Author(s): Lincoln Dale <ltd@cisco.com>

White Paper

Security Features of the Cisco MDS 9000 Family of Multilayer Storage Switches

rev 1.6, November 2003

1 Purpose

The purpose of this whitepaper is to highlight the advanced security features present within the Cisco MDS 9000 Family of Multilayer storage switches and their practical use within storage networking environments. The principle goal of this whitepaper is provide best practice information on designing and implementing a secure Storage Area Network (SAN), making use of the advanced security features within the Cisco MDS 9000 Family of Multilayer storage switches.

1.1 Introduction

Many organizations are wary of transmitting sensitive data over networks. Hospitals transmit sensitive patient information to insurance companies. Banks and stock exchange companies transfer vital financial information over networks. There is a valid fear that this data could be viewed or altered in transit or used by malicious people to harm patients, create lawsuits, or defraud corporations. People want these kinds of data communication to remain private. Almost every company has transactions that, viewed by an eavesdropper, could have negative consequences.

Traditionally, SANs have been considered 'secure' primarily due to the fact that SAN deployments had been limited to a subset of a single data centre – in essence, an isolated network. Fibre Channel (FC) has been considered 'secure' on the basis that FC networks have been isolated from other networks – in essence, physical security.

It isn't uncommon today to find a SAN that spans outside a datacenter. SAN extension technologies such as DWDM, CWDM and FCIP can be used to connect devices in multiple datacenters to storage in multiple datacenters. Transport technologies such as iSCSI decrease the cost associated with attaching hosts to a SAN and therefore accelerate the rate at which devices are connected to a SAN.

As the number of devices connected to a SAN increases and SANs become more commonplace, it becomes much more of a gamble to depend on 'security through isolation'. Just as security is a consideration when sensitive information passes over data networks, security should also be a consideration when deploying a SAN.

SAN Security should be thought of from three angles:

1. Securing the SAN from external threats (e.g. hackers and people with malicious intent)
2. Securing the SAN from internal threats (e.g. unauthorized staff and compromised devices)
3. Securing the SAN from unintentional threats from authorized users (e.g. mis-configurations and human errors)

The first two are relatively straight-forward and well understood from a security standpoint. The third angle is less straight-forward and only minimal or no attention in the past has been paid to unintentional security threats from authorized users. Just as in a UNIX or Windows environment it is prudent to minimize the use of performing all administrative tasks with *root* or *Administrator* privileges, the same cautious approach of granting the minimum amount of privileges to perform a task holds true when working with a SAN. There are many facets to this – the benefits of locking down 'operator' privileges on a switch using role-based authentication are easily understood, but others such as minimizing the probability of a disruptive fabric reconfiguration as a result of a mis-configured switch with an overlapping Domain_ID are less common-place. Many of these blur the boundaries between SAN Security, best-practice SAN Design and High Availability SAN Design, but all are important from the perspective that correctly configured *secure* switch can help prevent both deliberate and unintentional disruptions.

The Cisco MDS 9000 Family of Multilayer storage switches comes complete with advanced security features to help provide security within a SAN. The following sections will detail what these features are and how to best deploy them in a production environment. All features listed in this document are provided as standard base software (no-cost) license, unless noted otherwise.

1.2 How this document is organized

This document is organized into the following sections:

- Section 2 provides a summary of the security features within the Cisco MDS 9000 Family of Multilayer switches.
- Section 3 provides details of the security features make up Fibre Channel Fabric Security
- Section 4 covers the security implications of Storage over IP
- Section 5 covers Switch Management security
- Section 6 covers security from an Operational point-of-view.

While this document is split into multiple sections for clarity, all areas should be given equal weighting. Just as Lenin's Maxim states "A chain is only as strong as its weakest link", the same holds true for security.

2 Summary of Security Features

The following is a summary of the Security Features present within the Cisco MDS 9000 family of Multilayer switches. Unless otherwise noted, the features listed below are available on all products in the product family.

Fabric Security Features [3]	
Zoning [3.1]	
Supported Zoning Types [3.1.2]	Hard-Zoning and Soft-Zoning; Hard-zoning enforced via ACLs in hardware, soft-zoning via the name server. All members of the Cisco MDS 9000 product family currently support hardware-enforced zoning for 2,000 zones and 20,000 zone members.
Zone Membership [3.1.1]	Zone Membership in SAN-OS 1.3(1) may be defined by any of seven methods: 1. Device Port World Wide Name (pWWN) 2. Fabric Port World Wide Name (fWWN) 3. Fibre Channel ID (FC_ID) 4. Interface and Switch WWN (sWWN) 5. Domain ID and Port Number 6. Symbolic Node Name (iSCSI Qualified Name or IP Address of an iSCSI device) 7. FC Alias
Zoning Granularity [3.1.4]	Hardware LUN-based Zoning; Zoning membership may be applied to individual LUNs in a disk array *(LUN Zoning is a feature enabled by the Enterprise License Package).*
Zoning Capabilities [3.1.5]	All Models within the Cisco MDS 9000 product family can enforce 'Read-Only LUNs'. i.e. block all write traffic to a given LUN using hardware zoning. *(Read-Only LUNs is a feature enabled by the Enterprise License Package).*
Zoneset Distribution [3.1.7]	Configurable; distribute active zoneset only or full zoneset including fcaliases.
Virtual SANs (VSANs) [3.2]	
VSANs [3.2]	VSANs can be used to create multiple logical SANs over a common physical infrastructure. Each VSAN runs its own set of fabric services providing for absolute partitioning between virtual fabrics. Up to 256 VSANs can be configured on a single switch.
Inter VSAN Routing [3.2.4]	Inter-VSAN Routing (IVR) can be used to securely create a path from a device in one VSAN to one or multiple devices in a different VSAN, without merging the individual VSAN fabrics (and hence without creating a merged fault domain).
Port Security [3.3]	
Port Security locks down ports to given devices by by association of Device Port World Wide Name (pWWN) to Fabric Port World Wide Name (fWWN), Device Port World Wide Name (pWWN) to Interface, World Wide Node Name (nWWN) to Fabric Port World Wide Name (fWWN), including ranges of ports and logical interfaces such as Port-Channels. Port Security functionality includes separate Active and Configured Databases, Auto Learning and Logging of Intrusion Attempts. *(Port Security is a feature enabled by the Enterprise License Package).*	

Fibre Channel Security Protocols (FC-SP) ***(SAN-OS 1.3)*** [3.4]	
SAN-OS 1.3 adds support for the FC-SP standard, with switch-to-switch and host-to-switch authentication, via DH-CHAP integrated with RADIUS/TACACS AAA infrastructure. Authentication can be enabled in a per-port basis. Future versions will support FCsec frame encryption to provide per-frame origin authentication, integrity protection, anti-replay protection, and privacy protection. *(FC-SP DH-CHAP Authentication is a feature enabled by the Enterprise License Package).*	
Fibre Channel Addressing Security [3.5]	
Principal switch priority **[3.5.1]**	Supports setting of priorities for principal switch selection.
Disruptive Reconfigure Fabric (RCF) Rejection **[3.5.2]**	Ability to reject disruptive fabric reconfiguration requests.
FICON Fabric Binding **[3.5.3]**	Limit ISLs to a given set of switch WWNs (and optionally limit Domain_IDs) (***SAN-OS 1.3***). *(FICON Fabric Binding is a feature enabled by the FICON/Mainframe License Package).*
FC_ID Caching, Persistent FC_ID Allocation and Static FC_ID Assignment **[3.5.4]**	Ability to limit what FC_IDs are assigned to given pWWNs as well as offer persistent FC_IDs across system restarts.
Storage over IP Security [4]	
iSCSI Security [4.1]	
iSCSI Authentication **[4.1.5]**	CHAP authentication for incoming iSCSI initiators. Validation of authentication credentials can either be local (through an on-switch database), or centralized through RADIUS or TACACS+ (TACACS+ in *SAN-OS 1.3*).
Mapping of iSCSI initiators to Virtual FC Initiators **[4.1.2]**	iSCSI initiators may be dynamically or statically mapped to Virtual FC Initiators with a unique fWWN and pWWN per initiator per VSAN (Virtual N_Port). Dynamic Mapping of initiator can be either via iSCSI node name (IQN) or by IP-Address. Static Mapping can either use system-assigned persistent WWNs or manually-assigned WWNs. FC Targets may be mapped to iSCSI devices including LUN Mapping capabilities to map a FC LUN to a different LUN offered to the iSCSI initiator.
Presenting Fibre Channel Targets as iSCSI Targets **[4.1.1]**	Either dynamically map all available FC Targets to be possible iSCSI Targets (FC fabric zoning permitting), or explicitly manual mapping of individual FC Targets to iSCSI Targets.
Access Controls **[4.1.3]**	Initiators are explicitly put into VSANs thereby allowing for access-control based on VSAN. Standard FC fabric services apply to iSCSI inititator virtual N_Ports. (i.e. FC fabric zoning). Per-interface restrictions on iSCSI targets to limit individual targets to being advertised either globally or only on specific Gigabit Ethernet interfaces, sub-interfaces or VLANs.
Other Storage over IP Security Features	
VLAN Trunking **[4.4]**	Each Gigabit Ethernet Port on the IP Services Module may participate as a 802.1q trunk port.

Switch Management Security Features [5]	
Account Management [5.1]	Local User Database, RADIUS-based Centralized Account Management and TACACS+-based Centralized Account Management.
Secure Access to Switch Management [5.3]	SSH, SCP, SFTP.
Secure GUI Communication [5.4.2, 5.5, 5.7]	The Fabric Manager & Device Manager GUI Administration tools are both digitally-signed JAVA applications which communicate to the switch using SNMP/v3 (encrypted) with access-restrictions limited via Roles-Based Access and optionally centrally managed with RADIUS / TACACS+ (TACACS+ in *SAN-OS 1.3*).
IP ACLs [5.6]	Limit access to IP interfaces (e.g. mgmt0 and IP-over-FC) via IP Access Control Lists.
User Accounting [5.8]	Provides an accounting audit-trail of configuration commands.
Time Synchronization [5.9]	Ability to synchronize the Time/Date via NTP.
Roles-based Access [5.5]	Ability for different users to have different roles/responsibilities and management capabilities and restrictions.
VSAN-Based Roles Access [5.5.2]	SAN-OS 1.2 provides further granularity by providing per-VSAN Roles-based Access. *(VSAN-Based Roles Access is a feature enabled by the Enterprise License Package).*
SMI-S XML-CIM Management [5.10]	Secure access to SMI-S / WBEM / XML-CIM Management Interface via SSL / HTTPS (*SAN-OS 1.3*).
SNMP Security [5.4]	
SNMPv1/v2 [5.4.1]	Limit SNMP access through read-only and read-write community strings.
SNMPv3 [5.4.2]	Enhances SNMP security through SNMPv3 security model that provides Encryption, authentication, ensures data integrity and prevents masquerading.
SNMP IP Restrictions [5.4.1]	SNMP access may be restricted to a given set of IP addresses.
SNMP Groups [5.4.2.2]	SNMP groups (RFC 2575) can be used to tie into Roles-based Access.
Operational Security [6]	
Denial of Service (DoS) Attack Protection [6.1]	Protection against switch failure / control-plane failure through Denial-of-Service attempts against IP interfaces (e.g. mgmt0).
Control-Plane Scalability [6.2]	Scaleable modular control-plane with distributed processing per line card designed to handle large fabric deployments..
System Logs [6.3]	Ability to log multiple events to both internal logs and externally to syslog servers.
Call Home [6.4]	Ability to 'call home' to log error events and indicate problems requiring administrator attention.

3 Fibre Channel Fabric Security

There are several factors to be considered today when designing and deploying storage area networks (SANs). Attributes such as high availability, scalability and security of the network all must be carefully considered when designing and implementing the network itself. Security within Fibre Channel has traditionally only consisted of using 'Zoning' to restrict communication between devices, with little or no thought to securing other aspects of securing a Fibre Channel fabric such as limiting the propagation of disruptive reconfigure-fabric, countermeasures to prevent access to unauthorized data via spoofed or hijacked WWNs, or restrictions to prevent rogue switches from connecting into a fabric and intercepting traffic via specially-crafted routing updates.

From the perspective that a correctly configured *secure* switch can help prevent both deliberate and unintentional disruptions to a Fibre Channel fabric, Fibre Channel Fabric Security should be considered a mandatory piece of any SAN design.

The Cisco MDS 9000 Family of multilayer storage switches provide many features to provide Fibre Channel Fabric Security offering the SAN designer a more cost-effective, resilient and reliable SAN deployment.
Some of the Fibre Channel Fabric Security features provided in the Cisco MDS 9000 family include:

- **FC Zoning:** allows for devices segregation within a single FC fabric, with different levels of granularity (port- and device zoning, LUN Zoning, Read-only zones) (see section 3.1)
- **Virtual SANs (VSANs):** provide segregation between virtual fabrics (see section 3.2)
- **Port security:** provides access control at the port level (see section 3.3)
- **FC-SP authentication:** verifies switch and device identity preventing identity spoofing and unauthorized access (see section 3.4)
- **FCsec encapsulation:** provides frame by frame security (see section 3.4.2)
- **Numerous other FC security features**: FC Address Security (Principal switch selection, RCF rejection, Fabric Binding, FC_ID security) (see section 3.5)

3.1 Zoning

Zoning is the security mechanism within Fibre Channel used to restrict communication between devices within the same Fibre Channel fabric. Zoning carves a FC fabric into multiple partitions; devices in one zone cannot see devices in any other zone.

With many different types of servers and storage devices on the network the need for security is critical. For example, if a host were to gain access to a disk being used by another host, potentially with a different operating system, the data on this disk could become corrupted. To avoid any compromise of critical data within the SAN, zoning allows the user to overlay a security map dictating which devices, namely hosts, can see which targets thereby reducing the risk of data loss.

Typical uses for zoning include:

- Separate devices that use different operating systems (OSes)
  This is useful to protect different OSes from treating disks formatted by other OSes as ‘blank disks’ – and corrupt/re-use each others’ storage.

- Separate devices that have no need to communicate with other devices in the fabric or have classified data.

- Separate devices into department, administrative, or some other functional groupings.

3.1.1 Zone Membership

Zoning itself comprises a number of different concepts and characteristics. These are outlined below and illustrated in Figure 1:

- A *zone* consists of multiple *zone members*.

- Members in a zone (*zone members*) can access each other; members in different zones cannot access each other.

- If zoning is not activated, all devices are members of the *default zone*.
  Devices within the *default zone* are zoned according to the *default zone policy* (normally set to deny).

- If zoning is activated, any device that is not in an *active zone* (a zone that is part of an *active zone set*) is a member of the *default zone*.

- Devices can belong to more than one zone.

- A *zone set* consists of one or more *zones*.

  - A *zone set* can be activated or deactivated as a single entity across all switches in the fabric.

  - Only one *zone set* can be activated at any time.

  - A *zone* can be a member of more than one *zone set*.

- Zoning can be administered from any switch in the fabric.

  - Because zoning information is distributed to all switches in the fabric, zoning changes made on one switch are available in all switches.

  - If a new switch is added to an existing fabric, zone sets are acquired by the new switch and any zones configured on the new switch are merged into the full zone set. (Note that if there is a conflict which prevents a *zone merge*, the link isn’t brought up and the port is put into an isolated state).

- Zone changes can be configured non-disruptively.

  - New *zones* and *zone sets* can be activated without interrupting traffic on unaffected ports or devices.

- Default zone membership includes all ports or WWNs that do not have a specific membership association. Access between *default zone* members is controlled by the default zone policy.

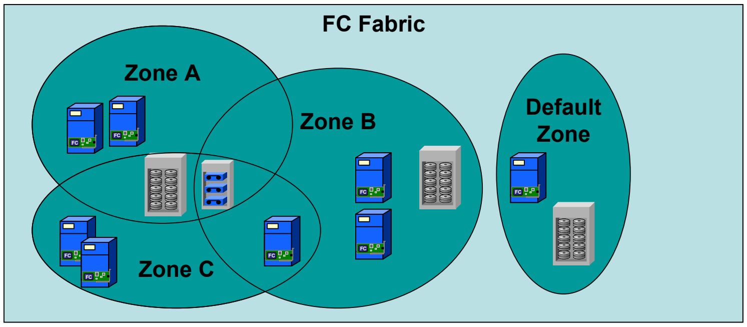


Figure 1 Fibre Channel Zoning

3.1.2 Types of Zoning

There are two types of Zoning within FC: 'Soft' Zoning and 'Hard' Zoning.

- Soft Zoning refers to software-based Zoning; that is, zoning is enforced through control-plane software on FC switches themselves – in the FC Name Server service. When devices connect to a FC fabric, they use the Name Server to correlate World Wide Names (WWNs) to Fibre Channel IDs (FC_IDs). With soft zoning, a FC switch responding to a Name Server query from a device will only respond with a list of those devices registered in the name server that are in the same zone(s) as that of the querying device.

- Hard Zoning refers to hardware-based Zoning. Hard Zoning is enforced through hardware Access Control Lists (ACLs) which are applied to every FC frame that is switched.

Since Soft Zoning doesn't enforce access-controls on a per-frame basis, Soft Zoning in itself isn't actually secure as there is nothing to protect against a rogue device scanning all FC_IDs.

The Cisco MDS 9000 product family support Hard Zoning enforced via ACLs in hardware and Soft Zoning via the name server. All members of the Cisco MDS 9000 product family support hardware-enforced zoning for 2,000 zones and 20,000 zone members.

3.1.3 Zone Membership Types

A 'Zone Member' is used to uniquely identify device(s) to be included in a Zone. On the Cisco MDS 9000 family of multilayer switches, Zone Members can be identified using any of 7 possible ways:

1. **Device Port World Wide Name (pWWN)**

   Zoning based on Device pWWN means that zone membership is determined using the Device pWWN of an N_Port attached to the FC switch – i.e. unique identification of a given FC port on a given device. Any new device plugged into the FC switch in the same FC port, will have a different pWWN.

2. **Fabric Port World Wide Name (fWWN)**

Zoning based on Fabric pWWN means that zone membership is determined using the pWWN of the N_port of the FC switch where the device is attached to – i.e. zone membership is determined by the device being plugged into a given port on a given linecard in a FC switch.

3. **Fibre Channel ID (FC_ID)**

   Zoning based on the FC_ID of an N_Port attached to the switch. This means that zone membership is determined by the FC_ID assigned to a device by the Fabric Domain Controller. (note: for this kind of zoning to be useful, one would typically make use of *Static FC_ID Allocation*)

4. **Interface and Switch WWN (sWWN)**

   Zoning based on the switch-port interface that a device is attached to. This form of zoning is typically referred to as interface-based zoning. This form of zoning allows zone membership to be globally specified based on a given switch name (sWWN) and interface on that switch.
   e.g. '*member interface fc3/10 swwn 20:00:00:05:30:00:91:9e*'.

   Note: Interface and sWWN-based Zoning is not yet part of any ANSI Fibre Channel standard, therefore this form of zoning is unavailable for multi-vendor fabrics.

5. **Domain_ID and Port Number**

   Zoning membership is based on the Domain_ID of the switch and the port number on the switch to which the device is attached to. Since Domain_IDs may be allocated dynamically, use of Static Domain_ID Allocations would be recommended. 'Port Number' is specified as a port index between 0 and 255. On many vendors' FC switches, it is difficult to associate a Port Number to a given module slot & port combination, so it is recommended to use zoning based on 'Interface and Switch WWN (sWWN)' whenever possible.

   Note: Domain_ID and Port Number-based Zoning is specified in the appropriate ANSI Fibre Channel standards, but given each FC vendor may number ports in different ways, this form of zoning is not recommended for multi-vendor switch fabrics and is therefore unavailable for standards-based *Interop* mode. It can function with other vendors' legacy switches that are compatible with *interop mode 1*.

6. **Symbolic Node Name**

   Symbolic Node Name zoning allows zone members to be defined using their unique symbolic node name, such as the iSCSI Qualified Name (IQN) or IP Address associated with iSCSI devices. In this manner, iSCSI devices can have dynamic pWWN and nWWNs associated with them, but continue to make use of zoning membership using their globally-unique IQN.
   e.g. '*member symbolic-nodename iqn.1987-05.com.cisco.01.bock-bock-au*'.

   Note: Symbolic Node Name-based Zoning is not yet part of any ANSI Fibre Channel standard, therefore this form of zoning is unavailable for multi-vendor fabrics.

7. **FC Alias**

   Zoning membership based on a previously defined fcalias.
   e.g. '*member fcalias host-bock-bock-au*'.

FC Zoning should always be deployed in a FC fabric, if not from a security perspective, then from the perspective of minimizing loss of data.

In general, it is recommended that as many Zones be used as there are hosts communicating with storage.
For example, if there are 2 hosts each communicating with 3 storage devices, it would be recommended to use 2 zones.

From an ease-of-configuration perspective, *FCAliases* should be used wherever possible, as they make identification of devices easier than trying to identify devices through their 64-bit World Wide Name.

It is also recommended that zoning administration generally be confined to a single FC switch within a FC fabric, in order to ensure that there is no possibility of activating an incomplete ZoneSet (as can happen if the Full ZoneSet is not consistent across FC switches). See section 3.1.6 details how the Full ZoneSet differs from the Active ZoneSet. Also see section 3.1.7 on how to minimize the possibility of this through the use of Full ZoneSet distribution.

3.1.4 LUN Zoning

Disk Arrays typically have multiple Logical Units on them. Standard FC Zoning extends down to the switch port level or down to the WWN of the port, but not down to the LUN level.

This means that any fabric containing disk arrays with multiple LUNs needs security policies configured on *both* the disk array (or multiple disk arrays) and on the FC switches themselves.

LUN Zoning is a feature specific to switches in the Cisco MDS 9000 Family introduced in SAN-OS 1.2 that allows zoning to extend to individual LUNs within the same WWN. This means that the centralized zoning policy configured on the FC switches can extend to hardware-enforcing zoning down to individual LUNs in disk arrays.

This is shown in Figure 2. In this case, there are 3 zones (zones A-C) that allow individual hosts to see LUNs assigned to them but not LUNs that aren't assigned to them. Zone D (a standard pWWN-based zone) allows a tape library to access all LUNs to perform a LAN- free backup.

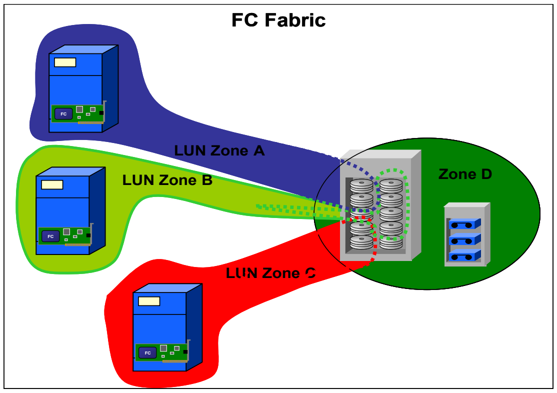


Figure 2 LUN Zoning

(Note: LUN Zoning is a feature enabled by the Enterprise License Package)

3.1.5 Read-Only Zones

Standard FC Zoning is used to permit devices to communicate with each other. Standard FC Zoning cannot perform any advanced filtering – for example, by blocking or allowing specific I/O Operations such as a Write I/O command.

The Cisco MDS 9000 Family provides the ability to enforce 'read only' zones in hardware. That is, the switch can enforce read-only access to a given device (e.g. a disk) and will block any write requests. When used in conjunction with LUN Zoning, read-only or read-write access can be granted for specific hosts to specific LUNs. Read-only Zoning was introduced with SAN-OS 1.2. This functionality is available on every port across the entire Cisco MDS 9000 product family.

(Note: Read-Only Zones is a feature enabled by the Enterprise License Package)

3.1.6 Active ZoneSet versus the Full ZoneSet

Before configuring a zone set, consider the following guidelines:

- A FC fabric can have multiple *zone sets* but only one *zone set* can be active at any given time.
- When a *zone set* is created, that *zone set* becomes a part of the *full zone set*.
- When a *zone set* is activated, a copy of the *zone set* from the *full zone set* is used to enforce zoning, and is called the *active zone set*. An *active zone set* cannot be modified. A *zone* that is part of an *active zone set* is called an *active zone*. You can activate a *zone set* using the 'zoneset activate name' command.
- The *full zone set* can be modified even if a *zone set* with the same name is active. The changes do not take effect until the *zone set* is activated with the 'zoneset activate name' command.
- When the activation is done, the *active zone* set is automatically stored in persistent configuration (Flash RAM). This enables the switch to preserve the *active zone set* information across switch resets. There is no need to issue the 'copy running-config startup-config' command to store the *active zone set*.
- To explicitly store the *full zone set*, the 'copy running-config startup-config' command is required otherwise the *full zone set* won't be available across switch resets.
- All switches in the FC fabric receive the *active zone set* so they can enforce zoning in their respective switches.
- Any device that is not part of the *active zone set* belongs to the *default zone*. This is not distributed to other switches.

3.1.7 Full ZoneSet Distribution

The Cisco MDS 9000 Family can be configured to distribute the *full zone set* to other switches within a FC fabric (as opposed to just the *active zone set)*, helping ensure that zoning is consistent across the fabric, regardless of the FC switch chosen to perform the zoning configuration on. The configuration command to enable this is 'zoneset distribute full vsan <n>'.

Since the *full zone set* is distributed, all zoning configuration including *fcaliases* are distributed also.

Note that *full zone set* distribution isn't supported by other FC switch vendors, and therefore cannot be enabled if the VSAN is set to *interop* mode. If a VSAN is to be connected to non-Cisco FC switches, *full zone set* distribution needs to be disabled using the configuration command 'no zoneset distribute full vsan <n>'.

From a security perspective, Full ZoneSet Distribution should be used wherever possible as it helps minimize the probability of inadvertent misconfigurations of zoning across multiple switches, potentially resulting in zone-merge failures when new zoning is applied.

