

Improve Security on the CSS 11000 and CSS 11500

Document ID: 25945

Contents

Introduction

Prerequisites

- Requirements
- Components Used
- Conventions

Password Management

- Local User Profiles

Control of Interactive Access

- Console Ports
- General Interactive Access
- Control of Console Access
- Control of VTYS
- SSH Support
- RADIUS
- TACACS+
- Warning Banners

Commonly Configured Management Services

- SNMP
- HTTP
- HTTPS

Management and Interactive Access Over the Internet (and Other Untrusted Networks)

- Packet Sniffers
- Other Internet Access Dangers

Logging

- Save Log Information
- Record Access List Violations

Secure the IP Routing

- Antispoofing
- Antispoofing with ACLs
- Control of Directed Broadcasts
- Path Integrity
- IP Source Routing
- ICMP Redirects
- Routing Protocol Filtering and Authentication

Flood Management

- Transit Floods

Possibly Unnecessary Services

- SNTP
- Cisco Discovery Protocol

Stay Up To Date

Related Information

Introduction

This document provides information about Cisco configuration settings that can improve security on the Cisco Content Services Switch (CSS) 11000 or CSS 11500. This document describes basic configuration settings that are almost universally applicable in IP networks and covers a few unexpected items of which you must be

aware.

This document does not present an exhaustive list of these items, nor can the information in the document be substituted for knowledge on the part of the network administrator. The document serves as a reminder of items that are sometimes forgotten.

This document mentions only the commands that are important in IP networks. Many of the services that you can enable on the CSS require careful security configuration. However, this document focuses on information for services that are enabled by default or that are almost always enabled by users and that can require disablement or reconfiguration.

Some of the default settings in Cisco WebNS software exist for historical reasons. These settings were applicable when they were chosen, but would probably be different if new defaults were chosen today. Other defaults are applicable for most systems, but can create security exposures if these defaults are used in devices that form part of a network perimeter defense. Still other defaults are actually required by standards, but are not always desirable from a security point of view.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

This document is not restricted to specific software and hardware versions.

Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

Password Management

Passwords and similar proprietary information, such as Simple Network Management Protocol (SNMP) community strings, are the primary defense against unauthorized access to your CSS. The best way to handle most passwords is to maintain them on a TACACS+ or RADIUS authentication server. However, almost every CSS still has a locally configured password for privileged access. The CSS can also include other password information in the configuration file. Any password that is configured in clear text appears in the configuration encrypted with Data Encryption Standard (DES).

Local User Profiles

This list describes the local user profiles:

- *Administrator* The Administrator profile includes these privileges:
 - ◆ Access to the Offline Diagnostics Monitor menu
 - ◆ Full access to the command line
 - ◆ Full directory access

These settings can be configured from either the command line or the Offline Diagnostics Monitor menu.

- *Technician* The Technician profile includes these privileges:

- ◆ Full access to the command line

- ◆ Full directory access

These settings can be configured with use of the command line. Do not use the Technician profile for CSS administrative purposes.

- *Superuser* The Superuser profile includes these privileges:

- ◆ Full access to the command line

- ◆ The ability to save directory access restrictions

These settings can be configured with use of the command line.

- *User* The User profile cannot make configuration changes and includes directory access restrictions.

These settings can be configured with use of the command line.

When you issue the **restrict user–database** command, you enforce directory access restrictions on each user. Only the Administrator and Technician user levels can perform these actions:

- Remove the **restrict user–database** command.

- Change the **local user–database** command.

- Issue the **clear running–config** command.

Control of Interactive Access

Any user who can log in to a CSS can display information that the general public does not necessarily need to view. In some cases, a user who can log in to the CSS can use the CSS as a relay for further network attacks. A user who gains privileged access to the CSS can reconfigure the CSS. In order to prevent inappropriate access, you need to control interactive logins to the CSS.

Although most interactive access is disabled by default, there are exceptions. The most obvious exceptions are interactive sessions from directly connected asynchronous terminals, such as the console terminal, and access to the Ethernet management port.

Refer to Configuring CSS Remote Access Methods for more information on how to control interactive access to the CSS.

Console Ports

An important item to remember is that the console port of a Cisco device has special privileges. In particular, suppose that someone sends an ESC (escape) character to the console port when the POST diagnostics run. After a reboot, this person can easily use the password recovery procedure in order to take control of the system. Attackers who can interrupt power or induce a system crash, and who have access to the console port through a hardwired terminal, a modem, a terminal server, or some other network device, can take control of the system. These attackers can take control even if they do not have physical access to the system or the ability to log in to the system normally.

