Release Note for the Cisco 11500 Series Content Services Switch

January 30, 2013

Note The most current Cisco documentation for released products is available on Cisco.com.

Contents

This release note applies to the following software versions for the Cisco 11500 Series Content Services Switch (CSS):

- 8.10.9.01 (version 8.10, release 9, build 01)
- 8.10.8.01 (version 8.10, release 8, build 01)
- 8.10.7.01 (version 8.10, release 7, build 01)
- 8.10.6.02 (version 8.10, release 6, build 02)
- 8.10.5.03 (version 8.10, release 5, build 03)
- 8.10.4.01 (version 8.10, release 4, build 01)
- 8.10.3.01 (version 8.10, release 3, build 01)
- 8.10.2.05 (version 8.10, release 2, build 05)
- 8.10.1.06 (version 8.10, release 1, build 06)
- 8.10.0.02 (version 8.10, release 0, build 02)

For information on version 8.10 commands and features, refer to the CSS 8.10 documentation located in http://www.cisco.com.

This release note contains the following sections:

- CSS Standard and Enhanced Feature Sets
- Before Upgrading the CSS Software
- Required Updates to Management Information Base (MIB) Files

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

- Features in Software Version 8.10
- Documentation Set for Software Version 8.10
- Operating Considerations
- Software Version 8.10.9.01 Open Caveats and Resolved Caveats
- Software Version 8.10.8.01 Open Caveats, Resolved Caveats, and Command Changes
- Software Version 8.10.7.01 Open Caveats, Resolved Caveats, and Command Changes
- Software Version 8.10.6.02 Open Caveats, Resolved Caveats, and Command Changes
- Software Version 8.10.5.03 Open Caveats, Resolved Caveats, and Command Changes
- Software Version 8.10.4.01 Open Caveats, Resolved Caveats, and Command Changes
- Software Version 8.10.3.01 Open Caveats, Resolved Caveats, and Command Changes
- Software Version 8.10.2.05 Open Caveats, Resolved Caveats, and Command Changes
- Software Version 8.10.1.06 Open Caveats, Resolved Caveats, and Command Changes
- Software Version 8.10.0.02 Open Caveats, Resolved Caveats, and Command Changes
- Obtaining Documentation and Submitting a Service Request

CSS Standard and Enhanced Feature Sets

The CSS software is available in a Standard or optional Enhanced feature set. The Enhanced feature set contains all of the Standard feature set and also includes Network Address Translation (NAT) Peering, Domain Name Service (DNS), Demand-Based Content Replication (Dynamic Hot Content Overflow), Content Staging and Replication, and Network Proximity DNS. Proximity Database and Secure Management, which includes Secure Shell Host and SSL strong encryption for the Device Management software, are optional features.

Software version 8.10 no longer requires that you enter a license key for the Standard software feature set. The Enhanced software feature set, as well as the optional Secure Management feature, still require a license key in order to be activated.

Before Upgrading the CSS Software

Before you upgrade the CSS software, refer to the information in the following sections:

- Required Minimum Maintenance Release Before Upgrading to Version 8.10
- Archiving Custom Scripts

Required Minimum Maintenance Release Before Upgrading to Version 8.10

For the CSS software version 8.10 to support the SSL compression (SSL-C) module, before you upgrade, the CSS **must** be at one of the following maintenance releases or higher:

- 7.50.1.03
- 7.40.2.02
- 7.30.4.02

If the CSS is not at one of these maintenance releases, you must perform the following upgrade sequence:

1. Upgrade the CSS to the required maintenance release.
2. Upgrade the CSS to software version 8.10.

Archiving Custom Scripts

Before you upgrade your CSS software, archive your custom scripts (including user profiles and custom script keepalives) by using the **archive script** or **save_profile** command. When you upgrade the software, the upgrade process creates a new /<current running version>/script directory, overwriting the current script directory.

After the upgrade is done, use the **restore** *filename* **script** command to restore the scripts you archived. Refer to the *Cisco Content Services Switch Administration Guide* for detailed software upgrade instructions.

Required Updates to Management Information Base (MIB) Files

The MIBs in 8.10 have been modified to be consistent with other Cisco products within the Cisco private enterprise branch of the MIB tree. The modifications include a change to the enterprise OIDs (Object Identifiers). If you have created any customized network management applications, you must modify these applications in order to use the new OIDs in the modified MIBs in 8.10. If you continue to use the former Arrowpoint enterprise OIDs (.2467), the CSS will not recognize SNMP requests.

The former Arrowpoint enterprise MIB branch was:

- iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).arrowPoint(2467)
  1.3.6.1.4.1.2467

The new Cisco enterprise MIB branch is:

- iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).cisco(9).ciscoMgmt(9).arrowPoint(368)
  1.3.6.1.4.1.9.9.368

Replace .2467 with 9.9.368 wherever it is used. For a graphical view of the updated MIB tree, refer to the *Cisco Content Services Switch Administration Guide*, Chapter 5, ‘Configuring Simple Network Management Protocol’, Figure 5-2.

After you upgrade the CSS software, you must unload the current CSS MIBs and load the latest CSS MIBs in your network management station. The CSS MIBs are included in the CSS GZIP file. During the software upgrade, the MIBs are loaded into the CSS /mibs directory.

To update the CSS MIBs on your management station after you upgrade the CSS:

1. FTP the specific MIBs or the GZIP file (which contains all the MIBs) from the CSS MIBs (/v1 or /v2) directory to your management station.
2. Unload the CSS MIBs from the management application.
3. Load the MIBs into the management application.

Features in Software Version 8.10

The following new features are supported in software version 8.10:

- CiscoView Device Manager (CVDM) installation - *Cisco Content Services Switch Getting Started Guide*
- Cisco Unique Device Identifier (UDI) display information - *Cisco Content Services Switch Content Administration Guide*
- SSL integrated compression hardware - *Cisco 11500 Series Content Services Switch Hardware Installation Guide* and *Cisco Content Services Switch SSL Configuration Guide*
- HTTP data compression configuration - *Cisco Content Services Switch SSL Configuration Guide*

Documentation Set for Software Version 8.10

The documentation set for software version 8.10 contains the publications listed below.

Document Title	Description
Cisco 11500 Series Content Services Switch Hardware Installation Guide	This guide provides information for installing, cabling, and powering the Cisco 11500 series CSS. In addition, this guide provides information about CSS specifications, cable pinouts, and hardware troubleshooting.
Cisco Content Services Switch Getting Started Guide	This guide describes how to perform initial administration and configuration tasks on the CSS, including: • Booting the CSS for the first time and a routine basis, and logging in to the CSS • Configuring the username and password, Ethernet management port, static IP routes, and the date and time • Configuring DNS server for hostname resolution • Configuring sticky cookies with a sticky overview and advanced load-balancing method using cookies • Installing the CSS Cisco View Device Manager (CVDM) browser-based user interface used to configure the CSS • Finding information in the CSS documentation with a task list • Troubleshooting the boot process

Document Title	Description
Cisco Content Services Switch Administration Guide	This guide describes how to perform administrative tasks on the CSS, including upgrading your CSS software and configuring the following: • Logging, including displaying log messages and interpreting sys.log messages • User profile and CSS parameters • SNMP • RMON • XML documents to configure the CSS • CSS scripting language • Offline Diagnostic Monitor (Offline DM) menu
Cisco Content Services Switch Routing and Bridging Configuration Guide	This guide describes how to perform routing and bridging configuration tasks on the CSS, including: • Management ports, interfaces, and circuits • Spanning-tree bridging • Address Resolution Protocol (ARP) • Routing Information Protocol (RIP) • Internet Protocol (IP) • Open Shortest Path First (OSPF) protocol • Cisco Discovery Protocol (CDP) • Dynamic Host Configuration Protocol (DHCP) relay agent
Cisco Content Services Switch Content Load-Balancing Configuration Guide	This guide describes how to perform CSS content load-balancing configuration tasks, including: • Flow and port mapping • Services • Service, global, and script keepalives • Source groups • Loads for services • Server/Application State Protocol (SASP) • Dynamic Feedback Protocol (DFP) • Owners • Content rules • Sticky parameters • HTTP header load balancing • Content caching • Content replication

Document Title	Description
Cisco Content Services Switch Global Server Load-Balancing Configuration Guide	This guide describes how to perform CSS global load-balancing configuration tasks, including: • Domain Name System (DNS) • DNS Sticky • Content Routing Agent • Client-Side Accelerator • Network proximity
Cisco Content Services Switch Redundancy Configuration Guide	This guide describes how to perform CSS redundancy configuration tasks, including: • VIP and virtual interface redundancy • Adaptive session redundancy • Box-to-box redundancy
Cisco Content Services Switch Security Configuration Guide	This guide describes how to perform CSS security configuration tasks, including: • Controlling access to the CSS • Secure Shell Daemon protocol • RADIUS • TACACS+ • Firewall load balancing
Cisco Content Services Switch SSL Configuration Guide	This guide describes how to perform CSS SSL configuration tasks, including: • SSL certificate and keys • SSL termination • Back-end SSL • SSL initiation • HTTP data compression
Cisco Content Services Switch Command Reference	This reference provides an alphabetical list of all CLI commands including syntax, options, and related commands.

Operating Considerations

The following operating considerations apply to software version 8.10 and greater.

- When you use the **advanced-balance** content rule command for sticky methods that access the sticky database, the **advanced-balance** command sticky methods that use the internal sticky database include: **sip-call-id**, **ssl**, **sticky-srcip**, **sticky-srcip-dstport**, and **wap-msisdn**.

