

IBM® Storage

IBM Storage Virtualize, IBM Storage FlashSystem, and IBM SAN Volume Controller Security Feature Checklist - For IBM Storage Virtualize 8.6

IBM Storage Team



© Copyright International Business Machines Corporation 2023, 2024.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Executive summary	1
System security capabilities	2
User authentication	2
Local authentication	3
Remote authentication	3
Role-based access control	4
Default users	5
Object-based access control	5
Login interfaces	6
Representational State Transfer API	7
Password policy	7
Locking users	8
Locking the superuser	8
Unlocking users	9
Unlocking the superuser	10
Session timeouts	11
Login banner	11
Multifactor authentication	11
Single sign-on	12
Two person integrity	13
System Time and Network Time Protocol	13
Auditing and reporting	14
Secure and trusted boot	14
Secure sockets and Secure Shell settings	14
SSL/TLS certificates	15
Certificate truststores	16
Disabling USB ports	17
IP address and network port allocation	17
Internal software execution protection mechanisms	17
Preventing access as root	18
File system protection	18
Disabling service assistant password reset	18
Code signing	19
Secure engineering (Security and Privacy by Design)	19
Data security capabilities	21
Encryption	21
Volume delete protection	27
Immutable snapshots with Safeguarded Copy	27
Logical port isolation	27
Authors	28
Notices	29
Trademarks	30
Terms and conditions for product documentation	31
Applicability	31
Commercial use	31
Rights	31
Privacy policy considerations	31



Executive summary

IBM® Storage Virtualize based storage systems are secure storage platforms that implement various security-related features, in terms of system-level access controls and data-level security features.

This document outlines the available security features and options of IBM Storage Virtualize based storage systems. It is not intended as a “how to” or best practice document. Instead, it is a checklist of features that can be reviewed by a user security team to aid in the definition of a policy to be followed when implementing IBM FlashSystem®, IBM SAN Volume Controller, and IBM Storage Virtualize for Public Cloud.

IBM Storage Virtualize features the following levels of security to protect against threats and to keep the attack surface as small as possible:

- ▶ The first line of defense is to offer strict verification features that stop unauthorized users from using login interfaces and gaining access to the system and its configuration.
- ▶ The second line of defense is to offer least privilege features that restrict the environment and limit any effect if a malicious actor does access the system configuration.
- ▶ The third line of defense is to run in a minimal, locked down, mode to prevent damage spreading to the kernel and rest of the operating system.
- ▶ The fourth line of defense is to protect the data at rest that is stored on the system from theft, loss, or corruption (malicious or accidental).

The topics that are discussed in this paper can be broadly split into two categories:

- ▶ System security

This type of security encompasses the first three lines of defense that prevent unauthorized access to the system, protect the logical configuration of the storage system, and restrict what actions users can perform. It also ensures visibility and reporting of system-level events that can be used by a Security Information and Event Management (SIEM) solution, such as IBM QRadar®.

These security features include, but are not limited to:

- Multifactor authentication (MFA)
- Role-based access control (RBAC)
- Object-based access control (OBAC)
- Disabling access to the command line interface (CLI), graphical user interface (GUI), and Representational State Transfer (REST) interface

- ▶ Data security

This type of security encompasses the fourth line of defense. It protects the data that is stored on the system against theft, loss, or attack. These data security features include Encryption of Data At Rest (EDAR) or IBM Safeguarded Copy (SGC).

This document is correct as of IBM Storage Virtualize 8.5.3.

System security capabilities

For the purposes of this document, the term *system* is used to mean any platform that runs the IBM Storage Virtualize software, including all IBM Storage FlashSystem 5000, 7000, 9000 systems, IBM SAN Volume Controller products, and IBM Storage Virtualize for Public Cloud. Where applicable, code levels are supported, which also relates to the IBM Storwize® family of products.

Physical security measures must be implemented in accordance with organizational mandates in addition to the logical, network, and hardware features that are discussed in this publication.

The system is designed to limit the number and means of user access to a restricted set of known, policed, and secured access points.

User authentication

An authenticated user is required to administer, configure, and monitor the system. The system supports the following types of users:

- ▶ Local

These users are defined on the system and managed internally by the system. The system supports up to 200 or 400 local users, depending on your platform.

- ▶ Remote

These users are defined on an external authentication server. Remote users can be defined in an external Lightweight Directory Access Protocol (LDAP) repository, such as Microsoft Active Directory. The details of the LDAP server and groups must be configured on the system.

Each user is assigned a role that defines the actions that they can perform on the system. For more information, see “Role-based access control” on page 4.

IBM Storage Virtualize supports local authentication, remote authentication, or a combination of both. When deciding which type of authentication to use, consider the benefits and drawbacks of each one.

Security guidelines (such as the ones that are described in [National Institute of Standards and Technology \(NIST\) Special Publication 800-209](#)) suggest disabling local authentication where possible. It is considered more secure because it provides a single point for managing user security. One drawback of relying on only remote authentication is that the system might not be accessible if the remote authentication service is unavailable. For this reason, security guidelines also recommend leaving a single local user for emergency purposes, who can be used to administer the system even when remote authentication is unavailable. However, the credentials and access of these local users requires more management.

Using a combination of local and remote authentication, with correct access controls and least privilege principles that are applied, provides the right balance between security and serviceability.

Local authentication

With *local authentication*, users can authenticate to the system by using credentials that are stored locally within IBM Storage Virtualize without an external authentication service. Local users can authenticate with the system by using a password, an SSH key, or both. Local user credentials, including a superuser, are one-way hashed with salt by using PBKDF2.

To configure local user authentication, user groups must be on the system (see “Role-based access control” on page 4). It is possible to use the existing default user groups, but it is a best practice to create user groups with suitable roles and authentication requirements. Then, local users can be created and assigned to a user group.

It is possible to have local authentication enabled for first-factor authentication, and native MFA that is configured to provide second-factor authentication.

It is not possible to fully disable local authentication, but it is possible to delete all local users from the system and disable the local superuser account (see “Default users” on page 5) so that there are no remaining local users.

Remote authentication

Remote authentication allows users to authenticate to the system by using credentials that are stored on an external authentication service. When you configure remote authentication, users do not need to be configured on the system or assigned more passwords. Instead, passwords and user groups that are defined on the external authentication service are used. This mechanism can be used to separate user management from storage management.

A remote user is authenticated on a remote server that implements LDAPv3. A remote user does not need to be added to the list of users on the system, although they can be added to configure optional SSH keys. Remote users have their groups that are defined by the remote authentication service. User groups must be created locally on the system to reflect the group names, and their roles that are defined by the remote authentication service.

If the remote authentication service is unavailable, remote users cannot log in to the system. There is no fallback mechanism for logins to the system in this scenario, other than to use local authentication. For this reason, it is a best practice to have a local user that can be used to gain access to the system.

LDAP authentication applies only to first-factor authentication and cannot be used for second-factor authentication. Often, native LDAP authentication is configured to provide first-factor authentication, and native MFA is configured to provide second-factor authentication.

In contrast, with single sign-on (SSO), the system delegates all authentication (first-factor and second-factor) to the remote Identity Provider and SSO service. However, unlike LDAP (which can authenticate logins through all interfaces), SSO can be used only to authenticate logins by using the GUI.

For more information, see [Configuring remote authentication](#).

Role-based access control

Role-based access control (RBAC) is a mechanism for restricting system access based on a user's role within an organization. User groups exist on the system so that a set of users can be assigned to the same role. A local user can be assigned to a single user group on the system. The system supports up to 256 user groups. Each user group is assigned a role that is associated with a set of privileges and commands that can be run.