Therefore, any modem or network device that gives access to the Cisco console port must be secured to a standard that is comparable to the security that is used for privileged access to the CSS. At a minimum, any console modem must be of a type that can require the dialup user to supply a password for access, and the modem password must be carefully managed.

General Interactive Access

There are more ways to get interactive connections to a CSS than users may realize. You can use these methods in order to manage the CSS:

- Telnet
- Secure Shell Host (SSH)
- SNMP
- Console
- FTP
- XML
- Web management

Issue the **restrict** command in order to enable or disable. The CSS still listens on the particular port, but closes the connection. So that the packets do not hit these ports, configure access control list (ACL) clauses to deny the packets.

It is difficult to be certain that all possible modes of access have been blocked. In most cases, administrators must use some sort of authentication mechanism in order to make sure that logins on all lines are controlled. Administrators must ensure that the logins are controlled even on machines that are supposed to be inaccessible from untrusted networks.

Control of Console Access

By default, the console authenticates against locally configured user profiles. In order to activate TACACS+ or RADIUS authentication, issue the **console authentication** global command and the associated options.

Control of VTYS

By default, the vtys authenticate against locally configured user profiles. In order to activate TACACS+ or RADIUS authentication, issue the **virtual authentication** global command and the associated options.

SSH Support

If your software supports an encrypted access protocol such as SSH, Cisco recommends that you enable only that protocol and disable Telnet access when you want to use the SSH server. In order to enable the SSH Daemon (SSHD), you need an SSHD server license, which enables SSHD functionality on both the Standard and Enhanced versions of CSS software. Issue the **sshd** commands. Refer to Configuring CSS Network Protocols for more information.

Note: SSH version 1 support started in 4.01. SSH version 2 support started in 5.20.

RADIUS

As of version 5.00 and later, you can configure the CSS to use RADIUS for user authentication. In order to configure the CSS for RADIUS authentication, refer to Configuring User Profiles and CSS Parameters.

Note: A user/group profile only requires Internet Engineering Task Force (IETF) RADIUS Attributes, [006] Service-Type = Administrative.

This list identifies the debug message codes:

PW_ACCESS_REQUEST	1
PW_ACCESS_ACCEPT	2
PW_ACCESS_REJECT	3
PW_ACCOUNTING_REQUEST	4
PW_ACCOUNTING_RESPONSE	5
PW_ACCOUNTING_STATUS	6
PW_ACCESS_CHALLENGE	11

In order to view the debugs that are associated with RADIUS logins, issue these commands:

```
logging subsystem radius level debug-7
logging subsystem security level debug-7
logging subsystem netman level debug-7
```

This is an example of a successful authentication debug:

```
JUL 23 02:30:41 7/1 165 SECURITY-7: SECMGR:SecurityAuth:Request from 0x30204b10
JUL 23 02:30:41 7/1 166 SECURITY-7: SECMGR:SecurityMgrProc:Try Primary
JUL 23 02:30:41 7/1 167 RADIUS-7: Auth Primary
JUL 23 02:30:41 7/1 168 RADIUS-7: The id is 1
JUL 23 02:30:41 7/1 169 RADIUS-7: Return Auth Primary
JUL 23 02:30:41 7/1 170 RADIUS-7: RADIUS attribute 0 received with bad length -2
JUL 23 02:30:41 7/1 171 SECURITY-7: Security Manager sending success 5 reply to
  caller 30201c00
JUL 23 02:30:41 7/1 172 SECURITY-7: SECMGR:SecurityMgrProc:Try Done, Send 0x30204b10
JUL 23 02:30:41 7/1 173 NETMAN-6: CLM: Login user1@172.16.20.200
JUL 23 02:30:45 7/1 174 NETMAN-6: CLM: Logout user1@172.16.20.200
```

This is an example of an authentication that failed because of an incorrect user name or password:

```
JUL 23 02:31:36 7/1 177 SECURITY-7: SECMGR:SecurityAuth:Request from 0x30204b11
JUL 23 02:31:36 7/1 178 SECURITY-7: SECMGR:SecurityMgrProc:Try Primary
JUL 23 02:31:36 7/1 179 RADIUS-7: Auth Primary
JUL 23 02:31:36 7/1 180 RADIUS-7: The id is 2
JUL 23 02:31:36 7/1 181 RADIUS-7: Return Auth Primary
JUL 23 02:31:36 7/1 182 SECURITY-7: Security Manager sending error 7 reply to
  caller 30201c00
JUL 23 02:31:36 7/1 183 SECURITY-7: SECMGR:SecurityMgrProc:Try Secondary
JUL 23 02:31:36 7/1 184 SECURITY-7: Security Manager sending error 7 reply to
  caller 30201c00
JUL 23 02:31:36 7/1 185 SECURITY-7: SECMGR:SecurityMgrProc:Try Done, Send 0x30204b11
```