  In a CSS11500 chassis with multiple Session Processors (SP), the sticky database must be synchronized between each SP to allow for the sticky feature to work. If several client connections are sent in a very short duration that use the same sticky key, for example the same source IP address for **advanced-balance sticky-srcip**, there is a possibility that the connections could be directed to two different servers. In this case, this behavior would be due to the fact that the sticky database was not synchronized between each SP in the CSS11500 chassis.
- For the CSS software version 8.10 to support the SSL compression (SSL-C) module, before you upgrade, the CSS **must** be at software version 7.50.1.03, 7.40.2.02, 7.30.4.02 or higher. Otherwise, the installation of software version 8.10 will fail. For more information, see the "Required Minimum Maintenance Release Before Upgrading to Version 8.10" section.
- The CVDM GUI is not part of the CSS software image. You must download the CVDM image separately.
- The global configuration mode **idle timeout web-mgmt** command does not apply to the CVDM GUI session.
- When the SSL modules are oversubscribed, you will see more failed connections. After the oversubscription stops, the number of failed connections will eventually decrease on the modules.
- When you configure the expiration time and date for a location cookie using the **location-cookie expiration** command, the CSS CPU may spike and the CSS may experience a degradation in its performance. Configure the **expiration** option with the **location-cookie** command only when necessary.
- When you configure the **arrowpoint-cookie expiration** command and the **advanced-balance arrowpoint-cookie** command, the CSS CPU may spike and the CSS may experience a degradation in its performance. Configure the **arrowpoint-cookie expiration** command only when necessary.
- When the CSS is processing an SNMP BULK_WALK request to obtain the ether-history table, the requesting application may time out due to the large amount of information it has to gather. To avoid having the requesting application time out, increase the requesting application's retransmission timer.
- When you configure redundant firewalls without configuring the firewalls to accept ICMP ECHO requests or replies (or pings), the CSS places the KAL in the down state. If the master firewall fails over to the backup, the CSS continues to send traffic to the MAC address of the old master except for self-initiated traffic, such as KALs. To establish traffic to the MAC address of the new master, manually clear the CSS MAC entry by using the **clear arp cache** command.
- If you configure a **balance** or **advanced-balance** method on a content rule that requires the TCP protocol for Layer 5 (L5) spoofing, you should configure a default URL string, such as **url "/*"**. The addition of the URL string forces the content rule to become an L5 rule and ensures L5 load balancing or stickiness. If you do not configure a default URL string, unexpected results can occur.

  In the following configuration example, if you configure a Layer 3 (L3) content rule with an L5 balance method, the CSS performs L5 load balancing, but will reject UDP packets.

```
content testing
vip address 192.168.128.131
add service s1
balance url
active
```

The **balance url** method is an L5 load-balancing method in which the CSS must spoof the connection and examine the HTTP GET content request to perform load balancing. The CSS rejects the UDP packet sent to this rule because a UDP connection cannot be L5. Though the CSS allows this rule configuration, its expected behavior would be more clear if you promote the rule to L5 by configuring the **url "/*"** command.

In the next example, if you configure an L3 content rule with an L5 advanced-balance method, L5 stickiness will not work as expected.

```
content testing
vip address 192.168.128.131
add service s1
advanced-balance arrowpoint-cookie
active
```

The **advanced-balance arrowpoint-cookie** method causes the CSS to spoof the connection, however, the CSS still marks it as an L3 rule. Thus, the CSS does not insert the generated cookie and the rule defaults to L3 stickiness (sticky-srcip). You must configure a URL like **url "/*"** to promote this rule to L5, ensuring that L5 stickiness works as expected.

Note There is a significant difference between hardware or software compression performance capability. We highly recommend that you do not use compression on an SSL module that does not have integrated hardware compression. This module performs compression through software, but it is not optimized for performance.

Use Table 1 to determine how the CSS performs compression based on the module type or the CSS 11501 platform.

Table 1 CSS Compression Method

		CSS Platform and SSL-Module Type				
		11501	**11501S**	**11501S-C**	**11503 or 11506 with an SSL module**	**11503 or 11506 with an SSL-C module**
Compression method	Software	Not available	Yes	No	Yes	No
	Hardware	Not available	No	Yes	No	Yes

- We do not recommend using custom scripted keepalive scripts that contain the ">" or ">>" file redirection characters (see DDTS CSCek55371 in the "Software Version 8.10.2.05 Open Caveats" section). These characters write the output of a CSS command to the named file on disk. For example, the following command writes the received data from the keepalive host to a file on the CSS disk named tmp:

  socket inspect ${SOCKET} >log/tmp

- The CSS generates a Lifetick Failure trap when ISC-port configuration changes occur.
- When the CSS has an uptime of 828 days, it cannot send packets to the management port for 18 minutes. This issue affects management port only. The circuit and VIP addresses works fine. We recommend that you reboot the CSS before its uptime is 828 days.
- When you configure the **max connections** command on the CSS, the number of connections is reduced by two. For example, if you configure the maximum connection to 6, the CSS allows only four connections.

Software Version 8.10.9.01 Open Caveats and Resolved Caveats

The following sections contain the open and resolved caveats in software version 8.10.9.01:

- Software Version 8.10.9.01 Open Caveats
- Software Version 8.10.9.01 Resolved Caveats

Software Version 8.10.9.01 Open Caveats

Software version 8.10.9.01 has no open caveats.

Software Version 8.10.9.01 Resolved Caveats

The following caveats were resolved in software version 8.10.9.01:

- **CSCtx68270**—The CSS SSL Module failed to do SSL header insert with the newer versions of the Chrome Browsers (v16 and v17). The behavior is also observed with Internet Explorer (IE) with Windows security patch KB2585542.
- **CSCty60767**—Windows security patch KB2585542 added the ability to split the HTTP method across 2 SSL encrypted blocks. The CSS drops the 2nd fragment and waits 3 seconds for the SSL module to retransmit the 2nd fragment, and this adds a large delay in the processing of the SSL connection.
- **CSCua70184**—The **show rule** command can cause the CSS to generates a core dump if another user was performing a dynamic content rule configuration at the same time.
- **CSCty80826**—Processing an SSL connection, under some error paths, can cause some buffers to not be freed properly. Overtime, this buffer leak could cause the encrypted keepalives to fail.

Software Version 8.10.8.01 Open Caveats, Resolved Caveats, and Command Changes

The following sections contain the open caveats, resolved caveats, and command changes in software version 8.10.8.01:

- Software Version 8.10.8.01 Open Caveats
- Software Version 8.10.8.01 Resolved Caveats
- Software Version 8.10.8.01 Command Changes

Software Version 8.10.8.01 Open Caveats

Software version 8.10.8.01 has no open caveats.

Software Version 8.10.8.01 Resolved Caveats

The following caveats were resolved in software version 8.10.8.01:

- **CSCtk82402**—When the CSS boots and there is no available disk space, this can cause the **show ssl files** command not to display any results in the associated **show** command output. This behavior occurs as a result of no available disk space which prevents the CSS from building the encrypted SSL database.

  With this CSS release, the following message will appear to flag this disk space issue on the CSS:

  ```
  syslog message - NETMAN-2: SSL file list could not be constructed - check disk space.
  ```

  In addition, the **show ssl files** command will return the error "Error: Invalid SSL file list detected - check disk space".

- **CSCtl42000**—In some cases, you may find that the SSL RSA key is invalid for use by the CSS because the P and Q values are not prime numbers. When this occurs, the CSS accepts this key in the **ssl associate** command, however, when the SSL Handshake initiates it fails during the SSL Client Key Exchange and the CSS logs this failure as critical error "SSLACCEL-3: CRYPTO HARDWARE INVALID PARAMETER".

  With this CSS release, the CSS rejects this key in the **ssl associate** command and generates a log to instruct the user about a problem with the RSA key ("%% RSA Key invalid, bad prime found"). The CSS then generates one or both of the following syslogs:

  ```
  NETMAN-3: Invalid RSA key - p not prime
  NETMAN-3: Invalid RSA key - q not prime
  ```

- **CSCtn14907**—The CSS can be configured for a content rule without a VIP or port, which, in some configurations, can cause network application issues. The CSS will now warn you about this potential misconfiguration through the display of one of the following errors:

  ```
  %% WARNING: Rule rule being activated without a VIP or Port configured
  %% WARNING: Rule rule being activated without a VIP configured
  ```

- **CSCtl94836**—A HTTP Persistent connection to a Layer 5 virtual IP address on the CSS fails when the CSS has to change the backend server due to cookie persistence. This failure allows for enough delay that the application has to retransmit one packet of a HTTP GET which spans four Ethernet packets due to the long length of the cookie. In this instance, the CSS mishandles the retransmission.

- **CSCtn01197**—The CSS is configured for a Layer 5 content rule with persistent due to the cookies balance method. A client HTTP content request (for example, POST or GET) spans multiple packets, and the CSS correctly receives those packets. However, the client retransmitted the middle parts of the spanned content (not the first packet and not the last), which causes the CSS to incorrectly handle the next HTTP content request on the persistent connection. When this occurs, the connection stalls.

  With this CSS release, the new **flow-tcp-expert-retransmit** Layer 5 content rule command has been added to global configuration mode. This command relates to the **flow persist-span-ooo** command (enables the reordering of persistent spanning packets) as a means to address this particular case of client retransmits.
- **CSCtn17683**—The client SSH connection to the CSS local IP address transmitted a NULL (or zero length) command string causing the CSS to become unresponsive. Command validity is now checked by the CSS before execution.
- **CSCtn17809**—PSIRT CVE-2009-3245 OpenSSL before 0.9.8m does not check for a NULL return value from bn_wexpand function calls in (1) crypto/bn/bn_div.c, (2) crypto/bn/bn_gf2m.c, 3) crypto/ec/ec2_smpl.c, and (4) engines/e_ubsec.c, which has unspecified impact and context-dependent attack vectors.
- **CSCtn56210**—With the following CSS configuration:
  - Layer 5 content rules for the same VIP and port combination.
  - At least one of the content rules is using the **ssl-accel-backed** or **ssl-accel-init: services** command.
  - The TCP SYN from the client contains the TCP Window Scale (WS) option.