The following roles are available:

- ▶ **Monitor**

Users with this role can view all objects, but cannot manage the system or its resources. Support personnel can be assigned this role to monitor the system and to determine the cause of a problem. This role is suitable for use by automation tools, such as the Storage Insights data collector for collecting status about the system.
- ▶ **Copy Operator**

Users with this role have monitor role privileges and can create, change, and manage all Copy Services functions (Remote Copy and IBM FlashCopy®), but cannot create consistency groups or modify host mappings.
- ▶ **FlashCopy Administrator**

Users can create, change, and delete all the FlashCopy mappings and consistency groups, and create and delete host mappings.
- ▶ **Administrator**

Users with this role can access all functions on the system, except those functions that involve managing users, user groups, and authentication. This standard role is assigned to users who administer the system and perform tasks, such as provisioning storage.
- ▶ **Security Administrator**

Users with this role can access all functions on the system, including managing users, user groups, all aspects of security, and user authentication.
- ▶ **Restricted Security Administrator**

When Two Person Integrity (TPI) is enabled, all users with a Security Administrator role are demoted to Restricted Security Administrators. All tasks that would normally be limited to a Security Administrator role require a TPI request to be approved to temporarily elevate them to a Security Administrator.
- ▶ **Service**

These users can delete dump files, add and delete nodes, apply service, and shut down the system. These users can also perform the same tasks as users in the Monitor role.
- ▶ **Restricted Administrator**

Users with this role can perform the same tasks as the Security Administrator role, but are restricted from deleting specific objects. Support personnel can be assigned this role to solve problems.
- ▶ **3-Site Administrator**

Users with this role can configure, manage, and monitor 3-site replication configurations by using specific command operations that are available only on the 3-Site Orchestrator. This role is the only role that is intended to be used with the 3-site orchestrator.
- ▶ **VASA Provider**

Users with this role can manage virtual volumes (VVOLs) that are used by VMware vSphere and managed by using IBM® Spectrum Connect software.

For more information, see [User roles](#).

Default users

Upon creation, the system defines a single local Security Administrator user that is called *superuser*. For newly created systems, the default superuser password must be changed on first login. Although this user cannot be deleted or configured for remote authentication, it can be disabled.

During initial setup, users for the system are defined. After this process is complete, the default superuser account can be disabled. This user can be reenabled with physical access to the system only by using the technician port, by another user with the Security Administrator role, or by a support engineer that uses remote support assistance (RSA). For more information, see “Locking users” on page 8.

If a SIEM solution is used, consider forwarding the system audit log so that the SIEM system can be triggered when this action occurs.

Some CLI commands and service procedures are available that can be performed only by the superuser. If the superuser account is disabled, it might be necessary to temporarily re-enable the account occasionally.

If you choose not to disable the superuser account, ensure that the highest level of protection is afforded to the superuser account. For example, use a strong password, limit the knowledge of the password, and configure multi-factor authentication.

For more information, see the following IBM Documentation web pages:

- ▶ [chuser](#)
- ▶ [chsecurity](#)
- ▶ [Password policy](#)

Object-based access control

In addition to role-based access control, other restrictions can be implemented on a per-user group basis to limit the set of objects that users can view, manage, and configure. For example, you might configure two different groups of administrators: the first group is limited to provisioning storage to test or development systems; the second group can provision storage to production systems. This approach is known as *object-based access control* (OBAC).

OBAC can be used to separate parts of the system between users; for example, different storage administrators can administer different pools of storage. OBAC can also be used by managed service providers to implement a form of multi-tenancy between multiple clients that are hosted on the same system.

Role-based and object-based access control can be configured simultaneously and provide two different ways of enforcing least privilege by restricting what users can do on the system.

Ownership groups are used to maintain groupings of users and objects. Only users with the Security Administrator role can define ownership groups.

Access to the following objects can be controlled by using an ownership group:

- ▶ Child pools
- ▶ Volumes
- ▶ Volume groups

- ▶ Hosts
- ▶ Host clusters
- ▶ FlashCopy mappings
- ▶ FlashCopy consistency groups
- ▶ User groups
- ▶ Port sets

The system supports up to 64 ownership groups.

For more information, see [Configuring ownership groups](#).

Login interfaces

The system provides a CLI, GUI, and a REST application programming interface (API), all of which authenticate users and allow them to administer the system. All interfaces implement data in-flight encryption to secure the login and all subsequent interactions with the system.

Command line interface

Users log in to the CLI by using an SSH client terminal window. Logging in to the system by using a terminal places the user in a highly restricted shell. For example, the user cannot run a change directory command and can run only the commands that are designated by IBM as required to administer the system.

Consider the following points:

- ▶ This login interface can be disabled on a per-user group basis.
- ▶ This login interface cannot be disabled for the superuser.

Note: SSH clients cannot use the batch mode setting to run commands non-interactively if there are any user groups that are defined on the system where a password and key are required or multifactor authentication is enabled.

For more information, see the following IBM Documentation web pages:

- ▶ [UNIX commands available in interactive SSH sessions](#)
- ▶ [Command-line interface](#)
- ▶ [Changing user groups](#)

Graphical user interface

Users log in to the GUI is by using a Hypertext Transfer Protocol Secure (HTTPS) connection from Transport Layer Security by using a supported web browser.

You can install certificates signed by a certificate authority (CA) with the suitable certificate chain. The system supports the following system certificate key types:

- ▶ RSA 2048-bit
- ▶ ECDSA 384-bit
- ▶ ECDSA 521-bit

The system presents the installed system certificate and chain to web browsers when they connect to the system.

This login interface can be disabled on a per user group basis.

By default, the superuser is exempt from disabling this interface. To disable this interface for the superuser, run the `chsecurity -disablesuperusergui yes` command.

The system currently supports a single certificate that is shared by the GUI, Encryption Key servers, IP Quorum, CIMOM, and Rest API interfaces.

For a list of supported key servers and versions, see [IBM Storage Virtualize Supported Key Servers](#).

For more information, see the following IBM Documentation web pages:

- ▶ [Managing certificates for secure communications](#)
- ▶ [Changing user groups](#)

Representational State Transfer API

The system provides a REST model API. This interface is accessed by using an HTTPS connection. The REST API implements a full function list that is equivalent to the CLI interface.

Connections to the REST API can be made only by using a suitable authenticated user, for which a session token is generated. The session timeout, the elapsed time before the user or application must reauthenticate can be defined in the security policy. The RBAC and OBAC user restrictions are enforced.

This login interface can be disabled on a per-user group basis.

By default, the superuser is exempt from disabling this interface. To disable this interface for the superuser, use the `chsecurity -disablesuperuserrest yes` command.

For more information, see the following IBM Documentation web pages:

- ▶ [Spectrum Virtualize RESTful API](#)
- ▶ [Changing user groups](#)

Password policy

The Security Administrator can define a set of policies regarding all users on the system. This set of attributes can be tailored to match your organizational mandates about password rules.

The following attributes can be defined:

- ▶ Minimum password length (6 - 64 characters)
- ▶ Minimum number of uppercase characters (1 - 3)
- ▶ Minimum number of lowercase characters (1 - 3)
- ▶ Minimum number of special characters (1 - 3)
- ▶ Minimum number of digits (1 - 3)
- ▶ History check (0 - 10) before password reuse
- ▶ Password expiry (0 - 365 days)
- ▶ Password expiry warning (0 - 30 days): Displayed on CLI at login only
- ▶ Password age (1 - 365 days): Minimum age before password can change
- ▶ Force change on next login: One time option by Security Administrator

The “Force password change on next login” setting can be used to ensure that a newly created user changes their password when logging in for the first time. It can also be used by a Security Administrator when changing the system-wide password policy, which forces all local users to change their password at next login to ensure that the password policy is enforced.