3.2 Virtual SANs (VSANs)

Virtual SANs (VSANs) can be used to achieve higher security and greater stability in FC fabrics by providing isolation among devices that are physically connected to the same physical fabric. VSANs can be used to create multiple logical SANs over a common physical infrastructure. Each VSAN can contain up to 239 switches and has an independent address space which allows identical Fibre Channel IDs (FC_IDs) to be used simultaneously in different VSANs.

Fabrics built using VSANs have the following advantages over fabrics built without VSANs:

- Traffic isolation —Traffic is contained within VSAN boundaries and devices reside only in one VSAN ensuring absolute separation between groups of devices – for all of unicast, broadcast and multicast traffic.
- Scalability—VSANs are overlaid on top of a single physical fabric. The ability to create several logical VSAN layers increases the scalability of the SAN.
- Per VSAN fabric services—Replication of fabric services on a per VSAN basis provides increased scalability and availability.
- Redundancy—Several VSANs created on the same physical SAN ensure redundancy. If one VSAN fails, redundant protection is provided (to another VSAN in the same physical SAN) by a configured backup path between the host and the device.
- Ease of configuration—Users can be added, moved, or changed between VSANs without changing the physical structure of a SAN. Moving a device from one VSAN to another only requires configuration at the port level, not at a physical level.
- Shared topology—Multiple VSANs can share the same physical topology.
- Independent FC_IDs—The same Fibre Channel IDs (FC_IDs) can be assigned to hosts in another VSAN, thus increasing VSAN scalability.
- Required protocols—Every instance of a VSAN runs all required protocols such as FSPF, domain manager, and zoning. Disruption of one of these services in one VSAN doesn't affect the other VSANs.
- Independent—Fabric-related configurations in one VSAN do not affect the associated traffic in another VSAN.

- Containment—Events causing traffic disruptions in one VSAN are contained within that VSAN, and are not propagated to other VSANs.
- Isolation—No communication is possible between VSANs, unless explicitly configured using Inter VSAN Routing.

One of the biggest benefits of using VSANs is that of fault-containment. VSANs allows the virtual segregation of a set of physical switches into multiple logical switches, the net benefit being that any disruption can be contained within a single segment of the SAN.

Figure 3 shows a typical High Availability (HA) SAN design employed today in architecting a SAN that connects two data centers.   While dual fabrics have been used within each data centre, connecting both data centers together has resulted in creating two FC fabrics that *merged* across a WAN link.

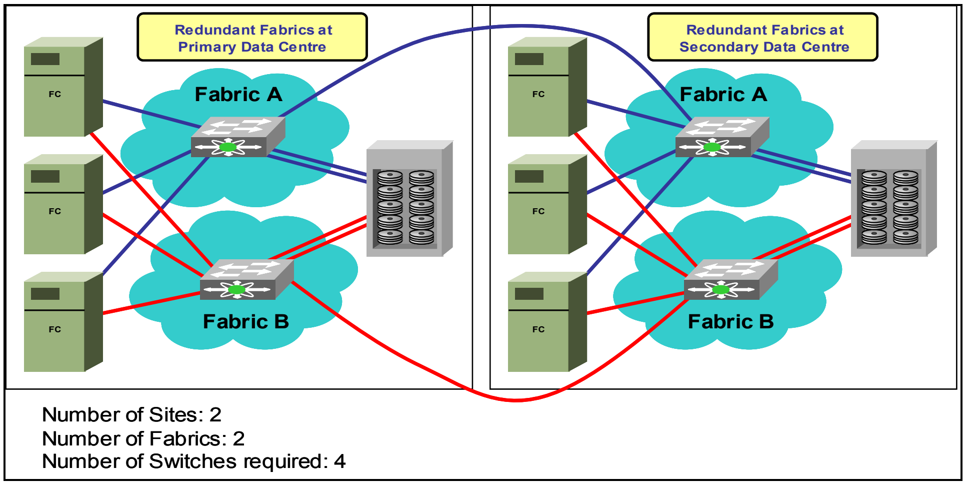
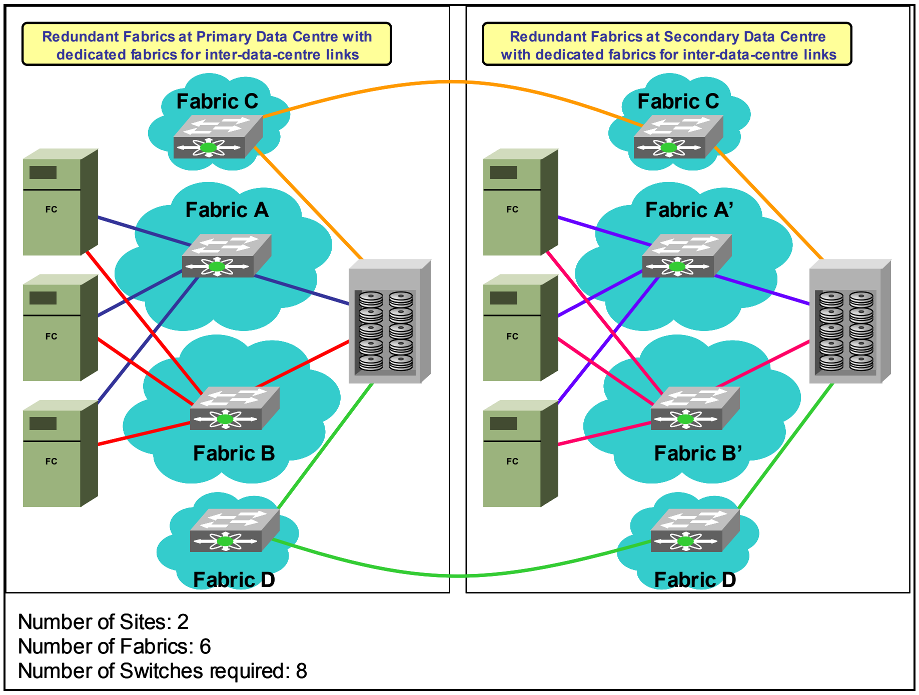

Redundant Fabrics at Primary Data Centre
Redundant Fabrics at Secondary Data Centre
Fabric A
Fabric B
Fabric A
Fabric B
FC
FC
FC
FC
FC
FC
Number of Sites: 2
Number of Fabrics: 2
Number of Switches required: 4

Figure 3 Dual Site HA SAN Design with dual fabrics at each site

From a security perspective, anyone with malicious intent can use a single compromised host in one data centre to potentially cause disruptions that span both data centers.

Because of this, many Enterprise HA designs utilize dedicated switches for the inter- data centre links – in essence, creating more FC fabrics. This is shown in Figure 4 below:


Redundant Fabrics at Primary Data Centre with dedicated fabrics for inter-data-centre links
Redundant Fabrics at Secondary Data Centre with dedicated fabrics for inter-data-centre links
Fabric C
Fabric C
Fabric A
Fabric A'
Fabric B
Fabric B'
Fabric D
Fabric D
FC
FC
FC
FC
FC
FC

Number of Sites: 2
Number of Fabrics: 6
Number of Switches required: 8

Figure 4 Dual Site HA SAN Design with dedicated inter- and intra- data centre fabrics

In this design, redundant FC fabrics are used for intra- data centre and inter- data centre traffic. This is one area where VSANs can be used to provide fault containment while decreasing the overall amount of infrastructure required. Figure 5 below shows the same logical SAN design as in Figure 4, but by using VSANs, the number of FC switches required is halved.

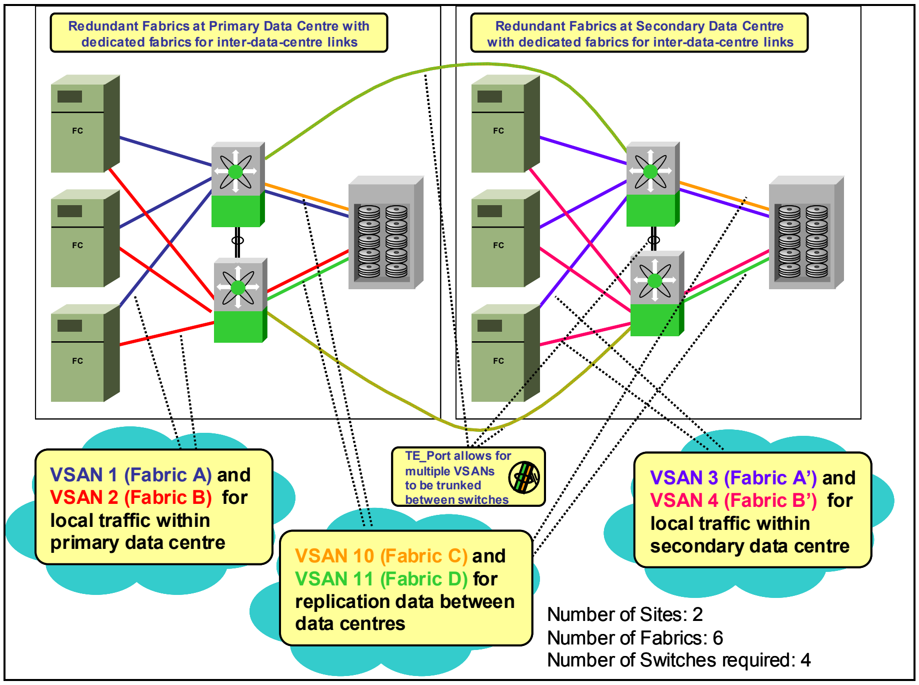

Redundant Fabrics at Primary Data Centre with dedicated fabrics for inter-data-centre links
Redundant Fabrics at Secondary Data Centre with dedicated fabrics for inter-data-centre links
FC
FC
FC
FC
FC
FC
TE_Port allows for multiple VSANs to be trunked between switches
VSAN 1 (Fabric A) and VSAN 2 (Fabric B) for local traffic within primary data centre
VSAN 10 (Fabric C) and VSAN 11 (Fabric D) for replication data between data centres
VSAN 3 (Fabric A') and VSAN 4 (Fabric B') for local traffic within secondary data centre
Number of Sites: 2
Number of Fabrics: 6
Number of Switches required: 4

Figure 5 Dual Site HA SAN Design with dedicated inter- and intra- data centre VSANs

In this design, the number of physical switches is kept to a minimum and VSANs are used to segregate the inter- data centre fabrics (dotted lines) from the intra- data centre fabrics (solid lines). In this manner, any fault in a single fabric won't impact other fabrics, allowing for maximized system availability. From a security perspective, any compromised devices in one fabric cannot impact devices in other fabrics.

Isolating inter- data centre traffic from intra- data centre traffic is just one use for VSANs. Other possible uses include:

- Different customers in storage provider data centers
- Production or test in an enterprise network
- Low and high security requirements
- Backup traffic on separate VSANs
- Replicating data from user traffic

Note that VSANs do not preclude FC Zoning; Zoning may still exist within each VSAN. The key is that VSANs provide a set of fabric services per VSAN and thus offer complete segregation of not only data traffic but also control-plane functionality.

It is recommended to use VSANs wherever strict security is required to isolate fabric devices from one another or where it is important to minimize the scope of potential disruptions to fabric services (e.g. loss of path to principal switch).

3.2.1 Relationship of VSANs to Zones

VSANs are used to divide a redundant physical SAN infrastructure into separate virtual SAN islands each with its own set of Fibre Channel fabric services. With each VSAN supporting an independent set of Fibre Channel services, a VSAN-enabled infrastructure can house numerous applications without the concern for fabric resource or event conflicts between these virtual environments. Once the physical fabric has been divided, zoning can then be used to implement a security layout within each VSAN that is tuned to the needs of each application within each VSAN. Figure 6 shows the relationship of VSANs to Zones.

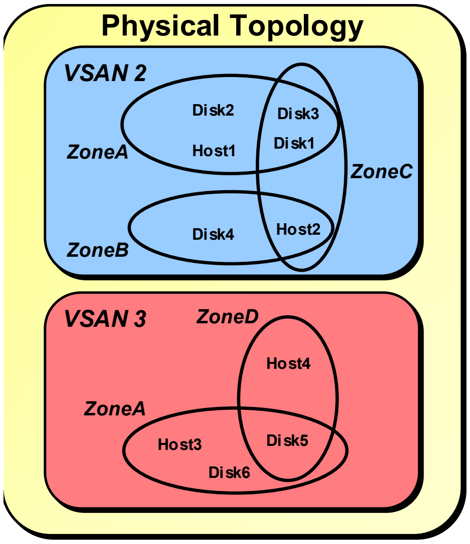


Figure 6 Relationship of Zones to VSANs

3.2.2 VSAN Membership

For most port types (F_Port, FL_Port, L_Port, E_Port, B_Port), VSAN membership on the Cisco MDS 9000 product family can be assigned on a port-by-port basis, with the given port belonging only to a single VSAN.

Trunking ports (TE_Port) between switches can trunk multiple VSANs and for those ports a list of VSANs that are allowed to be trunked is configurable.

For Gigabit Ethernet ports on the IP Storage Services (IPS), ports used with iSCSI may have multiple iSCSI sessions active on a single Gigabit Ethernet port. VSAN membership can be specified on a per-iSCSI session basis.

3.2.3 Default & Isolated VSAN

Up to 256 VSANs can be configured on a single switch. There is always a default VSAN (typically VSAN 1) and an isolated VSAN (VSAN 4094). User-specified VSAN IDs range from 2 to 4093.

Default VSAN

The factory settings for switches in the Cisco MDS 9000 Family have only the default VSAN 1 enabled. If you do not need more than one VSAN for a switch, use this default VSAN as the implicit parameter during configuration. If no VSANs are configured, all devices in the fabric are considered part of the default VSAN. By default, all ports are assigned to the default VSAN.

Isolated VSAN

VSAN 4094 is an isolated VSAN. All non-trunking ports are transferred to this VSAN when the VSAN to which they belong is deleted. This avoids an implicit transfer of ports to the default VSAN or to another configured VSAN. All ports in the deleted VSAN are isolated (disabled).

From a security perspective, it is recommended that spare (unused) FC ports be put in their own VSAN with the default FC Zoning policy on that VSAN be set to 'deny'. This would allow new devices being connected to a fabric to achieve 'link up' status thereby validating physical-layer connectivity, but would block devices from communicating with each other, thereby preventing any potential fabric disruptions or security violations.

This is also useful when Port Security (section 3.3), Fabric Binding (section 3.5.3), FC-SP-based Switch-to-Switch or Host-to-Switch (section 3.4.1) security features are used, as it allows one to gather HBA WWN details without adding the device into a production fabric.

3.2.4 Inter VSAN Routing (IVR)

VSANs enable a unique degree of isolation through the construction of virtual fabrics on top of a common physical fabric. However, SAN architects may wish some devices to be available to communicate to devices in multiple VSANs, but not at the expense of merging the multiple fabrics together. Prior to Inter VSAN Routing (IVR), the only way this could be accomplished was by using multiple interfaces on the device to connect the device into multiple VSANs by using multiple physical switch ports. With Inter VSAN Routing, it is possible to create a path from a device in one VSAN to potentially multiple devices in different VSANs – without merging the individual VSAN fabrics and hence fault domains.

Inter VSAN Routing works on the concept of using an '*Inter VSAN Routing Zone*'. *IVR Zones* may span multiple devices in multiple VSANs; any devices across VSANs which require connectivity need to exist within the same *IVR Zone*.

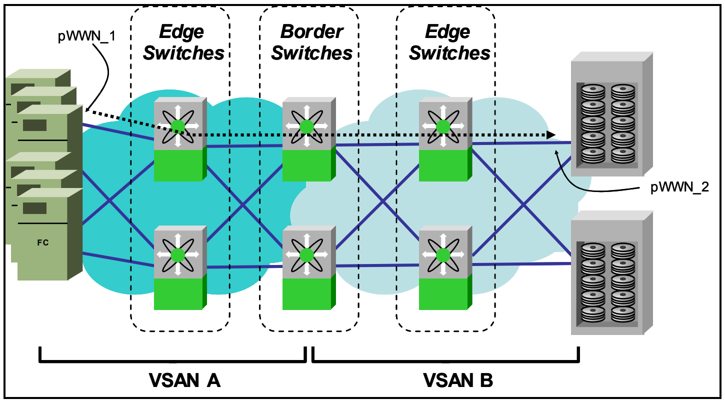


Figure 7 Inter VSAN Routing between pWWN_1 and pWWN_2

IVR Zones

Using the same zone administration procedures and management tasks already familiar to SAN administrators (see section 3.1), an Inter-VSAN Routing instance is created by simply defining a new type of zone called an Inter-VSAN zone. An inter-VSAN zone is defined as any other zone however containing devices from participating edge VSANs requiring cross-VSAN connectivity. The IVR technology automatically determines which routes and name server entries require cross-propagation based on the inter-VSAN zone definition. Using the familiar zoning mechanism, SAN administrators need not learn any new protocols or new management tools to enable and manage the IVR service.

IVR Border Switches

In order to accomplish IVR connectivity, at least one switch in the fabric must act as a transport, or *IVR border switch* providing a transport path between VSANs. As shown in Figure 7, device pWWN_1 in VSAN_1 needs to communicate with device pWWN_2 in VSAN_2. The *border* switches acts as transport between the devices in the separate VSANs and as is shown, multiple IVR *border* switches can be used to provide redundant paths. The *border* switches inject only selective FSPF routes into their respective local VSANs relating to devices within the defined IVR VSANs requiring IVR connectivity. Specifically speaking, only FSPF routes to domains (switches) in the defined IVR VSANs containing these devices are cross-propagated. In addition, name server database entries containing identification for the communicating devices within defined Inter-VSAN zones are also cross-propagated within the defined IVR VSANs. If all devices were contained within a single VSAN, all database information in every switch would be synchronized. However with the IVR feature, only a small subset of routing and name server information for the devices requiring IVR communication as defined by the Inter-VSAN zones is cross propagated – again without merging the participating virtual fabrics (VSANs).

IVR Transit VSANs

An additional IVR configuration option is the use of an IVR Transit VSAN. Transit VSANs are additional VSANs created to bridge two edge IVR VSANs thereby alleviating the need to extend one IVR VSAN to the another to setup the IVR service. In a sense, a transit VSAN extends the reach provided by a border switch to reach participating IVR VSANs. As shown in Figure 7, the two edge IVR VSANs which had devices needing to communicate were already connected common VSAN border switches. In that simple configuration, no transit VSAN is required. In fact, a transit VSAN is always optional, but is recommended to maintain additional isolation in some configurations such as wide area SAN extension solutions. Figure 8 shows how a transit VSAN could be used to connect devices from three different edge VSANs across a wide area network. Note that a transit VSAN could contain devices as well and multiple transit VSANs could exist between edge VSANs. As shown in Figure 8, the IVR service is established without the need to extend any of the edge IVR VSANs to a common border switch. This prevents the need to extend a local IVR VSAN across a higher latency long distance network. Even though selective connectivity is enabled through the IVR function, the three edge VSANs remain completely isolated from a control protocol perspective and even from a management perspective when VSAN-based

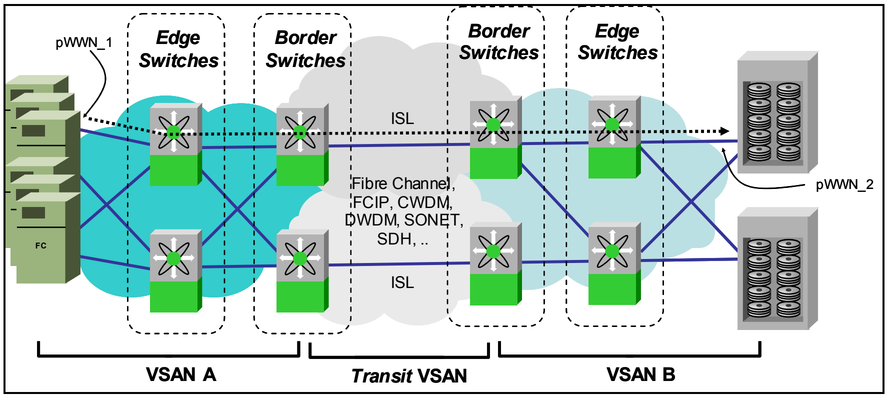


Figure 8 Inter VSAN Routing with Transit VSAN

Roles-Based Access Control (RBAC) (see section 5.5) is used. Legacy non-Cisco Fibre Channel switches can also participate in any edge IVR VSAN and can communicate across Inter-VSAN boundaries but cannot participate as border switches.

With the example in Figure 8, fabric control traffic for IVR VSAN_A and VSAN_B is terminated locally at the border switches and control traffic for IVR VSAN_C is terminated locally at remote border switches. The transit VSAN is the only VSAN that spans over the wide-area connection between border switches. Therefore, any fabric events that occur in either of the edge IVR VSANs, whether it be normal control traffic or fault events, remain within the respective edge VSANs and do not traverse nor impact other IVR VSANs. In a stable environment, all switches participating in IVR exchange negligible amounts of control traffic. During normal ISL link initialization between two given switches, the switches exchange a small amount of control protocol information. In addition to this information, two switches border a transit VSAN also exchange FSPF and name server database information strictly pertaining to devices that will be IVR connected over the transit VSAN. This exchange adds approximately two Fibre Channel frames worth of traffic for each device requiring IVR connectivity.

(Note: Inter VSAN Routing is a feature enabled by the Enterprise License Package)

3.3 Port Security

Typically, any Fibre Channel device in a SAN can attach to any SAN switch port and access SAN services based on zone membership. Port Security is a feature that was introduced into the Cisco MDS 9000 family in SAN-OS 1.2 that is used to prevents unauthorized access to a switch port by binding specific WWN(s) as having access to one or more given switch ports.

When Port Security is enabled on a switch port, all devices connecting to that port must be in the port-security database and must be listed in the database as bound to a given port. If both these criteria aren’t met, the port won’t ever achieve an operationally ‘active’ state and the devices connected to the port will be denied access to the SAN.

In the case of a storage device or host, the port name (pWWN) or node name (nWWN) can be used to lock authorized storage devices to a specific switch port. In the case of an E_Port/TE_Port, the switch name (sWWN) is used to bind authorized switches to a given switch port.

When Port Security is enabled on a port:

- Login requests from unauthorized Fibre Channel devices (Nx ports) and switches (xE ports) are rejected.
- All intrusion attempts are reported to the SAN administrator.

By default, the port security feature is disabled. To enable/disable the port security feature on a per-VSAN basis, the '*[no] port-security activate vsan <n> [no-auto-learn]*' CLI command is used.
The 'auto-learn' option allows allow for rapid migration across to Port Security when it is being activated for the first time. Rather than manually secure each port, auto-learn allows for automatically population of the port-security database based on an inventory of currently-connected devices.

Entries can be manually added into the port-security database, using the following commands:

switch# **configure terminal**	Enters configuration mode.
switch(config)# **port-security database vsan 1**	[Deletes]/Creates the port-security database for VSAN 1.
switch(config-port-security)# **[no] any-wwn interface fc3/1**	[Disables]/Enables any WWN to log in through the interface fc3/1.
switch(config-port-security)# **[no] pwwn 20:11:00:33:11:00:2a:4a fwwn 20:81:00:44:22:00:4a:9e**	[Disables]/Enables the specified pWWN to only log in through the specified fWWN.
switch(config-port-security)# **[no] nwwn 26:33:22:00:55:05:3d:4c fwwn 20:81:00:44:22:00:4a:9e**	[Disables]/Enables the specified nWWN to only log in through the specified fWWN.
switch(config-port-security)# **[no] pwwn 20:11:33:11:00:2a:4a:66**	[Disables]/Enables the specified pWWN to only login through any port on the local switch.
switch(config-port-security)# **[no] swwn 20:01:33:11:00:2a:4a:66 interface port-channel 5**	[Disables]/Enables the specified sWWN to only login through Port Channel 5.
switch(config-port-security)# **[no] any-wwn interface fc1/1 - fc1/8**	[Disables]/Enables any WWN to login through interfaces fc1/1-8.

The Port Security Database is much like the Zoning Database in that there are both 'Active' and 'Configured' databases; changes are only ever applied to the 'configured' database and when activated, it is atomically applied by becoming the 'active' database. To activate/deactivate the configured Port Security Database, the '*[no] port-security activate vsan <n> [force]*' CLI command is used.

The Configured Port Security Database is stored in NVRAM as part of the switch configuration when the '*copy running-config startup-config*' CLI command is issued. Since this is in text format, the configuration can be moved easily from one switch to another.

The '*port-security database copy vsan <n>*' CLI command can be used to copy from the active to configured database.

The '*port-security database diff <active|config> vsan <n>*' CLI command can be used to view the differences between the active and configured databases. This command can be useful when resolving conflicts.

The '*clear port security port-security statistics vsan <n>*' CLI command is used to clear all existing statistics from the port security database for a specified VSAN.

The '*clear port-security database auto-learn interface <fcX/Y> vsan <n>*' CLI command is used to clear any learnt entries in the active database for a specified interface within a VSAN. To remove all learnt entries in a given VSAN the '*clear port-security database auto-learn vsan <n>*' CLI command is used.

Numerous CLI commands exist to view the status of Port Security. Output from many of these is shown in Figure 9 below:

```
switch# show port-security status
VSAN 1 :Activated database, auto-learning is enabled
VSAN 2 :No Active database, auto-learning is disabled
switch# show port-security database
--------------------------------------------------------------------------------
Vsan Logging-in Entity Logging-in Point (Interface)
--------------------------------------------------------------------------------
1 * 20:81:00:44:22:00:4a:9e (fc3/1)
1 50:06:04:82:bc:01:c3:84(pwwn) 20:0c:00:05:30:00:95:de (fc1/12)[learnt]
1 21:00:00:e0:8b:06:d9:1d(pwwn) 20:0d:00:05:30:00:95:de (fc1/13)[learnt]
1 26:33:22:00:55:05:3d:4c(nwwn) 20:81:00:44:22:00:4a:9e (fc3/1)
 [Total 4 entries]
switch# show port-security violations
-------------------------------------------------------------------------------
VSAN Interface Logging-in Entity Time [Repeat count]
-------------------------------------------------------------------------------
switch# show port-security violations last 50
-------------------------------------------------------------------------------
VSAN Interface Logging-in Entity Time [Repeat count]
-------------------------------------------------------------------------------
```

Figure 9 Output from various Port Security Status CLI commands

(Note: Port Security is a feature enabled by the Enterprise License Package)

It is recommended that Port Security be enabled in a FC fabric to prevent both deliberate unauthorized access to switch ports and inadvertent mis-cabling of existing switch ports.