This is an example of an authentication that failed because the user-profile RADIUS attribute 006 service-type is not configured:

```
JUL 23 02:36:33 7/1 195 SECURITY-7: SECMGR:SecurityAuth:Request from 0x30204b13
JUL 23 02:36:33 7/1 196 SECURITY-7: SECMGR:SecurityMgrProc:Try Primary
JUL 23 02:36:33 7/1 197 RADIUS-7: Auth Primary
JUL 23 02:36:33 7/1 198 RADIUS-7: The id is 4
JUL 23 02:36:33 7/1 199 RADIUS-7: Return Auth Primary
JUL 23 02:36:33 7/1 200 RADIUS-7: RADIUS attribute 0 received with bad length -2
JUL 23 02:36:33 7/1 201 SECURITY-7: Security Manager sending success 5 reply to
  caller 30201c00
JUL 23 02:36:33 7/1 202 SECURITY-7: SECMGR:SecurityMgrProc:Try Done, Send 0x30204b13
JUL 23 02:36:33 7/1 203 NETMAN-6: CLM: Login user1@172.16.20.200
```

TACACS+

In version 5.03 and later, you can configure the CSS to use TACACS+ for user authentication. In order to configure the CSS for TACACS+ authentication, refer to the Release Notes for the CSS 11000 Series.

In order to view the debugs that are associated with TACACS+ logins, issue these commands:

```
logging subsystem security level debug-7
logging subsystem netman level debug-7
```

This is an example of a successful authentication debug:

```
JUL 23 01:53:32 7/1 89 SECURITY-7: SECMGR:SecurityAuth:Request from 0x30204b08
```

```
JUL 23 01:53:32 7/1 90 SECURITY-7: SECMGR:SecurityMgrProc:Try Primary
JUL 23 01:53:33 7/1 91 NETMAN-7: TACACS: tac_Authen:Final <Authen OK->
JUL 23 01:53:33 7/1 92 NETMAN-7: TACACS: tac_Authorize:Final <Author OK->
JUL 23 01:53:33 7/1 93 NETMAN-7: TACACS: tacacs_AuthorizeCommands <user1:vty1> Rsp:
  <Author OK> from10.66.79.241:49
JUL 23 01:53:33 7/1 94 NETMAN-7: TACACS: TACACS_AuthAgent:Rqst <user1:vty1:-2132790672>
  Rsp <Author OK:> <PRIV_ADMIN>
JUL 23 01:53:33 7/1 95 SECURITY-7: Security Manager sending success 0 reply to
  caller 30201c00
JUL 23 01:53:33 7/1 96 SECURITY-7: SECMGR:SecurityMgrProc:Try Done, Send 0x30204b08
JUL 23 01:53:33 7/1 97 NETMAN-6: CLM: Login user1@172.16.20.200
JUL 23 01:54:11 7/1 98 NETMAN-6: CLM: Logout user1@172.16.20.200
```

This is an example of a failed authentication because of an incorrect user name or password:

```
JUL 23 01:54:41 7/1 109 SECURITY-7: SECMGR:SecurityAuth:Request from 0x30204b0a
JUL 23 01:54:41 7/1 110 SECURITY-7: SECMGR:SecurityMgrProc:Try Primary
JUL 23 01:54:41 7/1 111 NETMAN-7: TACACS: tac_Authen:Final <Authen Rejected->
JUL 23 01:54:41 7/1 112 NETMAN-7: TACACS: TACACS_AuthAgent:Rqst <user1:vty1:-2132790672>
  Rsp <Authen Rejected:> <PRIV_DENIED>
JUL 23 01:54:41 7/1 113 SECURITY-7: Security Manager sending success 0 reply to
  caller 30201c00
JUL 23 01:54:41 7/1 114 SECURITY-7: SECMGR:SecurityMgrProc:Try Secondary
JUL 23 01:54:41 7/1 115 SECURITY-7: Security Manager sending error 7 reply to
  caller 30201c00
JUL 23 01:54:41 7/1 116 SECURITY-7: SECMGR:SecurityMgrProc:Try Tertiary
JUL 23 01:54:41 7/1 117 SECURITY-7: Security Manager sending error 7 reply to
  caller 30201c00
JUL 23 01:54:41 7/1 118 SECURITY-7: SECMGR:SecurityMgrProc:Try Done, Send 0x30204b0a
```

Warning Banners

In some jurisdictions, you can greatly ease the process of civil and/or criminal prosecution of crackers who break into your systems if you provide a banner that informs unauthorized users that their use is unauthorized. Other jurisdictions forbid the monitor of the activities of even unauthorized users unless you have taken steps to notify users of your intent to do so. One way to provide this notification is to put it into a banner message. You can configure a banner message with the CSS **set banner** command. This command was introduced in 5.03.