  As a result of the fix for CSCsq73004, and the SSL Module's TCP stack's inability to support TCP WS when a VIP and port belong to any content rule intended for the SSL Module, the TCP WS option is not returned in the TCP SYN/ACK to the client. When this issue occurs, the Layer 5 connection eventually matches a content rule which does not go to the SSL Module. When the Flow Manager (flowmgr) sends the TCP SYN to the backend server, it sends the client's TCP SYN/ACK and the server returns its own TCP SYN/ACK. This results in CSS performance issues because the client side connection and flow does not use TCP WS.
- **CSCtn63591**—With this CSS release, a few logs printed at SSL logging debug level 7 have been removed from the CSS because they are no longer necessary by the CSS.
- **CSCtn73256**— The CSS may become unresponsive and reboot if it has no available disk space and you attempt to import an SSL certificate or key from the CLI.
- **CSCto16588**—The CSS may write the **ip virtual-router** command to the wrong section of the circuit configuration. This issue can occur when the CSS code, which sorts the running configuration based on SNMP OID, incorrectly assumes that the active IP address mode was actually the last configured and ignores that the CSS transitions into another IP address mode. This behavior can corrupt the configuration so that at the next reboot the IP virtual router configuration is incorrect.
- **CSCtn29457**—The CSS VIP fails to respond to a TCP SYN packet with the SCPS capabilities TCP option. This issue can occur when the CSS incorrectly assumed that this was an internal Nat Channel Indication (NCI) option that the CSS no longer supports.
- **CSCto34867** —The **http-rspcode** configuration parameter takes effect only for a service that is configured with the **keepalive type http non-persistent** command. With this release, the CSS now displays an error if you configure the **http-rspcode** configuration parameter for a service configured with only the **keepalive type http** command. With this CSS release, the CSS instructs you that the **http-rspcode** configuration parameter will not take effect.

- **CSCts43617**— In some instances, the **show sticky-stats debug** command displays negative values for certain counters that should display larger values. When this issue occurs, the counters appear as signed integers when they should be unsigned integers.
- **CSCts77281**—With a CSS configured with SSL termination, when the CSS receives an SSL CLIENT HELLO with TLS 1.1, it may did not properly fall back to TLS 1.0. The CSS would reset the connection, which was an incorrect action.
- **CSCtu09137**— The following command:

```
clause # permit tcp any destination 1.2.3.4 eq 80 prefer
<service>|<service1>,<service2>
```

  may fail if the length of the string "<service1>,<service2>" exceeds 32 characters. Since each service name can be 32 characters, the length of the MIB OID has been increased.

Software Version 8.10.8.01 Command Changes

Table 2 lists the commands and options that have been added or changed in software version 8.10.8.01.

Table 2 CLI Commands Added in Version 8.10.8.01

Mode	Command and Syntax	Description
Global	**flow-tcp-expert-retransmit**	Per CSCtn01197, this new Layer 5 content rule command has been added to global configuration mode to relate to the **flow persist-span-ooo** command as a means to address issues when an HTTP content request (for example, a GET or POST) spans multiple packets and the client retransmits either: • The first packet of a spanned HTTP content request, which contains a GET or POST, is retransmitted after the CSS spanning packets code has delivered all the packets to the server. If the first packet is retransmitted, the CSS could incorrectly assume that retransmission was the start of a new content request. With the new **flow-tcp-expert-retransmit** global configuration parameter, the CSS will drop this unnecessary TCP retransmission action. • If any piece of the spanned HTTP content request, except the first and the last, is retransmitted after the CSS spanning packets code has delivered all the packets to the server, the CSS could incorrectly flag the next new HTTP content request as the retransmission. This action could only happen when the **flow persist-span-ooo** command is configured globally.

Software Version 8.10.7.01 Open Caveats, Resolved Caveats, and Command Changes

The following sections contain the open caveats, resolved caveats, and command changes in software version 8.10.7.01:

- Software Version 8.10.7.01 Open Caveats
- Software Version 8.10.7.01 Resolved Caveats
- Software Version 8.10.7.01 Command Changes

Software Version 8.10.7.01 Open Caveats

Software version 8.10.7.01 has no open caveats.

Software Version 8.10.7.01 Resolved Caveats

The following caveats were resolved in software version 8.10.7.01:

- **CSCsz04690**—The CSS does not look for and remove any of the headers that may be inserted as part of the **ssl-server** *number* **http-header client-cert** command. If you insert these headers prior to encryption before they arrived at the CSS and they were there after decryption, you could impersonate a different client, thus spoofing the client session. Workaround: If the client configures the **ssl-server** *number* **http-header prefix "Unique-"** command and the **"Unique-"** string is secret, the server looks for the "Unique-ClientCert-Subject-CN: CN=userY" header instead of the more generic "ClientCert-Subject-CN: CN=userY" header, therefore mitigating the exposure to spoofing.
- **CSCtd01636**—Summary: An industry-wide vulnerability exists in the Transport Layer Security (TLS) protocol that could impact any Cisco product that uses any version of TLS and SSL. The vulnerability exists in how the protocol handles session renegotiation and exposes users to a potential man-in-the-middle attack.

  This advisory is posted at http://www.cisco.com/warp/public/707/cisco-sa-20091109-tls.shtml.

  Workaround: The CSS 11500 Series Content Services Switches are affected by this vulnerability with default configurations. However, the client authentication feature can be enabled as mitigation/solution. To enable or disable client authentication on a virtual SSL server, use the **ssl-server** <*number*> **authentication** command under the ssl-proxy-list.

  Note: By default, client authentication is disabled. After you enable client authentication on the CSS, you must specify a CA certificate that the CSS uses to verify client certificates.
- **CSCtd32718**—The CSS does not respond to ICMP packets with a a TTL of 1. Workaround: None.
- **CSCtd34926**—The CSS acknowledges POST data from the client but never sends it to the back-end server. Eventually, it resets from timeouts that are seen from the web server. Workaround: None.
- **CSCtd92378**—When you dynamically configure source groups while traffic is flowing on the CSS, the CSS generates an unexpected crash file due to a source group Connection Block structure on a duplicate list.
- **CSCtd92684**—When you configure the CSS with HTTP Class B keepalives, an HTTP keepalive that closes unexpectedly may cause a memory leak. If enough memory is lost, the SCM may become unresponsive causing the CSS to reboot unexpectedly.

- **CSCte18094**—When you configure the CSS for Layer 5 content load balancing with the global **flow persist-span-ooo** command and the CSS receives HTTP POST data packets out of order (OOO), the CSS may not forward the retransmitted packets to the backend server.
- **CSCte63815**—When you configure the CSS with an SSL-C card and enable compression, the CSS allows the Content-Type HTTP header tag to have strings with up to 96 characters. Previously, the CSS allowed 64 characters.
- **CSCte64298**—When you configure the CSS for Layer 5 load balancing and the client sends an HTTP POST immediately followed by a TCP FIN/ACK on an HTTP connection, the connection may be torn down immediately and the CSS does not forward the HTTP POST to the server.
- **CSCte86000**—When you configure the CSS for DNS Proximity and DNS-record keepalives, if you dynamically reconfigure one of the DNS keepalives, the SCM may become unresponsive causing the CSS to reboot unexpectedly. When the dynamic reconfiguration occurs, the CSS incorrectly resets the internal hash pointer for the DNS keepalive causing the reboot.
- **CSCtf00487**—The commit_vip_redundancy sync script had failures that were traced to the handling of the **snmp** *name* command in the CSS configuration.
- **CSCtf62619**—Due to the resolution for CSCtd01636 which defaults SSL renegotiation to disabled, the SSL module may fail to respond to commands from the Command Line Interface (CLI), becomes unresponsive, and does not pass traffic.
- **CSCtf70895**—If you configure the **flow tcp-window-scale disabled** command, the CSS may incorrectly send the TCP Window Scale (WS) Option to the backend server.
- **CSCtf99785**—When you configure the **ssl pre-remove-http-hdr** command on an SSL-proxy list with the **http-header static** command, the CSS SSL module may become unresponsive when the static HTTP header is inserted.
- **CSCtg09231**—When the CSS11500 has SSL or SSL-C cards in the chassis and is running a code version with CSCSte10734 (for example, 8.10.60.4s), the following message occues at initialization time:

  ```
  SSLACCEL-3: Load_X509Cert_Mem error calling PEM_read_bio_X509
  ```

  This is a cosmetic issue only.
- **CSCtg20158**—When you configure the CSS for SSL termination with HTTP header insertion and a clear-text Layer 5 content rule on the backend, there may be a 200-millisecond (ms) delay making the connection to the backend server. One SSL packet on the front end may become multiple TCP packets on the backend (clear text rule) when the HTTP header is inserted. The SSL module is acting as the client to the SP and the 200-ms delay is expected because the SP waits 200 milliseconds to send the TCP ACK for the second TCP packet. The CSS detects that the client is actually the SSL module and skip the 200-ms timer on the SP.
- **CSCtg38327**—When you install an SSL module on the CSS and import SSL keys and certificates, there were differences in the output of the **show ssl files** command from the **dir** command of the /CertStore disk. This issue occurs when importing an SSL file that differed only in upper and lower case, for example foo.key and Foo.key. Now, when this occurs, the CSS displays the following error message:

  ```
  %%file Filename of similar name to filename that already exists
  ```

- **CSCtg52574**—When the CSS is processing an SNMP GET BULK request, it reboots unexpectedly. There is internal SNMP debug flag which may be in an undetermined state.
- **CSCtg73566**—When you configure the CSS for SSL termination with URL rewrite, if the CSS receives an HTTP Content Request where the Location: HTTP tag spanned two packets and scans for a string that does not exist in the HTTP Content Request, the CSS may fail to forward the client HTTP Content Request packets.