Port Security should be used in combination with FC-SP-based Switch-to-Switch or Host-to-Switch (section 3.4.1) security to provide appropriate countermeasures to prevent access to unauthorized data via spoofed or hijacked WWNs where traditional Port Security would be vulnerable.

When enabling Port Security for the first time, it is prudent to make use of the *auto-learn* capabilities to ensure that no disruption to traffic occurs as a result of a mis-typed WWN or switch-port. Once auto-learn has populated the port security database, auto-learn should be permanently disabled.

3.4 Fibre Channel Security Protocols (FC-SP)

Historically Fibre Channel has been a 'plug and play' technology with very little restrictions on what devices can be connected to a fabric and on who can send and receive frames. FC Security measures have been traditionally been specified through device configuration (e.g. restricting WWNs through techniques like LUN Security) and FC Zoning. Devices have typically been secured or exchange security information using weak authentication mechanisms, often using clear text passwords. The growth of SANs beyond the protected doors of the data center introduces a new set of requirements.

To address this concerns Cisco has widely contributed to the Fibre Channel Security Protocols (FC-SP) working group of the INCITS T11.3 (Fibre Channel) committee. The result is the draft of the future FC-SP standard that extends the Fibre Channel architecture with:

- switch-to-switch, switch-to-device, and device-to-device authentication within the login phase
- frame-by-frame FC-2 level encryption (generally referred as FCsec) that provides origin authentication, integrity, anti-replay and privacy protection to each frame sent over the wire
- consistent and secure policy distribution across the fabric

With FC-SP, switches, storage devices and hosts shall be able to prove their identity through a reliable and manageable authentication mechanism. FC-SP can protect against impersonation attacks from rogue hosts, disks, or fabric switches, as well as providing protection from common misconfigurations when cabling devices in a fabric. With FC-SP, Fibre Channel traffic can be secured on a frame-by-frame basis to prevent snooping and hijacking, even over untrusted links. A consistent set of policies and management actions are propagated through the fabric to provide a uniform level of security across the entire fabric.

3.4.1 FC-SP Authentication – Switch-to-Switch and Host-to-Switch Authentication

Fibre Channel Security Protocol (FC-SP) is being introduced into the Cisco MDS 9000 family in SAN-OS 1.3. This release will include support for Data Integrity (tamper-proof) and Authentication (non-repudiation) for both Switch-to-Switch and Host-to-Switch communication. Authentication is based on Challenge Handshake Authentication Protocol (CHAP) with Diffe-Hellman (DH) extensions (DH-CHAP).

Authentication can be performed locally in the switch or remotely through a centralized RADIUS or TACACS+ server. If the authentication credentials cannot be ascertained or the authentication check fails, a switch or host will be blocked from joining a FC fabric. Figure 10 shows how FC-SP can be used to authenticate servers and forbid untrusted devices from being able to join a fabric.

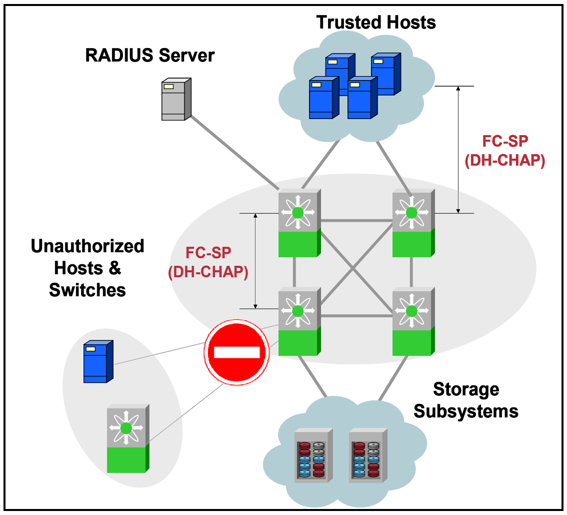


Figure 10 Switch-to-Switch and Host-to-Switch authentication within FC-SP

By default, the FC-SP feature is disabled in all switches in the Cisco MDS 9000 Family. When FC-SP is enabled, DH-CHAP is also automatically enabled. The following configuration excerpt shows how to enable FC-SP:

switch(config)# **fcsp enable**	Enables the FC-SP DHCHAP in this switch
switch(config)# **no fcsp enable**	Disables FC-SP DHCHAP in this switch (default)

Once FC-SP has been enabled on the switch, it can be activated on a per-port basis. Each port may be configured into one of four security modes:

- **Off**
  In this mode, the FC port will not support the FC-SP protocol. Authentication Requests received on this port will be rejected with a 'Command Not Supported' reject message and no FC-SP Authentication Requests will be initiated by the switch to devices on this port.

- **On**
  In this mode, the FC port is operating in strict authentication mode of FC-SP. The switch will enforce Authentication on this port and will refuse to enable the port if the attached device either fails authentication or does not support FC-SP.

- **Auto (active)**
  In this mode, FC-SP is used for E_Port negotiation regardless of whether FC-SP is enabled on the neighbor switch or not. If the attached device doesn't support FC-SP, the port is enabled. If the attached device does support FC-SP but

fails to authenticate, the port will be disabled. Ports in this mode are also allowed to try to re-authenticate within a specified time period. The default time period is 0 (does not allow re-authentication).

- **Auto (passive)**
  This is the default security mode of FC-SP. In this mode, FC-SP is supported on the port but the port is allowed to be enabled if the device attached to the port doesn't support FC-SP. If the device does support FC-SP but fails to authenticate, the port will be disabled.

The following configuration excerpt shows how to enable FC-SP on a per-port basis:

switch(config)# **interface fc3/1-12**	Enter configuration for interfaces fc3/1-12
switch(config-if)# **fcsp on**	Unconditionally enables FC-SP on interfaces fc3/1-12 ("on" mode)
switch(config-if)# **no fcsp on**	Reverts to the default (auto-passive) FC-SP mode on interfaces fc3/1-12
switch(config-if)# **fcsp auto-active 0**	Configures interfaces fc3/1-12 to initiate FC-SP authentication, but does not permit reauthentication.
switch(config-if)# **fcsp auto-active 120**	Configures interfaces fc3/1-12 to initiate FC-SP authentication and permits reauthentication within two hours (120 minutes) of the initial authentication attempt.

FC-SP DH-CHAP authentication is dependent on a key-based authentication based on password hashes. Supported password hash algorithms include MD-5 (default) and SHA-1. FC-SP DH-CHAP authentication requires that there is at least one common authentication algorithm between devices to ensure successful authentication.

The Cisco MDS implementation of DH-CHAP supports all DH groups specified in the standards: 0 (Null DH group which does not perform the DH exchange, and which is the default), 1, 2, 3, or 4.

The following configuration excerpt shows how to configure the DH-CHAP hash algorithm:

switch(config)# **fcsp dhchap hash sha1**	Configures the use of the SHA-1 hash algorithm.
switch(config)# **fcsp dhchap hash md5**	Configures the use of the MD-5 hash algorithm (default)
switch(config)# **fcsp dhchap hash md5 sha1**	Configures the switch to use both MD-5 and SHA-1, preferring to use MD-5
switch(config)# **fcsp dhchap group 2 3 4**	Prioritizes the use of DH group 2, 3 and 4 over Null and 1
switch(config)# no **fcsp dhchap group 2 3 4**	Reverts to the default of having NULL (0) priority which does not perform the exchange.

The shared secret passwords used by FC-SP DH-CHAP authentication can be derived from one of three places:

- **Local FC-SP User Database authentication**
  Local FC-SP User Database-based authentication can use either MD-5 (default) or SHA-1 authentication based on passwords stored within the configuration of the local switch.
  (the configuration to enable this is shown below)

- **RADIUS based authentication**
  This requires IP connectivity to centralized RADIUS server(s).
  Only MD-5 authentication is available when using RADIUS.

- **TACACS+ based authentication**
  This requires IP connectivity to centralized TACACS+ server(s).
  Only MD-5 authentication is supported when using TACACS+.

RADIUS and TACACS+ based authentication ties into Cisco's AAA (Authentication, Authorization and Accounting) security framework, providing a security architecture that is aligned with existing security infrastructure already typically deployed within an organization for the IP networking infrastructure. Sections 5.2.2 and 5.2.3 provide details on how to configure RADIUS and TACACS+ based authentication.

DH-CHAP authentication requires a shared secret password between the initiating device and the receiving device (if each device is authenticating each other then you have two shared secrets). To do this, the switch requires database of passwords for all devices that connect to it. This database may either reside on the switch itself (Local FC-SP User Database authentication) or reside on a centralized RADIUS or TACACS+ server.

There are three approaches to managing FC-SP device authentication passwords:

- **Approach 1**: Use the same password for all FC-SP enabled devices in the fabric.
  Using the same password across all FC-SP enabled devices in a fabric is the simplest approach. When a new FC-SP device is added to the fabric, the same common password is used to authenticate that FC-SP device in this fabric.
  Note that this is also the most vulnerable approach; if the security of one FC-SP enabled device is compromised, the shared password for all devices is then compromised. This rules out approach 1 from a security perspective.

- **Approach 2**: Use a single password for each FC-SP device and maintain the list of all passwords for all devices in each FC-SP enabled device in the fabric.
  When a new FC-SP enabled device is added into a fabric, its password must be added to every other FC-SP enabled device in the fabric.
  Note that this approach is also vulnerable; if the security of one FC-SP enabled device is compromised, the passwords to all other devices are known. This is equivalent to approach 1 from a security perspective.

- **Approach 3**: Use unique passwords for each FC-SP device communicating to every other FC-SP device in the fabric.
  When a new FC-SP enabled device is added into a fabric, multiple new passwords corresponding to each other FC-SP device in the fabric must be generated and configured in each device. Even if one device is compromised, the revealed passwords cannot be used to access any of the other devices in the fabric.
  This approach requires considerable administrative management to maintain each secret password but does offer true security.

> ☞ RADIUS or TACACS+ authentication is recommended for fabrics with more than five FC-SP-enabled devices. If centralized AAA authentication isn’t desired but approach 3 is still deemed mandatory, Cisco Fabric Manager can help ease some of the burden with maintaining the FC-SP DH-CHAP password database. Refer to the Cisco MDS 9000 Family Fabric Manager User Guide for further information.

The following configuration excerpt shows how to configure DH-CHAP Local FC-SP User Database authentication passwords within the switch:

switch(config)# **fcsp dhchap password foo 20:00:00:05:30:00:1d:1e**	Configures DH-CHAP to use the password ‘foo’ when authenticating with the switch with the WWN 20:00:00:05:30:00:1d:1e
switch(config)# **no fcsp dhchap password foo 20:00:00:05:30:00:1d:1e**	Removes the DH-CHAP authentication password for the switch with the WWN 20:00:00:05:30:00:1d:1e
switch(config)# **fcsp dhchap devicename 20:00:00:05:30:00:91:9e password bar**	Configures DH-CHAP to use the password ‘foo’ when being authenticated by the switch with the WWN 20:00:00:05:30:00:91:9e
switch(config)# **no fcsp dhchap devicename 20:00:00:05:30:00:91:9e password bar**	Removes the DH-CHAP authenticated password for the switch with the WWN 20:00:00:05:30:00:91:9e

The following configuration excerpt shows how to configure FC-SP DH-CHAP authentication using RADIUS within the switch:

switch(config)# **aaa group server radius fc_sp_group**	Creates a AAA server group called “fc_sp_group” for a group of RADIUS servers
switch(config-radius)# **server 10.67.16.5** switch(config-radius)# **exit**	Adds RADIUS server at 10.67.16.5 into the “fc_sp_group” RADIUS server group
switch(config)# **aaa authentication dhchap default group fc_sp_group**	Configures DH-CHAP to use the AAA server group “fc_sp_group” for centralized DH-CHAP authentication
switch(config)# **aaa authentication dhchap default group fc_sp_group local**	Configures DH-CHAP to use the AAA server group “fc_sp_group” for centralized DH-CHAP authentication, with fallback to local user authentication if centralized authentication cannot be completed.

Note that it uses Cisco’s AAA framework, so enabling TACACS+ based authentication is very similar.
For both RADIUS and TACACS+, FC-SP DH-CHAP Authentication will use the *switch WWN* (sWWN) as the username.

(Note: FC-SP DH-CHAP is a feature enabled by the Enterprise License Package)

It is recommended that FC-SP-based switch-to-switch authentication be used on all E_Port and TE_Port links wherever switches support FC-SP.

Likewise, it is recommended that FC-SP-based host-to-switch authentication be used on all host-facing ports as the various FC HBA vendors release drivers that support FC-SP-based authentication.

FC-SP-based authentication should be considered mandatory in a secure SAN in order to prevent access to unauthorized data via spoofed or hijacked WWNs where traditional Port Security would be vulnerable.

3.4.2 FCsec: per frame Security

Recognizing the need for a per-message protection that would secure each FC frame individually, Cisco (in conjunction with EMC, QLogic, and Veritas) proposed an extension to the FC-2 frame format that allow for frame-by-frame encryption. The frame format has been called the ESP Header, since is very similar to the Encapsulating Security Payload (ESP) used to secure IP packets in IPsec. Given the overall security architecture is similar to IPsec, this aspect of the security architecture for FC is often referred to as FCsec.

The goals of the FCsec architecture are to provide a framework to protect against both active and passive attacks using the following security services:

- **Data Origin Authentication**
  Ensure that the originator of each frame is authentic.
- **Data Integrity and Anti-Replay Protection**
  Provides integrity and protects against each frame transmitted over a SAN.
- **Optional encryption for data and/or control traffic**
  Protects each frame from eavesdropping.

Other benefits of FCSec include converging the storage industry on a single set of security mechanisms, regardless of whether the storage transport was based on iSCSI, FCIP, or FC and that FCSec could be layered onto existing applications with minimal or no changes to the underlying applications.

One of the main benefits behind the use of ESP to secure an FC network is its great flexibility; it can be used to authenticate a single control messages exchanged between two devices, to authenticate all control traffic between two nodes, or to authenticate the entire data traffic exchanged between two nodes. Optional encryption can be added to any of the steps above to provide confidentiality.

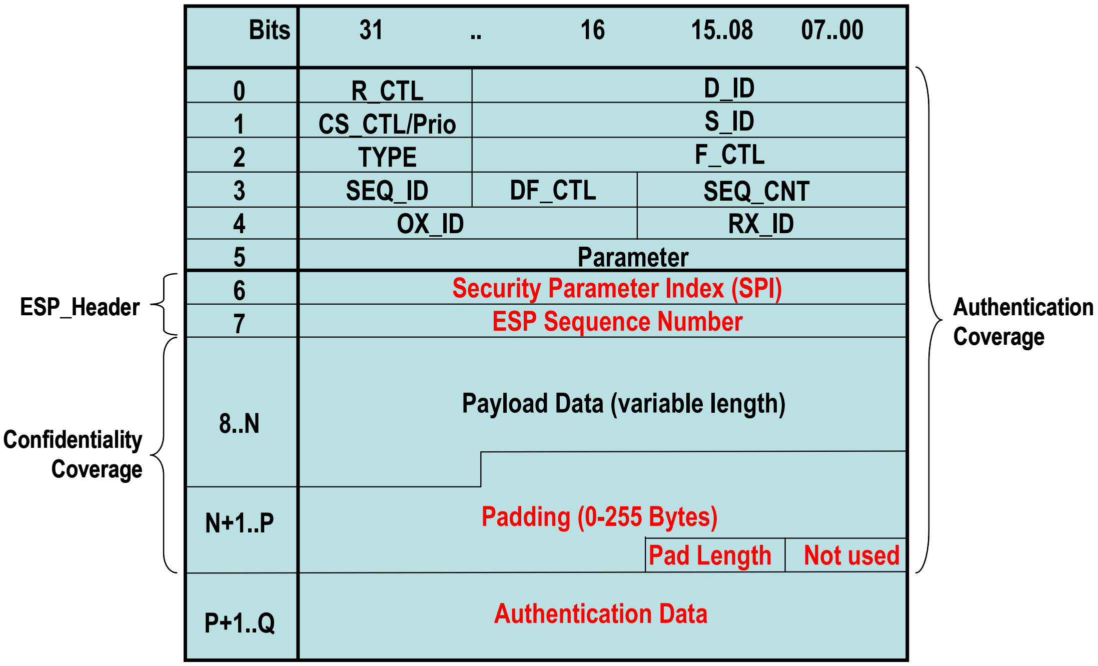
A per-entity authentication and key exchange protocol provides also a set of other services including the negotiation of the use of ESP for encapsulation of FC-2 frames, the exchange of security parameters to be used with the ESP encapsulation protocol, and the capability to update keys used by the two entity without any disruption to the underlying traffic flow.

ESP is used as a generic security protocol. Independently from the upper layers, ESP can provide the following:

- Per message integrity and authentication.
  When used with a NULL encryption algorithm and an HMAC as authentication algorithm it guarantees that the frames have not been altered in transit and authenticates the originating entity (i.e. “message is authentic”).

- Traffic encryption.
  When used with a non-NULL encryption algorithm such as AES, triple DES, or RC5, it allows the encryption of the frame content.

The format of a protected FC frame including ESP Header and ESP Trailer is shown in Figure 11 below. The specific areas covers by Authentication (non-repudiation) as well as areas that can optionally be encrypted (confidentiality) are shown:



Bits	31	.. 16	15..08	07..00
0	R_CTL	D_ID		
1	CS_CTL/Prio	S_ID		
2	TYPE	F_CTL		
3	SEQ_ID	DF_CTL	SEQ_CNT	
4	OX_ID		RX_ID	
5	Parameter			
6	Security Parameter Index (SPI)			
7	ESP Sequence Number			
8..N	Payload Data (variable length)			
N+1..P	Padding (0-255 Bytes)		Pad Length	Not used
P+1..Q	Authentication Data			

Figure 11 FC Frame format including ESP Header and Footers

The FCsec security architecture extension to FC has already been included in the FC-SP standard, and completes the panorama of security mechanisms available to a storage area network, offering a consistent level of security whether the underlying network is FC or IP.

Future versions of Cisco SAN-OS will support FCsec per-frame encryption for both control-plane (Class F) and data-plane (Class 2 & 3) frames, providing a complete set of security solutions that will cover the needs of the customer with the tightest security requirements.

(Note: FCsec per-frame encryption is a feature that will be enabled by the Enterprise License Package)

3.5 FC Addressing Security

Many of the potential threats to the stability and security of a FC fabric are through either inadvertent or deliberate mis-configuration of a FC switch within a FC fabric.

The Fabric Services Domain Controller within the Cisco MDS 9000 Family can be configured to mitigate the impact of a mis-configured switch on a fabric and help maintain stability within a fabric.

3.5.1 Principal switch selection

In a FC fabric, the allocation of Domain_IDs is performed by a 'master' switch. This is called the principal switch. If any FC switch within a FC fabric loses connectivity with the principal switch (typically failure of any 'principal ISLs'), a new principal switch needs to be elected. While a new principal switch can be re-elected in a way that is non-disruptive to traffic within the FC fabric, it is still recommended for stability reasons to set a preference for a *core* switch to become the principal switch. This is accomplished through the use of the FCdomain 'switch priority'.

The FCDomain 'switch priority' is a range between 1 and 254, with lower numbers having higher priority. The Cisco MDS 9000 family of multilayer switches use a default priority of 128. During the principal switch selection phase, the switch with the highest priority (lowest number) becomes the principal switch. If two switches have the same configured priority, the switch with the lower WWN becomes the principal switch.

Figure 12 shows a core/edge topology switch design making use of multiple VSANs to provide redundant fabric services. Note that the principal switch in each VSAN is explicitly assigned through the use of 'switch priority' and the each VSAN has a separate core switch as its principal switch. This ensures that only the core switches become the principal switches in each FC fabric respectively. From a security and stability standpoint, any inadvertent or deliberate mis-configuration on one VSAN / switch won't impact other VSANs / switches.

Note that any new switch joining an existing stable fabric cannot become the principal switch but will always be a subordinate switch.

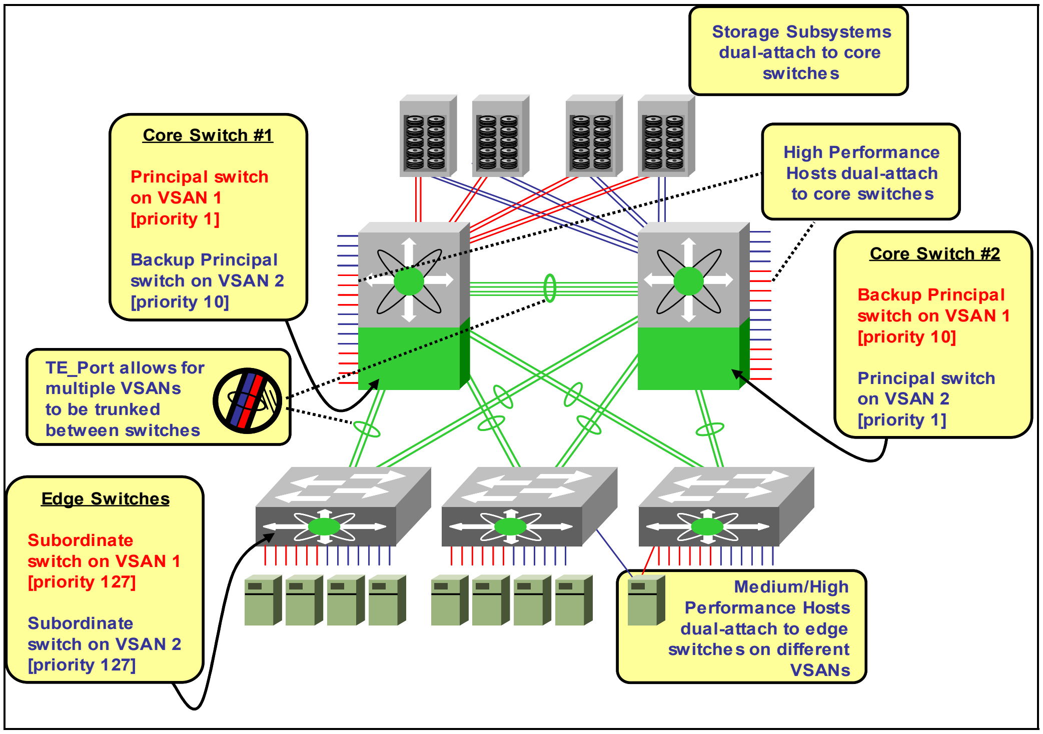

Storage Subsystems dual-attach to core switches
Core Switch #1
Principal switch on VSAN 1 [priority 1]
Backup Principal switch on VSAN 2 [priority 10]
High Performance Hosts dual-attach to core switches
Core Switch #2
Backup Principal switch on VSAN 1 [priority 10]
Principal switch on VSAN 2 [priority 1]
TE_Port allows for multiple VSANs to be trunked between switches
Edge Switches
Subordinate switch on VSAN 1 [priority 127]
Subordinate switch on VSAN 2 [priority 127]
Medium/High Performance Hosts dual-attach to edge switches on different VSANs

Figure 12 Core/Edge SAN design making use of multiple VSANs for resiliency with different core switches as principal switches

☞ The FCdomain 'switch priority' can be changed at any time on an active fabric, but won't actually take effect until such time as a Disruptive Restart command is issued ('*fcdomain restart disruptive vsan <n>*'), which will cause *Reconfigure Fabric* (RCF) frames to be sent to all switches. This will cause an outage to all traffic.

In order to prevent a rogue switch or compromised switch from causing FC fabric outages, the Cisco MDS 9000 Family contains "Disruptive Reconfiguration (RCF) Rejection" functionality. See section 3.5.2 for details on how to enable RCF-Rejection.

In a small SAN deployment (e.g. a pair of fabric switches), it isn't all that important to explicitly define which physical switches will be the principal switches.

In a larger SAN deployment (a 2-tier core/edge topology or 3-tier edge/core/edge toplogy), it should be considered compulsory to explicitly define the principal switch(es) to be the switches that both:

(a) have the most connectivity to other switches and core switch(es), and

(b) have the highest relative control-plane CPU capabilities (i.e. fastest processor, most RAM).

Typically this means that the principal switch in a large SAN fabric is almost always a 'core' switch.

In deploying a Cisco MDS-based FC SAN, it is recommended that VSANs be used to create multiple logical FC fabrics, with different FC fabrics using different physical chassis as the principal switch – as illustrated in Figure 12.
This kind of topology not only provides the highest possible fabric-service availability, thereby helping maintain a stable

fabric even in the event of switch or multiple-switch failure, but also allows for optimal future FC fabric growth, minimizing any future requirement for re-architecture of the SAN as the number of devices attached to the SAN grows.

3.5.2 Disruptive Reconfiguration (RCF) Rejection

In the FC standard for switch-to-switch connectivity (FC-SW-2, FC-SW-3 draft), there are many events that can cause an instability in the FC fabric services which requires a 'disruptive' reconfiguration to restore services.

One such event that may cause this is connecting a new switch into a SAN. If the new switch has an overlapping Domain ID with an existing switch in the fabric, net result may be a disruptive reconfiguration of the fabric – with the net-effect that all traffic will be suspended while a new Principal switch is elected (or the existing Principal switch is re-elected), Domain_IDs are re-allocated and Zone Merging is instigated. The overlapping Domain_ID would be assigned to which ever switch requests it first from the newly selected principal switch therefore the overlap would be resolved.