For more information, see the following IBM Documentation web pages:

- ▶ [Password policy](#)
- ▶ [chsecurity](#)

Locking users

Local users are managed by IBM Storage Virtualize and support locking. Remote users are not managed by IBM Storage Virtualize and can be locked only by the remote authentication service. Any user with the Security Administrator role can manually lock or unlock a local user account at any time. When a user account is locked, all login attempts to the user fail (these attempts do not increase the failed login attempts counter for the user).

The system supports two types of account locking; local users can either be locked manually, or automatically according to the defined policy on the system. In addition, the following system-wide policies can be set:

- ▶ Automatic lock after (0 - 10) consecutive failed login attempts.
A value of 0 means that the account does not lock because of failed login attempts.
- ▶ The set length of the time auto-lock applies for (0 - 10080) minutes (7 days maximum).
A value of 0 means that the account remains locked out indefinitely.
- ▶ Superuser account can be locked from a general login.

Manual user locking can be performed in the management GUI, or by using the **chuser** CLI command through SSH or the REST API. Possible use cases include the following ones:

- ▶ *Suspicious login activity has been detected.* In this scenario, a security administrator can lock the user indefinitely until the issue is resolved without needing to delete the user. They can simply unlock the user later if the incident is a false alarm.
- ▶ *Temporarily disable a user that has gone on vacation.* In this scenario, a security administrator can lock the user so that it cannot be used while the person is away. They can simply re-enable the user when the person returns.
- ▶ *New employee joining the company.* In this scenario, a security administrator can create a new user who is initially locked, and keep the user deactivated until the employee's first day at the company.

Automatic user locking can be enabled by defining a security policy in the management GUI, or by using the **chsecurity** CLI command through SSH or the REST API. When a user breaches the policy (for example, too many failed login attempts), then the user is locked automatically according to the defined policy.

For more information, see [chsecurity](#).

Locking the superuser

Most day-to-day storage and security administration tasks can be performed without needing the superuser by using other users with the Administrator or SecurityAdmin role. The superuser is a special local user with the SecurityAdmin role who is the only user that can run service assistant commands on the system. Because the superuser is the highest privilege user on the system, secure the user as much as possible. It is a best practice to keep the superuser unlocked for convenience, then consider using multifactor authentication, a strong password, and limit knowledge of the password. Alternatively, it is possible to disable the superuser by locking the user; in this scenario, the superuser is unavailable to perform service assistant tasks and must be temporarily unlocked when required.

Special care must be taken to not lose access to the superuser. Before adding security controls to the superuser, it is a best practice to review the documentation and form a plan for recovering access and reenabling the superuser if it becomes necessary. By default, the superuser is exempt from the system-wide policy for manual or automatic account locking. This setting is a safeguard to prevent accidental or unintentional locking the superuser account. To apply the system locking policy to the superuser, use the **chsecurity -superuserlocking enable** command.

Note: The superuser locking feature is supported only on platforms with a permanent technician port. It is not supported on platforms with a configurable technician port that can be turned off because the system depends on the availability of the technician port as a recovery mechanism. To protect the superuser on platforms that do not support superuser locking, consider enabling MFA for the superuser and using a strong password.

After superuser locking is enabled, the superuser can be locked manually by any user with the Security Administrator role by using the management GUI, or by running the **chuser -lock superuser** CLI command through SSH or the REST API. After the superuser is locked, all login attempts for the user fail (these attempts do not increase the failed login attempts counter for the user).

If the Remote Support Assistance feature is enabled, a privileged support engineer can log in to the system on request and lock the superuser.

If you intend to keep the superuser unlocked for day-to-day use, automatic locking for the superuser can be a useful safeguard to prevent brute force attacks. For example, configure the security policy on the system so that users are locked out for a fixed period after the maximum number of failed login attempts is reached.

If you plan to lock the superuser, ensure that at least one extra user with the Security Administrator role exists (either locally or remotely) so that it can be used to log in and unlock the superuser when needed, and then lock it again when finished. Consider making the additional SecurityAdmin user as secure as possible, for example, by using an obscure username, a long password, and enabling multifactor authentication.

For more information, see the following IBM Documentation web pages:

- ▶ [Password policy](#)
- ▶ [The chsecurity command](#)

Unlocking users

Local users are managed by IBM Storage Virtualize and can be unlocked by a Security Administrator at any time. Remote users are not managed by IBM Storage Virtualize and can be unlocked only by the remote authentication service.

Regardless of how a user becomes locked (manually, or automatically by the system policy), the user can be unlocked manually at any time by a Security Administrator who uses the management GUI, or uses the **chuser** command through SSH or the REST API.

If the user is locked automatically for a defined period, the system automatically unlocks the user after the period expires. The time at which a user automatically unlocks can be viewed by using the management GUI, or by using the **lsuser** command through SSH or the REST API.

Unlocking a user account resets any failed login attempts by the user. If the Security Administrator requires that the user changes their password on the next login, this policy is enforced.

For more information, see the following IBM Documentation web pages:

- ▶ [Password policy](#)
- ▶ [chsecurity](#)

Unlocking the superuser

If the superuser account is locked, occasionally it must be unlocked to run service assistant tasks on the nodes within a system. There are many ways to unlock the superuser, depending on your configuration.

If the superuser is locked automatically for a defined period, the system automatically unlocks the superuser when the period expires. The time at which a user automatically unlocks can be viewed by using the management GUI, or by running the `lsuser` command through SSH or the REST API.

If the superuser is locked manually or automatically locked out indefinitely, then the simplest method to re-enable the superuser is to use another SecurityAdmin user. If one exists locally on the system or on the remote authentication service, log in as this user and manually unlock the superuser by using the management GUI, or by using the `chuser -unlock superuser` command through SSH or the REST API.

If the Remote Support Assistance feature is enabled, a privileged support engineer can log in to the system on request and unlock the superuser.

There might be scenarios where another SecurityAdmin user cannot be used, for example, if the user is defined on a remote authentication service that is inaccessible, or because there are no other SecurityAdmin users that are defined. In these circumstances, if Remote Support Assistance is disabled, then physical access to the system is required to unlock the superuser.

If physical access to the system is available, the technician port can be used to unlock the superuser. When connecting to the service assistant GUI through the technician port, if the superuser is locked, then a button to unlock the superuser is displayed on the login page. This procedure is required only on one of the nodes in the system.

This method can also be used to reset the credentials for the superuser. When connecting to the service assistant GUI through the technician port, if the superuser is unlocked, then a button to reset the credentials of the superuser is displayed on the login page. This procedure is required only on one of the nodes in the system.

For more information, see the following IBM Documentation web pages:

- ▶ [Password policy](#)
- ▶ [The chsecurity command](#)

Session timeouts

The CLI and GUI can include defined independent session timeouts. These timeouts can be 5–240 minutes.

The REST interface can also include a defined token session timeout of 10–120 minutes.

The grace time for SSH sessions can be configured to specify the duration of time a user must authenticate per SSH connection before the connection is terminated (15–18000 seconds).

The maximum number of login retries that can be attempted per single SSH session also is configurable (1–10 attempts).

For more information, see [chsecurity](#).