- **CSCth28944**—The new debug **show ip internal-interface** command has been added to display the vxWorks shell **mbufShow** command and mbuffer error statistics from the CSS application trying to send it to the vxWorks kernel.
- **CSCth31484**—When you configure a service on the CSS for a scripted keepalive, if the scripted keepalive performs a search for data at the end of the internal 20,000-byte buffer and does not find the data, the CSS may reboot.
- **CSCth95170**—When the CSS has the default setting for flow enabled, SIP sticky is configured on the content rule, the sticky entry exists in the sticky database, and the incoming SIP traffic is IP UDP fragments, the CSS corrupts the first packet of the fragment chain and the rest of the fragment chain are not sent.
- **CSCti11803**—When the CSS has the default setting for flow enabled, SIP sticky is configured on the content rule, the incoming SIP traffic is IP UDP fragments, and the incoming (ingress) and outgoing (egress) ports are the same gigabyte port, some IP UDP fragment packets are not sent on the wire. If the IP fragment chain is transferred between slots (either SP or CPU), the fragments are lost in the Prism fastpath.
- **CSCti12615**—The debug **xmask wcc set global 0x400000** command has been added as a Web Conversation and Control (WCC) debug flag to be used as a diagnostic when the configured sorry service is chosen on a content rule.
- **CSCti75402**—This CDETS ports the changes from CSCsx37430 to the SSL-C (D3GMAC) SSL Chip. The SSL module becomes unresponsive due to a duplicate block-free crash. The SSL module code has been modified to correct the duplicate block-free condition to resolve this issue. This issue is also displayed by the debug **show ssl statistics** command under the nicDuplicateBlockFree counter.
- **CSCti99853**—The CSS does not support SSL certificates or keys greater than or equal to 4096 bits. When you configure the CSS as a SSL client with SSL initiation and the SSL server sends the CSS a certificate signed by a 4096 bit public key, the CSS logs the SSLACCEL-3: CRYPTO HARDWARE INVALID PARAMETER error and leaks memory that would eventually cause the SSL module to become unresponsive. Now, the CSS gracefully closes the SSL connection with a fatal alert and increments a counter displayed by the **show ssl statistics** command.
- **CSCtj28637**—The CSS does not support SSL certificates or keys greater than or equal to 4096 bits. Now, the CSS prevents the **ssl associate** command from importing a 4096-bit or greater RSA or DSA key along with the certificates signed by the key. The CSS displays an error message similar to the following:

```
%%  RSA Key size exceeds 2048, too big.
```

- **CSCtj38660**—When you configure the CSS with a DNS record associated with a kal-icmp keepalive to a VIP and a service associated with the VIP reports a load of 254, the CSS incorrectly marks the keepalive as down impacting traffic. A load of 254 is high but valid and the service should still be considered alive.

Software Version 8.10.7.01 Command Changes

Table 3 lists the commands and options that have been added or changed in software version 8.10.7.01.

Table 3 CLI Commands Added in Version 8.10.7.01

Mode	Command and Syntax	Description
Debug	**show ip internal-interface**	Per CSCth28944, this new command assists with the debuggin of an mBuffer leak.
Debug	**xmask wcc set global 0x400000**	Per CSCti12615, this new command was added as a Web Conversation and Control (WCC) debug flag to be used as a diagnostic when the configured sorry service is chosen on a content rule.
Exec	**show ssl-proxy-list**	Per CSCtd01636, this command displays the renegotiation setting for the SSL servers.
Global	[**no**] **flow drop-content-fin**	By default, when the CSS receives a FIN that immediately follows data as it attempts to make a Layer 5 load-balancing decision, it sends an RST to the client. Per CSCte64298, the **new flow drop-content-fin** command enables the dropping of a FIN that immediately follows data used for a Layer 5 load-balancing decision. By allowing the client to retransmit the FIN, the CSS can handle it more effectively when it does not immediately follows the data. To reset the default behavior of resetting the connection, use the **no flow drop-content-fin** command.
Global	**ssl associate ...**	Per CSCtj28637, this command no longer allows you to configure an SSL key or certificate that exceeds 4096 bits.
Global	**ssl renegotiation enable | disable**	Per CSCtd01636, this new command enables or disables renegotiation on the CSS. The default setting is to disable renegotiation.
Global	[**no**] **ssl pre-remove-http-hdr**	By default, the CSS always insert SSL certificate information headers at the end of the request header. However, the same headers may be inserted earlier in the request header. Some servers may act upon the first instance of the header and the spoofed header may be processed, not the header that the CSS is inserting. Per CSCsz04690, the new **ssl pre-remove-http-hdr** command enables the CSS to search for the headers it will insert in the request header. And, if the CSS finds the headers, it removes them. The inserted headers include client and server certificate information, and session information. If the CSS finds any of these headers, it removes them. This functionality could affect performance if many of these headers are present. By default, the CSS does not search for these headers before insertion. This feature does not work with prefixes. To reset the default behavior, use the **no ssl pre-remove-http-hdr** command.

Software Version 8.10.6.02 Open Caveats, Resolved Caveats, and Command Changes

The following sections contain the open caveats, resolved caveats, and command changes in software version 8.10.6.02:

- Software Version 8.10.6.02 Open Caveats
- Software Version 8.10.6.02 Resolved Caveats
- Software Version 8.10.6.02 Command Changes

Software Version 8.10.6.02 Open Caveats

The following caveats apply to software version 8.10.6.02:

- **CSCsz04690**—The CSS does not look for and remove any of the headers that may be inserted as part of the **ssl-server** *number* **http-header client-cert** command. If you insert these headers prior to encryption before they arrived at the CSS and they were there after decryption, you could impersonate a different client, thus spoofing the client session. Workaround: If the client configures the **ssl-server** *number* **http-header prefix "Unique-"** command and the **"Unique-"** string is secret, the server looks for the "Unique-ClientCert-Subject-CN: CN=userY" header instead of the more generic "ClientCert-Subject-CN: CN=userY" header, therefore mitigating the exposure to spoofing.
- **CSCsz13210**—When the CSS stops passing SSL traffic, the syslog displays that multiple services are going down, however, the CSS continues to process all other HTTP traffic. Workaround: Reboot the CSS.
- **CSCtd01636**—Summary: An industry-wide vulnerability exists in the Transport Layer Security (TLS) protocol that could impact any Cisco product that uses any version of TLS and SSL. The vulnerability exists in how the protocol handles session renegotiation and exposes users to a potential man-in-the-middle attack.

  This advisory is posted at http://www.cisco.com/warp/public/707/cisco-sa-20091109-tls.shtml.

  Workaround: The CSS 11500 Series Content Services Switches are affected by this vulnerability with default configurations. However, the client authentication feature can be enabled as mitigation/solution. To enable or disable client authentication on a virtual SSL server, use the **ssl-server** <*number*> **authentication** command under the ssl-proxy-list.

  Note: By default, client authentication is disabled. After you enable client authentication on the CSS, you must specify a CA certificate that the CSS uses to verify client certificates.
- **CSCtd32718**—The CSS does not respond to ICMP packets with a a TTL of 1. Workaround: None.
- **CSCtd34926**—The CSS acknowledges POST data from the client but never sends it to the back-end server. Eventually, it resets from timeouts that are seen from the web server. Workaround: None.

Software Version 8.10.6.02 Resolved Caveats

The following caveats were resolved in software version 8.10.6.02:

- **CSCso12766**—When you configure the CSS for scripted keepalives and dynamically configure the scripted keepalive to go down through the **suspend** command and then bring it back through the **active** command, on rare occasions, the scripted keepalive would remain in the down state due to a race condition in which the CSS deletes the scripted keepalive process. Workaround: Reboot the CSS.
- **CSCsq62300**—Symptom: CSS does not validate remote SSL Server identity. Conditions: The CSS validates the remote SSL Server identity by determining it has a certificate issued by a recognized Certificate Authority. Workaround: None
- **CSCsu75564**—When a CSS uses arrowpoint cookies and the server retransmits 200 OK, the CSS may corrupt the retransmission packet by 52 bytes with arrowpoint-cookie injection and causes the connection to fail. Workaround: None.
- **CSCsu80676**—When you configure a service with a named (global) HTTP keepalive on the CSS and the keepalive fails, the **show keepalive** command displays it in the Keepalive Error field but the **show service** command does not. Workaround: None.
- **CSCsv06328**—When you configure the CSS with RIP on an interface and it receives an invalid RIP route of 0.0.0.0 with a non-zero subnet mask, the CSS reboots and generates a core dump within three minutes of receiving the bad RIP route. Workaround: None.
- **CSCsv12580**—When you configure the CSS for Layer 5 (L5) rules and the client TCP SYN contains the TCP Option Window Scale (WS), with a large configuration and a high traffic rate, the CPU could be higher than expected and may cause overall performance degradation. Workaround: The CSS now allows you to configure the ability to propagate the TCP Window Scale (WS) option to the back-end server through the **flow tcp-window scale** [**enabled | disabled**] command.
- **CSCsv21454**—During the last days of each month, the CSS incorrectly reports that the CRL has expired. The SSL module incorrectly converts day 30 and 31 of a month to day 1 and 2 of the next month. Workaround: Disable CRL expiration or client authentication, if it is not required.
- **CSCsv30030**—When you incorrectly configure OSPF on the redundancy-protocol link, the OSPF packet fails to transmit and the CSS displays the following log message:

  `IPV4-4: ospf_transmit: could not forward` *`ip_address`* `on interface` *`interface_ip circuit_number`*

  Each time this log message occurs, a small amount of system heap memory leaks. Over time, as the amount of free memory decreases, the CSS could eventually become unresponsive. Workaround: Remove OSPF from the redundancy-protocol VLAN.
- **CSCsv79835**—When you configure the CSS with virtual RADIUS authentication and then use the Denial of Service (DoS) Nessus scanner tool to scan the CSS, the CSS could stop successfully authenticating users through RADIUS and the users cannot log into the CSS. Workaround: Do not run the Nessus tool on the CSS in a working network.
- **CSCsw25443**—When a content rule is configured for Layer5 and the rule does not terminate SSL, the **flow tcp-window-scale disabled** command does not fully disable the propagation of the TCP Window Scale (WS) option to the server. Also the TCP WS option from the client may not have been correctly set in a spoofed TCP SYN to the server if it was not present in the original TCP SYN. Workaround: Configure the content rule as a Layer4 rule, or terminate SSL using an internal SSL module.