In order to prevent a rogue or compromised switch from causing FC fabric outages, the Cisco MDS 9000 Family contains "Disruptive Reconfiguration (RCF) Rejection" functionality whereby the switch may be configured to reject incoming RCFs on a per-VSAN and port level basis (enabling the rcf-reject option) and simply isolate ports or switches that request a fabric reconfiguration. This is enabled through the per-port command '*fcdomain rcf-reject vsan <vsan-range>*'. If the security of other switches in a fabric cannot be guaranteed, then it is recommended that this be enabled.

As a minimum, it is recommended that RCF-Rejection be enabled on FC ports prior to connecting a new FC switch into an existing fabric.

When High Availability is considered mandatory, it is recommended that RCF-Rejection be enabled on every port that can possibly be a switch-to-switch port (E_Port, TE_Port).

👍 Changes within the (currently in draft) FC-SW-3 standard help ensure that RCF Frames are only generated through administrative actions and are never generated automatically. However, since FC-SW-2 is the current standard and doesn't have many of the newer protection mechanisms that FC-SW-3 has, it is still recommended to make use of RCF-Rejection to help minimize potential disruptions.

3.5.3 FICON Fabric Binding

Fabric Binding is a feature introduced in SAN-OS 1.3 with FICON that is used to ensure that FICON ISLs between switches are only enabled for switches which are configured within the fabric binding database.

The fabric binding database contains a set of switch WWNs (sWWNs) along with an optional persistent Domain_ID. This list of sWWNs represents the set of all switches that may be part of a fabric; all other switches will be denied.

The Fabric Binding database is similar to Port Security database and FC Zoning database in that there are both 'Active' and 'Configured' databases; changes are only ever applied to the 'Configured' database and when activated, it is atomically applied to become the 'Active' database.

To configure/[clear] the Fabric Binding database, the ‘*[no] fabric-binding database vsan <n>*’ CLI command is used. Once in the fabric-binding configuration, switches can be added/[removed] with the command ‘*[no] swwn <n> [domain-id <n>]*’. To activate/[deactivate] Fabric Binding, the ‘*[no] fabric-binding activate vsan <n>*’ CLI command is used.

Other commands available include:

switch# **show fabric-binding [active [vsan <vsan-id>]] [vsan <vsan-id>]**	Display the fabric binding database
switch# **show fabric-binding statistics [vsan <vsan-id>]**	Show the fabric binding statistics
switch# **show fabric-binding violations [1-100]**	Show the last 1-100 violations

(Note: FICON Fabric Binding is a feature that will be enabled by the FICON/Mainframe License Package)

3.5.4 FC_ID Allocation

Fibre Channel FC_IDs are normally assigned dynamically by a Fibre Channel switch when devices including hosts, disks, and tape arrays log into the fabric. FC_ID assignments can therefore change and be re-assigned as devices are removed and added to the fabric, or switches are reloaded. Because of this, security-enforcement in FC (FC Zoning within the fabric and LUN Security on devices) usually uses the globally unique and persistent WWN in identifying devices. Devices on a fabric can discover the WWN of devices by making use of the FC Name Server to discover the mapping of FC_ID to WWN.

Some operating systems and older FC devices only utilize the FC_ID for mapping block devices to the SAN storage targets. This method of binding between OS-based block devices and Fibre Channel targets raises some concern for the integrity of the binding and the recovery from failure events in the fabric. In addition, the task of migrating servers and storage to different SAN switches becomes complicated.

The problem with FC_ID-based target-binding method is that it binds to dynamically assigned and non-persistent addresses, when there are several possible cases where a new FC_ID may be assigned to a storage device thereby invalidating the binding held by a given server. These cases may involve a simple move of a storage device, or perhaps a port failure requiring the storage device to be moved to a working switch port. It could even be something as simple as a SAN switch being rebooted.

All of these conditions may cause new FC_IDs to be assigned to existing storage devices – or worse, may result in a new device (e.g. a compromised host) inheriting the FC_ID of an existing device, allowing for security to be compromised through the use of man-in-the-middle (MITM) attacks. A SAN designer must pay very close attention to the details of FC_ID allocation when deploying operating systems in a SAN when there is FC_ID-based binding, as this can represent a significant high availability and security risk.

The Cisco MDS 9000 has a number of features to help mitigate the HA and security risks associated with FC_ID-binding, namely FC_ID Address Caching, Persistent FC_ID Allocation and Static FC_ID Allocation. Each of these is covered below:

3.5.4.1 FC_ID Address Caching

FC_ID Address Caching is the default addressing mechanism used in the Cisco MDS 9000. In this mode, FC_IDs are assigned in a sequential order as devices issue Fabric Logins (FLOGI) into the switch. The switch maintains an active cache of assignments based on the WWN, so devices that go offline or are moved from one port in the switch to another port keep the same FC_ID.

If a switch port or SFP failure were to occur, FC_ID Address Caching means that no pre-configuration is required; the device connected to the failed port could simply be moved to another port and would assume the same FC_ID.

3.5.4.2 Persistent FC_ID Allocation

By default, FC_ID Address Caching is maintained in switch dynamic memory. If a switch were to be powered off, the cache assignments are not maintained when the switch is powered up again.

The Persistent FC_ID Allocation feature builds upon the FC_ID Address Caching mechanism by recording the binding in non-volatile memory within the switch. This binding remains intact until it is explicitly purged by the switch administrator. Therefore, as new devices are attached to the MDS 9000 switch, they are dynamically assigned FC_IDs and this relationship of WWN->FC_IDs is recorded on persistent non-volatile memory.

This configuration can be extracted as part of the running configuration in the switch and easily copied to a new switch, even if the new switch does not have the same port configuration as the copied switch. This sort of flexibility in the Cisco MDS 9000 Family significantly reduces the management complexity and availability risks associated with deploying servers into the SAN that depend on FC_ID binding.

Persistent FC_ID Allocation is enabled on a per-VSAN basis allowing different VSANs to have different addressing policies or practices. Persistent FC_ID Allocation requires that the switch Domain_ID be *statically* configured.

The following configuration shows enabling Persistent FC_ID feature with 3 devices in the fabric being assigned persistent FC_IDs:

switch(config)# **fcdomain domain 2 static vsan 1**	Enable a static Domain_ID of 2 in VSAN 1
switch(config)# **fcdomain fcid persistent vsan 1**	Enable Persistent FC_IDs within VSAN 1
switch(config)# **fcdomain restart vsan 1**	Force a non-disruptive restart of fabric services in VSAN 1 to start populating the FC_ID Database.
switch(config)# **exit**	Exit configuration mode
switch# **show running-configuration** **..** fcdomain fcid database vsan 1 wwn 21:00:00:e0:8b:08:14:5a fcid 0x7b0200 area dynamic vsan 1 wwn 21:01:00:e0:8b:29:81:46 fcid 0x7b0300 area dynamic vsan 1 wwn 21:01:00:e0:8b:29:7b:46 fcid 0x7b0400 area dynamic	The FC_ID Database will be populated automatically as new devices are attached to the SAN and are assigned a FC_ID

Note that the '*running-configuration*' will grow as the number of devices connected to the fabric increases.

> 👍 Since the first octet in the FC_ID is the Domain_ID of the switch, the Persistent FC_ID feature is dependent upon the switch having a static or insistent Domain_ID. A static or insistent Domain_ID may be assigned on a per-VSAN basis.

3.5.4.3 Static FC_ID Assignment

For those administrators who keep tight records and want more control over the actual allocation of the FC_ID addresses, the Cisco MDS 9000 Family supports static FC_ID assignments. Using static FC_ID assignments, the addresses are not only persistent and stored in non-volatile memory, but the actual area and port octet in the FC_ID can be assigned by the administrator as well. Using this feature, SAN administrators can use custom numbering or addressing schemes to divide the FC_ID domain address space amongst available SAN devices.

Static FC_ID Assignment is similar to Persistent FC_ID Allocation. It is enabled on a per-VSAN basis allowing different VSANs to have different addressing policies or practices. Also like Persistent FC_ID Allocation, Static FC_ID Assignment requires that the switch Domain_ID be *statically* configured.

The following configuration shows enabling Static FC_ID Assignment for two devices in the fabric:

switch(config)# **fcdomain domain 2 static vsan 1**	Enable a static Domain_ID of 2 in VSAN 1
switch(config)# **fcdomain restart vsan 1**	Force a non-disruptive restart of fabric services in VSAN 1
switch(config)# **fcdomain fcid database**	Activate FC_ID Persistency in VSAN 1
switch(config-fcid-db)# **vsan 1 wwn 33:e8:00:05:30:00:16:df fcid 0x070128**	Configures the device in VSAN 1 with WWN 33:e8:00:05:30:00:16:df with the FC_ID 0x070128
switch(config-fcid-db)# **vsan 1 wwn 11:22:11:22:33:44:33:44 fcid 0x070123 dynamic**	Configures the device in VSAN 1 with WWN 11:22:11:22:33:44:33:44 with the FC_ID 0x070123 in Dynamic Mode.
switch(config-fcid-db)# **vsan 1 wwn 11:22:11:22:33:44:33:44 fcid 0x070100 area**	Configures the device in VSAN 1 with WWN 33:e8:00:05:30:00:16:df with the FC_IDs 0x070100 through 0x701FF

> 👍 Since the first octet in the FC_ID is the Domain_ID of the switch, the Persistent FC_ID feature is dependent upon the switch having a static Domain_ID. A static Domain_ID may be assigned on a per-VSAN basis.

From a security standpoint, it doesn’t matter whether FC_ID Caching, Persistent FC_ID Allocation or Static FC_ID Assignment is used.

The choice on which is most suitable is dependent on the devices in the fabric, and whether it is beneficial to use a custom numbering scheme or persistent scheme that aids configuration and storage availability with the Host OSes in use on the fabric and the storage devices connected to the fabric.

4 Storage over IP Security

In the past, security as it pertains to storage devices and storage networks, has not been a major consideration. Either storage devices were directly attached to hosts, or connected via a separate SAN, independent of user accessible networks.

With the advent of Storage over IP transport technologies such as Fibre Channel over IP (FCIP) and iSCSI, a lot more focus has been put on security – particularly security as it applies to storage over IP networks. The main rationale behind this is that IP networks are far more ubiquitous than SANs have traditionally been – and thus far less specialized equipment is required to snoop traffic on an IP network. Knowing this, the IP Storage working group within the IETF has also developed a framework for securing IP based storage communications. This work is contained in the 'Securing Block Storage Protocols over IP' (draft-ietf-ips-security-19).

The Cisco MDS 9000 family of multilayer intelligent switches not only provides all the security functionality recommended by the IP Storage working group, but also has additional security features/functionality as it pertains to Storage over IP. These features are covered in this section below.

4.1 SCSI over IP (iSCSI) Security

4.1.1 Presenting Fibre Channel Targets as iSCSI Targets

IP Services modules within the Cisco MDS 9000 family of multilayer switches can present physical Fibre Channel targets as iSCSI targets allowing them to be accessed by iSCSI initiators (hosts). It can present the FC targets in one of two ways:

1. **Dynamic Importing**
   Used if all logical units (LUs) in all Fibre Channel storage targets are made available to iSCSI hosts (subject to VSAN and zoning).

   To enable dynamic importing of Fibre Channel targets into iSCSI, the '*iscsi import target fc*' command is used. Each IPS module maps each physical Fibre Channel target port as one iSCSI target. That is, all LU accessible via the physical storage target port are available as iSCSI LUs with the same LU number (LUN) as in the storage target.

   For example, if an iSCSI target was created for Fibre Channel target port with pWWN 31:00:11:22:33:44:55:66 and that pWWN contains LUN 0 through 2, those LUNs would become available to an IP host as LUNs 0 through 2 as well.

   Dynamically Imported FC targets are imported to become virtual iSCSI targets with the following naming conventions:

   IPS ports that are NOT part of a VRRP group use this format:
   iqn.1987-05.com.cisco:05.<mgmt-ip-address>.<slot#>-<port#>-<sub-intf#>.<Target-pWWN>

   IPS ports that are part of a VRRP group use this format:
   iqn.1987-05.com.cisco:05.vrrp-<vrrp-ID#>-<vrrp-IP-addr>.<Target-pWWN>

Ports that are part of a PortChannel use this format:
iqn.1987-05.com.cisco:05.PC-<port-ch-intf#>-<port-ch-sub-intf#>.<Target-pWWN>

Note that with Dynamic Importing, each FC target may appear as multiple iSCSI virtual targets, since importing is performed on a per-interface basis on each IPS module.

2. **Static Mapping**

Static Mapping should be used if iSCSI hosts are to be restricted to subsets of LUs in the Fibre Channel targets and additional iSCSI access control is needed. Also, static import allows automatic failover if the Fibre Channel targets' LU is reached by redundant Fibre Channel ports.

With static importing, iSCSI targets are created and named manually. A statically-mapped iSCSI target can either contain the whole FC target port, or it can contain one or more LUs from a Fibre Channel target port.

The following excerpt shows how to configure iSCSI for static importing:

switch(config)# **iscsi virtual-target name iqn.abc**	Creates the iSCSI target name called 'iqn.abc'
switch(config-iscsi-tgt)# **pWWN 26:00:01:02:03:04:05:06**	Maps iSCSI virtual target 'iqn.abc' to map to FC target pWWN 26:00:01:02:03:04:05:06. All LUNs on this FC target will be available in this iSCSI virtual target. Note: an iSCSI target may only map to a single FC target.

If no specific LUN is specified in the mapping, all FC LUNs from the FC target are exposed as iSCSI targets. The following shows the LUN option being used to map specific FC LUNs to specific iSCSI virtual targets. This is sometimes referred to as "LUN Mapping":

switch(config)# **iscsi virtual-target name iqn.abc**	Creates the iSCSI target name called 'iqn.abc'
switch(config-iscsi-tgt)# **pWWN 26:00:00:00:00:11:00:11 fc-lun 5 iscsi-lun 0**	Maps the iSCSI virtual target 'iqn.abc' LUN 0 to FC target pWWN 26:00:00:00:00:11:00:11 LUN 5.

With static importing, iSCSI targets can be advertised out specific Gigabit Ethernet ports on IP Service modules or made available to all Gigabit Ethernet ports.

By default, iSCSI targets are advertised on all Gigabit Ethernet interfaces, subinterfaces, PortChannel interfaces, and PortChannel subinterfaces. To limit the scope of iSCSI target advertisements, the following configuration is used:

switch(config)# **iscsi virtual-target name iqn.abc**	Creates the iSCSI target name called 'iqn.abc'
switch(config-iscsi-tgt)# **advertise interface**	Limit iSCSI target 'iqn.abc' to be advertised on

GigabitEthernet 2/5	interface GigabitEthernet 2/5 only.
switch(config-iscsi-tgt)# **advertise interface GigabitEthernet 7/1-4**	Advertise iSCSI target ‘iqn.abc’ on interfaces GigabitEthernet 7/1 through 7/4.
switch(config-iscsi-tgt)# **advertise interface GigabitEthernet 2/5, GigabitEthernet7/1-2**	Advertise iSCSI target ‘iqn.abc’ on interfaces GigabitEthernet 2/5 and GigabitEthernet 7/1 and 7/2.
switch(config-iscsi-tgt)# **no advertise interface GigabitEthernet 2/5**	Remove interface-specific advertisement for iSCSI target ‘iqn.abc’. If no other advertised interfaces are specific, iSCSI target ‘iqn.abc’ will now be advertised out all Gigabit Ethernet interfaces.

By default, the IP Services modules do not import any FC targets to be presented as iSCSI targets. Either dynamic or static mapping must be configured before the FC targets are available to iSCSI initiators.

Both static importing and dynamic importing can be used at the same time.

From a security perspective, either Dynamic Importing or Static Mapping may be used. Static Mapping can be considered more secure as it allow for more fine-grained security policies to be put into place, at the expense of additional configuration management.

With Dynamic Importing, care needs to be taken to ensuring that security is enforced via VSANs and FC Zoning in the FC fabric. If further restrictions to individual LUNs are required, then either LUN Zoning in the fabric and/or LUN Security on intelligent disk arrays need to be used.

With Static Mapping, there is a consistent iSCSI virtual target name, regardless of what interface(s) an iSCSI virtual target is advertised out. Static Mapping doesn’t preclude the use of VSANs, FC Zoning, LUN Zoning or LUN Security, but can be used as another layer of security to prevent visibility of storage beyond where it should be.

Static Mapping can have explicit security measures put in place to permit/deny initiators on a per-IQN, per-IP-address or per-subnet range also. This is covered in section 4.1.3.

4.1.2 Presenting iSCSI Hosts (initiators) as virtual FC Hosts (initiators)

As well as mapping FC Targets to appear as iSCSI Targets, the same needs to be configured in reverse – that is, the mapping of iSCSI initiators to appear as virtual FC initiators. In order for an iSCSI initiator to appear as a virtual FC initiator, it needs to behave just like a FC initiator would – namely, to appear as a N_Port device and have its own unique identity in a FC fabric, a nWWN and pWWN which are the FC identity of the iSCSI device.

This mapping of iSCSI initiator to FC initiator occurs within the IP Services module(s) installed within switches. This mapping involves the creation of a ‘virtual N_Port’, with its own nWWN and pWWN. A virtual N_Port is created when an iSCSI initiator successfully completes an iSCSI Login into the switch for the first time. The virtual N_Port behaves just like a FC HBA would, registering in the nameserver, PLOGIing and PRLIing into devices.

This virtual N_Port will stay logged into the fabric until such time as the iSCSI host is no longer attached to any iSCSI targets, at which point the N_Port will logout from any remaining fabric services.

> 👍 If an iSCSI initiator is connecting into multiple Gigabit Ethernet ports, each GE port will independently require a virtual N_Port. This is taken care of automatically with Dynamic Mapping. With System-Assigned Static Mapping, care should be taken to allocate a sufficient number of pWWNs to cover the total number of GE interfaces an iSCSI initiator may connect in to. For Manually-Assigned Static Mapping, multiple pWWNs should be specified for as many GE ports the iSCSI initiator will be connecting in to.

The nWWN and pWWN associated with this virtual N_Port can be obtained in a number of different ways. These are as follows:

1. **Dynamic Mapping**

   Dynamic mapping may be used if there is no access control on the FC target (e.g. LUN Security isn't used on the storage array, or it's a non-intelligent storage array such as a JBOD).

   With dynamic mapping, the virtual N_Port will be assigned a pWWN and nWWN derived from a pool of WWNs stored in an EEPROM within the MDS chassis. The WWNs assigned to it may be different each time it connects to a FC target.

   The WWNs will only remain consistent while there is an active iSCSI session. When there is no longer an active iSCSI session, the WWNs will be released back into the pool and made available for assignment to other iSCSI hosts requiring access to FC fabric(s).

   In order to map an iSCSI initiator to a virtual N_Port, each iSCSI initiator needs to be uniquely identified in some way. The Cisco MDS 9000 family of multilayer switches supports two ways of identifying iSCSI initiators:

   A. Map by Initiator Name

   In this mode, the iSCSI initiator node name (iSCSI Qualified Name) is used to identify the iSCSI initiator. Regardless of how many network interface cards (NICs) the initiator have, it will only have a single virtual N_Port assigned to it. This is the default.

   The following configuration excerpt shows configuring an iSCSI interface to map iSCSI initiators by name:

switch(config)# **interface iscsi7/1**	enter configuration for interface iscsi7/1
switch(config-if)# **switchport initiator id name**	match iSCSI initiator by iSCSI Qualified Name (IQN) (default)

   B. Map by Initiator IP Address

   In this mode, iSCSI initiators are identified by the IP Address of the incoming iSCSI session. This is useful to map different network interfaces (NICs) (with different IP addresses) within a single host to different nWWN/pWWNs.

The following configuration excerpt shows configuring an iSCSI interface to map iSCSI initiators by IP Address:

switch(config)# **interface iscsi7/1**	enter configuration for interface iscsi7/1
switch(config-if)# **switchport initiator id ip-address**	match iSCSI initiator by iSCSI Qualified Name (IQN) (default)

Note that all dynamic iSCSI initiators are only ever members of the default VSAN (VSAN 1). If this doesn't meet the configuration requirements, then Static Mapping is necessary.

2. Static Mapping

Static Mapping should be used if an iSCSI initiator should always have the same pWWN and/or nWWN each time it connects to a Fibre Channel target.

With static mapping, the nWWN/pWWN(s) will be consistent every time an iSCSI initiator connects to a FC target, thus allowing for nWWN/pWWN-based FC Zoning to be used in the fabric and LUN Security on a storage array to identify the iSCSI host/initiator.

Static Mapping also allows an iSCSI initiator to reside in a VSAN other than the default VSAN. Static Mapping also allows an iSCSI initiator to reside in multiple VSANs based on the configuration. By default, a host is only in VSAN 1 (default VSAN). A virtual N_Port will created for each VSAN to which an iSCSI initiator belongs.

Static mapping can be implemented in one of two ways: system assignment or manual assignment:

A. System Assignment

System assignment is similar to dynamic mapping in that the WWNs are assigned from a pool of WWNs stored in an EEPROM within the MDS chassis. Unlike dynamic mapping, however, the WWN used is persistent and will always be the same.

A system-assigned nWWN will always be persistent for a given VSAN.
A system-assigned pWWN is assigned statically from the pool of addresses available within the EEPROM. If an iSCSI host is connecting into multiple Gigabit Ethernet ports, then multiple virtual N_Ports are required, and therefore multiple pWWNs are required. Because of this, anywhere between 1 to 64 pWWNs may be allocated to an iSCSI initiator.

The following configuration excerpt shows a Static Mapping using System Assignment:

switch(config)# **iscsi initiator name iqn.1987-05.com.cisco.01.mel-stglab-host12**	Creates a mapping for iSCSI initiator 'iqn.1987-05.com.cisco.01.mel-stglab-host12'
switch(config-(iscsi-init))# **static nWWN system-assign**	Uses the switch's WWN pool to allocate the nWWN for this iSCSI initiator and keeps it persistent.
switch(config-(iscsi-init))# **static pWWN**	Uses the switch's WWN pool to allocate 2

system-assign 2	pWWN for this iSCSI initiator and keeps it persistent.
switch(config-(iscsi-init))# **vsan 3**	Assigns the iSCSI initiator to VSAN 3.
switch(config-(iscsi-init))# **exit**	Exit configuration for this iSCSI mapping
switch(config)# **iscsi initiator ip-address 10.67.16.5**	Creates a mapping for an iSCSI initiator at the ip-address 10.67.16.5.
switch(config-iscsi-init)# **static nWWN system-assign**	Uses the switch's WWN pool to allocate the nWWN for this iSCSI initiator and keeps it persistent.
switch(config-(iscsi-init))# **static pWWN system-assign 8**	Uses the switch's WWN pool to allocate 8 pWWN for this iSCSI initiator and keeps it persistent. 8 pWWNs would allow this iSCSI initiator to connect to up to 8 Gigabit Ethernet ports.
switch(config-(iscsi-init))# **vsan 3** switch(config-(iscsi-init))# **vsan 4**	Assigns the iSCSI initiator to both VSAN 3 and VSAN 4.

B. Manual Assignment

With manual assignment, the nWWN and pWWN(s) for a given iSCSI initiator may be manually assigned. The Cisco MDS 9000 family allows any valid WWN to be manually configured.

The following configuration excerpt shows a Static Mapping using Manual Assignment:

switch(config)# **iscsi initiator name iqn.1987-05.com.cisco.01.mel-stglab-host12**	Creates a mapping for iSCSI initiator ‘iqn.1987-05.com.cisco.01.mel-stglab-host12’
switch(config-(iscsi-init))# **static nWWN 01:02:03:04:05:06:02:01**	Assign the static nWWN 01:02:03:04:05:06:02:1 for this iSCSI initiator.
switch(config-(iscsi-init))# **static pWWN 11:22:33:44:55:66:77:88** switch(config-(iscsi-init))# **static pWWN 11:22:33:44:55:66:77:89** switch(config-(iscsi-init))# **static pWWN 11:22:33:44:55:66:77:90**	Assign three static pWWNs for this iSCSI initiator.
switch(config-(iscsi-init))# **vsan 3** switch(config-(iscsi-init))# **vsan 4**	Assigns the iSCSI initiator to VSAN 3 and VSAN 4.
switch(config-(iscsi-init))# **exit**	Exit configuration for this iSCSI mapping
switch(config)# **iscsi initiator ip-address**	Creates a mapping for an iSCSI initiator at the

10.67.16.5	ip-address 10.67.16.5.
switch(config-iscsi-init)# **static nWWN system-assign**	Uses the switch's WWN pool to allocate the nWWN for this iSCSI initiator and keeps it persistent.
switch(config-(iscsi-init))# **static pWWN system-assign 8**	Uses the switch's WWN pool to allocate 8 pWWN for this iSCSI initiator and keeps it persistent. 8 pWWNs would allow this iSCSI initiator to connect to up to 8 Gigabit Ethernet ports.
switch(config-(iscsi-init))# **vsan 3** switch(config-(iscsi-init))# **vsan 4** switch(config-(iscsi-init))# **vsan 5**	Assigns the iSCSI initiator to VSAN 3, VSAN 4 and VSAN 5.

Note that both System-Assigned and Manual-Assigned Static Mappings may be used at the same time for the same initiator. e.g. use a system-assigned nWWN and a manually-assigned set of pWWNs.

All of Dynamic Mappings and System-Assigned/Manually-Assigned Static Mappings may be used for different iSCSI initiators, optionally across VSANs at the same time; the configuration is on a per-initiator basis.

From a security perspective, either Dynamic or Static mapping may be used. The use of one or the other is determined by whether one is more suitable for the architecture being deployed and whether it is considered important to have a security policy configured within the storage arrays.

Dynamic Mapping provides the most flexibility in terms of minimizing the number of configuration tasks necessary to connect an iSCSI initiator to a FC target, but it does have a number of preconditions on its use:

(a) Dynamic mapping can be used if there no access control is on the FC target (i.e. LUN Security isn't used on the storage array, or it's a non-intelligent storage array such as a JBOD), and

(b) Device isolation with Dynamic mapping is made available through the use of FC Fabric Zoning using Symbolic-Node-Name-based zoning (the IQN for the initator is included in the zoning). Symbolic-Node-Name zoning is currently only available on Cisco MDS switches and cannot be used in a fabric with switches requiring *interop* mode to be enabled.