Login banner

A customizable login banner can be modified to match your organizational requirements and display on the CLI or GUI login window before logging in.

For more information, see [Changing the login message](#).

Multifactor authentication

The system supports the configuration of *multifactor authentication*. Two different methods are available to configure multifactor authentication: by using the native multifactor authentication feature, or by configuring single sign-on and then using a multifactor authentication solution that integrates with the single sign-on identity provider.

The system supports native multifactor authentication by using one of the following multifactor authentication providers:

- ▶ IBM Security® Verify: Cloud-based
- ▶ Duo Security: Cloud-based

These providers can be configured to enforce a wide range of other authentication options. The system connects to the multifactor authentication service by using the industry-standard OpenID Connect protocol.

After multifactor authentication is enabled system-wide, it can be enabled on a per-user group basis.

Note: By default, the superuser is exempt from the default user group's multifactor authentication setting. To enable multifactor authentication for the superuser, use the `chsecurity -superusermultifactor yes` command.

The superuser is the only user account that can run service assistant commands. Special care must be taken not to lose access. Before adding security controls to the superuser, review the documentation and form a plan for recovering access and reenabling the superuser.

Multifactor authentication is not supported on the REST API. User groups with higher privileges should be configured to disable REST API access, and user groups with lower privileges should be configured to enable REST API access for monitoring and automation purposes.

If the multifactor authentication service is unavailable, users belonging to user groups with MFA enabled cannot log in to the system. There is an optional fallback mechanism that can be configured to permit logins to the system in this scenario that is disabled by default and can be configured only by using the CLI. To permit logins to the system when the multifactor authentication service is unavailable, use either one of the following approaches:

- ▶ For IBM Security Verify, use the following command:

```
chauthmultifactorverify -failmode insecure.
```

- ▶ For Duo Security, use the following command:

```
chauthmultifactorduo -failmode insecure.
```

For a list of supported MFA providers and versions, see [IBM Storage Virtualize Supported Authentication Providers](#).

For more information, see [Configuring multifactor authentication](#).

Single sign-on

The system supports the configuration of *single sign-on*. The system connects to the single sign-on provider by using the industry-standard OpenID Connect protocol and supports the following providers:

- ▶ IBM Security Verify: Cloud-based
- ▶ Microsoft Active Directory Federation Services (AD FS): On-premises
- ▶ Microsoft Azure Active Directory: Cloud-based
- ▶ Duo Security: Cloud-based
- ▶ Okta Workforce Identity Cloud: Cloud-based

After single sign-on is configured, it can be enabled on a per-user group basis. Single sign-on is *not* supported for local user accounts (including superuser) because it requires the delegation of all authentication to an external Identity Provider (IdP).

Single sign-on applies only to logins to the GUI and does *not* apply to the CLI or REST interfaces. If you have any local users on the system, it is recommended to disable CLI and REST access for those user groups or enable the “password and key required” feature, which enforces CLI users to provide a password (something you know) and public key (something you have) during authentication.

For user accounts that are used for monitoring and automation, it is recommended not to enable passwords. Instead, put these users in a separate user group so they can be assigned different access controls. Also, assign these users the least privileged role that is required for the functions that they are performing.

For a list of supported SSO providers and versions, see [IBM Storage Virtualize Supported Authentication Providers](#).

For more information, see [Configuring single sign-on](#).

Two person integrity

The system supports the enabling of two-person integrity (TPI) checking to prohibit critical and potentially destructive tasks from being run by a single security administrator. When TPI is enabled, certain tasks require the involvement of two security administrators.

When TPI is enabled, a security administrator can either request elevated privilege for a certain amount of time, or will be prompted to request elevation if they attempt to perform a restricted task. The request can only be approved or rejected by a different security administrator.

By implication, any user that previously had a SecurityAdmin role is changed to a RestrictedSecurityAdmin role when TPI is enabled.

The system ensures there are at least two security administrators before the feature can be enabled.

The request includes a period of time that the user is granted elevated privileges. This can range from 10 minutes to 24 hours. The approver of the request can change the time period. Any security administrator can revoke an approved request at any time while it is active.

Any tasks that are normally limited to be actionable only by a user with the SecurityAdmin role are restricted. In addition, any commands that promote a user to SecurityAdmin are also restricted. For example, the following actions require elevated privileges after TPI is enabled:

- ▶ Create, change, or remove security administrator user groups
- ▶ Change the non-security administrator user group attribute on an existing local user to a security administrator user group
- ▶ Modify attributes on existing local users that are members of the security administrator user groups
- ▶ Change the role of existing non-security administrator user groups to the security administrator role
- ▶ Change the security administrator role of an existing user group to a non-security administrator role
- ▶ Remove and change Safeguarded backups and Safeguarded backup locations
- ▶ Delete Safeguarded snapshots
- ▶ Use a provisioning policy to define a set of rules that are applied when volumes are created within a storage pool or child pool
- ▶ Change the single sign-on credentials that are used for the system
- ▶ Remove the Safeguarded snapshot policy association from a volume group

For more information, see [Two person integrity](#).

System Time and Network Time Protocol

It is important that the system time and date are set correctly to maintain system security.

The system time and date are used when establishing secure connections to other servers to perform validity checks on their certificates, if the system time and date are incorrect then these validity checks might fail, which prevents the connections.

To mitigate against attacks that are based on the system time and date being out of sync with other services, the system supports a configurable Network Time Protocol (NTP) server to control the system time.

An NTP server can only be configured by a user with the SecurityAdmin role. The system ensures that any modifications to system time do not affect the duration that a user account will remain locked or the retention period for immutable safeguarded copies.

Auditing and reporting

The system includes an internal tamper-proof audit log that can be viewed and exported if required. The internal audit log traces all successful configuration command execution on the system, along with details of the user, where they connected from (IP address), and date and time stamps.

The system can also send audit information to an external server by using syslog or SNMP.

When the audit log is exported, more information is provided. In addition to logging successful configuration commands, unsuccessful configuration commands are logged. The exported audit log can also include entries for successful and attempted logins by using the GUI, CLI, or REST interfaces.

For more information, see [Audit log commands](#).

Secure and trusted boot

Secure boot encrypts the file systems and relies on a hardware root of trust that extends all the way through to the operating system and initrd (initial ramdisk) to unlock the file system.

The system checksums and validates the file systems at boot time to protect against system files and executable files being corrupted (maliciously or by hardware or software faults).

For more information, see [Secure boot](#).

Secure sockets and Secure Shell settings

Secure sockets (SSL/TLS) and *Secure Shell (SSH)* are used to establish authenticated and encrypted connections to the system for management interfaces, such as the GUI, CLI, REST, and when the system connects to remote servers, such as email, LDAP, or external key managers. Secure sockets and Secure Shell define a set of protocols, cipher suites, and key exchange algorithms that can be used.

When a connection is established, the local and remote system negotiates which protocol, cipher, and key exchange algorithm is used. Some users might want to restrict the permitted choices to increase security by disabling algorithms that provide weaker security.

The system defines levels of security for secure sockets and SSH. The lower levels implement a wider choice of algorithms to maximize compatibility. The higher levels restrict the choice of algorithms to provide higher levels of security, but might be incompatible with older software.

Note: Existing systems that are created on older software versions and upgraded over time might still use weaker protocol levels. Each level must be considered carefully and set according to your current security requirements, and adjusted over time as your requirements change or newer levels are added.

Consider the following points:

- ▶ SSL security levels are 1 - 4. The default for new systems is level 3.
- ▶ SSH security levels are 1 - 3. The default for new systems is level 3.