- **CSCsw47504**—Due to memory corruption, the CSS may reboot and generate a core dump. Workaround: None. You can now display corruption of the Memory Quick Pools that are of sizes 16, 32, and 48 bytes through the debug **xmask mpool set global 0x2** command. The CLI hint for the 0x2 option is Memory Small Pool.
- **CSCsw73978**—In a back-end SSL configuration while waiting to receive the server certificate, the SSL module becomes unresponsive and generates a core dump. The fix for CSCsm9935 in software release 8.20.2.04s incorrectly adds a delay in the certificate that is processing the SSL code. During the delay period when the SSL connection is closed, the module would reboot and generate a core dump. CSCsm99353 will be fixed differently with protection checks placed in for NULL (0x0) pointers when processing the server certificate. Workaround: None.
- **CSCsw75856**—When you configure the CSS with the **flow-disable** command and a source group and the UDP packet with a well-known source port hits the source group, it does not perform port address translation on source ports below 1024. Workaround: This behavior is now configurable through the source group [**no**] **portmap well-known** command. The default setting is enabled. To disable this setting, use the **no portmap well-known** command. To reenable this setting, use the **portmap well-known** command.
- **CSCsw79162**—When you configure the CSS11500 for HTTP keepalives, on rare occasions if the keepalive application attempts to pend a socket to read data on the network but the socket is closed, the CSS could generate a core dump due to a NULL socket pointer. Workaround: This resolved caveat is the final fix for a set of reboots related to the same issue for CSCso41083 and CSCsq99227.
- **CSCsx05640**—When you configure the CSS for a Layer 5 (L5) content rule and it receives an HTTP method POST with the HTTP header in one packet that is quickly followed by many packets of POST data or payload, it could fail to deliver all the data to the back-end server. The CSS Flow Manager (FM) application could incorrectly handle the POST and the data packet as a spanned content request and could cause the data to be mishandled. Workaround: Use less than 1-Gb connections in the network; a 100-Mb link does not exhibit this issue.
- **CSCsx33407**—When you configure SSL termination on the SSL module and it receives a TCP RST when the state of the TCP connection is LAST-ACK, in which the TCP RST would not be propagated in all cases, the SSL module could leave connections unnecessarily in the TCP CLOSE or CLOSE WAIT states. Workaround: None. A new **ssl process-rst-last-ack** command has been added to the CSS CLI. The **show ssl statistics** command in debug mode includes the new TCP resets recv in state last_ack, TCP recv SYN in state closed, and TCP Reprocess SYN in state closed counters.
- **CSCsx37430**—The SSL module becomes responsive due to a duplicate block free crash and the root cause could not be determined. Workaround: None. The SSL module code has been modified to correct the duplicate block free condition. The debug **show ssl statistics** command includes a new nicDuplicateBlockFree counter to display the count for this condition.
- **CSCsx40586**—If the CSS exceeds 828 days of uptime, SNTP may stop working. Workaround: None.
- **CSCsx41962**—OpenSSL 0.9.8i and earlier does not properly check the return value from the EVP_VerifyFinal function, which allows remote attackers to bypass validation of the certificate chain through a malformed SSL/TLS signature for DSA and ECDSA keys. This issue is documented in CVE-2008-5077. This product is affected by this vulnerability. All versions prior to the fixed system software are vulnerable. Workaround: None.
- **CSCsx43587**—When you configure a CRL URL, the CSS does not allow the % character because it accepts an unquoted string. You cannot configure a URL that contains a space, space character, or its URL encoded escape sequence (%20). Workaround: None. The SSL CRL URL string is now quoted text to allow for embedded % characters. For example, you can represent a SPACE (0x20) as "%20".

- **CSCsx44453**—When you configure the CSS11500 for IP static routes which overlap with a local VLAN subnet that was dynamically configured and the VLAN comes up, the CSS may generate a core dump due to the invalid static routes. Workaround: None. The CSS no longer generates a core dump due to the invalid static routes.
- **CSCsx50794**—When you configure the CSS11500 for compression, the CSS attempts to compress a page that had no Content-Length or Transfer-Encoding HTTP tag. This action results in the inability to know the length of time to compress the data and causes the CSS to send invalid compressed data to the server. You can display this condition in the Content-length or Transfer-Encoding counter through the debug **show ssl statistics compression** command. Workaround: None.
- **CSCsy01342**—When a Prism Buffer state transition error occurs due to an invalid field in the buffer, the CSS11500 could reboot and generate a core dump. Workaround: None. Further error checking was added to the CSS.
- **CSCsy21994**—When the Flow Manager (FM) or Flow Agent applications are processing an accounting report to clean up completed flows and the accounting report contains invalid data, the CSS11500 may reboot and generate a core dump. Workaround: None. Verification of the accounting report was added to prevent the reboot of the CSS.
- **CSCsy32611**—If you enter the debug **facet show sram_cnts** command on an SSL module slot number, the CSS11500 can generate a core dump. This command should not be allowed for the compression or noncompression SSL module. Workaround: Do not enter the command for the SSL module slot number. The CSS11500 now displays an error message if you enter the **facet show sram_cnts** commands on any type of SSL module.
- **CSCsy32925**—A pair of CSSs were configured for VIP/Interface redundancy and OSPF, and one of the redundant-interface addresses is configured with the **ospf advertise** command. If you run the commit_vip_redundancy script on the master CSS, the **clear running-config** command occurs on the backup CSS that is configured for the redundant VIP and it immediately reboots. The reboot occurs when the CSS attempts to delete a route entry for the redundant interface that was being advertised by OSPF. Workaround: None. Preventive code was added for this condition.
- **CSCsy38035**—When you configure the CSS for VIP/Iinterface redundancy with multiple virtual routers and the CSS exceeds 828 days of uptime, the master CSS stops sending VRRP packets and the backup CSS transitions to become the master CSS for some or all of the VIPs. This issue could cause a duplicate mastership issue and network connectivity problems. Workaround: Reboot both CSSs.
- **CSCsy57143**—The Denial of Service (DoS) LAND Attacks counter displayed by the debug **facet show sram_cnts** *slot subslot* command does not increment and is inconsistent with the counter that is displayed by the **show dos** command. Workaround: None. The LAND Attacks counters are now consistent in the **show dos** and **facet show sram_cnts** command.
- **CSCsy86356**—Introduced by the fix for CSCsk43344 in software release 8.20.201, when you configure two content rules which were identical except one had a URL string that contained a port number (:*port*) and activate the rule without the port number first, activating the rule with the port number fails and the CSS displays the "%% Content already exists" error message. The URL check process stops at the ":" after a same URL length is activated.

  For example:

  - content rule1—url "//www.example.com:443/secure/app/"
  - content rule2—url "//www.example.com/internal/app/"

  If content rule1 is activated after content rule2, it fails with the "%% Content already exists" error which is incorrect. Workaround: Activate the URL with the :*port* first. In this example, suspend the content rule 2. Then, activate content rule 1. Finally, activate content rule 2.

- **CSCsz05578**—When the fix for CSCek57234 went into in software release 8.20.0.01, it added the ability to configure the TCP MSS that the SSL module for both SSL termination and back-end SSL sends to the server. Workaround: The following commands allow the SSL module to set its TCP MSS lower than 1460 were added to the CSS CLI:
  - [**no**] **ssl-server** *number* **tcp server advertise-mss** *tcp_mss*
  - [**no**] **backend-server** *number* **tcp server advertise-mss** *tcp_mss*

  For the *tcp_mss* argument, enter an integer from 200 to 1460. The default is 1460.
- **CSCsz07676**—When the CSS exceeds 828 days of uptime and then one of the interface links flaps, the **show interface** command displays the incorrect time of the last link transition in the Last Change field. Workaround: None.
- **CSCsz10540**—When you configure the CSS for SSL termination, the SSL module cannot handle an SSL Client Hello that spanned two packets and the first packet only had 5 bytes which is exactly the size of the SSLv3 Record Header. The SSL module sends a TCP RST which is incorrect. Workaround: Do not have client hello span multiple packets. The debug mode **ssl statistics ssl** command includes the new Bad message type in ssl23_get_client_hello counter.
- **CSCsz65488**—When the CSS is using a SSL-C module for compression, if you enable services with and without compression and apply them to the same SSL-proxy list, the last service on the list with compression has its attributes applied to all the services including the services without compression. Workaround: This configuration is not allowed and is now prevented with the error message:

  ```
  Cannot have compression and non-compression services using the same ssl-proxy-list.
  ```

  Configure two separate SSL-proxy lists: one list with compression services and another with noncompression services.
- **CSCsz66388**—When you configure the CSS11501 with ports 7 and 8 as the ISC ports and if the port fails, the port number displayed in the SNMP Trap Enterprise:Inter-Switch Communications Lifetick Failure: *slot*/*subslot* is incorrect; port 7 (e7) is displayed as 1/15 and port 8 (e8) is displayed as 1/16. Workaround: None.
- **CSCsz69456**—The **show reboot-reason** command displayed Primay instead of Primary. Workaround: None.
- **CSCsz75285**—When you configure a content rule with the **advanced-balance ssl** command, the **show rule** command does not display the statistics that are related to using the SSL Layer 4 (L4) hash value rather than the SSL Session ID. Workaround: None. The SSL sticky Total SSLv2 Hits, Total SSLv3 Hits, and Total SSL L4 Hits counters were added to the following commands:
  - **show rule** *owner_name rule_name* **all**
  - **show rule** *owner_name rule_name* **statistics**
- **CSCta04885**—Symptom: Information is inserted in the client certificate header when a Carriage Return-Line feed (CRLF)/CRLF terminator is received in the HTTP header. This behavior is according to the specification. When a Line Feed (LF)/LF terminator is received, the client certificate headers are not inserted. Conditions: A LF/LF terminator must be sent by the client. Workaround: Ensure that you are using a client that sends CRLF/CRLF as a terminator.
- **CSCta06871**—When you configure the **protocol** and **port** commands within a content rule, the **show** {**running** | **startup**}-**config** command may not display these commands in the order in which they were configured. Workaround: None.