Legal notification requirements are complex and vary in each jurisdiction and situation. Even within jurisdictions, legal opinions vary. Discuss this issue with your legal counsel. In cooperation with counsel, consider which of these notices to put into your banner:

- A notice that specifically states only authorized personnel are to log in to or use the system and perhaps information about who can authorize use.
- A notice that any unauthorized use of the system is unlawful and can be subject to civil and/or criminal penalties.
- A notice that any use of the system may be logged or monitored without further notice and that the resulting logs may be used as evidence in court.
- Specific notices that are required by local laws.

For security (rather than legal) reasons, do not include in your login banner this information about your CSS:

- Name
- Model
- Software that runs
- Owner

Commonly Configured Management Services

Many users manage their networks with the use of protocols other than interactive remote login. The most common protocols for this purpose are SNMP and HTTP. The most secure option is not to enable these protocols at all. However, if you have enabled one of the protocols, secure it as this section describes.

SNMP

SNMP is very widely used for network device monitoring and, frequently, for configuration changes. SNMP has two major standard revisions, SNMPv1 and SNMPv2. Your CSS supports SNMP version 2C (SNMPv2C), which is known as community-based SNMP. The CSS generates traps in SNMPv1 format.

In order to control SNMP access to the CSS, issue the **no restrict snmp** command and the **restrict snmp** command. Access through SNMP is enabled by default. If you disable access through SNMP, the CSS still listens on the particular port 1, but closes the connection. Configure ACL clauses to deny the packets so that packets do not hit the SNMP port.

Unfortunately, SNMPv1 and SNMPv2C use a very weak authentication scheme that is based on a community string. The authentication amounts to a fixed password that is transmitted over the network without encryption. If you must use SNMPv2C, be careful to choose obscure community strings (and do not use, for example, public or private). If at all possible, avoid use of the same community strings for all network devices. Use a different string or strings for each device, or at least for each area of the network. Do not make a read-only string the same as a read-write string. If possible, do periodic SNMPv2C polling with a read-only community string. Use read-write strings only for actual write operations.

SNMPv2C is not suitable to use across the public Internet for these reasons:

- SNMPv2C uses cleartext authentication strings.
- SNMPv2C is a datagram-based transaction protocol that is easily spoofed.
- Most SNMP implementations send those strings repeatedly as part of periodic polling.

Carefully consider the implications before you use SNMPv2C across the public Internet.

In most networks, legitimate SNMP messages only come from certain management stations. If legitimate SNMP messages only come from certain management stations in your network, consider the use of ACLs that are applied to the circuit VLANs in order to deny unwanted SNMP messages.

SNMP management stations often have large databases of authentication information, such as community strings. This information can provide access to many CSSs and other network devices. This concentration of information makes the SNMP management station a natural target for attack. Secure the SNMP management station accordingly.

HTTP

The CSS supports remote configuration via the HTTP protocol with use of Extensible Markup Language (XML) documents. In WebNS version 4.10 or earlier, you can reach access to the WebNS device management user interfaces in clear text if you browse to TCP port 8081. In general, HTTP access is equivalent to interactive access to the CSS. The authentication protocol that is used for HTTP is equivalent to the send of a cleartext password across the network. Unfortunately, there is no effective provision in HTTP for challenge-based or one-time passwords. Therefore, HTTP is a relatively risky choice for use across the public Internet.

If you choose to use HTTP for management, restrict access to the appropriate IP addresses with the use of ACLs that are applied to the circuit VLANs. In order to control HTTP XML access to the CSS, issue the **no restrict xml** command and the **restrict xml** command. In later versions of WebNS, the command has changed to **web-mgt state [disable | enable]**. Access through HTTP XML is disabled by default. In order to control the HTTP WebNS device management user access, issue the **no restrict web-mgmt** command and the **restrict web-mgmt** command. The WebNS device management user interface is disabled by default. You must configure both the **no restrict xml** command and the **no restrict web-mgmt** command in order to browse to the CSS on port 8081.

In version 5.00 and later, if you HTTP-browse to the circuit address on port 8081, the browser is redirected to use HTTPS and connect to the same circuit address.

HTTPS

The CSS supports remote configuration through the HTTP Secure (HTTPS) protocol. This Secure Socket Layer (SSL) protects data transfers (which can include passwords) between the WebNS device management user interface and your web browser.

In order to control HTTPS WebNS device management user access, issue the **no restrict web-mgmt** command and the **restrict web-mgmt** command. The WebNS device management user interface is disabled by default. If it is disabled, the CSS continues to listen on the particular port but closes the connection. So that packets do not hit the SSL TCP port 443, configure ACL clauses to deny the packets.