For more information, see these IBM Documentation web pages:

- ▶ [Changing SSL/TLS/SSH levels for the system using the CLI](#)
- ▶ [Security levels and supported security ciphers](#)

SSL/TLS certificates

SSL certificates are used to establish secure communications for many services. The system uses a certificate to identify itself when authenticating with other devices. Depending on the scenario, the system might be acting as either the client or the server. In one-way TLS authentication scenarios where the system is acting as the server (for example, a client web browser connects to the IBM Storage Virtualize management GUI), the client verifies the system certificate. In two-way TLS authentication, also known as mutual authentication, which is used in scenarios where the system acts as the client (for example, the system connects to an encryption key server), the system verifies the key server certificate, and the key server also verifies the system certificate.

When a system is initially configured, a default certificate is generated. The default certificate is signed by the system's internal root CA. Replace the default certificate with one that contains more information that is specific to their organization and meets their security policy. The system can generate a certificate that is either signed by the system's own root CA or by an external CA, such as a third-party CA or an internal company CA. The system certificate is used by the following features and interfaces:

- ▶ Management GUI and service GUI
- ▶ REST API
- ▶ Encryption key servers
- ▶ IP quorum applications
- ▶ Common Interface Model Object Manager (CIMOM)
- ▶ IPsec for secured IP partnerships
- ▶ Policy-based replication
- ▶ VMware APIs for Storage Awareness (VASA) provider
- ▶ Multifactor Authentication with IBM Security Verify

Note: The system certificate should be configured before using any features or interfaces that require it. Any changes to the system certificate might result in the need to reconfigure software or services to use the new certificate.

If you use a system-signed certificate, you do not need to generate and manage a certificate signing request (CSR) when updating the certificate. In this scenario, the user provides the information that is stored in the certificate, and then the system automatically generates the

certificate signing request and private key, signs the request by using the system's own CA, and installs the new certificate automatically.

The system also supports the automatic renewal of system-signed certificates when they are within 30 days of expiry. A new system-signed certificate is generated and installed automatically, and it contains the same information as the old certificate, so that there is no disruption to services that use certificates. Automatic renewal of certificates is useful if using features such as policy-based replication or VMware vSphere Virtual Volumes (vVols), where lack of a working certificate causes automated tasks to fail.

If you use an externally signed certificate, the security administrator must provide the information that is stored in the certificate to generate a CSR, copy the CSR file from the system, get the certificate request signed by a CA, copy the signed certificate to the system, and install the new signed certificate.

The system does not support the automatic renewal of externally signed certificates. The system raises a warning event 30 days before the system certificate expires, and another alert event when it expires. When the certificate expires, the system experiences disruption to services that authenticate by using certificates. The security administrator should proactively monitor the system certificate and plan for a replacement certificate to be generated and installed before the current certificate expires to avoid disruption to services.

For security reasons, the system does not support importing a certificate private key that is generated by another device. When the system generates a certificate request, it also generates the corresponding private key, which never leaves the system.

For more information, see [Managing certificates for secure communications](#).

Certificate truststores

An *SSL certificate*, also known as a public key certificate, is a digital file that is used to authenticate the identity of an entity, such as a web server or a CA. Certificates contain a public key and can either be self-signed or signed by a CA. When a CA signs a certificate, it is certifying that a public key is owned by the entity that is described in the certificate. By signing certificates, trust can be established between entities. For example, if the certificate of a web server A is signed by a CA B, then anybody who trusts B also now trusts A.

A certificate chain is a collection of related certificates, consisting of an end-point certificate (sometimes known as the leaf certificate) that identifies a device, and one or more CA certificates. If an end-point certificate is signed directly by a root CA, then the chain has a depth of 2 and contains the end-point certificate and the root CA certificate that signed it. If an end-point certificate is signed by one or more intermediate CAs, all the intermediate CA certificates are included, and the root CA certificate that signed the final intermediate CA certificate in the chain. For example, if an end-point certificate A is signed by intermediate CA certificate B, and intermediate certificate B is signed by a root CA certificate C, then the chain has a depth of 3 and contains A > B > C.

A certificate bundle is a collection of non-related root CA certificates, which are combined into a single file. Bundles are useful for supplying many trusted CA certificates to a truststore in one operation.

For a client to establish a secure TLS connection with a server, the client must verify that it trusts the certificate of the server. Typically, the server presents its end-point certificate, including any intermediate CA certificates that signed it but excluding the root CA certificate of the chain. The client has a truststore containing the certificates of all the root CAs it trusts. During verification, the client validates the structure of the certificate chain and searches its

truststore for the root certificate that signed the final intermediate CA certificate in the chain. If a matching root certificate is found in the truststore, the client trusts the server, and the connection succeeds. If the truststore does not contain the root certificate, the client rejects the connection to the server.

The system supports truststores for installing and managing root CA certificates for external services. Each truststore supports 12 KiB of certificate data and can be assigned to one or more services. For example, truststores can be used to enable mutual TLS authentication for IPsec between two systems in an IP replication partnership. First, a truststore is created on system A with the IPsec tag enabled, which contains the root certificate of system B. Second, a truststore is created on system B with the IPsec tag enabled, which contains the root certificate of system A.

Currently, certificate truststores can be managed only by using the CLI.

For more information, see [Creating and managing certificate authority store by using CLI](#).

Disabling USB ports

For companies with a security policy that restricts the usage of USB ports, these ports can be disabled on a per-node basis on systems with a supported BIOS. After the ports are disabled, they are electronically disabled at the BIOS level and rendered non-operational. Only the superuser can be enabled or disable USB ports.

This feature can be combined with encryption key management on local USB flash drives so that the ports remain disabled during day-to-day use but can be temporarily reenabled in scenarios when a key must be provided.

For more information, see [chnodeusb](#).

IP address and network port allocation

Because the system uses several IP addresses for management, iSCSI, and other secure interfaces, consider whether your company firewall rules require modification.

If specific interfaces or features are not used, access might need to be disabled by using firewall rules that are provided by the network team. Similarly, if specific interfaces are used, ensure that the firewall rules are set correctly.

Alternatively, a proxy server can be configured on the system to manage connections between your internal network and any entity outside of your network.

For more information, see the following IBM Documentation web pages:

- ▶ [IP address allocation and usage](#)
- ▶ [Defining an HTTP proxy server](#)

Internal software execution protection mechanisms

The IBM Storage Virtualize software runs on a hand-picked Linux installation with a bare minimum of carefully selected packages. The kernel configuration is locked down and tightly controlled. Only software that is required for the interaction between the hardware and the IBM Storage Virtualize software is installed.

By limiting the packages that are installed and preventing the installation of extra packages, many operating system vulnerabilities are prevented from affecting the system.

Preventing access as root

On traditional Linux installations, a user with root privileges can perform any system action and bypass any security control.

The root access is not permitted on the system. Logging in as root is not permitted by way of the network, console, or any other back-door methods.

The management software in the system runs as an internal Linux user, without root privileges. Although external users cannot access this software, an attacker might leverage a weakness to compromise the system. Even in this circumstance, they do not have root privileges.

File system protection

Secure boot encrypts the file systems and relies on a hardware root of trust. The boot drives are tied to the Trusted Platform Module chip.

Checksums and validation of the file systems at boot time provide security against files and executable files being corrupted maliciously or by hardware or software faults.

Most file systems are mounted as read-only; the others are mounted with 'noexec', 'nosuid', and 'nodev' attributes to minimize security risks.