(c) Dynamic mapping only applies to devices in the default VSAN (VSAN 1).

If all of these conditions aren't met, then Static mapping is necessary.

With static mapping, the choice is between using system-assigned WWNs (from a pool of WWNs programmed into an EEPROM on the chassis backplane), or manually-assigned WWNs.

System-assigned static WWNs minimize the amount of administrator configuration required, but don't provide portability in terms of being able to take a working configuration from one switch and apply it verbatim to another switch (since the chassis will have different pools of WWNs).

Thus, if explicit configuration of WWNs deemed necessary, then manually-assigned WWNs should be used. If this is not a consideration, then system-assigned static WWNs should be used.

With Static mapping, care should always be taken to ensure that iSCSI initiators are only ever configured into the VSANs that they're entitled to, and should always have FC Zoning applied in the FC fabric. If further restrictions to individual LUNs are required, then either LUN Zoning in the fabric and/or LUN Security on intelligent disk arrays need to be used.

4.1.3 iSCSI Access Control

Additional access controls can be applied to Statically Mapped iSCSI virtual targets to further limit what iSCSI initiators may use a given iSCSI virtual target. An iSCSI Virtual Target may be restricted to a set of iSCSI initiators, based on any/all of:

- iSCSI node name (IQN)
- IP address(es)
- IP subnet(s)

By default, static virtual iSCSI targets are not accessible to any iSCSI host and must be explicitly configured to allow access by any hosts. Any number of restrictions may be placed on a given virtual target.

The following configuration excerpt shows an iSCSI virtual target with various restrictions put in place:

switch(config)# **iscsi virtual-target name iqn.abc**	Creates the iSCSI target name 'iqn.abc'.
switch(config-(iscsi-tgt))# **pWWN 26:00:01:02:03:04:05:06**	Maps the iSCSI Virtual target 'iqn.abc' to FC target at pWWN 26:00:01:02:03:04:05:06
switch(config-(iscsi-tgt))# **initiator iqn.1987-02.com.cisco.initiator1 permit**	Allows iSCSI initiator 'iqn.1987-02.com.cisco.initiator1' to access the virtual target 'iqn.abc'. This command may be issued multiple times to allow multiple initiators.
switch(config-(iscsi-tgt))# **no initiator iqn.1987-02.com.cisco.initiator1 permit**	Prevents iSCSI initiator 'iqn.1987-02.com.cisco.initiator1' from accessing the virtual target 'iqn.abc'.
switch(config-(iscsi-tgt))# **initiator ip address 10.50.1.1 permit**	Allows the host at IP-address 10.50.1.1 access to virtual target 'iqn.abc'. This command may be issued multiple times to allow multiple initiators.
switch(config-(iscsi-tgt))# **no initiator ip address 10.50.1.1 permit**	Prevents the host at IP-address 10.50.1.1 from accessing virtual targets 'iqn.abc'.
switch(config-(iscsi-tgt))# **initiator ip address 10.50.1.1 255.255.255.0 permit**	Allows all initiators in this IP subnet 10.50.1.0/24 to access virtual target 'iqn.abc'.
switch(config-(iscsi-tgt))# **no initiator**	Prevents all initiators in IP subnet 10.50.1.0/24 from

ip address 10.50.1.1 255.255.255.0 permit	accessing virtual target ‘iqn.abc’.
switch(config-(iscsi-tgt))# **all-initiator-permit**	Allows any initiators to access this virtual target ‘iqn.abc’
switch(config-(iscsi-tgt))# **no all-initiator-permit**	Prevent any initiator from accessing virtual target ‘iqn.abc’ (default).

Note that with both Statically Mapped and Dynamically Mapped iSCSI Targets, iSCSI Access Control is enforced at both iSCSI Session Creation (iSCSI Login) and iSCSI Session Discovery using both iSCSI access-control (if present) and FC Zoning to enforce access:

- **At iSCSI session creation**
  When an IP host initiates an iSCSI session, the IPS module verifies if the specified iSCSI target (in the session login request) is a static mapped target, and if true, verifies if the IP host's iSCSI node name is allowed to access the target. If the IP host does not have access, its login is rejected.
- **Within iSCSI discovery**
  When an iSCSI host creates an iSCSI discovery session and queries for all iSCSI targets, the IPS module returns only the list of iSCSI targets this iSCSI host is allowed to access based on any access control policies in place.

In both iSCSI session creation and iSCSI discovery, the IPS module uses the FC virtual N_Port for this IP host and performs a lookup in the FC name server to query for the FC_ID of the Fibre Channel target pWWN that is being accessed by the IP host. It uses the IP host virtual N port's pWWN as the requester of the name server query, thereby allowing the name server to perform a zone-enforced query (soft-zoning) for the pWWN and responds to the query. If the FC_ID is returned by the name server, then the iSCSI session is accepted. Otherwise, the login request is rejected.

If there was already an active iSCSI session between an IP host and a FC target and a new zoneset was activated which denied access between the two devices, Hard Zoning would enforce access restrictions between the IP Host’s virtual N_Port and the FC target.

4.1.4 Enforcing security on iSCSI Initiators using VSANs & FC Zoning

On the Cisco MDS 9000 family of multilayer switches, both VSANs and FC Zoning can be used to enforce security for iSCSI initiators, thereby providing for both uniform and flexible access control mechanism across an entire SAN.

Dynamically Mapped iSCSI Initiators
Dynamically Mapped iSCSI Initiators are only ever in VSAN 1. Within VSAN 1, they In order to use FC Zoning on a dynamically mapped iSCSI initiator (which will have a dynamic fWWN/pWWN), the ‘Symbolic Node Name’ of the iSCSI initiator should be used to include the device in a FC Zone. In the case of IP-Address based mapping, the ‘Symbolic Node Name’ should consist of the IP-Address (e.g. ‘member symbolic-nodename 10.50.1.1’), otherwise it should consist of the iSCSI node name (IQN) of the initiator (e.g. ‘member symbolic-nodename iqn.1987-02.com.cisco.initiator1’).

Statically Mapped iSCSI Initiators
Statically Mapped iSCSI Initiators are in as many VSANs as they have been configured to exist in. Since Statically

Mapped iSCSI Initiators may have a permanently-mapped/fixed nWWN/pWWN, they may be included in FC Zones using their pWWN/fWWN, interface or symbolic node name.

It is recommended that zoning based on symbolic node-name is used wherever possible, as this provides a consistent security policy, regardless of whatever Gigabit Ethernet port the iSCSI initiator is connecting in on at any given time. Symbolic Node Name-based zoning can also provide a consistent security policy in the event that an iSCSI initiator is connecting in on multiple switches/chassis.

4.1.5 iSCSI Authentication

The Cisco MDS 9000 family supports Challenge Handshake Authentication Protocol (CHAP) authentication of iSCSI hosts. When authentication is enabled, iSCSI hosts must provide a user name and password each time an iSCSI session is established. CHAP allows for two-way authentication; that is, iSCSI hosts may also challenge the Cisco MDS 9000 to provide its authentication credentials.

iSCSI Authentication may either be enabled (‘*iscsi authentication chap*’), or be disabled (‘*iscsi authentication none’*).

If Authentication is enabled, AAA is used to perform the authentication. AAA is Cisco’s architectural framework for configuring a set of three independent security functions (Authentication, Authorization and Accounting) in a consistent, modular manner. AAA services may be used to allow iSCSI hosts to authenticate either against a local authentication database in the switch (local user database), centralized RADIUS server(s) or (in SAN-OS 1.3) centralized TACACS+ server(s). If RADIUS or TACACS+ authentication is enabled and the centralized authentication servers cannot be contacted or are unavailable, the local database will be used.

iSCSI Authentication may be enabled either globally, or on a per-interface basis. Per-interface authentication will override the global setting for that particular interface.

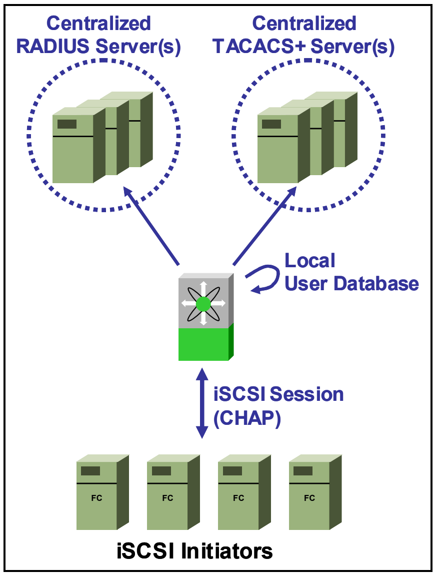


Figure 13 iSCSI Authentication and AAA Methods

The following configuration excerpt shows iSCSI Authentication being enabled and applied to AAA services:

switch(config)# **aaa authentication iscsi radius**	Globally enable RADIUS as the iSCSI authentication method.
switch(config)# **aaa authentication iscsi tacacs**	Globally enable TACACS+ as the iSCSI authentication method.
switch(config)# **aaa authentication iscsi local**	Globally enable the local database as the iSCSI authentication method.
switch(config)# **iscsi authentication none**	Disables iSCSI Authentication globally
switch(config)# **iscsi authentication chap**	Configures CHAP as the default authentication mechanism globally
switch(config)# **interface GigabitEthernet 2/1.100**	Enters the configuration for interface Gigabit Ethernet 2/1.100
switch(config-if)# **iscsi authentication none**	Specifies that no authentication is required for iSCSI sessions connecting to interface GigabitEthernet 2/1.100

Section 5.1 provides details on how to configure the AAA services on the Cisco MDS 9000 product family.

It is recommended that iSCSI Authentication always be used to authenticate iSCSI hosts (initiators). When there are only a handful of iSCSI hosts, the local user database may be used; if there is more than half a dozen hosts, a centralized RADIUS or TACACS+ Server should be utilized.

4.2 Fibre Channel over IP (FCIP) Security

Fibre Channel over IP (FCIP) transports Fibre Channel data across an IP network by tunneling the FC frames across a pair of TCP connections; the raw FC Frames (including the complete FC header) are encapsulated into TCP segments at the sending end of the tunnel and reconstructed into FC frames at the receiving end. There is a single tunnel for all local-to-remote storage traffic with FCIP; the remote (receiving) end is responsible for switching each FC frame to its appropriate Fibre Channel end device.

👍 FC-SP may be applied to FCIP interfaces to authenticate the remote FCIP tunnel endpoint. Section 3.4.1 provides details on how to configure FC-SP based switch-to-switch authentication.

FCIP itself doesn’t provide any security of the data being transported; if a rogue device in the path were able to eavesdrop, they could view all of the storage data being transferred across the link. Knowing this, the IP Storage working group within the IETF has also developed a framework for securing IP based storage communications. This work is contained in the ‘Securing Block Storage Protocols over IP’ document (draft-ietf-ips-security-19.txt). In essence, the IETF mandates the use of IPsec if the data network cannot be trusted. Section 4.3 contains details on IPsec and how to best deploy it.

4.3 IPsec

IP Security (IPsec) is a framework of open standards developed by the Internet Engineering Task Force (IETF) to provide privacy, integrity, and authenticity to information transferred across IP networks. IPsec is applied at the network layer, protecting and authenticating IP packets between participating IPsec devices ("peers"). IPsec provides these security features:

- **Data Confidentiality**
  The IPsec sender can encrypt packets before transmitting them across a network.

- **Data Integrity**
  The IPsec receiver can authenticate packets sent by the IPsec sender to ensure that the data has not been altered during transmission.

- **Data Origin Authentication**
  The IPsec receiver can authenticate the source of the IPsec packets sent. This service is dependent upon the data integrity service.

- **Anti-Replay**
  The IPsec receiver can detect and reject replayed packets.

With IPsec, data can be transmitted across a public network without fear of observation, modification, or spoofing. This enables applications such as Virtual Private Networks (VPNs), including intranets, extranets, remote user access, and remote transport of Storage over IP.

The IETF's Internet Engineering Steering Group (IESG) is mandating IPsec be used with the iSCSI and FCIP to provide secure private exchanges at the IP layer. In order to compliant, an iSCSI network element must implement IPsec tunnel mode with the Encapsulating Security Protocol (ESP). Confidentiality is obtained by encrypting the IPsec tunnel using Triple Digital Encryption Standard (3DES) or Advanced Encryption Standard (AES) in cipher block chaining (CBC) mode. An iSCSI node must support Internet Key Exchange (IKE) to provide authentication, security association negotiation, and key management. A separate IKE Phase 2 security association protects each TCP connection within an iSCSI session

The IP Services module within the Cisco MDS 9000 family doesn’t yet natively support IPsec encryption. If IPsec encryption is desired, then an external IPsec device should be used.

Some of the devices available include:

- Cisco VPN Acceleration Module 2 (VAM2) for the Cisco 7200 Series router provides hardware-based encryption in excess of 260 Mbps and also includes compression
- Cisco PIX 535 Firewall with integrated hardware VPN acceleration can deliver up to 440 Mbps of 3DES or AES IPSec throughput
- IPsec VPN Services Module for the Catalyst 6500 & Cisco 7600 provides high-performance hardware-based encryption at up to 1.9Gbps

The Cisco SAFE Blueprint on VPN security provides information on best-practice for designing and implementing Enterprise IP Security (IPsec) virtual private networks (VPNs). The SAFE VPN blueprint is available at:
<http://www.cisco.com/application/pdf/en/us/guest/netsol/ns128/c654/cdccont_0900aecd800b05ad.pdf>

For a complete list of Cisco's security product offerings, please see http://www.cisco.com/go/security.

A future firmware release for the Cisco MDS 9000 family may enable IPsec encryption for iSCSI and FCIP sessions on the IP Services module. However, it is not expected that the internal implementation of IPsec will be capable of sustaining wire-rate performance on the Gigabit Ethernet ports on the IPS module.

Given the potential throughput bottlenecks that may occur as a result of using IPsec, if IPsec is considered desirable, then a half-way measure may be to first look at creating a separate "storage" IP network that is secured from the rest of the data network. Cisco provides numerous deployment guides in the form of the "Cisco SAFE Blueprint" (http://www.cisco.com/go/safe) that detail how to deploy Ethernet/IP infrastructure in a secure manner – without necessarily requiring end-to-end encryption.

If encryption of IP Storage traffic is deemed mandatory, then an external encryption device such as those listed above is recommended.

4.4 VLAN Trunking

Virtual LANs (VLANs) are a mechanism to allow network administrators to create logical broadcast domains that can span across a single ethernet switch or multiple ethernet switches, regardless of physical proximity. VLANs are useful for reducing the size of broadcast domains, or allowing groups or users to be logically grouped without being physically located in the same place.

Gigabit Ethernet ports on the IP Services module(s) within the Cisco MDS 9000 family of multilayer switches can be configured as a trunking port and uses the IEEE 802.1Q standard for encapsulation of multiple VLANs over a single physical Gigabit Ethernet port.

VLAN Trunking may be useful in scenarios where different end users shouldn't be allowed to communicate with each other (e.g. IP Services supplied within a *Storage Service Provider* scenario) and where Jumbo Frames (Ethernet frames with up to a 9000 byte MTU) are used (it is generally recommended to avoid layer-3 routing hops for jumbo frames).

5 Switch Management Security Features

The Cisco MDS 9000 family of multilayer intelligent switches contains a comprehensive framework of management security features designed to ensure that security cannot be compromised through weaknesses in the switch management interface.

Security features such as centralized RADIUS or TACACS+-based account management through Cisco's AAA (Authentication, Authorization and Accounting) security framework provide for centralized user account management. NTP-based time synchronization ensures that all devices run from a consistent time base, aiding correlation of events on multiple devices. Other aspects of switch management include automatic accounting logs which record every configuration command, secure management of logs and software images, IP access-control-lists (ACLs) to restrict IP packets from entering management interfaces and role-based authorization to limits access to switch operations by assigning users to roles.

All of these features and more are standard across the entire Cisco MDS 9000 family in the base system configuration. No additional software licenses are required to enable any of this functionality.

The complete set of switch management security features is listed in this section below:

5.1 Authentication & Authorization

5.1.1 Management Paths

Management Security can be independently configured for each of the following management paths:

- **Command-line interface (CLI)**
  The CLI can be accessed using one of three connection options:
  - **Console (serial connection)**
  - **Telnet**
  - **Secure Shell Protocol (SSH)** (see section 5.3 for details on SSH)
- **Simple Network Management Protocol (SNMP)**
  The SNMP agent supports security features for versions 1, 2c, and 3.
  SNMP security mechanisms apply to all applications that use SNMP (for example, Cisco MDS 9000 Fabric Manager and Device Manager make use of SNMP).
  (see section 5.4 for details on SNMP security)

Each management path (console, Telnet, SSH and SNMP), can use different authentication mechanisms – for example, console access could use local-database only whereas Telnet, SSH and SNMP could be configured to use RADIUS or TACACS+.

Authentication mechanisms include:

• **Local user database**	Look up the username/password in an on-switch database
• **RADIUS (centralized)**	Authenticate the username/password against centralized RADIUS server(s)
• **TACACS+ (centralized)**	Authenticate the username/password against centralized TACACS+ server(s)
• **None**	Allow a valid username to connect in with no password

The following table shows the relative security strengths and security features available with each management path:

Security Features	CLI (Console or Telnet/SSH Access)	SNMP (SNMPv1, SNMPv2c and SNMPv3 access)
User authentication	Local and RADIUS	Local only
Role-based authorization		
Accounting		Local and RADIUS (logging of configuration commands)
Encryption management access	SSH only (not applicable for console or Telnet access)	SNMPv3
Anti-replay attack and prevention of man-in-middle attack		

☞ Up until SAN-OS 1.2, users and roles configured through the CLI were different to users and roles configured through SNMP and did not necessarily correspond to each other. It was possible (typically, very desirable) to configure roles and roles for both CLI and SNMP to be identical.

Beginning with SAN-OS 1.3, users and roles can be tied together to allow for a consistent security policy across all available management paths in the Cisco MDS 9000 product family.

Individual management paths may be disabled. The following configuration excerpt shows enabling/disabling various management paths:

switch(config)# **no telnet server enable**	Disable telnet access
switch(config)# **telnet server enable**	Enable telnet access (default)
switch(config)# **no ssh server enable**	Disable SSH server
switch(config)# **ssh server enable**	Enable SSH server

From a security perspective, it would be prudent to only use management paths that provide encrypted management access and employ anti-replay attack protection – i.e. SSH access to the CLI and SNMPv3 network-based management. If maximum security is required, then one would be prudent to disabling the other forms of management (i.e. disabling the Telnet service and not defining any SNMPv1/v2 community strings).

5.1.2 Authentication & Authorization Process

Authentication is the process of verifying the identity of the person managing the switch. This identity verification is based on the user ID and password combination provided by the person trying to manage the switch. The Cisco MDS 9000 Family switches allow you to perform local authentication (using the lookup database) or remote authentication (using one or more RADIUS servers or TACACS+ servers).

The steps shown in Figure 14 below illustrate authorization and authentication process.

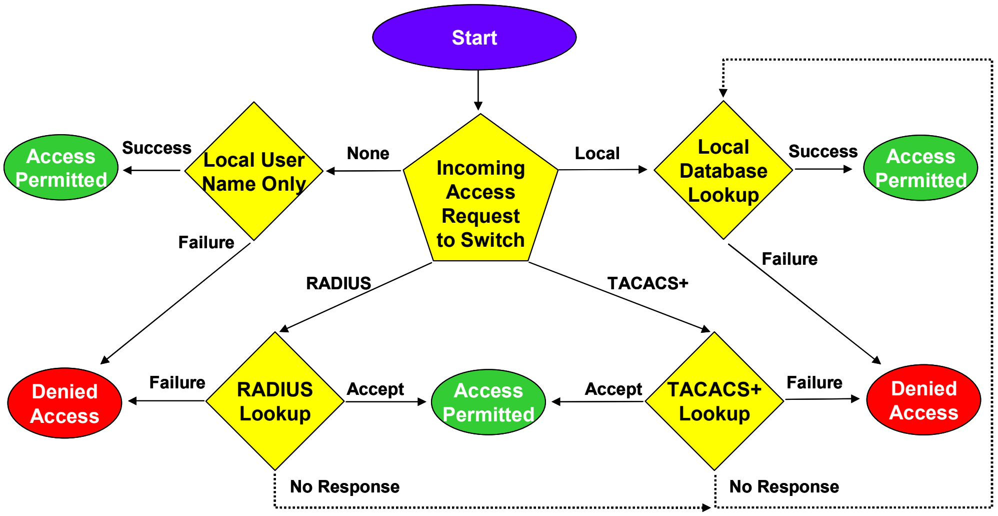


Figure 14 Authentication & Authorization Process

The first decision on an incoming authentication/authorization request is to determine what authentication methods are most suitable for the incoming request. This is determined by what authorization method are configured for the given management path. The following configuration excerpt shows the options available:

switch(config)# **aaa authentication login local console**	Set local authentication for incoming connection attempts on the console port
switch(config)# **aaa authentication login none console**	Allow any valid username to connect in on the console without any password
switch(config)# **aaa authentication login**	Use RADIUS to authenticate username/passwords for

radius console	connections on the console port
switch(config)# **aaa authentication login radius telnet**	Use RADIUS to authenticate username/passwords for incoming connections via telnet or SSH
switch(config)# **aaa authentication login tacacs telnet**	Use TACACS+ to authenticate username/passwords for incoming connections via telnet or SSH

If the authorization method is configured as RADIUS or TACACS+ and there is no response from the centralized server(s), the local user database will be used. This allows for successful authentication in the event that the centralized servers cannot be reached due to any outages.

For specific details on configuring the Local User Database, RADIUS or TACACS+, refer to section 5.2 below.

The decision to whether to use the local user database on the switch or a centralized user database via RADIUS or TACACS+ is a function on the number of switches and the number of users/administrators requiring access and whether any existing corporate standards exist for securing network infrastructure.

If there is only a small number of users/administrators and just a couple of switches then the Local User Database can easily be used.

If there are a larger number of users/administrators (e.g. more than 5), more than just a couple of switches, or an existing network corporate authentication service, then a centralized account management system built using RADIUS and/or TACACS+ should be used.

5.2 Account Management

5.2.1 Local User Database

The Local User Database is a set of locally configured user profiles that are stored in the configuration of each switch. The local user database contains entries for username, password, password expiration date and role membership. All entries are stored in plain text except for the password which is stored in encrypted form.

The same configuration command (‘username’) is used to create a local user account and to update an existing user account. The following configuration excerpt shows the local user database being modified:

switch(config)# **username admin password foo**	Create new user ‘admin’ user with password ‘foo’, or update existing ‘admin’ user’s password to be ‘foo’.
switch(config)# **username bob password bar role operator**	Create/update user ‘bob’ with password ‘bar’ with security capabilities limited to ‘operator’ role
switch(config)# **no username bob**	Remove user ‘bob’ from the local user database

switch(config)# **username bob expire 2004-04-01 password bar role operator**	Create/update user ‘bob’ with password ‘bar’ with security capabilities limited to ‘operator’ role with account expiry on 1 April 2004.

A user profile will be disabled once the system date reaches the password-expiration date configured for a given user profile. By default, the user accounts do not expire unless you explicitly configure to expire.

> The following words are reserved and cannot be used to configure users:
> bin, daemon, adm, lp, sync, shutdown, halt, mail, news, uucp, operator, games, gopher, ftp, nobody, nscd, mailnull, rpc, rpcuser, xfs, gdm, mtsuser, ftpuser, man, sys.

5.2.2 RADIUS-based Centralized Account Management

One way of centralizing account management is to make use of RADIUS authentication. With RADIUS, it is possible to take advantage of one-time passwords (OTP) and password-token devices.

RADIUS is a set of protocols for remote authentication, standardized within the IETF. The RADIUS protocol is primarily documented in RFC 2865 (Remote Authentication Dial In User Service) and RFC 2866 (RADIUS Accounting)

With RADIUS authentication, the Cisco MDS 9000 switch acts as a RADIUS client and can contact up to 5 RADIUS server(s) to validate the authorization credentials of a user. If one of the RADIUS servers is configured as a *primary* server, then it will always be contacted first. If no single server is marked as being a *primary* server, then multiple servers will be tried in the order in which they are configured.

The RADIUS based authorization process is as follows:

1. The switch sends an Access-Request packet to a RADIUS server.
2. The RADIUS server either fails to respond, or responds with an Accept or Reject message
   - In the case of a failure to respond, the switch may retry the Access-Request multiple times (as many iterations have been configured). Once the maximum number of iterations has been reached, other RADIUS server(s) will tried.
     If no RADIUS servers are contactable, authentication will fall back to the local user database.
   - If Access-Reject is received, that means authentication has failed and the user will be denied access.
   - If Access-Accept is received, that means authentication is successful and the user will be permitted access.

In the case of a successful Authentication, the role/capability of the user may optionally be included in the form of Vendor Specific Attributes (VSA) (VSAs are covered in more detail later in this section):

   - If VSA data exists, the user will be made a member of all groups indicated in the role-list attribute within the VSA response.

- If no VSA data is sent, local authorization is used.

If there is no Local user profile associated with the username used as a successful login via RADIUS, a new local user account will be created. This new account is locked and cannot be used for local login (i.e. authentication via RADIUS is enforced). The local account will automatically be removed from the configuration after 24 hours.

There are many configuration options available when configuring communication between the switch and RADIUS server(s). The following options may be configured on a global basis or a per-radius-server basis:

RADIUS Preshared Key
A RADIUS preshared key is necessary for the switch to communicate with the RADIUS server. The length of the key is restricted to 65 characters and can include any printable ASCII characters (white spaces are not allowed). Preshared keys can both be configured on a global basis (used for all RADIUS servers), or configured on a per-radius-server basis (overriding the global key).

RADIUS Server Time-Out Interval
The time between retransmissions to the RADIUS servers can both be configured both on a global basis and overridden on a per-radius-server basis. The timeout interval may configured to anywhere between 1 to 60 seconds. The default timeout period is 1 second.