Management and Interactive Access Over the Internet (and Other Untrusted Networks)

Many users manage their CSSs remotely, and sometimes this is accomplished over the Internet. Any unencrypted remote access carries some risk, but access over a public network such as the Internet is especially dangerous. All remote management schemes, which include interactive access, HTTP, and SNMP, are vulnerable.

The attacks that this section discusses are relatively sophisticated ones, but they are by no means out of the reach of the crackers of today. Public network providers who take the proper security measures can often thwart these attackers. Evaluate your level of trust in the security measures that all the providers who carry your management traffic use. Even if you trust your providers, take at least some steps to protect yourself from the results of any mistakes these providers might make.

All the cautions in this section apply as much to hosts as to the CSS. While this document discusses how to protect CSS login sessions, also look into the use of analogous mechanisms in order to protect your hosts if you administer those hosts remotely. Remote Internet administration is useful, but it requires careful attention to security.

Packet Sniffers

Crackers frequently break into computers that Internet service providers own, or into computers on other large networks. The crackers install packet sniffer programs, which monitor traffic that passes through the network. These packet sniffer programs steal data, such as passwords and SNMP community strings. Network operators have begun to improve their security, which makes this theft more difficult. However, this theft is still relatively common. In addition to the risk from outside crackers, rogue ISP personnel can also install sniffers. Any password that is sent over an unencrypted channel is at risk, which includes the login and enable passwords for your CSSs.

If you can, avoid logging into your CSS with the use of any unencrypted protocol over any untrusted network. If your CSS software supports it, use an encrypted login protocol such as SSH.

If you do not have access to an encrypted remote access protocol, another possibility is to use a one-time password system such as S/KEY or OPIE, together with a TACACS+ or RADIUS server, in order to control both interactive logins and privileged access to your CSS. The advantage is that a stolen password is of no use. A stolen password is made invalid by the very session in which it is stolen. Data that is transmitted in the session and not related to passwords remain available to eavesdroppers, but many sniffer programs are set up to concentrate on passwords.

If you must send passwords over cleartext Telnet sessions, change your passwords frequently, and pay close attention to the path that your sessions traverse.

Other Internet Access Dangers

In addition to packet sniffers, the remote Internet management of a CSS presents these security risks:

- In order to manage a CSS over the Internet, you must permit at least some Internet hosts to have access to the CSS. These hosts can be compromised, or their addresses can be spoofed. When you permit interactive access from the Internet, you make your security dependent, not only on your own antispoofing measures, but on the antispoofing measures of the service providers that are involved.

You can reduce these dangers if you perform these actions:

- ◆ Make sure that all the hosts that are permitted to log in to your CSS are under your own control.
- ◆ Use encrypted login protocols with strong authentication.
- Sometimes, access to an unencrypted TCP connection (such as a Telnet session) is possible to obtain. Someone who gets access to this type of session can actually take control away from a user who is logged in. Such attacks are not nearly as common as simple packet sniffing and can be complex to mount. However, such attacks are possible, and an attacker who has your network specifically in mind as a target can use them. The only real solution to the problem of session theft is to use a strongly authenticated, encrypted management protocol.
- Denial of service (DoS) attacks are relatively common on the Internet. If your network is under a DoS attack, you can be unable to reach your CSS in order to collect information or take defensive action. Even an attack on the network of someone else can impair management access to your own network. Although you can take steps to make your network more resistant to DoS attacks, the only real defense against this risk is to have a separate, out-of-band management channel (such as a dialup modem) for use in emergencies.

Logging

Cisco CSSs can record information about a variety of events, many of which have security significance. Logs can be invaluable for the characterization and response to security incidents. You can issue the **logging subsystem** command in order to enable logs on the CSS. The default logging level is warning-4 for all subsystems.

Issue these commands for subsystem logging in order to collect this information:

- User logins
- Logouts
- RADIUS authentication
- TACACS+ authentication

```
logging subsystem radius level debug-7
logging subsystem security level debug-7
logging subsystem netman level debug-7
```

Note: The **netman subsystem** command covers TACACS+ debugs.

From a security point of view, the most important events that system logging usually records include these events:

- Interface status changes
- Changes to the system configuration
- ACL matches

```
logging subsystem netman level info-6

!--- Note that the default logging level is warning-4, which does
!--- not appear in the configuration.

logging commands enable
logging subsystem acl level debug-7
```

Remote Monitoring (RMON) allows you to remotely monitor and analyze the activity of packets on CSS Ethernet ports. RMON also allows alarm configuration for the monitor of MIB objects and allows event configuration to notify you of these alarm conditions. An RMON event is the action that occurs when an associated RMON alarm is triggered. You can configure an alarm event such that, when an alarm event occurs, it generates one or both of these items:

- A log event
- A trap to an SNMP network management station

Save Log Information

By default, the CSS saves boot and subsystem event log messages to log files on the hard or Flash disk. The content of these files is recorded in ASCII text. You can also configure the CSS to send log messages to an active CSS session, email address, or another host system.