New software can be installed only by using the built-in software update technology. All software packages are validated and must be signed by a private key that is held within IBM servers. This configuration ensures that the tight control on software is maintained for all systems that are deployed at customers.

These techniques are used to prevent an attacker from running arbitrary code, which is a prerequisite for most security vulnerabilities. They also minimize the attack surface as much as possible.

Disabling service assistant password reset

The superuser is the only user account that is permitted to run service assistant commands on the system. When the superuser password is lost, the following methods can be used to reset the password:

- ▶ Gaining physical access to the system and inserting a suitably prepared USB flash drive to run the **resetpassword** command.
- ▶ The use of another SecurityAdmin user on the system to change the superuser password
- ▶ Connecting physically by way of the technician port and selecting **Reset Superuser Credentials** in the service assistant GUI.
- ▶ The use of the remote support assistant feature to allow the engineer with RestrictedAdmin role to log in to the system by way of a challenge and response and then reset the password.

For organizations with a security policy that requires this ability to be disabled, the password reset feature can be disabled on a per-system basis. After this ability is disabled, it is no longer possible to reset the superuser password.

To view the status of the password reset feature, use the `setpwdreset -show` command. To disable the password reset feature, use the `setpwdreset -disable` command. To enable the password reset feature, use the `setpwdreset -enable` command.

Note: Care must be taken when disabling the superuser password reset feature. If the superuser password is lost and is unrecoverable, and if the superuser password reset feature was disabled, the superuser cannot be accessed and IBM Support *must* be engaged.

The recovery procedure is disruptive and involves system downtime.

For more information, see [Maintaining passwords using the CLI](#).

Code signing

Code signing is the process of digitally signing software packages to confirm that IBM produced them and to guarantee that the software has not been modified since it was signed. All IBM Storage Virtualize software packages are signed by IBM to ensure that the software has not been tampered with.

The effectiveness of code signing depends on how well the keys that are used to sign software packages are secured. Released software packages are signed by using a set of production keys, which are stored on a secure build server. Access to the secure build machine is tightly controlled and accessible only to members of the IBM build team that manages the software build process. Requests to sign software packages by using the production keys are tightly controlled. Requests are generated by a build process that compiles the code, and they must originate from a specific build machine. Production signing requests can be made only for code branches with special settings, which are controlled by an access-controlled repository, and can be configured only by trusted members of the build team.

When a software package is downloaded from IBM Fix Central and installed on a system, the IBM Storage Virtualize software checks the digital signature on the package before extracting the contents, and refuses to install any software package that has not been signed by using those production keys. IBM is in control of which software packages are digitally signed and which packages can be installed on a system running IBM Storage Virtualize.

Secure engineering (Security and Privacy by Design)

Security and Privacy by Design (SPbD) is a set of practices that ensures that security and privacy are embedded into the design of IBM products and services. SPbD is aligned with the NIST [Secure Software Development Framework](#), and it is required across all business units within IBM. The SPbD principles are followed when designing and developing security features for the IBM Storage Virtualize family of products:

- ▶ Threat modeling
- ▶ Privacy assessment
- ▶ Vulnerability management
- ▶ Code scan
- ▶ Penetration test
- ▶ Secure release process

IBM uses an exercise that is called *threat modeling* to identify and understand security threats that might harm an IBM Storage Virtualize system. Threat modeling is a form of risk assessment that models aspects of the attack and defense sides of the system. As part of this exercise, a document that is called a threat model is produced. A threat model is a structured representation of all the information that affects the security of an application, which includes an inventory of the network and management interfaces, and the assets of the system (such as stored data at rest). The threat model is a living document within IBM that is reviewed for every major software release to ensure that no new threats were introduced without an appropriate mitigation or safeguard to protect against them.

A privacy assessment is an analysis of how personally identifiable information is handled to ensure compliance with appropriate regulations, such as General Data Protection Regulation (GDPR), Payment Card Industry Data Security Standard (PCI-DSS), and Health Insurance Portability and Accountability Act (HIPAA). As part of every major software release, IBM completes a privacy assessment to determine any privacy risks and evaluate ways to reduce them.

Security vulnerabilities in all IBM products are managed by a program that is called the IBM Product Security Incident Response Team (PSIRT). IBM PSIRT is the centralized process through which IBM customers, security researchers, industry groups, government organizations, or vendors report potential IBM security vulnerabilities. IBM is committed to responding to new threats and risks. IBM Secure Engineering practices are designed so that IBM can act in a timely fashion to a reported security vulnerability affecting an IBM product or solution. To help protect our customers, IBM does not publicly disclose or confirm security vulnerabilities until IBM has conducted an analysis of the product and issued fixes or mitigations.

For every Continuous Delivery (CD) and Long-Term Support (LTS) release of the software, IBM performs a Static Application Security Test (SAST), Dynamic Application Security Test (DAST), and Open Source Vulnerability Scan (OSS).

SAST, or static code analysis, is the analysis of software when it is not running, that is, the source code. It aims to find security issues in the code, such as buffer overflows or programming errors.

DAST, or dynamic code analysis, is the analysis of software while it is running on a real system. Its purpose is to find security issues in the code that might be exploited by a malicious user, such as SQL injection or cross-site scripting issues in the management GUI. The IBM Storage Virtualize team engages IBM X-Force® Red, which is an external security team to perform penetration testing of the software. IBM X-Force Red uses a combination of standard tools, such as Qualys and Rapid7, and bespoke tools. Because standard scanners do not understand IBM bespoke storage controller software, they often flag false positives or fail to effectively scan the system. The additional bespoke tools aim to find security issues that standard scanners typically cannot find. These DAST tools have a deeper understanding of the product and can perform targeted testing. Historical results have shown that these bespoke tests tend to find better security issues. Any security issues that are found are categorized as low, medium, high, or critical vulnerabilities. The IBM Storage Virtualize team ensures that any high or critical vulnerabilities are resolved before releasing the software, and aims to address the low and medium vulnerabilities where possible.

Open-source vulnerability scanning is a method that is used to identify known security vulnerabilities in open-source software packages that are used by IBM Storage Virtualize. If an issue is found in an open-source package, IBM either updates the package to a new version containing the fix or fixes the vulnerability in the open-source package. IBM ensures that the latest versions of open-source packages are used where possible, and the development team undergoes training on developing and using open-source code in IBM products.

Third-party researchers and other security entities can report potential security vulnerabilities in IBM Storage Virtualize through [HackerOne](#).

When upgrading IBM Storage Virtualize systems or installing new code levels, it is a best practice to use the most recent PTF that is available for the chosen release. Using the newest software available is important because PTFs often contain the latest security fixes.

For more information about the processes that IBM development teams follow when developing software, see *Security in Development: The IBM Secure Engineering Framework*, REDP-4641.

For more information, see the following IBM Trust Center web pages:

- ▶ [Security and Privacy by Design \(SPbD\)](#)
- ▶ [IBM Security Vulnerability Management](#)
- ▶ [IBM Privacy](#)

Data security capabilities

The system provides external access to data by way of the SCSI, iSCSI, SAS, or NVMe-based protocols over your chosen SAN network connectivity. It is assumed that this physical connectivity is secured by using normal data center means.

In this section, the extra data security features that are provided by the system are described.

Encryption

Encryption is a technology that uses cryptography to ensure confidentiality of sensitive information. Encryption uses keys to encode information so that it cannot be understood by unauthorized parties.