RADIUS Server Iterations
The number of attempts to contact a configured RADIUS server can be configured both on a global basis and overridden on a per-radius-server basis. The maximum number of iterations per server can be configured between 1 and 5. By default, a switch will retry a RADIUS server authentication attempt only once.

The following configuration excerpt shows many of the RADIUS configuration options available:

switch(config)# **radius-server key bar**	Set the global shared secret to 'bar' for all RADIUS servers
switch(config)# **no radius-server key bar**	Clear the global shared secret
switch(config)# **radius-server timeout 10**	Set the global timeout between retries to 10 second. The time range in seconds is 1 to 60. seconds. The default time-out is 1 second.
switch(config)# **no radius-server timeout**	Set the global timeout to the default of 1 second.
switch(config)# **radius-server retransmit 3**	Set the global number of retries to contact a RADIUS server to 3 retries.
switch(config)# **no radius-server retransmit**	Set the global number of retries to the default of 1 time.
switch(config)# **radius-server host 10.10.10.100**	Adds the RADIUS server with the IP Address 10.10.10.100 to the RADIUS server list.
switch(config)# **no radius-server host**	Removes the RADIUS server with the IP Address 10.10.10.100

10.10.10.100	to the RADIUS server list.
switch(config)# **radius-server host 10.10.10.101 primary**	Adds the RADIUS server with the IP Address 10.10.10.101 to the RADIUS server list as the primary server (that will be tried first)
switch(config)# **radius-server host 10.10.10.101 key HostKey**	Use the preshared key 'HostKey' when communicating with the RADIUS server at 10.10.10.101. This key overrides any global key (if defined)
switch(config)# **radius-server host 10.10.10.101 auth-port 2003**	Configures the switch to communicate with the RADIUS server at 10.10.10.101 using UDP port 2003 as the destination port where the RADIUS service is running. This overrides the standard RADIUS authentication port of UDP Port 1812.
switch(config)# r**adius-server host 10.10.10.101 acct-port 2004**	Configures the switch to communicate with the RADIUS server at 10.10.10.101 using UDP port 2004 as the destination for RADIUS accounting messages. This overrides the default standard RADIUS accounting port of UDP Port 1813.
switch(config)# **radius-server host 10.10.10.102 accounting**	Configures the RADIUS server at 10.10.10.102 to be used for accounting only and not for authentication. Note: If neither the authentication option nor the accounting options are specified, the server is used for both accounting and authentication purposes

👍 If the authorization method is configured as RADIUS and there is no response from the centralized RADIUS server(s), the local user database will be used. This allows for successful authentication in the event that the centralized servers cannot be reached due to any outages.

Vendor-Specific Attributes

The RADIUS protocol contains a method for vendors to communicate device-specific or vendor-specific information that isn't covered as part of the standard RADIUS attributes between clients and servers. These are called vendor-specific attributes (VSAs) and are included as attribute 26 in the RADIUS standard.

The Cisco MDS 9000 family currently supports one vendor-specific option using the format recommended in the specification. The Cisco vendor ID is 9, and the supported option is vendor type 1, which is named 'cisco-avpair'. The value is a string with the following format:

```
protocol : attribute sep value *
```

where protocol is a Cisco attribute for a particular type of authorization, and sep is = for mandatory attributes, and * is for optional attributes. If a value contains any white spaces, it should be put within double quotation marks.

When you use RADIUS servers for authentication, the RADIUS protocol directs the RADIUS server to return user attributes like authorization information, along with authentication results. This authorization information is specified through VSAs.

The Cisco MDS 9000 family supports the following VSA protocol options:

- Shell protocol
  used in Access-Accept packets to provide user profile information.
- Accounting protocol
  used in Accounting-Request packets.

The following attributes are supported:

- **Roles**
  This attribute lists all the roles to which the user belongs. The value field is a string storing the list of group names delimited by white space.
  For example, if you belong to roles 'vsan-admin' and 'storage-admin', the value field would be "vsan-admin storage-admin." This sub-attribute would sent in the VSA portion of the Access-Accept frames from the RADIUS server, and it would only be used with the shell protocol value.
  The following shows an example of using the roles attribute:

  Cisco-AVPair = "shell: roles = "network-admin vsan-admin" "
- Accountinginfo
  This attribute stores additional accounting information besides the attributes covered by a standard RADIUS accounting protocol. This attribute is only sent in the VSA portion of the Account-Request frames from the RADIUS client on the switch, and it can only be used with the accounting protocol value.

Please consult your RADIUS Server documentation for further details on configuring VSAs.

5.2.3 TACACS+-based Centralized Account Management

In SAN-OS 1.3, TACACS+ can also be used to provide centralized account management.

TACACS+ (Terminal Access Controller Access Control System +) was defined to provide AAA functionality for Cisco routers, network access servers and other networked devices via one or more centralized servers. TACACS+ predates RADIUS and evolved from two earlier versions (TACACS and XTACACS) to provide a flexible, secure and powerful modular authentication, authorization and accounting framework. The main difference between TACACS+ and RADIUS is that each service (authentication, authorization, accounting) can be tied into its own database or performed independently by different centralized servers based on requirements and server capabilities. It allows authorization to be done without authentication. TACACS+ also has the option to encrypt the whole payload in a security exchange, thereby providing a higher level of data confidentiality than the shared-secret-password offered by RADIUS.

The Cisco MDS 9000 switch acts as a TACACS+ client and can contact up to 5 TACACS+ server(s) to validate the authorization credentials of a user. TACACS+ Servers will be contacted in the order configured in the AAA group command.

The TACACS+ based authorization process is as follows:

1. The switch sends an Authentication Login request to the TACACS+ server.
2. The TACACS+ server will either fail to respond, or responds with an AuthorizedPermitted or DeniedAuthorization message
    - In the case of a failure to respond, the switch will use other TACACS+ servers listed in the 'AAA group server tacacs'. Note that since TACACS+ uses TCP, there is no need for retransmissions. If no TACACS+ servers respond, authentication will fall back to the local user database.
    - If an DeniedAuthorization response is received, that means authentication has failed and the user will be denied access.
    - If an AuthorizationPermitted message is received, that means authentication is successful and the user will be permitted access.

    In the case of a successful Authentication, the role/capability of the user may optionally be included in the form of Attribute-Value (AV) pairs. (TACACS+ AV pairs are covered in more detail later in this section):

    - If AV pairs indicate any special service/shell/role, this will be applied to the login session
    - If no AV pairs are present in the response, the roles will be as specified in the local user database.

    If the login attempt is successful and there is no local user profile associated with the username, a new local user account will be created. This new account is locked and cannot be used for local login (i.e. authentication via TACACS+ is enforced). The local account will automatically be removed from the configuration after 24 hours.

There are many configuration options available when configuring communication between the switch and TACACS+ server(s). The following options may be configured on a global basis or a per-server basis:

TACACS+ Key
A TACACS+ key is necessary for the switch to communicate with the TACACS+ server. The length of the key is restricted to 65 characters and can include any printable ASCII characters (white spaces are not allowed). TACACS+ keys can both be configured on a global basis (used for all TACACS+ servers), or configured on a per -server basis (overriding the global key).

TACACS+ Server Time-Out
The timeout interval to wait for a TACACS+ server can both be configured both on a global basis and overridden on a per -server basis. The timeout interval may configured to anywhere between 1 to 60 seconds. The default timeout period is 5 seconds.

The following configuration excerpt shows many of the TACACS+ configuration options available:

switch(config)# **tacacs-server key bar**	Set the global key to 'bar' for all TACACS+ servers

switch(config)# **no tacacs-server key bar**	Clear the global key; if no key is set then TACACS+ messages will be in clear text
switch(config)# **tacacs-server timeout 10**	Set the global timeout waiting for a TACACS+ response to 10 second. The time range in seconds is 1 to 60. seconds. The default time-out is 5 seconds.
switch(config)# **no tacacs-server timeout**	Set the global timeout to the default of 5 seconds.
switch(config)# **tacacs-server host 10.10.10.100**	Adds the TACACS+ server with the IP Address 10.10.10.100 to the TACACS+ server list.
switch(config)# **no tacacs-server host 10.10.10.100**	Removes the TACACS+ server with the IP Address 10.10.10.100 to the TACACS+ server list.
switch(config)# **tacacs-server host 10.10.10.101 key HostKey**	Use the key 'HostKey' when communicating with the TACACS+ server at 10.10.10.101. This key overrides any global key (if defined)
switch(config)# **tacacs-server host 10.10.10.101 port 555**	Configures the switch to communicate with the TACACS+ server at 10.10.10.101 using TCP port 555 as the destination port where the TACACS+ server is running. This overrides the standard TACACS+ TCP port 49.
switch(config)# **tacacs-server host 10.10.10.101 key foo port 556 timeout 3**	Adds the TACACS+ server with the IP Address 10.10.10.101 to the TACACS+ server list. Use the key 'foo' when communicating with the TACACS+ server, connect to the server on TCP port 556 and use a timeout of 3 seconds

Beginning with SAN-OS 1.3, the global AAA syntax will change slightly; rather than AAA services being applied on a per-interface basis (e.g. telnet interface, console interface, SNMP interface), they are moving more towards a per-service interface. With this in mind, the syntax to enable TACACS+ against a 'service' is as follows:

switch(config)# **aaa authentication login default group tacacs+ local**	Enable TACACS+ authentication as the default for all management interfaces; if no TACACS+ servers can be contacted, fall back to the local user database
switch(config)# **aaa authentication login default group tacacs+**	Enable TACACS+ authentication as the default for all management interfaces; if no TACACS+ servers can be contacted, deny access
switch(config)# **aaa authentication login console group tacacs+ group radius local**	Enable TACACS+ authentication for the console interface; if no TACACS+ servers can be contacted, fall back to trying the RADIUS servers. If none of them can be contacted, fall back to using the local user database
switch(config)# **aaa authentication**	Enable RADIUS authentication for FC-SP-based DHCHAP

dhchap default group radius local	authentication. If no RADIUS servers can be contacted, fall back to using the local user database
switch(config)# **aaa accounting default group radius local**	Enable RADIUS-based and local accounting

> Fallback to using the local user database is prudent, at least for console access, as it allows for successful authentication in the event that the centralized servers cannot be reached due to any outages.

Attribute/Value (AV) Pairs

TACACS+ uses AV pairs to send authorization information from the server to the client. Based on AV pairs, the client can determine access privileges and other information pertaining to the entity to be authorized. AV pairs have a fixed ASCII format that provides a fair degree of flexibility in providing authorization credentials.

AV pairs are specified in either the format "*<attribute>=<value>*" or "*<attribute>*<value>*". The former indicates a mandatory attribute in the AV pair, the latter an optional attribute. Mandatory attributes require that the receiving end understand the attribute and must act on it. If they cannot oblige or do not understand the attribute, authorization is considered to have failed.

The Cisco MDS 9000 family currently supports the following TACACS+ AV pair:

- **Roles**
  This attribute lists all the roles to which the user belongs. The value field is a string storing the list of group names delimited by white space.
  For example, if you belong to roles 'vsan-admin' and 'storage-admin', the AV pair would be:

  roles="vsan-admin storage-admin"

Please consult your TACACS+ Server documentation for further details on configuring AV pairs.

5.3 Secure Shell (SSH)

SSH provides the following benefits over traditional non-encrypted administration transports such as telnet/rlogin/ftp/tftp:

- Strong authentication: Closes several security holes (e.g., IP, routing, and DNS spoofing).
- Improved privacy: All communications are automatically and transparently encrypted.
- No retraining needed for normal users – tools such as *ssh* are similar to *rsh*/*telnet*; *scp* similar to *rcp*.
- Never trusts the network: Minimal trust on the remote side of the connection. Minimal trust on domain name servers. Pure RSA authentication never trusts anything but the private key.

- Client RSA-authenticates the server machine in the beginning of every connection to prevent trojan horses (by routing or DNS spoofing) and man-in-the-middle attacks, and the server RSA-authenticates the client machine before accepting .rhosts or /etc/hosts.equiv authentication (to prevent DNS, routing, or IP-spoofing).
- Host authentication key distribution can be centrally by the administration, automatically when the first connection is made to a machine.
- The SSH Server has its own server RSA key which is automatically regenerated every hour in order to provide perfect forward secrecy to each management session.
- An authentication agent, running in the user's laptop or local workstation, can be used to hold the user's RSA authentication keys.
- Optional compression of all data with gzip, which may result in significant speedups on slow connections.

5.3.1 SSH Server

SSH enables secure remote login for interactive command-line administration or for automated scripting of CLI commands. The entire family of the Cisco MDS 9000 product family implements a Secure Shell (SSH) Server.

Before the SSH Server may be enabled, appropriate SSH host key pairs need to be generated. The SSH server accepts three types of key pairs for use by SSH versions 1 (RSA) and 2 (RSA and DSA). The number of bits specified for each key pair ranges from 768 to 2048.

By default, the SSH service is disabled. The following configuration excerpt shows enabling/disabling the SSH server and generation of key-pairs:

switch(config)# **ssh server enable**	Enable SSH server
switch(config)# **no ssh server enable**	Disable SSH server (default)
switch(config)# **ssh key rsa1 1024**	Generate RSA key pair for SSHv1
switch(config)# **ssh key dsa 1024**	Generate DSA key pair for SSHv2
switch(config)# **ssh key rsa 1024**	Generate RSA key pair for SSHv2
switch(config)# **no ssh key rsa 1024**	Remove RSA key pair for SSHv2
switch(config)# **ssh key dsa 1024 force**	Regenerate a DSA key pair for SSHv2, replacing an existing key

The Command Line Interface (CLI) within the Cisco MDS 9000 product family even allows CLI commands to be issued remotely via SSH. The following excerpt shows how it is possible to script automated data collection from a Cisco MDS switch:

```
bock-bock-au% ssh -l admin mel-stglab-mds9509-1 "show version"
admin@mel-stglab-mds9509-1's password: *********
Cisco Storage Area Networking Operating System (SAN-OS) Software
TAC support: http://www.cisco.com/tac
Copyright (c) 2002-2003 by Cisco Systems, Inc. All rights reserved.
```

```
The copyright for certain works contained herein are owned by
Andiamo Systems, Inc. and/or other third parties and are used and
distributed under license.


Software
  BIOS:      version 1.0.8
  loader:    version 1.2(2)
  kickstart: version 1.2(2) [build 1.2(1.3)]
  system:    version 1.2(2) [build 1.2(1.3)]

  BIOS compile time:       08/07/03
  kickstart image file is: bootflash:/m9500-sf1ek9-kickstart-mz.1.2.1.3.bin
  kickstart compile time:  9/4/2003 4:00:00
  system image file is:    bootflash:/m9500-sf1ek9-mz.1.2.1.3.bin
  system compile time:     9/4/2003 4:00:00


Hardware
  RAM 1024576 kB

  bootflash: 503808 blocks (block size 512b)
  slot0:          0 blocks (block size 512b)

  mel-stglab-mds9509-1 uptime is 6 days 3 hours 11 minute(s) 11 second(s)

  Last reset at 405703 usecs after Thu Sep 25 16:48:31 2003
    Reason: Reset Requested by CLI command reload
    System version: 1.2(2)


>>SSH Session closed
bock-bock-au%
```

In combination with ssh-key and the user's credentials from the '.ssh/authorized_hosts' file, it is possible to allow selected users to login without requiring them to present a password. Consult your SSH documentation for details on how to use authorized_keys.

5.3.2 SSH Client

A Secure Shell Client is available on all models within the Cisco MDS 9000 product family. This enables one to initiate a secure session outbound from the switch.

The following CLI excerpt shows how to make use of the SSH client:

switch# **ssh ltd@bock-bock-au.cisco.com** ltd@bock-bock-au.cisco.com's password: **** bock-bock-au% **logout** switch#	Initiate a SSH connection using username 'ltd' to the 'bock-bock-au.cisco.com'

5.3.3 Secure management of Logs & Images using Secure Copy (scp) and SFTP

The Secure Shell Client within the Cisco MDS 9000 product family also enables a secured channel for file transfers when moving log files and system images to/from the switch. Both Secure Copy (scp) and Secure FTP (FTP via a SSH tunnel) are supported.

The following CLI excerpt shows how to make use of the SSH client for secure transfer of log files and system images to/from the switch:

switch# **copy scp://ltd@bock-bock-au.cisco.com/~ltd/ mds_images/m9500-sf1ek9-kickstart-mz.1.2.1a.bin bootflash:m9500-sf1ek9-kickstart-mz.1.2.1a.bin**	Use SSH (secure copy) to copy an image file from the host 'bock-bock-au' using the username 'ltd' and store the image within bootflash.
switch# **copy running-config scp://ltd@bock-bock-au.cisco.com/~ltd/config/ 9509cfg.txt**	Use SSH (secure copy) to save a copy of the current running-configuration onto the host 'bock-bock-au' in the file '~ltd/config/9509cfg.txt'.
switch# **copy running-config sftp://ltd@bock-bock-au.cisco.com/~ltd/config/ 9509cfg.txt**	Use SFTP to save a copy of the current running-configuration onto the host 'bock-bock-au' in the file '~ltd/config/9509cfg.txt'.
switch# **show tech-support > dump.txt** switch# **copy volatile:dump.txt scp://ltd@bock-bock-au.cisco.com/~ltd/debug/dump.txt**	Generate a tech-support dumo file and store it with the filename 'dump.txt' within the volatile: filesystem. Use SSH (secure copy) to copy the dump file to the host bock-bock-au.

5.4 SNMP Security

The Simple Network Management Protocol (SNMP) is an application-layer protocol designed to facilitate the exchange of management information between network devices. By using SNMP-transported data (such as packets per second and network error rates), network administrators can more easily manage network performance, find and solve network problems, and plan for network growth.

There are three versions of SNMP: Version 1, Version 2 and Version 3.

SNMP Version 1 (SNMPv1) was initially designed as a stop-gap measure for network management until a more sophisticated method could be developed. It was created to allow simple yet effective network management. The SNMPv1 standard is documented in Internet RFCs RFC 1155, RFC 1157, and RFC 1212.

SNMPv1 was rapidly superceded by SNMPv2c, predominantly due to the lax security measures in place with SNMPv1. SNMPv2c builds upon SNMPv1, with most of the changes introduced in SNMPv2c targeting additional security capabilities and increasing interoperability by more rigorously defining the implementation specifications. The SNMPv2c standard is documented in Internet RFCs 1441 through 1452 and later RFC 1902.

While SNMPv2c provided a security improvement over SNMPv1, it still used a relatively weak form of security for access control – a community string – and lacked mechanisms to protect against session hijacking.

SNMPv3 was created to provide a significant improvement over SNMPv2c access controls by making use of strong authentication and encryption. The SNMPv3 standard is documented in Internet RFCs 2271-2275 and again in 3410-3415. The additional security benefits provided by SNMPv3 over SNMPv2c include:

- **Message integrity**
  Ensures that a packet has not been tampered with in-transit.
- **Authentication**
  Determines the message is from a valid source.
- **Encryption**
  The contents of the packet are scrambled to prevent it from being seen by unauthorized sources.

Key to this functionality was that security in SNMPv3 is covered both from the perspective of both message-level security and user-level-security:

- **User Based Security Model (USM)**
  The SNMPv3 "User Based Security Model" (RFC 3414) ensures that there is message-level security and provides the basis for message integrity, authentication and encryption. This layer ensures that management messages and traffic are not tampered with and that the communication between devices cannot be hijacked for malicious intent.
- **View-based Access Control Model (VACM)**
  The SNMPv3 "View-based Access Control Model" (RFC 3415) is designed to control access to management information based on a user's identity. VACM allows different access levels (read, write, notify) to be defined for different users and for each piece of MIB information.

SNMPv3 requires a username/password combination for all management traffic. A user at a management station provides this username/password, which in turn is used as a key with either MD5 (Message Digest 5) or SHA (Secure Hash Algorithm) to create an authentication values applied to the SNMP packets.

The SNMP agent receiving the SNMPv3 packet applies the same algorithm with the same key and checks if its produced value is the same as the one in the message. The key used for the authentication is associated to the username, which is present within the SNMP message. The authentication functionality ensures that each SNMP message comes from an authorized user or agent, and that it was not tampered with in transit.

SNMPv3 messages are encrypted with DES (Data Encryption Standard) to prevent eavesdropping and ensure privacy of management information. When sending or receiving a message encrypted with DES, both the sender and receiver of the message must have the same private key.

SNMPv3 USM also has mechanisms for checking the timeliness of SNMP packet delivery using synchronization and time-window checking techniques. This detects messages that have been delayed, which is important because delay is often an indicator that packets have been altered.

After a user authenticates as specified in the USM, all SNMPv3 commands generated carry his/her credentials. SNMP agents check the user's information against a pre-configured access control database before allowing access to any MIB object. This gives network managers the ability to define different access rights for different administrators.

The entire family of the Cisco MDS 9000 product family supports all of SNMPv1, SNMPv2 and SNMPv3.

For security reasons, it is recommended that only SNMPv3 be used and SNMPv1/v2 access be denied. This can be accomplished by ensuring that there is no SNMP community strings configured on the platform.

Both Cisco Fabric Manager and Cisco Device Manager, the main JAVA WebStart-based management applications that Cisco ships with the MDS 9000 product family support SNMPv3.

If SNMPv1/v2 access is required due to a 3rd party application which doesn't support SNMPv3, then restrictions should be put in place to ensure that only selected IP addresses may issue SNMPv1/v2 requests and that hosts that applications that don't require 'write'-access are explicitly only given read-only access.

5.4.1 SNMPv1 / SNMPv2c

With SNMPv1 and SNMPv2c, a 'community string' is used to identify whether a user is authorized to access the switch with read-only or read-write access. By default, no SNMPv1/SNMPv2c community string is defined, therefore SNMPv1/SNMPv2c access is denied.
The following configuration excerpt shows how this is configured:

switch(config)# **snmp-server community foo ro**	Enables read-only access for the SNMP community 'foo'
switch(config)# **snmp-server community bar rw**	Enables read-write access for the SNMP community 'bar'
switch(config)# **no snmp-server community foo**	Removes the SNMP community 'far'

Further restrictions on what IP Addresses can connect with SNMPv1 and SNMPv2c is possible by directly modifying the *snmpTargetAddressTable* and *snmpTargetAddressTagList* tables from a management host using SNMP. The process to do this is as follows:

1. Create an entry for each host that requires access in the snmpTargetAddressTable. Ensure that all hosts have the same *snmpTargetAddressTagList* tag. (e.g. 'noc_hosts').
2. For each community name in the *snmpCommunityTable* for which access to the switch is required, *snmpCommunityTransportTag* must be set to the same tag as the *snmpTargetAddressTagList* value (e.g. 'noc_hosts').

SNMP access to the switch using the SNMP community listed in the *snmpTargetAddressTable* will now be limited to just those hosts with the specified tag name. Consult the Cisco MDS 9000 family Configuration Guide for additional details on how to apply IP Access Restrictions.

5.4.2 SNMPv3

5.4.2.1 SNMPv3 Username and Password

SNMPv3 requires a valid username and successful password authentication before management access to the switch is granted.

The following configuration excerpt shows how this is configured:

switch(config)# **snmp-server user joe network-admin auth sha abcd1234**	Creates or modifies the settings for a user (joe) in the network-admin role using the HMAC-SHA-96 authentication password (abcd1234).
switch(config)# **snmp-server user sam network-admin auth md5 abcdefgh**	Creates or modifies the settings for a user (sam) in the network-admin role using the HMAC-MD5-96 authentication password (abcdefgh).
switch(config)# **snmp-server user Bill network-admin auth sha abcd1234 priv abcdefgh**	Creates or modifies the settings for a user (network-admin) in the network-admin role using the HMAC-SHA-96 authentication level and privacy encryption parameters.
switch(config)# **no snmp-server user usernameA**	Deletes the user (usernameA) and all associated parameters.
switch(config)# **snmp-server user user1 network-admin auth md5 0xab0211gh priv 0x45abf342 localizedkey**	Specifies the password to be in localized key format (see RFC2574). The localized key is provided in the hex format (for example, 0xacbdef).

> 👍 "*localized keys"* are not portable across devices as they contain device engine ID information. If a configuration file is copied to the device, the passwords will not be set correctly if the configuration file was generated at a different device. Explicitly configure the desired passwords after copying the configuration into the device if the same password is desired across multiple devices.

It is possible to use different passwords for SNMPv3 access and CLI access for the same username. It is possible to synchronize the CLI and SNMPv3 password by making use of the *'update-snmpv3'* keyword on the *username* prompt. Note that SNMPv3 requires a password of at least 8 characters in length.
This is shown in the configuration excerpt below:

switch(config)# **username joe password wxyz6789 update-snmpv3 abcd1234**	Updates the SNMPv3 password for the specified user (joe). The local CLI password and the SNMP password are

	updated. If user Joe does not exist, the command fails.

5.4.2.2 SNMPv3 Groups

User management capabilities are limited according to the capabilities assigned to the user in the common role database. The common role database enables specific users to belong to specific groups and be permitted/denied access to various management aspects of the switch.

The common roles database ties into SNMPv3 groups by being able to limit specific roles (and therefore groups of users) on the items that they can configure. Within SNMPv3, access for any specific subsystem can be limited to read-only access, read and write access, and notification-access only.

For details on configuring roles, see section 5.5.

5.5 Roles-based Access

5.5.1 Roles

All management access within the Cisco MDS 9000 Family is based upon roles. Role-based authorization limits access to switch operations by assigning users to roles. Users are restricted to performing the management operations that are explicitly permitted based on the roles to which they belong.

By default, two roles exist in all switches:

- **Network operator (network-operator)**
  Has permission to view the configuration only. The operator cannot make any configuration changes.
- **Network administrator (network-admin)**
  Has permission to execute all commands and make configuration changes.

These two default roles cannot be changed or deleted. Users belonging to the *network-admin* role are authorized to create and customize up to an additional 64 roles and add other users to those roles.