The maximum size of a local log file is 50 MB for hard disk–based systems and 10 MB for Flash disk–based systems.

The subsystem log messages are subsystem events that occur during the operation of the CSS. The CSS saves these messages in the `sys.log` file. The CSS creates this file when the first subsystem event occurs that must be logged. The CSS determines which subsystem messages to log by its configured logging level.

Most larger installations have syslog servers. You can issue the **logging host** command in order to send logging information to a syslog daemon on the host system. Even if you have a syslog server, you should still enable local logging to disk.

All logs are time–stamped with the month, day, and time to the second. If you configure a common time source such as Simple Network Time Protocol (SNTP) for your logs, you can more easily track the sequence of logged events. In order to configure the SNTP server on the CSS, issue the **sntp** command. SNTP was introduced in 5.00 code.

Record Access List Violations

If you use ACLs to filter traffic that accesses circuit addresses or content rule virtual IP (VIP) addresses, you

can choose to log packets that violate your filter criteria. In order to enable logging on the ACL clause, issue the **clause # log enable** command. Also, issue the **logging subsystem acl level debug-7** command. The CSS logs this information:

- Protocol
- Source port
- Destination port
- Source IP address
- Destination IP address

Try to avoid the configuration of logging for ACL entries that match very large numbers of packets. This configuration causes log files to grow excessively large and can cut into system performance.

You can also use ACL logging to characterize traffic that is associated with network attacks. In this case, you configure ACL logging to log the suspect traffic. You can characterize on the Cisco router on the Internet side of the CSS in order to craft an ACL. Refer to *Characterizing and Tracing Packet Floods Using Cisco Routers* for more information.

Note: CSS ACLs are only applied on inbound packets. The ACL does not check packets that are outbound from an interface.

Secure the IP Routing

This section discusses some basic security measures that relate to the way in which the router forwards IP packets. Refer to *Cisco ISP Essentials – Essential IOS Features Every ISP Should Consider* for more information about these issues.

By default, a configuration of the CSS:

- Restricts the number of SYN packets that go to a VIP before CSS logs it as a DoS attack

Note: This behavior cannot be disabled.

- Denies directed broadcasts
- Denies packets with the same source and destination IP address
- Denies multicast source IP addresses
- Denies source or destination port 0 packets

Antispoofing

Many network attacks rely on an attacker that falsifies, or spoofs, the source addresses of IP datagrams. Some attacks rely on spoofing in order for the attack to work. Other attacks are much harder to trace if the attackers can use the address of someone else instead of their own address. Therefore, to prevent spoofing wherever it is feasible is valuable for network administrators.

Antispoofing should be done at every point in the network where it is practical. But antispoofing is usually both easiest to do and most effective at the borders between large address blocks or between domains of network administration. Antispoofing on every router in a network is usually impractical because determination of which source addresses can legitimately appear on any given interface is difficult.

If you are an Internet service provider (ISP), you may find that effective antispoofing, together with other effective security measures, causes expensive, problem subscribers to take their business to other providers. If you are an ISP, be especially careful to apply antispoofing controls at dialup pools and other end-user connection points.

Note: Refer to RFC 2267 .

Administrators of corporate firewalls or perimeter routers sometimes install antispoofing measures so that hosts on the Internet cannot assume the addresses of internal hosts. However, internal hosts can still assume the addresses of hosts on the Internet. Try to prevent spoofing in both directions. There are at least three good reasons to install antispoofing in both directions at an organizational firewall:

- Internal users are less tempted to try to launch network attacks and less likely to succeed if they do try.
- Internal hosts that are accidentally misconfigured are less likely to cause trouble for remote sites. Therefore, they are less likely to generate customer dissatisfaction.
- Outside crackers often break into networks as launching pads for further attacks. These crackers may be less interested in a network with outgoing spoofing protection.

Antispoofing with ACLs

Unfortunately, to simply list commands that provide appropriate spoofing protection is not practical. ACL configuration depends too much on the individual network. The basic goal is to discard packets that arrive on interfaces that are not viable paths from the supposed source addresses of those packets. For example, on a two-circuit CSS that connects a server farm to the Internet, you want to discard any datagram that arrives on the Internet circuit, but has a source address field that claims that it came from a machine on the server farm.