The system supports encryption of both data at rest and data in flight. To enable the encryption feature, an encryption license for each machine must be purchased. A machine is defined as a node (for node-based platforms such as IBM SAN Volume Controller) or a control enclosure (for enclosure-based platforms such as IBM FlashSystem). For example, if the system consists of two SAN Volume Controller Nodes, two licenses must be purchased and activated. If the system consists of one IBM FlashSystem control enclosure, then one license must be purchased and activated. When a license for every machine is activated on the system, various data encryption features can be enabled on the system.

IBM sells encryption licenses only in geographies that permit the usage of encryption technology.

Encryption of Data at Rest

Data at rest is static data that is stored on an internal drive or external storage, as opposed to data in flight, which is data that is being transmitted from one location to another one.

To enable encryption of data at rest (EDaR), you must first purchase encryption licenses and activate the encryption feature on the system. When enabling EDaR, some form of key management must be configured. Key management can be handled by local USB flash drives, external key management servers (with the use of IBM GKLM, SafeNet KeySecure, or Thales CipherTrust Manager), or a combination of both. When using local USB flash drives, ensure that they are not formatted with NTFS or APFS as the file system type. Encrypted USB flash drives and USB hub devices are not supported.

Regardless of the key management method that initially is chosen, it is possible to migrate between key management methods later. For example, it is possible to enable encryption with local USB flash drives as the only enabled key management method, and later migrate to using a network key server instead. Alternatively, both key management methods can be enabled simultaneously, keeping local USB flash drives as a backup method for urgent scenarios in which the key server might be disabled.

After encryption has been configured on the system, migrating between key management methods is non-disruptive and does not require any stored data-at-rest to be re-encrypted. You can migrate between key management methods by using the management GUI or the command-line interface.

IBM Storage Virtualize software supports both hardware-based and software-based EDaR. If you use a platform that supports built-in hardware encryption, the data encryption is applied without any impact to performance. Where possible, hardware-based encryption should be used. The system automatically uses hardware-based encryption if an encrypted array is created by using internal drives; the data on that array is always encrypted by using hardware with no performance penalty.

All NVMe drives that are supported by the system, including IBM FlashCore® Modules (FCMs) and a range of other third-party drives, are self-encrypting drives (SEDs) that encrypt data within the electrical circuit of each individual drive.

NVMe SEDs automatically lock on power loss and require unlocking by using an access key. On systems with encryption enabled, the individual SED access keys are protected by the overall system master key. If encryption is disabled, then NVMe SEDs continue to encrypt the data, but it is protected only by a well-known access key, so if encryption is disabled, the data that is stored on these drives is not protected from physical theft, but can still be deleted by using a cryptographic erasure.

IBM FlashCore Module is a FIPS-validated, self-encrypting, and self-compressing NVMe drive. At the time of writing, the FIPS validation status is as follows:

- ▶ IBM FlashCore Module 1 is FIPS 140-2 Level 1 [validated](#).
- ▶ IBM FlashCore Module 2 is FIPS 140-2 Level 2 [validated](#), and it is undergoing FIPS 140-3 Level 2 [validation](#).
- ▶ IBM FlashCore Module 3 is undergoing FIPS 140-3 Level 2 [validation](#).

When drives are connected through a Serial Attached SCSI (SAS) network, the SAS protocol chip provides data encryption capabilities. The system uses a PMC Sierra PM8073 SPCve+12G chip, which uses algorithms that are compliant with FIPS 140-2 Level 1.

With this data encryption capability, data at rest (on the drive) can be stored as encrypted, and in all cases is applied after data reduction technologies.

All data encryption keys (DEKs) and key encryption keys (KEKs) on the system are AES-256 bit keys. Data encryption keys are used to perform AES-XTS, and all keys are protected by using an AES wrap-key operation.

The IBM Storage Virtualize software can also apply encryption to devices that do not support built-in encryption. In this scenario, the software offloads the job of data encryption to the Advanced Encryption Standard New Instructions (AES-NI)-capable CPU within the node. This capability mainly applies to the IBM SAN Volume Controller platform, but it can be used by any platform that virtualizes external storage. This feature can be used to encrypt underlying storage that does not support encryption itself (for example, external storage that is connected by using other protocols, such as Fibre Channel (FC) or iSCSI). The system automatically uses software-based encryption if an encrypted storage pool is created that contains external storage that is not capable of encryption itself. For each managed disk in the storage pool that does not report as self-encrypting, the data on that external storage is automatically encrypted by using software. Software-based encryption incurs a small performance penalty that depends on your configuration and the workload.

IBM Storage Virtualize automatically detects the encryption capabilities of other IBM back-end storage controllers, such as IBM FlashSystem. It is possible to override the encryption status of a managed disk within a storage pool by using the `chmdisk -encrypt yes/no` command, for example, to indicate that a third-party back-end storage controller is performing EDaR. In this scenario, IBM Storage Virtualize software notices that the managed disk is performing its own EDaR, so it does not use software-encryption to encrypt the data again.

The system uses FIPS-compliant data encryption algorithms to perform software-based encryption.

Encryption of Data in Flight

Data in flight is data that is being transmitted from one location to another one, as opposed to data at rest, which is static data that is stored on an internal drive or external storage.

IBM Storage Virtualize supports *Encryption of Data in Flight* (EDiF) for data that is being replicated between two systems that are connected by Ethernet and configured in a secure Internet Protocol (IP) partnership. When a secured IP partnership is created, for example, between a production system and a recovery system, the data is secured as it travels through the network between the production system and the recovery system.

Secured IP partnerships use a combination of IPsec and IKEv2 to secure data in flight. IKEv2 is an IPsec-based tunneling protocol that uses secure key exchange algorithms to establish a secure connection to the partner system. IPsec is a suite of security protocols that ensures packets that are transmitted over the network are authenticated and encrypted.

In secured IP partnerships, the partner systems authenticate with each other, negotiate the security parameters, exchange encryption keys, and establish secured network tunnels through which encrypted data travels. Partner systems are authenticated by certificates that are issued by either the system's internal root CA or a trusted third-party root CA or intermediate CA.

To enable secured partnerships, you must purchase encryption licenses and activate the encryption feature on both partnered systems. To create a secured IP partnership, the necessary certificates and authorities must be installed on the partner systems. To do this task, a certificate truststore must be created on each partner system with the IPsec flag enabled. The truststore contains the correct certificate of the partner system. For example, if the partner systems are each using a system-signed certificate, then the signing CA

certificate must be exported and installed into a truststore on the partner system with the IPsec flag enabled.

One benefit of using system-signed certificates rather than third-party signed certificates on partner systems is that replication over secured IP partnerships is not interrupted when the system certificates expire. When automatic renewal is enabled, the system certificate that is used by the secured IP partnership automatically is renewed before it expires, and replication activity continues unaffected.

IBM Storage Virtualize does not support EDiF between host servers and the system, the system and back-end storage devices, or systems that are connected by FC that are configured in an FC replication partnership.

For more information, see [Configuring encryption](#).

Secure data deletion

Secure data deletion refers to the process of erasing or overwriting existing data from a data storage device so that the data is rendered inaccessible and cannot be reconstructed.

To comply with European Regulation EU2019/424, the system provides methods to erase data securely from media that is being decommissioned. The [NIST Special Publication 800-88 - Guidelines for Media Sanitization](#) provide best practices for securely deleting data from storage media, and categorizes media sanitation into three categories: Clear, Purge, and Destroy.

Various methods exist to perform a secure erasure of the storage media and the internal boot drives, depending on the use case and hardware being decommissioned. Often, these methods can be used only when the hardware is functional and capable of being erased by using software commands, for example, when repurposing an existing system, returning a loan or demo system to the vendor, or sending hardware to IBM for failure analysis. Any non-functional or damaged hardware that cannot be reliably erased should be physically destroyed.