Each role can contain multiple users and each user can be part of multiple roles.

> ☞ If you belong to multiple roles, you can execute a superset of all the commands permitted by these roles. Access to a command takes priority over being denied access to a command. For example, if the *sys-admin* group explicitly denied access to configuration commands and the *san-operators* group explicitly permitted access to configuration commands, and a user belonged to both of these groups, then they would have access to configuration commands.

The '*rule*' command specifies operations that can be performed by a specific role. Each '*rule*' consists of a rule number, a rule type (permit or deny), a command type (config, clear, show, exec, debug), and an optional feature name (FSPF, zone, VSAN, fcping, interface, …). The ‘exec’ command type refers to all other exec commands other than ‘clear’, ‘show’ and ‘debug’.

The user-specified rule number determines the order in which the rules are applied. For example, rule 1 is applied before rule 2 which is applied before rule 3 etc. Up to 16 rules can be configured for each role.

The following configuration excerpt shows a role being defined:

switch(config)# **role name sangroup**	Create a role called “sangroup”
switch(config-role)# **rule 1 permit config**	Allow users of the “sangroup” role to perform configuration commands
switch(config-role)# **rule 2** **deny config feature fspf**	Deny users of the “sangroup” role from being able to configure any FSPF-related commands
switch(config-role)# **rule 3** **permit debug feature zone**	Allow users of the “sangroup” role to be able to debug Zoning
switch(config-role)# **rule 4** **permit exec feature fcping**	Allow users of the “sangroup” role to be able to use ‘fcping’
switch(config-role)# **no rule 4**	Remote rule 4 (users of the “sangroup” role can no longer use ‘fcping’)

In this configuration, rule 1 is applied first, thus permitting all config commands to *sangroup* users. Rule 2 is applied next, denying FSPF configuration to *sangroup* users. As a result, *sangroup* users can perform all other config commands, except fspf configuration commands.

The optional ‘feature’ can refer to any top-level command for that section. For example, ‘config’ command features include any commands available from the context-sensitive help at the config prompt (i.e. ‘switch(config)# ?’).

As of SAN-OS 1.3, valid features are listed below:

exec	config		show		debug	clear
attach	aaa	iscsi	aaa	kernel	aaa	arp-cache
callhome	arp	isns	accounting	klm	all	cdp
cd	boot	ivr	arp	license	bootvar	cores
clock	callhome	kernel	boot	line	callhome	counters
copy	cdp	line	callhome	loadbalancing	cdp	debug-logfile
delete	cimserver	logging	cdp	loader	cimserver	fcanalyzer
dir	clock	ntp	cimserver	logging	core	fcflow
discover	fabric-binding	port-security	clock	module	ethport	fcns
fcping	fc-tunnel	power	copyright	ntp	exceptionlog	fcs
fctrace	fcalias	poweroff	cores	platform	fc-tunnel	fspf
find	fcanalyzer	qos	debug	port	fc2	ip
format	fcc	radius-server	environment	port-channel	fc2d	ips
gunzip	fcdomain	role	ethport	processes	fcc	iscsi
gzip	fcdroplatency	rscn	fc-tunnel	qos	fcdomain	license
install	fcflow	snmp-server	fc2	radius-server	fcfwd	line
ips	fcinterop	span	fc2d	rlir	fcns	logging
mkdir	fcip	ssh	fcalias	role	fcs	ntp

modem move ping purge pwd quiesce reload rmdir run-scrip send setup sleep ssh system tail telnet terminal test traceroute write zone	fcns fcroute fcs fcsp fctimer ficon fspf in-order-guara interface ip ips switchname system tacacs+ tacacs-server telnet trunk username vsan wwn zone zoneset	fcanalyzer fcc fcdomain fcdroplatency fcflow fcip fcns fcroute fcs fctimer fdmi file flogi fspf hardware hosts in-order-guarant incompatibility install interface ip ipconf ipfc ips iscsi isns rscn running-config scsi-target security sensor snmp span sprom ssh startup-config switchname system tacacs-server tech-support telnet terminal tlport trunk user-account users version vrrp vsan wwn zone zoneset	fdmi flogi fm fspf hardware idehsd ipacl ipconf ipfc ips klm license logfile module ntp platform port port-channel qos radius rib rlir rscn scsi-target security snmp span system tacacs+ tlport vni vrrp vsan wwn zone	processes qos rlir rscn screen ssh user vrrp zone

Prior to SAN-OS 1.3, Roles for CLI access and SNMP access were defined separately. As of SAN-OS 1.3, there is a single common roles database for both CLI and SNMP. When a role is created through CLI or SNMP, it will be created for both CLI and SNMP users.

5.5.2 VSAN-Based Roles

Beginning in SAN-OS 1.2, Role-based authorization has been enhanced to provide further granularity of roles within VSANs, thereby introducing the concept of 'VSAN administrators'. VSAN Administrators can perform configuration changes (for example, zoning changes and domain-controller changes) within their permitted VSAN(s) but cannot configure any global settings (VSAN configuration, Port Channel configuration etc) nor can they perform any configuration on VSANs outside their allowed scope. This ensures that VSAN Administrators cannot perform any configuration changes which impact any other VSAN(s) other than the ones that they control.

Roles can be configured to only allow a specific set of commands to be performed for a selected set of VSANs. By default, all rules within a role are permitted across all VSANs. In order to selectively allow VSANs for a role, the VSAN policy needs to be set to deny and then the appropriate VSANs need to be permitted.

The following configuration excerpt shows how the VSAN policy is configured for a role:

switch(config)# **role name sangroup**	Create a role called “sangroup” (or enter the group if it already exists)
switch(config)# **vsan policy deny**	Configures the policy for role “sangroup” to deny for all VSANs except those explicitly permitted.
switch(config-role)# **no vsan policy deny**	Deletes the configured ‘deny’ VSAN role policy and reverts to the factory default (permit)
switch(config-role-vsan)# **permit vsan 10-30**	Permits rules within the role “sangroup” role to be allows fro VSANs 10 through 30
switch(config-role-vsan)# **no permit vsan 15-20**	Removes the permission for this role to perform commands for vsan 15 to 20 (thereby reducing the rules within the role “sangroup” to be allowed for VSANs 10-14 and 21-30).

VSAN Administrators cannot perform any configuration or look at any statistics/configuration outside of their list of allowed VSANs. For example, a VSAN Administrator who is only allowed to access VSANs 5-10 would not be able to execute any CLI command which references any VSANs outside that range. i.e. if they tried the CLI command “*zone name jbod5_host57 vsan 2*”, the command would be rejected. Similarly, if the VSAN Administrator tried to configure “*zone default-zone permit vsan 1-10*”, the command would be rejected since it includes VSANs outside the permitted range.

<u>Configuration/Clear/Exec Command Restrictions</u>

When VSAN-based roles are being used, the following restrictions apply to configuration, clear and EXEC commands associated with interfaces:

- **FC Interfaces**
  VSAN Administrators can only modify the configuration of FC Interfaces configured for F, FL_Port or Fx_Port. Configuration of E_Port interfaces is explicitly denied so as to prevent such users from modifying configurations that may impact the core topology of a given FC fabric.
  Note that if a role allows a VSAN Administrator configuration control in multiple VSANs, then the VSAN administrator *can* move a port from one VSAN to another (change VSAN membership) among any VSANs that they administer.

- **Port-Channel Interfaces**
  VSAN Administrators cannot modify Port-Channel interfaces, since Port-Channels are used for E_Port connectivity

- **VSAN Interfaces**
  VSAN Administrators can configure VSAN Interfaces on any VSANs that they are explicitly permitted access to. Access to all other VSAN(s) is explicitly denied.

- **FCIP Interfaces**
  VSAN Administrators cannot modify FCIP interfaces, since Port-Channels are used for E_Port connectivity
- Management Interfaces (mgmt 0)
  VSAN Administrators cannot modify management interfaces, since management access is shared across all VSANs.

‘Show’ Command Restrictions

VSAN Administrators can issue ‘show’ commands for multiple VSANs and Interfaces, but the displayed results will only include information from VSANs and Interfaces that the VSAN Administrator is allowed to access.

For example, if a VSAN Administrator issues the command “*show zone active vsan 1-10*” and they were VSAN Administrators on VSANs 1-5, the ‘show’ command will be permitted but only the active zoning for VSANs 1-5 will be displayed.
Likewise, if a VSAN Administrator issued the command “*show interface fc1/1-16*”, only those interfaces that are *within* the VSANs permitted to the VSAN Administrator will be shown.

Note that FC and Port-Channel interfaces that are E_Port *are* viewable to the user, provided the port is within that VSAN or the port transports that VSAN (in the case of a TE_Port or Port-Channel interface).

‘Debug’ Command Restrictions

VSAN-specific ‘debug’ CLI commands are available to VSAN Administrators for those VSANs that are within their permitted list.

Startup-Configuration Restrictions

The startup-configuration is treated as a single entity, therefore it isn’t possible to view/modify it on a per-VSAN basis. VSAN Administrators are therefore explicitly denied from performing the following commands:

1. Commands that modify the startup-configuration
   e.g. “copy <f> startup-config”
2. Commands that view/display the startup-configuration
   e.g. “copy startup-config <file>”, “show startup-config”, “show running-config diff”

(Note: VSAN-based Roles is a feature that is enabled by the Enterprise License Package)

It is recommended that Roles-Based Access be used to enforce operational boundaries on what users may execute what commands. In using Roles-Based Access, one can enforce change-control procedures thereby minimizing the probability of a disruption (intentional or otherwise) due to configuration change.

5.5.3 Example Roles

The following configuration excerpts show the flexibility in management offered by Roles Based Access and how they can best be used:

Example 1: using Roles to limit administrator access

The following example shows the use of roles-based authentication to create different classes of users.

Company A is using roles-based authentication to limit the scope of configuration and have implement change-management procedures. Company A has allocated VSANs 1-100 for their production network and has allocated VSANs above 4000 as 'scratch' VSANs for their storage staff to experiment with.

They make use of the following roles:

- **Network-admin** (built-in role)
  Has permission to execute all commands and make configuration changes across all VSANs. The SAN Architects have this role.
- **Host-operator**
  Has permission to make zoning changes and issue show/debug commands on production VSANs. Cannot change fabric parameters on production VSANs
- **Network-operator** (built-in role)
  Has permission to view the configuration only; cannot make any configuration changes.
- **Scratch-Pad**
  Can perform all/any tasks on non-production (scratch) VSANs

Users are assigned to roles through the centralized RADIUS servers using AAA. Users are assigned to one of Network-admin, Host-operator or Network-operator based on their responsibility / job function. All users are assigned to the 'Scratch-Pad' role.

The following configuration excerpt shows the configuration to implement this role policy:

```
switch(config)# role name Host-operator
switch(config-role)# rule 1 permit config feature zone
switch(config-role)# rule 2 permit config feature zoneset
switch(config-role)# rule 3 deny config
switch(config-role)# rule 4 permit exec
switch(config-role)# rule 5 permit show
switch(config-role)# rule 6 permit debug
switch(config-role)# rule 7 permit clear
switch(config-role)# vsan policy deny
switch(config-role-vsan)# permit vsan 1-100
switch(config-role-vsan)# exit
switch(config-role)# exit
switch(config)# role name Scratch-Pad
switch(config-role)# rule 1 permit config
switch(config-role)# rule 2 permit exec
switch(config-role)# rule 3 permit show
switch(config-role)# rule 4 permit debug
switch(config-role)# rule 5 permit clear
switch(config-role)# vsan policy deny
switch(config-role-vsan)# permit vsan 4000-4092
switch(config-role-vsan)# exit
switch(config-role)# exit
```

Example 2: collapsing multiple (previously physically separate) fabrics into a collapsed-core design using VSANs.

Company A merged with Company B. Each previously had a SAN deployed (2 fabrics), now they wish to merge their infrastructure into a single physical data centre. The first phase of this is to deploy the existing fabrics within the same physical switches and have SAN Administrators from each company manage their own fabrics. The topology for this migration is shown in Figure 15 below:

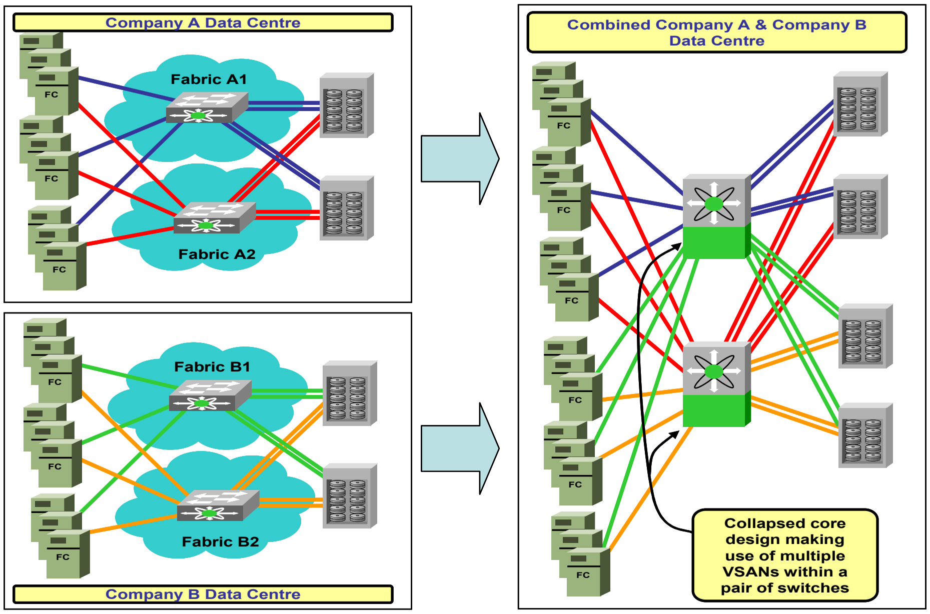

Company A Data Centre
Fabric A1
Fabric A2
FC
Fabric B1
Fabric B2
Company B Data Centre
Combined Company A & Company B Data Centre
Collapsed core design making use of multiple VSANs within a pair of switches

Figure 15 SAN Island Migration into a Collapsed Core design

The following configuration excerpt shows the configuration to implement this role policy. Note that there are policies for both SAN Administrators (SANAdmin_A, SANAdmin_B) and also for System Administrators (SysAdmin_A, SysAdmin_B) who have view-only access:

switch(config)# **vsan database**	Enter the VSAN database
switch(config-vsan-db)# **vsan 11 name Fabric_A1** switch(config-vsan-db)# **vsan 11 interface fc1/1-8**	Use VSAN 11 as Fabric ‘A1’. use ports fc1/1-8.
switch(config-vsan-db)# **vsan 12 name Fabric_A2** switch(config-vsan-db)# **vsan 12 interface fc2/1-8**	Use VSAN 11 as Fabric ‘A2’. use ports fc2/1-8.

switch(config-vsan-db)# **vsan 21 name Fabric_B1** switch(config-vsan-db)# **vsan 21 interface fc1/9-16**	Use VSAN 11 as Fabric ‘B1’. use ports fc1/9-16.
switch(config-vsan-db)# **vsan 22 name Fabric_B2** switch(config-vsan-db)# **vsan 22 interface fc2/9-16**	Use VSAN 11 as Fabric ‘B2’. use ports fc2/9-16.
switch(config-vsan-db)# **exit**	exit VSAN database
switch(config)# **role name SANAdmin_A** switch(config-role)# **rule 1 permit config** switch(config-role)# **rule 2 permit exec** switch(config-role)# **rule 3 permit show** switch(config-role)# **rule 4 permit debug** switch(config-role)# **rule 5 permit clear** switch(config-role)# **vsan policy deny** switch(config-role-vsan)# **permit vsan 11-19** switch(config-role-vsan)# **exit** switch(config-role)# **exit**	Configure role for SAN Administrators from Company A. Permit full configuration within VSANs 1x (11-19)
switch(config)# **role name SANAdmin_B** switch(config-role)# **rule 1 permit config** switch(config-role)# **rule 2 permit exec** switch(config-role)# **rule 3 permit show** switch(config-role)# **rule 4 permit debug** switch(config-role)# **rule 5 permit clear** switch(config-role)# **vsan policy deny** switch(config-role-vsan)# **permit vsan 21-29** switch(config-role-vsan)# **exit** switch(config-role)# **exit**	Configure role for SAN Administrators from Company B. Permit full configuration within VSANs 2x (21-29)
switch(config)# **role name SysAdmin_A** switch(config-role)# **rule 1 deny config** switch(config-role)# **rule 2 deny exec** switch(config-role)# **rule 3 permit exec feature terminal** switch(config-role)# **rule 4 permit show** switch(config-role)# **rule 5 permit debug** switch(config-role)# **rule 6 deny clear** switch(config-role)# **vsan policy deny** switch(config-role-vsan)# **permit vsan 11-19** switch(config-role-vsan)# **exit** switch(config-role)# **exit**	Configure role for System Administrators from Company A. Allow read-only access to show/debug EXEC commands, deny all config and other EXEC commands. (note that the exec feature ‘term’ is allowed so as to permit SysAdmin’s to use “term mon”). Limit access to VSANs 1x (VSAN 11-19).
switch(config)# **role name SysAdmin_B** switch(config-role)# **rule 1 deny config** switch(config-role)# **rule 2 deny exec** switch(config-role)# **rule 3 permit exec feature terminal** switch(config-role)# **rule 4 permit show** switch(config-role)# **rule 5 permit debug** switch(config-role)# **rule 6 deny clear** switch(config-role)# **vsan policy deny** switch(config-role-vsan)# **permit vsan 21-29** switch(config-role-vsan)# **exit** switch(config-role)# **exit**	Configure role for System Administrators from Company B. Allow read-only access to show/debug EXEC commands, deny all config and other EXEC commands. (note that the exec feature ‘term’ is allowed so as to permit SysAdmin staff to use “term mon”). Limit access to VSANs 2x (VSAN 21-29).

5.6 IP ACLs

IP Access control lists (ACLs) are used to provide basic network security (packet filtering) on the out-of-band management Ethernet interface and in-band IP management via IP-FC. IP ACLs can be used are used to restrict traffic from unknown and untrusted sources and even restricts network use based on user identity or device type.

IP ACLs function just like 'access-lists' on Cisco IOS routers:

- An ACL is a sequential collection of 'permit' and 'deny' conditions that apply to IP address(es).
  Each address is tested against the conditions in the list with the first match determining if the packet is accepted or rejected. Since the address testing stops after the first match, the order of the conditions in the list is critical. If no conditions match, the packet is dropped.
- An IP protocol can be configured using an integer ranging from 0 to 255 to represent a particular IP protocol. Alternatively, the name of an protocol (ip, icmp, tcp, or udp). 'ip' means all IP protocols.
- The source/source-wildcard and destination/destination-wildcard is specified in one of two ways:
  - Using the 32-bit quantity in four-part, dotted decimal format (10.1.1.2/0.0.0.0 is the same as host 10.1.1.2).
  - Using the 'any' option as an abbreviation for a source/source-wildcard or destination/destination-wildcard (0.0.0.0/255.255.255.255)

> ☞ IP ACLs can only be applied to the management interface and not to Gigabit Ethernet interfaces on the IP Services module.

The following configuration excerpt shows how IP ACLs can be used to restrict management access to the Cisco MDS 9000 family:

switch(config)# **ip access-list restrict_mgmt permit ip 10.67.16.0 0.0.0.255 any**	Add/Create an entry on access-list "restrict_mgmt" allowing all addresses in the 10.67.16.0/24 subnet
switch(config)# **ip access-list restrict_mgmt permit icmp any any eq 8**	Add entry on access-list "restrict_mgmt" to allow any device to ping the MDS (icmp type 8)
switch(config)# **ip access-list restrict_mgmt deny ip any any**	Explicitly block all other access for access-list "restrict_mgmt"
switch(config)# **interface mgmt 0**	Enter management interface configuration
switch(config-if)# **ip access-group restrict_mgmt in**	Apply "restrict_mgmt" access-list on inbound traffic on interface mgmt0.
switch(config-if)# **no ip access-group restrict_mgmt in**	Remove "restrict_mgmt" access-list on inbound traffic on interface mgmt0 (default)

It is recommended that IP ACLs be applied to the management interface to restrict access on the management interface to management hosts or subnets that contain management hosts.

5.7 Cisco Fabric Manager Security

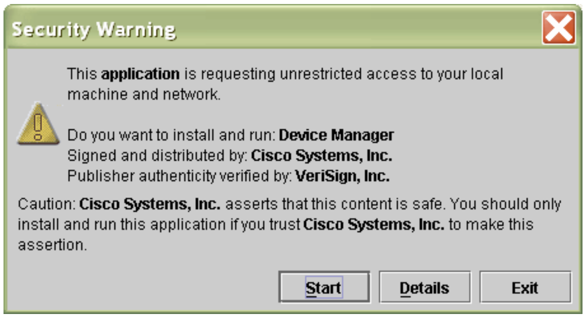
All products within the Cisco MDS 9000 family of multilayer switches come standard with complete command-line (CLI) and graphical (GUI) management tools. The GUI management tools are collectively referred to as 'Cisco Fabric Manager'.

Cisco Fabric Manager is a set of JAVA applications that can run on any host or management station supported by Java J2SE / J2RE (http://java.sun.com/downloads/). Cisco Fabric Manager provides a graphical user interface (GUI) that displays real-time views of the network fabric/topology and allows full configuration of Cisco MDS 9000 Family devices and third-party switches. The Cisco Fabric Manager tools consist of:

- Fabric Manager
  Fabric Manager displays a map of your network fabric, including Cisco MDS 9000 Family switches, third-party switches, hosts, and storage devices.

- Device Manager
  Device Manager presents two views of a switch. The 'Device View' displays a graphic representation of the switch configuration and provides access to statistics and configuration information for a single switch. The 'Summary View' displays a summary of xEPorts (Inter-Switch Links), Fx Ports (fabric ports), and Nx Ports (attached hosts and storage) on the switch, as well as FC and IP neighbor devices.

Cisco Fabric/Device Manager is downloaded to management hosts by pointing a web-browser at the management interface of a Cisco MDS switch. While the web-browser connection is plain HTTP, Cisco Fabric/Device Manager itself is a 'digitally signed' application who authenticity and integrity is digitally verified by the web-browser. As shown in Figure 16, a security dialog box will be displayed prior to any code being downloaded and activated. Clicking on the 'details' option on the dialog box will show details of who has provided the certificate used to digitally sign the Fabric/Device Manager application.



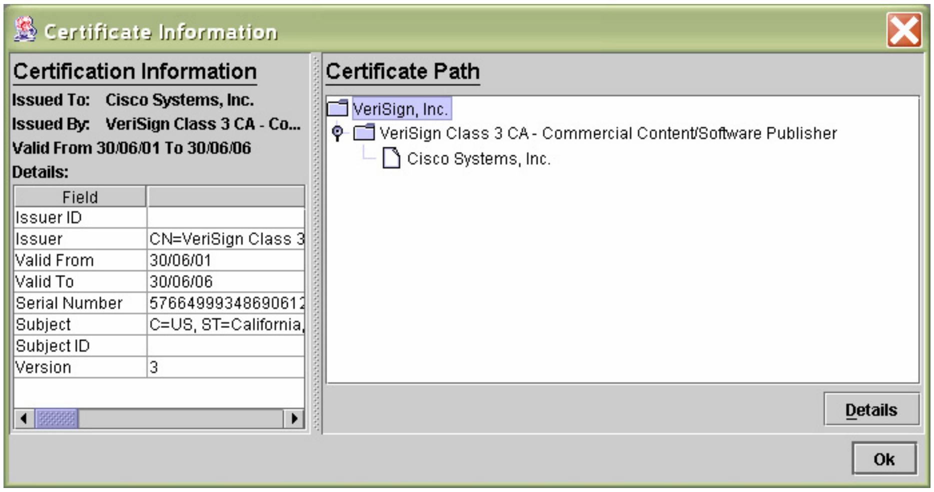
This **application** is requesting unrestricted access to your local machine and network.

Do you want to install and run: **Device Manager**
Signed and distributed by: **Cisco Systems, Inc.**
Publisher authenticity verified by: **VeriSign, Inc.**

Caution: **Cisco Systems, Inc.** asserts that this content is safe. You should only install and run this application if you trust **Cisco Systems, Inc.** to make this assertion.

Figure 16 Security Dialog box displayed with loading Cisco Fabric Manager

Figure 17 show an example of a digital certificate used by Cisco Systems to sign the application code.



Certification Information

Issued To: Cisco Systems, Inc.
Issued By: VeriSign Class 3 CA - Co...
Valid From 30/06/01 **To** 30/06/06
Details:

Field	
Issuer ID	
Issuer	CN=VeriSign Class 3
Valid From	30/06/01
Valid To	30/06/06
Serial Number	5766499934869061
Subject	C=US, ST=California,
Subject ID	
Version	3

Certificate Path

- VeriSign, Inc.
  - VeriSign Class 3 CA - Commercial Content/Software Publisher
    - Cisco Systems, Inc.

Figure 17 Typical security certificate issued to Cisco Systems

Cisco Fabric/Device Manager uses SNMP to communicate with the switches it is managing. By default, this communication uses SNMPv3 (see section 5.4.2), so provides protection with origin authentication, immunity to eavesdropping/playback attacks and ties into Cisco's AAA framework allowing for local or centralized authorization, authentication and accounting (see section 5.1). Cisco Fabric/Device Manager can be configured to use SNMPv1/v2 if desired, but obviously this is not recommended due to the inherent security limitations present in SNMPv1/v2..

Note that all the same policies as configured for Roles-based access (see section 5.5) apply to Cisco Fabric Manager also, as do audit/accounting logs of what users executed what administration tasks (see section 5.8).
It is recommended that IP ACLs (section 5.6) be used to limit what devices can communicate through the management interface.

> Managing Cisco MDS 9000 switches using Cisco Fabric/Device Manager with SNMPv3 isn't any *less* secure than using a secure command-line method such as SSH. As such, it is safe and secure to use Cisco Fabric Manager as a graphical management tool for managing Cisco MDS 9000 family of multilayer switches.