Similarly, you want to discard any datagram that arrives on the interface that is connected to the server farm, but that has a source address field that claims that it came from a machine outside the server farm. If CPU resources allow, apply antispoofing on any circuit where a determination of what traffic can legitimately arrive is feasible.

ISPs that carry transit traffic can have limited opportunities to configure antispoofing ACLs, but such ISPs can usually filter outside traffic that claims to originate within the address space of that ISP.

In general, antispoofing filters must be built with input ACLs. Packets must be filtered at the circuits through which the packets arrive. The CSS can only apply ACLs to inbound packets.

When antispoofing ACLs exist, they should always reject datagrams with broadcast or multicast source addresses. By default, the CSS denies these datagrams. Antispoofing ACLs should also reject datagrams that have the reserved loopback address as a source address. In addition, you should usually have an antispoofing ACL filter out all Internet Control Message Protocol (ICMP) redirects, regardless of the source or destination address. The CSS ACL does not allow you to specify the ICMP type to deny. Instead, issue the **no redirects** command in order to configure all circuit IP addresses to not accept ICMP redirects. These are the commands:

```
clause # deny any 127.0.0.0 255.0.0.0 destination any
clause # deny any 0.0.0.0 0.0.0.0 destination any
```

Note: The **clause # deny any 0.0.0.0 0.0.0.0 destination any** command filters out packets from many Bootstrap Protocol (BOOTP)/DHCP clients. Therefore, the command is not appropriate in all environments.

Control of Directed Broadcasts

Extremely common and popular smurf DoS attacks, and some related attacks, use IP directed broadcasts. By default, the CSS is configured with the **no ip subnet-broadcast** command, which denies directed broadcasts.

An IP directed broadcast is a datagram that is sent to the broadcast address of a subnet to which the sending machine is not directly attached. The directed broadcast is routed through the network as a unicast packet until the directed broadcast arrives at the target subnet. At the subnet, the directed broadcast is converted into a link

layer broadcast. Because of the nature of the IP addressing architecture, only the last router or Layer 3 network device in the chain can conclusively identify a directed broadcast. This device is the one that is connected directly to the target subnet. Directed broadcasts are occasionally used for legitimate purposes, but such use is not common outside the financial services industry.

In a smurf attack, the attacker sends ICMP echo requests from a falsified source address to a directed broadcast address. As a result, all the hosts on the target subnet send replies to the falsified source. When an attacker sends a continuous stream of such requests, the attacker can create a much larger stream of replies, which can completely inundate the host whose address is falsified.

Refer to *The Latest in Denial of Service Attacks: "Smurfing" Description and Information to Minimize Effects* for a strategy to block smurf attacks on some firewall routers (which depends on the network design). The document also provides general information on the smurf attack.

Path Integrity

Many attacks depend on the ability to influence the paths that datagrams take through the network. If crackers control routing, there is a chance that they can spoof the address of the machine of another user and have the return traffic sent to them. In some cases, crackers can intercept and read data that are intended for someone else. Routing can also be disrupted purely for DoS purposes.

IP Source Routing

The IP protocol supports source routing options that allow the sender of an IP datagram to control the route that the datagram takes toward the ultimate destination, and generally, the route that any reply takes. These options are rarely used for legitimate purposes in real networks. Some older IP implementations do not process source-routed packets properly. Someone can send datagrams with source routing options and, possibly, crash machines that run these implementations.

The CSS is configured by default with the **no ip source-route set** command. The CSS never forwards an IP packet that carries a source routing option. Leave the default command configured unless you know that your network needs source routing.

ICMP Redirects

An ICMP redirect message instructs an end node to use a specific router as the path to a particular destination. In an IP network that functions properly, a router sends redirects only to hosts on the local subnets of the router. End node never send a redirect, and redirects never traverses more than one network hop. However, an attacker can violate these rules, and some attacks are based on these rules. Filter out incoming ICMP redirects at the input interfaces of any router that lies at a border between administrative domains. In addition, you can have any ACL that is applied on the input side of a Cisco router interface filter out all ICMP redirects. This filtering causes no operational impact in a network that is configured correctly.

This type of filtering prevents only redirect attacks that remote attackers launch. In addition, attackers can use redirects to cause significant trouble if the attacker host is directly connected to the same segment as a host that is under attack.

By default, the CSS is configured to accept redirects on each circuit IP address that is configured. Issue the **no redirect** command under the circuit IP address in order to turn off this function.

Routing Protocol Filtering and Authentication

If you use a dynamic routing protocol that supports authentication, enable that authentication. The authentication prevents some malicious attacks on the routing infrastructure and can also help to prevent damage that misconfigured rogue devices on the network can cause.