- **CSCta27379**—When you configure the **ip management route** command, the CSS sends the syslog packet to the old source address. The command change does not take effect for syslogd traffic until the CSS is rebooted or the CSS Management Port route is bounced. The CSS has been modified allowing a new management route to take effect for syslogd traffic immediately after its configuration. Workaround: Reboot the CSS or remove the logging host command setting from the configuration and then readd it.
- **CSCta49265**—You cannot configure the CSS for duplicate content rules based on the Virtual IP address (VIP), Port, Protocol and URL. The code changes for CSCsk43344 in software release 8.20.2.01 and CSCsy86356 in software release 8.20.2.08s broke the duplicate rule checking and may cause the CSS to generate a core dump when you use the **no content** command to delete or suspend one of the duplicate rules. Workaround: Reconfigure the CSS without duplicate content rules.
- **CSCta60140**—When you configure the CSS for SSL termination and the **header-insert-per-request** command, if an HTTP POST is received, the CSS properly inserts the configured HTTP header at the end of the HTTP header but it also incorrectly inserts it in the POST data. The SSL module misinterprets the actual data in the POST payload at the start of a new HTTP content request. Workaround: None. The HTTP header insertion code has been modified to do additional verification to be certain that the module finds a new HTTP content request header instead of data that might look like an HTTP header.
- **CSCta85214**—When you configure the CSS for Layer 5 (L5) content rules and it processes a spanned content request and sends it to the back-end server, if the TCP window goes to zero and a CSS interface goes down before the server TCP ACK is received, the CSS could generate a core dump. Workaround: None.
- **CSCtb05310**—The **show service** command displays a negative number in the Total Reused Conns field when the counter exceeded 0x7fffffff (2147483647). Workaround: None.
- **CSCtb05442**—When you configure the CSS for SSL initiation or back-end SSL, if the server requests DSA certificate verification, the CSS may reboot due to a NULL pointer reference. The CSS does not support DSA server side certificate verification. Workaround: None. The CSS now properly closes down the SSL connection.
- **CSCtb45641**—When you configure the CSS for a scripted keepalive, if the returned keepalive response is greater than 10,000 bytes of data, on rare occasions, the CPU Utilization becomes 100% and the CSS eventually reboots due to an ONDM Lifetick failure because of buffer depletion. The first character of the search string was in the 10,000th character received and the second socket buffer did not contain enough characters to reach the search string length. The CSS fails to exit the search and release the CPU. Workaround: Do one of the following:
  - Change the scripted keepalive to point to a static web page.
  - Remove the scripted keepalive from the service configuration and use a TCP or ICMP keepalive temporarily.
- **CSCtb50496**—The CSS is introducing a global system variable concept. Currently, the only way for CSS scripts to communicate is by reading and writing to a common file on the CSS disk. This behavior can place stress on the disk especially if those scripts are used within scripted keepalives and are executed frequently. Workaround: None. The following commands have been added to the CSS CLI:
  - [**no**] **systemVariable** *string*
  - **show systemVariable**
- **CSCtb99618**—When you configure the CSS for SSL termination and HTTP header insertion, if the HTTP method header is terminated with a line feed (LF) sequence and the LF terminator is split across two Ethernet packets, the CSS SSL module may fail to properly insert the configured HTTP header string. Workaround: None.

- **CSCtc23135**— When the CSS is in a Session Level Redundancy (SLR) configuration in which messages are sent between the two CSSs to replicate connections, if the receiver of the message from the peer encounters an expected error, it may not free the buffer and the CSS reboots with an Online Diagnostic Monitor (ONDM) Lifetick failure. Workaround: None.
- **CSCtc36048**—When a Secure Shell (SSH) connection is made to the local interface and the moduli file stored in c:/CertStore/ssh and c:/*image*/info directories is corrupted, the CSS reboots. This file seeds the random number function used over the SSH connection. The CSS does not detect the error and continues the connection with an alternate random function. Workaround: Reinstall the image on disk and unpack it for an uncorrupted version of the moduli file.
- **CSCtc73736**—The CSS incorrectly logs the following syslog message:

  ```
  SYSSOFT-4: SysTimerHandler: Function issue - 0x809f2650 takes too long (8)
  ```

  Workaround: None.
- **CSCtc89684**—If the c:/CertStore/filedb file is not present because the CSS disk has been reformatted or the database has not been created, the CSS incorrectly tries to unlock the file and the following error appears in the boot.log or sys.log file:

  ```
  NETMAN-7: Unlock Failed.
  ```

  Workaround: None.
- **CSCtd07288**—When you configure the CSS for SSL termination with the **ssl-server** *number* **http-header insert-per-request** command and a malformed HTTP GET HTTP/1.0 content request is received, the status for the SSL module displayed by the **show chassis** command transitions to Bad. Also the module stops processing any traffic. This problem was caused by the fix for CSCta60140 in software release 8.10.5.10s. Workaround: Reboot the CSS.

Software Version 8.10.6.02 Command Changes

Table 4 lists the commands and options that have been added or changed in software version 8.10.6.02.

Table 4 CLI Commands Added in Version 8.10.6.02

Mode	Command and Syntax	Description
Exec	**show rule** *owner_name rule_name* **all** **show rule** *owner_name rule_name* **statistics**	Per CSCsz75285, these commands now include the following new counters: • Total SSLv2 Hits—Number of SSL Layer 4 (L4) Hash that were inserted into the sticky database • Total SSLv3 Hits—Number of SSL Session Id Hash that were inserted into the sticky database • Total SSL L4 Hits—Number of SSL L4 Hash that were used to select the SSL Service
Exec	**show systemVariable**	Per CSCtb50496, this new command allows you to display the global system variable (systemVariable) as configured with the global configuration mode **systemVariable** command.

Table 4 CLI Commands Added in Version 8.10.6.02 (continued)

Mode	Command and Syntax	Description
Global	**flow tcp-window-scale** [**enabled** | **disabled**]	Per CSCsv12580, the propagation of the TCP Window Scale (WS) option to the backend server when a TCP client SYN that contains the option hits a Layer 5 rule is now configurable through the new **flow tcp-window-scale** [**enabled** | **disabled**] command. The keywords are as follows: • **disabled**—Disables the propagation of the TCP WS option to the backend server. • **enabled**—Enables the propagation of the TCP WS option to the backend server. By default, this behavior is enabled on the CSS.
Global	**ssl crl-record** *crl_name* **"***url***"** *sign_cert hours*	Per CSCsx43587, the *url* argument is now a quoted string to allow the inclusion of spaces and the % character.
Global	**ssl process-rst-last-ack** **no ssl process-rst-last-ack**	Per CSCsx33407, this new command allows you to configure the CSS to process RSTs when they are received in TCP LAST-ACK state. By default, the CSS ignores RSTs. To reset the default setting, use the **no** form of this command.
Global	**systemVariable** *string* **no systemVariable**	Per CSCtb50496, this new command allows you to configure and save a global system variable. Previously, the only way for CSS scripts to communicate was by reading and writing to a common file on the CSS disk. This behavior could place stress on the disk especially if those scripts are used within scripted keepalives and executed frequently. For the *string* argument, enter quoted text with a maximum of 128 characters. Use the **no** form of this command to remove the global system variable.
Group	[**no**] **portmap well-known**	Per CSCsw75856, this new command allow you to configure NAT for UDP flow-disable well known ports. Previously, CSCsq59829 allowed the CSS to perform NAT on well-known source ports when you configured the CSS with the **flow-disable** command and a source group, and the UDP packet with a well known source port hits the source group. Now, this behavior is configurable through the **portmap well-known** command. By default, this behavior is enabled. To disable this behavior and not allow NAT on UDP flow-disable well known ports, use the **no portmap well-known** command. Use the the **portmap well-known** command to reenable this behavior.

Table 4 CLI Commands Added in Version 8.10.6.02 (continued)

Mode	Command and Syntax	Description
SSL-proxy-list	**backend** *number* **server-cert-verify-str** *string* **no backend** *number* **server-cert-verify-str**	Per CSCsq62300, this new command allows you to configure verification of the server certificate CN subject. This command applies to back-end SSL only. By default, the CSS does not check the subject CN of the certificate returned by the server side to validate the identity of the remote SSL server. The arguments are as follows: • *number*—Index number for the SSL server. • *string*—Subject CN of the server return certificate. Enter quoted text string with a maximum of 31 characters that exactly matches the Subject CN in the certificate. Use the **no** form of this command to disable the CSS from verifying the subject CN of the certificate returned by the server side.
SSL-proxy-list	**backend-server** *number* **tcp server advertise-mss** *tcp_mss* **ssl-server** *number* **tcp server advertise-mss** *tcp_mss* **no backend-server** *number* **tcp server advertise-mss** **no ssl-server** *number* **tcp server advertise-mss**	Per CSCsz05578, the **advertise-mss** *tcp_mss* option allows you to set the SSL module TCP MSS lower than 1460. For the *tcp_mss* argument, enter an integer from 200 to 1460. The default is 1460. Use the **no** form of these commands to reset the default MSS to 1460.

Software Version 8.10.5.03 Open Caveats, Resolved Caveats, and Command Changes

The following sections contain the open caveats, resolved caveats, and command changes in software version 8.10.5.03:

- Software Version 8.10.5.03 Open Caveats
- Software Version 8.10.5.03 Resolved Caveats
- Software Version 8.10.5.03 Command Changes

Software Version 8.10.5.03 Open Caveats

The following caveats apply to software version 8.10.5.03:

- **CSCso44058**—When spanning tree is enabled on the CSS, the Port Cost output displayed by the **show bridge status** command becomes -1 after rebooting the CSS. Workaround: Disable spanning tree on the CSS.
- **CSCsu75564**—When a CSS is using arrowpoint cookies and the server retransmits the 200 OK, the CSS may corrupt the retransmission packet by 52 bytes with the arrowpoint-cookie injection causing the connection to fail. Workaround: None.
- **CSCsu97359**—When a page with a large file size is requested by a Firefox client, if the page is loading when a Netscape client requests it, the CSS can reboot. Workaround: None.
- **CSCsv21454**—During the last few days of each month, the CSS incorrectly reports that the CRL is expired. The SSL module incorrectly converts day 30 and 31 of a month to day 1 and 2 of the next month. This bug is related to fixes for CSCsm68656 and CSCso29536 and also applies to the expiration of client authentication certificates. Workaround: Disable CRL expiration or client authentication, if it is not required.