5.8 Accounting Log / Audit Trail

All products within the Cisco MDS 9000 product family internally generate an audit-trail accounting log for any configuration operations. The accounting log can both be stored inside the switch itself (local accounting log) in non-volatile storage (NVRAM) (stored across reboots) and automatically transferred to an external centralized accounting server through the use of Cisco AAA services. The audit trail may e used to generate reports for troubleshooting purposes and for user accountability.

As well as storing an audit trail of configuration operations, important system events such as save-configuration (‘*copy running-config startup-config*’) and system-switchover (among others) are also recorded.

By default, the local accounting log is enabled with a 15,000 byte buffer; that is, the last 15,000 bytes of the log are kept recorded. This can be increased to log up to a maximum of 35,000 bytes.
The following configuration excerpt shows how to configure the accounting log:

switch(config)# **aaa accounting logsize 35000**	Modify the local accounting log file size to be 35000 bytes
switch(config)# **no aaa accounting logsize 35000**	Reset the local accounting log file size to the default (15000 bytes)
switch(config)# **aaa accounting default local**	Configure system to use local accounting only (default)
switch(config)# **aaa accounting default none**	Disable local accounting
switch(config-if)# **aaa accounting default group aaa_group**	Enable external accounting to AAA servers in AAA group “aaa_group”. (an AAA group may include multiple external RADIUS and TACACS+ servers)
switch(config-if)# **aaa accounting default group aaa_group local**	Enable both local accounting and external accounting to AAA servers in AAA group “aaa_group”.

For RADIUS-based accounting, ‘Interim-Update’ RADIUS accounting-request packets are used to communicate accounting log information to the RADIUS server. The RADIUS server must be appropriately configured to log the information communicated in these packets. Several servers typically have a ‘Log Update’/’Watchdog Packets’ flag in the AAA client configuration that needs to be enabled.

The following is an example of the type of information stored in the accounting log:

```
Wed Oct 22 16:13:01 2003:start:snmp_1066803181_64.104.224.230:admin:
Wed Oct 22 16:13:01 2003:update:snmp_1066803181_64.104.224.230:admin:Zone Zone1 is created on VSAN 400
Wed Oct 22 16:13:38 2003:update:snmp_1066803218_64.104.224.230:admin:Added member [ WWN: 21:00:00:e0:8b:06:ab:1e
                              ID: 1] to zone Zone1 on VSAN 400
Wed Oct 22 16:13:45 2003:update:snmp_1066803225_64.104.224.230:admin:Zoneset ZoneSet1 is created on VSAN 400
Wed Oct 22 16:13:51 2003:update:snmp_1066803231_64.104.224.230:admin:Added zone Zone1 to zoneset ZoneSet1 on VSAN 400
Wed Oct 22 16:13:51 2003:stop:snmp_1066803231_64.104.224.230:admin:
Wed Oct 22 16:14:00 2003:start:snmp_1066803240_64.104.224.230:admin:
Wed Oct 22 16:14:00 2003:update:snmp_1066803240_64.104.224.230:admin:Activation of zoneset ZoneSet1 attempted
                              on VSAN 400
Wed Oct 22 16:14:22 2003:update:snmp_1066803262_64.104.224.230:admin:Deactivation of zoneset LTD_Lab_vsan300 attempted
                              on VSAN 300
Wed Oct 22 16:16:17 2003:update:snmp_1066803377_64.104.224.230:admin:Interface fc1/5 state updated to down
Wed Oct 22 16:16:20 2003:update:snmp_1066803380_64.104.224.230:admin:Interface fc1/6 state updated to up
Wed Oct 22 16:16:20 2003:stop:snmp_1066803380_64.104.224.230:admin:
..
Wed Oct 22 16:38:25 2003:start:/dev/pts/0_1066804316:admin:
Wed Oct 22 16:33:28 2003:update:/dev/pts/0_1066804316:admin:updated TACACS+ parameters for server:10.64.37.2
Wed Oct 22 16:34:43 2003:update:/dev/pts/0_1066804316:admin:updated TACACS+ parameters for group:tacacs-group
Wed Oct 22 16:37:09 2003:update:/dev/pts/0_1066804316:admin:modified the configuration for authentication login default
Wed Oct 22 17:04:58 2003:stop:/dev/pts/0_1066804316:admin:shell terminated
```

Note that the date/time of the log entry, the user who performed the command ('admin' in the examples above) as well as the IP address that the action was performed from and the management interface used (SNMP or CLI). 'Start' and 'Stop' records are used to indicate when the user started and ended a management session.

It is recommended that the local accounting log never be disabled and to make use of external accounting logging whenever there are centralized AAA services available.

5.9 Time Synchronization

Network Time Protocol (NTP) is widely used in the Internet to synchronize computer clocks to national standard time. The NTP architecture, protocol and algorithms have evolved for over two decades. NTP as it stands today is generally referred to as NTP version 3 and is an internet standard documented in Internet RFC 1305.

The architecture and security models of NTP provide for operation in point-to-point (unicast) and point-to-multipoint (multicast) modes, and include provisions for secure authentication using both symmetric key and public key cryptography. NTP is transported over UDP within an IP network with all communications based on time specified in UTC. Time synchronization happens when several frames are exchanged between NTP clients and NTP servers.

Previous funded research has resulted in a continuous series of improvements in accuracy and reliability of the protocol and supporting algorithms. Used in the Internet of today with computers ranging from personal workstations to supercomputers, NTP provides accuracies generally in the range of a millisecond or two in LANs and up to a few tens of milliseconds in global WANs. When combined with OSes with in- kernel support for precision timing signals such as pulse-per-second (PPS) signaling, the accuracy can be improved ultimately to the order of one nanosecond in time and one nanosecond per second in frequency.

The basis for how NTP works is that there are NTP server(s) that receive its time from a reference time source, such as a radio clock or atomic clock, attached to the time. These NTP servers are referred to as *stratum-1* servers, since they have an accurate time source. Additional downstream NTP servers (*stratum-2* servers) are organized into a hierarchy that contact NTP servers above them (the *stratum-1* servers) and also other *peer* servers (other *stratum-2* servers). Further downstream NTP clients then communicate with these *stratum-2* servers and so on. This is illustrated in Figure 18 below.

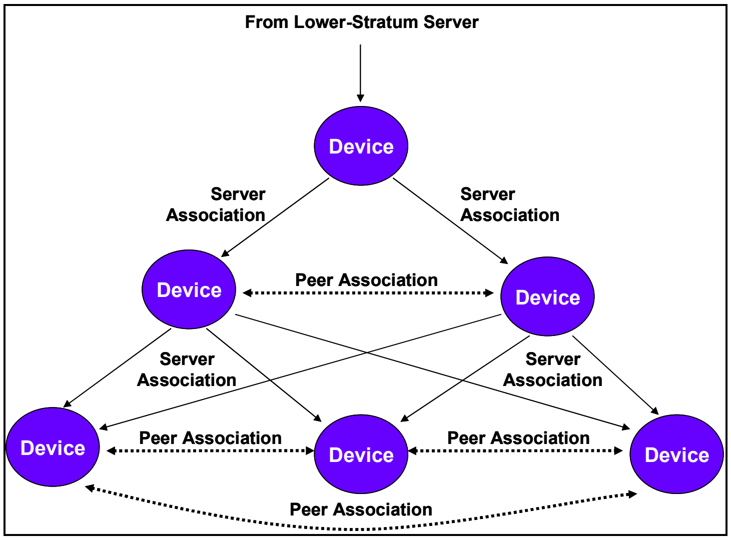


Figure 18 Recommended NTP hierarchical configuration to ensure accurate time syncronization

In a large enterprise network, having one time standard for all network devices is critical for management reporting and event logging functions when trying to correlate interacting events logged across multiple devices. Many enterprise customers with extremely mission-critical networks maintain their own *stratum-1* NTP source. For others without such extreme needs, they may reference time source(s) available from their corporate internet connection, typically provided by their Internet provider.

NTP clients need to be configured with the IP address(es) of one or more NTP servers. The servers act as the time source and receive client synchronization requests.

NTP clients may also be configured with the IP address(es) of one or more peers. A peer is capable of providing time on its own and is capable of having a server configured. Using NTP with multiple time sources – such as one instance receiving time from a server and another instance configured to receive time from a peer – can provide a higher degree of accuracy and reliability. Even if the active server link is lost, the correct time is still maintained due to the presence of the peer.

If NTP is configured to only communicate with peer(s), the most accurate peer takes on the role of the NTP server and the other peer(s) act as a peer(s). Both peers will end up with a correctly synchronized time providing they have the right time source or if they point to the right NTP source.

The Cisco MDS 9000 Family of multilayer intelligent switches supports Time Synchronization with support for the NTPv3 protocol and the ability to behave as both a NTP Client and an NTP Server. Additionally, the MDS 9000 product family hardware contains all the necessary precision timing signals to support time-synchronization, with accuracy to greater than a microsecond.

The following configuration excerpt shows how to configure NTP:

switch(config)# **ntp server 10.10.10.10**	Forms a server association with a NTP server at 10.10.10.10

switch(config)# **ntp peer 10.20.10.5**	Forms a peer association with a peer at 10.20.10.5. There can be multiple peer associations.

The following guidelines apply to using NTP on the Cisco MDS 9000 family for time synchronization:

- You should have a peer association with another switch only when you are sure that your clock is reliable (which means that you are a client of a reliable NTP server).
- Though a peer configured alone will be the most accurate peer taking on the role of a server, the configured peer should be used more as a back-up support. If more than one server is present, you should split the switches such that half point at one NTP server and have at the other NTP server, and configure both groups with peer association between these two sets. This will provide the most accurate time synchronization.
- If you only have one server, it's better for all the switches have a client association with that server.
- If the network is configured robustly, even a server down time will not affect well-configured switches in the network.

It is recommended that NTP be used at all times to provide a consistent accurate time source for correlating events from multiple devices.

5.10 SMI-S XML-CIM Management Interface

Web-Based Enterprise Management (WBEM) is a set of management and Internet standard technologies developed by the Distributed Management Task Force (DTMF) to unify the management of enterprise computing environments. The core set of standards that make up WBEM are the Common Information Model (CIM) (the data model), xmlCIM (a xml CIM encoding specification) and CIM-XML (the CIM over HTTP specification).

Storage Management Initiative Specification (SMI-S) is the common interfaces based on CIM to allow multi-vendor interoperability in a SAN environment. SMI-S enables SAN management clients to link with CIM servers to manage large number of storage resources with a unified interface.

The Cisco MDS 9000 Family will support CIM, SMI-S and WBEM standards in SAN-OS 1.3. Support includes:

- A subset of the classes described by the SNIA standards
- All interconnect device profiles in the SMI-S:
  - Switch profile
  - Fabric profile
  - Zoning Profile
  - Topological Profile

- Support for use of secure socket layer (SSL/TLS)

All products within the Cisco MDS 9000 Family will be fully compliant with SNIA SMI-S version 1.0. The embedded agent inside the MDS includes a CIM Object Manager (CIMOM) and CIM Provider to instrument the MDS without use of external proxy functionality.

Access to the SMI-S XML-CIM Management Interface is via SSL (HTTPS).

User authentication, authorization and accounting is tied into the standard AAA management framework within SAN-OS (see section 5.1).

SMI-S data is transported using SSLv2/TLS 1.0.
SSLv2/TLS 1.0 provides connection security that has three basic properties:

- The connection is private. Encryption is used after an initial handshake to define a secret key. Symmetric cryptography is used for data encryption (e.g. DES, RC4, AES etc)
- The peer's identity can be authenticated using asymmetric (public key) cryptography (e.g. RSA, DSS etc.).
- The connection is reliable. Message transport includes a message integrity check using a keyed MAC. Secure hash functions (e.g. SHA, MD5, etc.) are used for MAC computations.

In addition to these security mechanisms, IP ACLs can also be used to limit the scope for communication. Section 5.6 contains details on how to deploy IP ACLs on the management interfaces.

6 Operational security

Traditionally, there has been minimal attention paid to Operational security. In many cases, all that operational security meant was that "someone changed the passwords every now and again".

Cisco has a comprehensive set of guidelines to operational security in the form of Cisco's secure blueprint for enterprise networks (SAFE) (http://www.cisco.com/go/safe). The principle goal of SAFE is to provide best practice information to interested parties on designing and implementing secure networks. SAFE serves as a guide to network designers considering the security requirements of their network. SAFE takes a defense-in-depth approach to network security design. This type of design focuses on the expected threats and their methods of mitigation, rather than on "Put the firewall here, put the intrusion detection system there." This strategy results in a layered approach to security where the failure of one security system is not likely to lead to the compromise of network resources. SAFE is based on Cisco products and those of its partners.

From the perspective of operational security, an operationally-secure network is one which provides:

- the network keeps passing legitimate customer traffic (availability)
- traffic goes where its supposed to go (availability, confidentiality)
- the network elements remain manageable (availability)
- only authorized users can manage network elements (authorization)
- there is record of all security related events (accountability)
- a network operator has the necessary tools to detect and respond to illegitimate traffic

All of the operational security features are standard across the entire Cisco MDS 9000 family in the base system configuration. No additional software licenses are required to enable any of this functionality.

6.1 Denial of Service (DoS) Attack Protection

A "denial-of-service" attack is characterized by an explicit attempt by attackers to prevent legitimate users of a service from using that service. Examples of this include:

- attempts to "flood" a network, thereby preventing legitimate network traffic
- attempts to disrupt connections between two machines, thereby preventing access to a service
- attempts to prevent a particular individual from accessing a service
- attempts to disrupt service to a specific system or person

The Computer Emergency Response Team (CERT) have published a paper that provides additional details on DoS attacks. This paper is at <http://www.cert.org/tech_tips/denial_of_service.html>

Some of the most common forms of Denial of Service attacks go under the following names:

- TCP SYN Flood
- Ping of Death
- Tribe Flood Network (TFN) and Tribe Flood Network 2000 (TFN2K)
- Trinoo
- Stacheldraht
- Trinity

- Shaft

Up until recently, the most trivial denial-of-service attacks such as a flood-ping (icmp-flood), broadcast-storm or ARP-cache-flood could overwhelm the control-plane of most deployed FC switches.

All products within the Cisco MDS 9000 family contain protection against Denial of Service attacks in two ways:

1. The control-plane and network stack of the Cisco MDS 9000 has been hardened to by immune from all known denial-of-service attack methods used today.
2. The software/firmware which makes up the control-plane capabilities of the Cisco MDS 9000 family is modular. That is, there isn't a single "monolithic" piece of code running, but rather there is functional separation of code between different control-plane functions. The code within each module not only runs independently (with enforced limits on the internal CPU and memory resources that each module may use), but even in the case of a software bug or software crash, that crash will be isolated to that given module which will be restarted and pick up where it left off previously, restoring its *persistent state* to a previous checkpoint.

6.2 Control-Plane Scalability

As SANs continue to scale and more features/functionality are added to FC switches, control-plane CPU and memory resources on FC switches themselves are becoming the biggest limiting factor on the number of devices that can exist inside a FC fabric.

As the number of devices attached to a fabric increases, the amount of CPU time spent within the core fabric services (such as issuing RSCNs, FSPF updates and calculations, FLOGIs/PLOGIs) increases. Insufficient CPU resources within a FC switch will have a negative impact on the re-convergence time of a fabric (how long it takes for the network to reach 'steady state' of all devices being able to communicate with one another after some event such as a port failure) and can in itself result in compromised fabric stability.

Data-plane security is typically applied against switched frame, so it is mandatory to ensure that it is performed in hardware, regardless of the security policies being applied. Examples of data-plane security include zoning (hard zoning) (see section 3.1) and VSANs (see section 3.2)..
Control-plane security is less intensive, typically applied on a per-session or per-host/per-port basis, and can therefore run in software. Examples of control-plane security include Port and Fabric binding (sections 3.3 and 3.5.3), switch-to-host and switch-to-switch security (section 3.4). Note that many of the more advanced security techniques and algorithms such as FC-SP, SSH, SSL and SNMPv3 which make use of advanced cryptology algorithms do typically require lots of CPU and memory resources to ensure strong/safe cryptology.

Cisco has addressed this need with powerful processing capabilities in the MDS 9000 Family. All members of the Cisco MDS 9000 product family utilize an Intel Pentium-3 processor operating at 1.3 GHz as the 'control plane' processor. In relative CPU terms, this provides approximately 3477 Dhrystone 2.1 MIPS of computing power. There is 1 GB RAM on the supervisor for control-plane functionality.

Contrasting this to other FC switch vendors, one Core switch vendor uses an IBM PowerPC 405GP operating at 200MHz as their 'control-plane' processor. This provides approximately 282 Dhrystone 2.1 MIPS of computing power. Another Director switch vendor uses an Intel i960-HD80 operating at 80MHz as their 'control-plane' processor. This provides approximately 160 Dhrystone 2.1 MIPS of computing power. Both these vendors have 128 MB RAM available on the supervisor for control-plane functionality.

The control-plane performance available across the Cisco MDS 9000 product family range is sufficient to handle the advanced security techniques and cryptographic algorithms in use today, and any that may be available in future.

6.3 System Logs

All products within the Cisco MDS 9000 product family provide System Logging. System Logging can be used to provide logging for monitoring and troubleshooting purposes. Log messages may be sent to none/some/all of the console, remote console sessions (VTY sessions), internal system logfile or an external syslog server.

The types of system messages that can be saved can be set on a per-facility basis and the severity. All log messages are time-stamped to enhance real-time debugging and management.

As of SAN-OS 1.3, the following log facilities are available:

Facility Keyword	Description	Facility Keyword	Description
acl	ACL manager	ntp	NTP
all	All facilities	platform	Platform manager
auth	Authorization system	port	Port
authpriv	Authorization (private) system	port-channel	PortChannel
bootvar	Bootvar	qos	QoS
callhome	Call Home	rdl	RDL
cron	Cron or at facility	rib	RIB
daemon	System daemons	rscn	RSCN
fcc	FCC	securityd	Security
fcdomain	fcdomain	syslog	Internal syslog messages
fcns	Name server	sysmgr	System manager
fcs	FCS	tlport	TL port
flogi	FLOGI	user	User process
fspf	FSPF	uucp	Unix-to-Unix copy system
ftp	File Transfer Protocol	vhbad	Virtual host base adapter daemon
ipconf	IP configuration	vni	Virtual network interface
ipfc	IPFC	vrrp_cfg	VRRP configuration
kernel	Kernel	vrrp_eng	VRRP engine
local0-local7	Locally defined messages	vsan	VSAN syslog
lpr	Line printer system	vshd	vshd
mail	Mail system	wwn	WWN manager
mcast	Multicast	xbar	Xbar syslog
module	Switching module	zone	Zone server
news	USENET news		

As of SAN-OS 1.3, the following severity levels are available:

Level Keyword	Level	Description	Syslog Definition
emergencies	0	System unusable	LOG_EMERG
alerts	1	Immediate action needed	LOG_ALERT
critical	2	Critical conditions	LOG_CRIT
errors	3	Error conditions	LOG_ERR
warnings	4	Warning conditions	LOG_WARNING
notifications	5	Normal but significant condition	LOG_NOTICE
Informational	6	Informational messages only	LOG_INFO
debugging	7	Debugging messages	LOG_DEBUG

Log message always begin with a ‘%’ sign and are displayed in the following format:

```
month dd hh:mm:ss switchname facility-severity-MNEMONIC description
```

e.g.:

```
Nov 8 14:07:58 excal-113 %LOG_MODULE-5-MOD_OK: Module 1 is online
Nov 8 14:07:58 excal-113 %LOG_PORT-3-IF_UNSUPPORTED_TRANSCEIVER: Transceiver for interface fc1/13 is not supported
Nov 8 14:07:59 excal-113 %LOG_PLATFORM-5-PS_OK: Power supply 1 ok
Nov 8 15:21:44 excal-113 %LOG_VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from pts/0 (171.71.58.72)
```

By default, normal but significant system messages are sent to both the internal system log file, system console and any remote console sessions (VTY sessions) that have enabled terminal monitoring (‘*term mon*’). The internal system log file may be accessed using the CLI or viewed on an external syslog server. The internal system log file can be configured to save the last 4 MB of log messages.

Log messages are not saved across system reboots, however the system does keep up to 100 log messages with a severity level of critical and below (levels 0, 1, and 2) stored in NVRAM.

The following configuration excerpt shows system logging being used:

switch(config)# **logging console 3**	Configures console logging at level 3 (error). Logging messages with a severity level of 3 or above will be displayed on the console.
switch(config)# **logging console**	Reverts console logging to the factory set default severity level of 2 (critical). Logging messages with a severity level of 2 or above will be displayed on the console.
switch(config)# **logging level fspf 4**	Configures logging for the FSPF facility at level 4 (warning). As a result, logging messages with a severity level of 4 or above will be displayed on remote console sessions.
switch(config)# **logging logfile ManagerLog 3**	Configures logging information for errors or events above severity level 3 to be logged in a file named ManagerLog.

size 3000000	By configuring a size, you are restricting the file size to 3000000 bytes. The maximum upper limit is 4194304 (default).
switch(config)# **logging server 10.67.16.5**	Configures the switch to forward log messages according to the specified facility types and severity levels to remote multiple servers specified by its hostname or IP address (10.67.16.5). Up to 3 syslog servers may be specified.
switch(config)# **logging server 10.67.16.5 facility local0**	Configures the switch to forward log messages according to the specified facility (local1) for the server IP address (10.67.16.5). The default outgoing facility is local7.
switch(config)# **no logging server 10.67.16.5**	Removes the specified server (10.67.16.5) and reverts to factory default.

Internal Log files are stored in a special log file system inside the switch. The '*show logging*' CLI command can be used to view the internal log.

Log files can be saved to external hosts by making use of the '*copy*' CLI command with the source-filename in the '*log:*' file system. Section 5.3.3 provides details on how to securely transfer log files from the switch.

The last 100 Critical messages stored in NVRAM may be viewed using the '*show logging nvram'* CLI command.

It is highly recommended that at least one (preferably more) external syslog server be used to log critical system messages.

6.4 Call Home

Call Home provides e-mail-based notification of critical system events. Common uses of this feature may include direct paging of a network support engineer, e-mail notification to a Network Operations Center, and utilization of Cisco AutoNotify services for direct case generation with the Cisco Technical Assistance Center (TAC).

The Call Home function can even leverage support from Cisco Systems or another support partner. Flexible message delivery and format options make it easy to integrate specific support requirements. Call Home includes the following functionality

- Fixed set of predefined alerts and trigger events on the switch.
- Automatic execution and attachment of relevant command output.
- Multiple message format options:

  - Short Text—Suitable for pagers or printed reports.
  - Plain Text—Full formatted message information suitable for human reading.
  - XML—Matching readable format using Extensible Markup Language (XML) and Document Type Definitions (DTDs) named Messaging Markup Language (MML). The MML DTD is published on the Cisco Connection Online (CCO) website at http://www.cisco.com/. The XML format enables communication with the Cisco Systems TAC group.
- Multiple concurrent message destinations. Up to 50 E-mail destination addresses are allowed for each format type.
- Message categories include system, environment, switching module hardware, supervisor module, hardware, inventory, and test.

The actual configuration of Call Home depends on how you intend to use the feature. Some points to consider include:

- E-mail server and at least one destination profile must be configured. The destination profile(s) used depends on whether the receiving entity is a pager, email, or automated service such as Cisco AutoNotify.
- The contact name (SNMP server contact), phone, and street address information must be configured before Call Home is enabled. This is required to determine the origin of messages received.
- The Cisco MDS 9000 switch must have IP connectivity to an E-mail server for the feature to operate
- If Cisco AutoNotify is used, an active service contract must cover the device being configured.

The Call Home feature can be used to provide timely alerts to critical system problems. It is highly recommended that Call Home be used in secure SAN environments.

6.5 Switch Port Analyzer (SPAN)

Switched Port Analyzer (SPAN) is a feature that is specific to switches in the Cisco MDS 9000 Family. SPAN allows for any interfaces or VSANs being monitored to have a copy of each frame taken and sent out a SPAN destination port (SD port). Any Fibre Channel port in a switch can be configured as an SD port. A Fibre Channel Analyzer (such as a Finisar) or a Cisco Port Adapter Analyzer (DS-PAA) may be connected to the SD port and receive a copy of all monitored traffic.

Since the SPAN feature effectively allows one to take a copy of any FC frames being switched inside the switch, this may be considered a security risk given sensitive data that previously couldn't be snooped can now be captured. Because of this, it is recommended that Roles Based Access (see section 5.5) be used to protect against unauthorized users being able to enable SPAN.

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

European Headquarters
Cisco Systems Europe
11 Rue Camille Desmoulins
92782 Issy-les-Moulineaux
Cedex 9
France
www-europe.cisco.com
Tel: 33 1 58 04 60 00
Fax: 33 1 58 04 61 00

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-7660
Fax: 408 527-0883

Asia Pacific Headquarters
Cisco Systems, Inc.
Capital Tower
168 Robinson Road
#22-01 to #29-01
Singapore 068912
www.cisco.com
Tel: +65 317 7777
Fax: +65 317 7799

Cisco Systems has more than 200 offices in the following countries and regions. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices

**Argentina • Australia • Austria • Belgium • Brazil • Bulgaria • Canada • Chile • China PRC • Colombia • Costa Rica • Croatia
Czech Republic • Denmark • Dubai, UAE • Finland • France • Germany • Greece • Hong Kong SAR • Hungary • India • Indonesia • Ireland
Israel • Italy • Japan • Korea • Luxembourg • Malaysia • Mexico • The Netherlands • New Zealand • Norway • Peru • Philippines • Poland
Portugal • Puerto Rico • Romania • Russia • Saudi Arabia • Scotland • Singapore • Slovakia • Slovenia • South Africa • Spain • Sweden
Switzerland • Taiwan • Thailand • Turkey • Ukraine • United Kingdom • United States • Venezuela • Vietnam • Zimbabwe**