage Array Clocks) enables you to manually set the time zone, date, and time on the storage systems by synchronizing the clock to match the computer station running the browser (see Figure 11). It also supports NTP, but it does not support secure NTP. Without having the capability for the packets to be cryptographically signed for authentication, the NTP server can be susceptible to man-in-the-middle attacks.

Caution

It is recommended to manually set the time zone, date, and time on the storage systems if security is the priority.

Figure 11 System Manager window on how to manually synchronize the storage system clock



11. Securing Protocols and Ports

In addition to performing on-box security operations and functions, the hardening of a solution must also include off-box security mechanisms. Leveraging additional infrastructure devices, such as firewalls, intrusion prevention systems (IPSs), and other security devices, for filtering and limiting access to SANtricity is an effective way to establish and maintain a stringent security posture. Table 4 lists the common protocols and ports used in the SANtricity solution. This information is a key component for filtering and limiting access to the environment and its resources.

Table 4 Commonly used protocols and ports

Service	Port/Protocol	Description
SSH	22/TCP	Secure Shell login
SMTP	25/TCP	Simple Mail Transfer Protocol
HTTP	80/TCP	Administrative REST interface (redirects to 8443)
NTP	123/UDP	Network Time Protocol
SNMP	161/UDP	Simple Network Management Protocol
SNMP	162/UDP	Simple Network Management Protocol
LDAP	389/(UCP/TCP)	Local directory
HTTPS	443/TCP	Secure HTTP for administrative REST interface
Syslog	515/UDP	Syslog server
LDAPS	636/TCP	Secure LDAP
SYMbol	2463/TCP	Legacy Management Interface
iSCSI	3260/TCP	iSCSI target port
External Key Mgmt.	5696/TCP	External Key Management
HTTP	8080/TCP	Administrative REST interface (redirects to 8443)
HTTPS	8443/TCP	Administrative REST interface

12. Denial of Service Capabilities

Incoming HTTP requests can be rate-limited through a group of settings that control how many requests the web server will accept per second before it starts to reject requests. This feature must be managed through the REST API–it is not exposed in System Manager.

- The **enabled** parameter turns rate limiting on or off. The default is off.
- The **maxRequestsPerSec** parameter controls how many total incoming requests the web server will accept per second before starting to reject requests. The request rejection does not consider which IP address originates the request. Requests will be rejected in the order they are received after the rate limit is exceeded, regardless of which client originated the request.
- The **ipExcludeList** parameter allows a list of IPv4 or IPv6 addresses to be specified that will be excluded from any rate limiting restrictions. This prevents legitimate clients from being DoS'd during an overflow attack. The addresses in this list allow CIDR notation to be used.

After the incoming rate exceeds the limit, requests will start to be rejected with a 429 (Too many requests) result status. An audit log entry will also be generated, which will contain the IP addresses of the requests that were rejected. The **ipExcludeList** parameter provides a means to allow important traffic to continue to flow even if the web server is being overwhelmed with incoming requests.

To obtain the rate limit settings, use the following REST API:

- API
  `Administration > GET /devmgr/v2/settings`
- GET return payload

```
{
  "serverSettings": {
    "httpResponseHeaders": [],
    "relativeRedirectAllowed": true,
    "tls13Disabled": false,
    "rateLimit": {
      "enabled": true,
      "maxRequestsPerSec": 50,
      "ipExcludeList": []
    }
  }
}
```

To change the rate limit settings, use the following REST API:

- API
  `Administration > POST /devmgr/v2/settings`
- POST request body

```
{
  "serverSettings": {
    "httpResponseHeaders": [],
    "relativeRedirectAllowed": true,
    "tls13Disabled": false,
    "rateLimit": {
      "enabled": true,
      "maxRequestsPerSec": 50,
      "ipExcludeList": []
    }
  }
}
```

To change the rate limit settings, manipulate the `rateLimit` object in the payload. The `ipExcludeList` list value is a collection of IP addresses that will be excluded from the rate limit functionality.

13. Conclusion

Cyber security threats have become more prevalent in the recent years because attackers are in search of vulnerable systems. Following and implementing the recommendations specified in this guide will help reduce the security risk on the ETERNUS AB/HB series storage systems by eliminating potential attack vectors and securing the attack surfaces.

A. Security Resources

For information regarding the reporting of vulnerabilities and incidents, security responses, and customer confidentiality, contact our support division.

ETERNUS AB series All-Flash Arrays, ETERNUS HB series Hybrid Arrays
Security Hardening Guide for SANtricity

P3AG-6042-03ENZ0

Date of issuance: March 2025
Issuance responsibility: Fsas Technologies Inc.

Fsas Technologies Inc.