Software Version 8.10.5.03 Resolved Caveats

The following caveats were resolved in software version 8.10.5.03:

- **CSCsh91610**—When a CSS is configured with a Layer 5 (L5) content rule and receives an HTTP content request that has a URI that exceeds 252 characters, the CSS now attempts a best effort content rule match instead of sending a TCP RST for this connection.
- **CSCsj99178**—When a CSS is configured with the DNS server functionality, it reboots when it receives a very long domain name.
- **CSCsk18254**—When a content rule contains 64 services or 24 location services, the CSS may reboot.
- **CSCsk43018**—A CSS may hang when you configure it with router-discovery and configure the management interface.
- **CSCsk80734**—If a CSS is configured for RADIUS authentication and there is repeated abnormal termination of the connection to the RADIUS server, the CSS may build up RADIUS tasks and eventually reboot.
- **CSCsk92868**—The Windows Vista Operating System (OS) can use the TCP Window Scale (WS) option in the TCP SYN. The TCP WS option is not propagated to the back-end server and this may cause the application to fail.

- **CSCsk95448**—An SSL-proxy list configured with all-cipher-suites and client authentication may cause the CSS to reboot.
- **CSCsl23853**—If you enter the **traceroute** command repeatedly from the CSS command line interface (CLI), the buffers may be lost and the CSS may reboot.
- **CSCsl35996**—If a CSS is configured for RIP on the interface and invalid routes were received, it may reboot.
- **CSCsl48284**—When a CSS is configured with an L5 content rule, it spoofs the backend connection causing a route change after the TCP SYN is sent. If a different CSS port receives the returned TCP SYN/ACK, the CSS may reboot.
- **CSCsl57105**—When a CSS is configured for OSPF and receives a bad OSPF packet with an incorrect packet type, it may reboot.
- **CSCsl57690**—When a CSS with an SSL module is configured for SSL termination with an L5 content rule and the module does not receive the SSL Client Hello, the CSS may incorrectly forward the retransmitted internal TCP SYN/ACK packet between the SSL module and the CSS to the client.
- **CSCsl61242**—When a CSS containing an SSL module is configured with the **ssl associate rsakey** command with a corrupted PKCS8 private key, the CSS may reboot.
- **CSCsl72651**—A CSS with an SSL module attempts to validate intermediate certificates if it cannot verify the root certificate against the configured CA certificates.
- **CSCsl85753**—When a CSS is configured with HTTP keepalives, it may incorrectly access a closed file descriptor and may reboot.
- **CSCsm11230**—When a CSS is configured with four or more VLANs and multiple hardware-applied ACLs, reconfiguring the ACLs may cause the CSS to apply the clauses in the incorrect order.
- **CSCsm39951**—When a CSS with an SSL module is processing a Finished message from the client at the same time it receives an application data packet from the client and the connection is in FIN_WAIT_1 state, the CSS may reboot.
- **CSCsm50650**—When a CSS with an SSL module is configured for HTTP-header insert, the CSS may become unresponsive.
- **CSCsm53153**—When the **copy ftp … startup startup-config** command fails, the CSS may clear the local startup-config file from the disk.
- **CSCsm58924**—When a CSS is configured with ACLs and running more than 414 days, dynamically configuring an ACL may cause the **apply** command to appear in the middle of the running-config ACL section. If you reboot the CSS, the ACL may not function properly.
- **CSCsm62595**—When a CSS is configured with an L5 rule with a URL of the //www.ab*.com/*xx*/bar* form, removing or suspending this rule may cause the CSS to reboot.
- **CSCsm68656**—When a CSS with an SSL module is configured for client authentication, it may incorrectly calculate the calendar date during the authentication of a certificate and then improperly reject the certificate as an expired certificate.
- **CSCsm73591**—When a CSS with an SSL module is configured for SSL termination and the TCP Initial Sequence Number (ISN) is too predictable, security warnings may occur on network security scanners.
- **CSCsm84515**—The **copy sftp … boot-image** command has been enhanced to add support for the .adi image. Previously, the command supported only the .adi-gz image.
- **CSCsm97211**—When a CSS is configured for DFP, it now accepts malformed DFP packets that other Cisco devices accept.

- **CSCsm97273**—When a CSS is configured with the **dns-server** command and the management port subnet to the circuit address of the CSS receives a DNS request, the CSS may reboot.
- **CSCsm99353**—When a CSS with an SSL module is configured for client authentication and you modify the date through the **clock date** command, the new date does not take effect on the SSL module until the second authentication request.
- **CSCsm99462**—When a CSS is configured with the **advanced-balance arrowpoint-cookie** and **arrowpoint-cookie expiration** commands, the CSS may insert an invalid expiration date on a leap day.
- **CSCso36251**—When a CSS is configured with a URQL containing a domain entry of the *domain*:*port* form that is associated with a content rule, suspending the URQL or the content rule may cause the CSS to reboot.
- **CSCso41083**—A CSS configured with HTTP GET or HEAD persistent keepalives may cause the CSS to reboot.
- **CSCso53400**—When a CSS is configured with a content rule containing a header-field rule and a specific URL with no wildcard, for example "/cisco," the CSS may load balance to the wrong server.
- **CSCso53528**—When a CSS with an SSL module is configured for client authentication, the CSS rejects a client connection that does not provide a client certificate. To allow or reject a client connection that does not provide a client certificate, the **ssl-server** *number* **no-client-cert ignore|reject** command in ssl-proxy-list configuration mode has been added to the CSS CLI. The default setting is to reject the connection.
- **CSCso53545**—When a CSS with an SSL module is configured for client authentication and client certificate header insert, the **ssl-server** *number* **no-client-cert ignore** command allows the client connection without producing a client certificate. To insert specified text into the Subject-CN field when the client does not provide a certificate, the **ssl-server** *number* **http-header no-client-cert** *text* command in ssl-proxy-list configuration mode has been added to the CSS CLI.
- **CSCso48009**—When a CSS service is configured with the **keepalive type http non-persistent** command and the service goes down, the Keepalive Error: field displayed by the **show service** command shows OK instead of the actual reason for the failure.
- **CSCsq02268**—Some CSS CLI commands (for example, the **clear ip statistics** command) may cause a small memory leak. If the command is repeated continuously, the CSS may run out of memory and reboot.
- **CSCsq37677**—When a CSS is configured with multiple content rules with the same VIP address and the services go down in a content rule, the CSS may not respond to ping requests for that VIP address.
- **CSCsq59829**—When a CSS is configured for flow-disabled ports and a source group, a UDP packet with a well-known source port hits the source group and may cause the CSS to incorrectly NAT the well-known source port.
- **CSCsq62191**—When you enter the CSS **show** command with the MORE option enabled and the MORE buffer has one free byte, entering the **/** character for a forward search may cause the CSS to reboot.
- **CSCsq69016**—When a CSS is configured with scripted keepalives, a keepalive script that encounters an error in the process of cleanup causes a resource to be freed twice and causes the CSS to reboot.

- **CSCsq72608**—When a CSS is configured for VIP and interface redundancy with a large configuration of redundant VIPs and interfaces, an interface transition failover may cause the backup CSS to incorrectly send a Gratuitous ARP (GARP) for a VIP address. The debug mode **show redundant-interface** command displays the arpMasterSent flag with the incorrect setting of True and the log file contains Duplicate IP Address errors.
- **CScsq73004**—When a CSS with an SSL module is configured with a content rule configured with a service type of **ssl-accel**, **ssl-accel-backend**, or **ssl-init**, and a TCP SYN with the TCP Window Scale (WS) option hits the content rule, the SSL module performance becomes very slow.
- **CSCsq78562**—When a CSS with an SSL module is configured with a front-end and back-end SSL content rule and ACLs, an SSL flow is back-end remapped causing the CSS to not match the appropriate ACL and reject the SSL connection.
- **CSCsq80585**—When a CSS with an SSL module receives a Client Hello that is divided into two packets, the CSS may reset the SSL connection.
- **CSCsq94788**—When a CSS with an SSL module is configured for SSL termination and client authentication and receives the client Certificate verify message, it may improperly report a decrypt error and reset the SSL connection.
- **CSCsq99227**—When a CSS is configured with HTTP GET or HEAD persistent keepalives and the keepalive closes down, the keepalive task may continue to use a freed resource causing the CSS to reboot.
- **CSCsr05163**—When the following log message exceeds the host entries limit of 5,120, it may display the wrong IP address.

  ```
  Ipv4UnicastSubmit: exceeded max support 5120 host entries for x.x.x.x
  ```

- **CSCsr20304**—When a CSS is configured for SNMP and the **ip management route** command for SNMP incoming queries and outgoing responses on the management port, the CSS may send the SNMP responses out the front panel instead of the management port.
- **CSCsr48042**—When a CSS is configured with the global configuration mode **flow-state** *port_number* **udp flow-disable**|**flow-enable nat-enable** command and a source group, the CSS may improperly NAT all the UDP packets.
- **CSCsr53577**—When a CSS with an SSL module is configured with the **ssl-server** *number* **authentication enable**, **ssl-server** *number* **no-client-cert ignore**, **ssl-server** *number* **http-header client-cert**, and **ssl-server** *number* **http-header no-client-cert** *text* commands, and the client sends a NULL certificate, the CSS may reboot.
- **CSCsr58467, CSCso76023**—When a CSS is authenticating a Telnet session and terminates the session the same time it checks the Telnet idle session timeout, the console and all management functions may hang due to a semaphore deadlock.
- **CSCsr61464**—The number of configured IP management routes has increased from 8 to 64.
- **CSCsr67391**—An SSL performance decrease has been corrected as related to the fix for CSCsq73004.
- **CSCsu35876**—When a CSS is configured for an L5 content rule and a persistent connection, the CSS may mishandle the original POST retransmission causing the connection to fail and the client and server to TCP RST.
- **CSCsu38563**—When you add a new static route through SNMP or the **ip route** command to a CSS that has a full routing table, the CSS does not add the route to the table. It incorrectly adds the route to the running-config file and does not display errors for the insertion failure. If you subsequently remove this route through SNMP or the **no ip route** command, the CSS reboots.