For the same reasons, service providers and other operators of large networks can consider the use of route filtering. With route filtering, the network routers do not accept clearly incorrect routing information. For route filtering, use the **distribute-list** parameter in the command. Excessive use of route filtering can destroy the advantages of dynamic routing. But selective use often helps to prevent bad results. For example, if you use a dynamic routing protocol in order to communicate with a stub customer network, do not accept any routes from that customer other than routes to the address space that you have actually delegated to the customer.

The CSS cannot filter routes. Instead, configure routing peers of the CSS with this function.

This document does not provide detailed instruction on the configuration of routing authentication and route filtering. Such documentation is available on Cisco.com and elsewhere. You can refer to the document Cisco ISP Essentials – Essential IOS Features Every ISP Should Consider. Because of the complexity, seek experienced advice if you are a novice before you configure these features on important networks.

Flood Management

Many DoS attacks rely on floods of useless packets. These floods congest network links, slow down hosts, and can overload routers as well. Careful router configuration can reduce the impact of such floods.

An important part of flood management is awareness of where performance bottlenecks can occur. If a flood overloads a T1 line, filter out the flood on the router at the source end of the line. There is little or no effect if you filter at the destination end in this case. If the router itself is the most overloaded network component, you can make matters worse if you filter protections that place heavy demands on the router. Keep this in mind when you consider an implementation of the suggestions in this section.

Transit Floods

You can use Cisco QoS features on upstream Cisco IOS[®] routers in order to protect the CSS, hosts, and links against some kinds of floods. Unfortunately, this document does not provide a general treatment of this sort of flood management. Also, the protection depends heavily on the attack. The only simple, generally applicable advice is to use weighted fair queuing (WFQ) wherever CPU resources can support WFQ. WFQ is the default for low-speed serial lines in later versions of Cisco IOS Software. Other features of possible interest include:

- Committed access rate (CAR)
- Generic traffic shaping (GTS)
- Custom queuing

Sometimes, you can configure these features when under an active attack.

The CSS can reduce the impact of SYN flooding attacks on the VIP and real servers. By default, the CSS restricts the number of SYNs and incomplete three-way handshakes and logs them as DoS attacks.

Refer to Security Reference Information for more information.

Possibly Unnecessary Services

As a general rule, disable any unnecessary service in any router that is reachable from a potentially hostile network. The services that this section lists are sometimes useful. But disable these services if they are not in active use.

SNTP

SNTP is not especially dangerous, but any unnecessary service can present a path for penetration. If you actually use SNTP, be sure to explicitly configure the trusted time source. SNTP does not use authentication. A corruption of the time base is a good way to subvert certain security protocols. The best method is to use a source that is internal and less likely to be spoofed.

Cisco Discovery Protocol

Cisco Discovery Protocol (CDP), which was introduced in WebNS 5.10, is used for some network management functions. CDP is dangerous because any system on a directly connected segment can perform these actions:

- Learn that the router is a Cisco device
- Determine the model number and the software version that runs

An attacker can use this information in order to design attacks against the CSS. CDP information is accessible only to directly connected systems. The CSS only advertises CDP information. The CSS does not listen. You can issue the **no cdp run** global configuration command in order to disable the CDP protocol. You cannot disable CDP on the CSS on a per-interface basis.

Stay Up To Date

Like all software, Cisco WebNS software has bugs. Some of these bugs have security implications. In addition, new attacks continue to be invented. And behavior that was considered correct when a piece of software was written can have bad effects when the behavior is deliberately exploited.

When a major new security vulnerability is found in a Cisco product, Cisco generally issues an advisory notice about the vulnerability. Refer to the Security Vulnerability Policy for information about the process through which these notices are issued. Refer to Security Advisories for the notices.

Almost any unexpected behavior of any piece of software can create a security exposure somewhere. Advisories only mention bugs that have direct implications for system security. You can enhance your security if you keep your software up to date, even in the absence of any security advisory.

Some security problems are not the result of software bugs, and network administrators must stay aware of trends in attacks. There are a number of websites, Internet mailing lists, and Usenet newsgroups that are concerned with these trends.

Related Information

- [RFC 2267](#)
- [Security Advisories](#)
- [Security Vulnerability Policy](#)
- [Security Reference Information](#)
- [Configuring CSS Network Protocols](#)

- **Configuring CSS Remote Access Methods**
 - **Configuring User Profiles and CSS Parameters**
 - **Release Notes**
 - **Characterizing and Tracing Packet Floods Using Cisco Routers**
 - **Cisco ISP Essentials – Essential IOS Features Every ISP Should Consider**
 - **The Latest in Denial of Service Attacks: "Smurfing" Description and Information to Minimize Effects**
 - **Technical Support & Documentation – Cisco Systems**
-

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2010 – 2011 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

Updated: Dec 27, 2006

Document ID: 25945