Erasing internal boot drives

The internal SATA boot drives are within the node or node canister and contain the installed IBM Storage Virtualize software image, configuration data, and occasionally other data, such as a hardened cache data. Depending on the platform that is used, there might be a single boot drive or redundant boot drives installed, but the secure erase procedure erases all boot drives simultaneously.

When decommissioning a node or node canister, it is a best practice to erase securely the internal boot drives. To erase the internal boot drives, issue the command `satask rescuename -secureerase`. This command starts an ATA Sanitize operation with Block Erase set, repartitions the device, and then reinstalls a new IBM Storage Virtualize software image. This action is equivalent to the Purge sanitation method that is defined in section 5 of NIST SP 800-88.

Erasing SAS drives

The SAS drives are within a control enclosure or expansion enclosure and might contain customer data or system metadata (if the drive was configured as a quorum device). Depending on the drive vendor and model, the SAS drive might not support the SCSI Sanitize command and therefore cannot be securely erased.

When decommissioning a single SAS drive, it can either be formatted, securely erased, or physically destroyed (depending on your requirements). When decommissioning multiple SAS drives, repeat the erase process for each drive individually.

SAS drives can be formatted by issuing the command `chdrive -task format <id>`. This command issues standard SCSI write commands to overwrite the data with zeros and aims to be equivalent to the Clear sanitation method that is defined in section 5 of NIST SP 800-88.

SAS drives can be securely erased by issuing the command `chdrive -task erase <id>`. This command issues a SCSI Sanitize command with either Crypto Erase, Block Erase, or Overwrite set (in that priority order), depending on what operations the device supports. This action is equivalent to the Purge sanitation method that is defined in section 5 of NIST SP 800-88.

The Sanitize command is designed and implemented to ensure that data is erased and cannot be recovered. The format command might overwrite data with zeros before destroying it, or it might only reinitialize mapping tables so that future reads of the device return zeros, but without destroying the data that is stored on the media. The implementation of the format operation is vendor-specific and might vary across drive types and firmware levels. For SAS drives that do not support the secure erase procedure, manual procedures (such as overwriting the data by using a host-based tool) should be used to securely erase the data on the device.

The system raises events to indicate when a secure erase task starts, completes successfully, or fails, for a SAS drive.

Erasing NVMe drives

The NVMe drives are within a control enclosure or expansion enclosure and might contain customer data or system metadata (if the drive was configured as a quorum device). Depending on the drive vendor and model, the NVMe drive might not support the NVMe Sanitize command, and therefore cannot be securely erased.

When decommissioning a single NVMe drive, it can either be formatted, securely erased, or physically destroyed (depending on your requirements). When decommissioning multiple NVMe drives, repeat the erase process for each drive individually.

NVMe drives can be formatted by issuing the command `chdrive -task format <id>`, which issues a Trusted Computing Group (TCG) Opal Revert to restore to factory settings and an NVMe Format with Secure Erase Setting (SES) bits 001b set to select User Data Erase; take ownership of the device; create locking ranges; and then issue a TCG Opal GenKey to cryptographically erase the locking range. This action is equivalent to the Purge sanitation method that is defined in section 5 of NIST SP 800-88.

NVMe drives can be securely erased by issuing the command `chdrive -task erase <id>`, which issues a TCG Opal Revert to restore to factory settings and an NVMe Sanitize with Crypto Erase and Block Erase set; take ownership of the device; create locking ranges; and then issue a TCG Opal GenKey to cryptographically erase the locking range. If the drive does not support the NVMe Sanitize command, an NVMe Format with SES bits 001b set to User Data Erase is issued instead. This action is equivalent to the Purge sanitation method that is defined in section 5 of NIST SP 800-88.

The system raises events to indicate when a format or secure erase task starts, completes successfully, or fails, to an NVMe drive.

For more information, see [Secure data deletion](#).

Volume delete protection

The system can be enabled with volume delete protection. This feature can be configured at the system-wide or pool and child pool level.

Volume delete protection prevents any user from deleting a volume if the system received read/write I/O requests for the volume within the defined period. The Security Administrator can set the period 5 minutes - 24 hours.

For more information, see [Volume protection](#).

Immutable snapshots with Safeguarded Copy

The system provides a mechanism to set a policy on a group of volumes to enable automated immutable snapshot volumes to be taken at regular intervals. These volume snapshots cannot be read from or written to by any host system. A restore or recovery of a Safeguarded copy must be performed before access is granted.

Users cannot delete or modify immutable snapshots.

An external REST API is available to trigger instant immutable copies by using an external scheduler tool, for example, if your SIEM tool detects an imminent security threat.

Safeguarded policies define how frequently Safeguarded backups are created, and how long to retain backups before they are deleted.

Safeguarded Copy can be combined with IBM CyberVault or IBM Spectrum® Sentinel to build a fast recovery environment in the case of ransomware attacks.

For more information, see [Safeguarded Copy function](#).

Logical port isolation

In addition to traditional SAN Fabric zoning mechanisms, the system can define logical portsets. A *portset* is a grouping of logical addresses that is associated with a specific type of traffic.

FC-based and IP address-based Ethernet portsets can be defined. The system supports both FC and IP portsets for host attachment, and IP address portsets for back-end storage connectivity and IP address replication traffic.

For example, portsets can be used to further restrict the login traffic from a host, or set of hosts, to isolate traffic to specific SAN paths.

For more information, see [Portsets](#).

Authors

This blueprint guide was produced by a team of specialists from around the world.

Bill Scales is a Distinguished Engineer for IBM Systems who is based in the IBM Hursley Lab, UK. He has over 25 years of experience with IBM Storage, and is one of the architects for the IBM Storage Virtualize products.

James Whitaker is a software engineer for IBM Systems who is based in the IBM Manchester Lab, UK. James graduated from The University of Manchester in 2011 with a B.Sc (Hons) degree in Computer Science. He has 11 years of experience designing, developing, and testing various software features across the IBM Storage Virtualize family of products. In his current role, he is the technical lead of a team that is responsible for delivering new security and encryption features.

Barry Whyte is an IBM Master Inventor working in the IBM Systems Group. Based in Auckland, New Zealand, Barry is an IBM Principal Storage Technical Specialist in the Asia Pacific region. Barry primarily works with the IBM Storage Virtualize (IBM SAN Volume Controller and IBM FlashSystem) family of virtual disk systems. Barry graduated from The University of Glasgow in 1996 with a B.Sc (Hons) degree in Computing Science. Barry joined the IBM SAN Volume Controller development team soon after its inception and held many positions, including performance architect, during his 20 years in development. Barry has over 25 years experience developing, designing, and selling IBM Storage,

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.


COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

FlashCopy®	IBM Security®	Storwize®
IBM®	IBM Spectrum®	X-Force®
IBM FlashCore®	QRadar®	
IBM FlashSystem®	Redbooks (logo)  ®	

The following terms are trademarks of other companies:

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

Microsoft, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

VMware, VMware vSphere, and the VMware logo are registered trademarks or trademarks of VMware, Inc. or its subsidiaries in the United States and/or other jurisdictions.

Other company, product, or service names may be trademarks or service marks of others.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Privacy policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at <http://www.ibm.com/privacy> and IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details> in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.



© Copyright IBM Corporation

February 2024

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.



Please recycle

ISBN 0738461474

REDP-5716-00