- **CSCsu67098**—When the size of the SSL Client Hello Message length randomly increases in length by adding extra bytes within the data portion, the CSS SSL module reboots. This issue was related to the fix for CSCsq80585.
- **CSCsu82159**—To allow a CSS to reset an L5 connection if the arrowpoint cookie maps to a down service, the **advanced-balance arrowpoint-cookies** command now works with the **balance** and **reject** options of the **sticky-serverdown-failover** command.
- **CSCsu86628**—When a CSS is configured with encrypted keepalives through the service configuration mode **keepalive type http encrypt** command, management traffic from the CSS (for example, a **socket connect** or **copy ftp** command) may fail because the CSS incorrectly sends the management traffic from the source IP port to the SSL module as if it is an encrypted keepalive.

Software Version 8.10.5.03 Command Changes

Table 5 lists the commands and options that have been added or changed in software version 8.10.5.03.

Table 5 CLI Commands Added in Version 8.10.5.03

Mode	Command and Syntax	Description
Service	**advanced-balance arrowpoint-cookies**	Per CSCsu82159, this command now works with the default **balance** option and the **reject** option of the **sticky-serverdown-failover** command.
Global	[**no**] **flow tcp-window-scale** [*value*]	When the client uses the TCP Window Scale (WS) option in the TCP SYN, the CSS did not propagate the TCP WS option to the backend server and this may cause the application to fail. CSCsk92868 adds support for the TCP WS option. A TCP connection that hits a Layer 5 content rule now parses the TCP WS option from the initial TCP SYN. If the TCP WS option is present in the original TCP SYN from the client, when the backend server connection is spoofed, the CSS inserts the TCP WS option in the initial TCP SYN to the backend server. The WS shift count value inserted into the server TCP SYN is the same value received from the client. The new **flow tcp-window-scale** *value* command allows you to modify the WS shift count and the CSS inserts the TCP WS option in the TCP SYN/ACK sent back to the client. By default, the window scale is zero (0) and is not sent sent back to the client. The *value* argument is the number in shift count that the true receive window size is left shifted when the CSS sends this option. Enter an integer from 0 to 14. The default is 0. To reset the TCP window scale shift count to default of 0, use the **no flow tcp-window-scale** command.
Global	**ip management route**	Per CSCsr61464, the number of routes configured with this command increased from 8 to 64.
Global	[**no**] **restrict outgoing-ftp|outgoing-tftp|outgoing-sftp**	Per CSCsm32522, the new **outgoing-ftp|outgoing-tftp|outgoing-sftp** keywords have been added to restrict outgoing FTP, TFTP and SFTP access, respectively.

Table 5 CLI Commands Added in Version 8.10.5.03 (continued)

Mode	Command and Syntax	Description
SSL-proxy-list	**ssl-server** *number* **http-header no-client-cert** *text*	Per CSCso53545, the new **no-client-cert** *text* option inserts the specified text into the Subject-CN field when the client does not provide a certificate.
	ssl-server *number* **no-client-cert** [**ignore**|**reject**]	Per CSCso53528, this command allows or rejects a client connection that does not provide a client certificate. The default setting is to reject the connection.

Software Version 8.10.4.01 Open Caveats, Resolved Caveats, and Command Changes

The following sections contain the open caveats, resolved caveats, and command changes in software version 8.10.4.01:

- Software Version 8.10.4.01 Open Caveats
- Software Version 8.10.4.01 Resolved Caveats
- Software Version 8.10.4.01 Command Changes

Software Version 8.10.4.01 Open Caveats

The following caveats apply to software version 8.10.4.01:

- **CSCsj92088**—The CSS loses network connectivity due to internal network buffer exhaustion.
- **CSCsj99178**— The CSS reboots when it receives a very long domain name.
- **CSCsk00135**—The CSS reboots when it uses an expired CRL (Certificate Revocation List) file.
- **CSCsk05982**—The CSS reboots when you attempt to FTP to it.
- **CSCsk18254**—A content rule that contains 64 services causes the CSS to reboot.
- **CSCsk43018**—A CSS may hang when you configure it with router-discovery and configure the management interface.
- **CSCsk80734**—RADIUS authentication may cause a task buildup, which eventually causes the CSS to reboot.
- **CSCsk87217**—A CSS may reboot when attempting to transmit a DNS packet.
- **CSCsk92868**—The Windows Vista Operating System (OS) can use the TCP Window Scale (WS) option in the TCP SYN. The TCP WS option is not propagated to the back-end server and this may cause the application to fail.
- **CSCsk95448**—An ssl-proxy-list configured with all-cipher-suites and client authentication may cause the CSS to reboot.
- **CSCsl21429**—The CSS may reboot when you enter the **enable** command.
- **CSCsl23853**—The CSS may experience a traffic outage due to the SCM Buffer Pool 1 Buffers being held by tImmRx.

Software Version 8.10.4.01 Resolved Caveats

The following caveats were resolved in software version 8.10.4.01:

- **CSCsi27491**—When an SSHv2 client makes a connection to the CSS and SSH terminates the connection, a file descriptor is closed twice. This may cause the CSS to reboot.
- **CSCsi34222**—When the CSS is configured for redundancy and the **ip advanced-route-remap** command is configured, a connection is routed through the master CSS. The backup CSS takes over mastership, and the routed flow gets garbage-collected on the original master CSS. The original CSS takes over mastership again, and this causes the routed flow to fail.

- **CSCsi43711**—When the CSS is configured for SSL header insert, it would incorrectly insert the header at the end of the packet instead of at the end of the header.
- **CSCsi62218**—CSS core dumps after receiving malformed snmp packets. An attacker would need to know the community string to successfully exploit this vulnerability. Workaround: While there are no workarounds to prevent this issue from happening, unauthorized snmp access to the system can be filtered.

  Infrastructure Access-Lists (iACL): Although it is often difficult to block traffic transiting your network, it is possible to identify traffic which should never be allowed to target your infrastructure devices and block that traffic at the border of your network. Infrastructure ACLs are considered a network security best practice and should be considered as a long-term addition to good network security as well as a workaround for this specific vulnerability. The white paper entitled "Protecting Your Core: Infrastructure Protection Access Control Lists" presents guidelines and recommended deployment techniques for iACLs: http://www.cisco.com/warp/public/707/iacl.html
- **CSCsi80599**— Running the clock time command may cause a file descriptor leak. The CSS may reboot if 256 file descriptors are leaked.
- **CSCsj06141**—When a CSS was configured with an ssl-proxy-list with **backend-servers**, the ssl-proxy-list was associated with multiple services of type **backend-ssl**. Activating the ssl-proxy-list may result in the display of the following error message: "%% Service's SSL proxy list contains a duplicate server."
- **CSCsj07250**—When the SSL module was processing a client hello and it received an application data packet instead of the client hello, it caused the CSS to reboot.
- **CSCsj18667**—If the CSS was configured with scripted keepalives and the scripts were using the **socket waitfor** command, a script failed and the socket was freed, but the script still used the socket. This caused memory corruption and caused the CSS to reboot.
- **CSCsj71697**—The offdm and technician username and password is a maximum of 15 characters.
- **CSCsj78548**—When you are connected to the CSS using the GUI, an illegal SSL disconnect sequence may cause the CSS to reboot.
- **CSCsj88034**—A new command was added to enable you to configure a second Certificate Revocation List (CRL) on an SSL server. The new command is:

  ```
  ssl-server <number> crl2 <name> [expiration-enabled|verification-enable]
  ```
- **CSCsk09805**—When the CSS is configured with Layer 4 and Layer 5 content rules configured to use the same VIP, and you suspend the Layer 4 content rule, the Layer 5 content rule may stop working.
- **CSCsk24574**—If the BANNER variable is set to NULL, it may cause a volume descriptor leak. When 32 of these volume descriptors are leaked, the disk could appear to be unmounted.
- **CSCsk43344**—When the CSS has two content rules configured with similar content rule domain names, (such as, the domain name URLs are of the form "//domain:port"), the only difference in the content rule configuration is the port value in the URL. If the length of the URLs was the same, the CSS considers the rules to be duplicates. Suspending one of the rules may cause the CSS to reboot.
- **CSCsk46813**—Issuing the **copy ssl sftp** command or the **copy sftp** command may cause a file descriptor leak.
- **CSCsk53888**— Running the **show cdp** command may cause the CSS to reboot even if CDP is not configured on the CSS.

- **CSCsk73631**—If the CSS contains an SSL module and an SSL connection is being shut down using a "Close Notify" alert at the same time a data packet is received for that connection, the SSL module may be reset.
- **CSCsk74674**—When an SSL connection is closing down, a race condition existed where the extra SSL Proxy Information (SPI) structure may be freed and then accessed by the closing functions. This results in the SSL module rebooting with a core dump due to a NULL pointer reference.

Software Version 8.10.4.01 Command Changes

Table 6 lists the commands and options that have been added or changed in software version 8.10.4.01.

Table 6 CLI Commands Added in Version 8.10.4.01

Command and Syntax	Description
no arp duplicate-ip-garp-response	This global command enables the CSS to respond to a GARP for an IP address it owns. This feature is disabled by default. (requested by CSCek68199)
ssl-server *number* **crl2** *name* [**expiration-enabled**|**verification-enable**]	This command enables you to configure a second Certificate Revocation List (CRL) on an SSL server within an ssl-proxy-list.

Software Version 8.10.3.01 Open Caveats, Resolved Caveats, and Command Changes

The following sections contain the open caveats, resolved caveats, and command changes in software version 8.10.3.01:

- Software Version 8.10.3.01 Open Caveats
- Software Version 8.10.3.01 Resolved Caveats
- Software Version 8.10.3.01 Command Changes

Software Version 8.10.3.01 Open Caveats

The following caveats apply to software version 8.10.3.01:

- **CSCek60702**—Denial of Service (DoS) anomalies require further investigation.
- **CSCek65804**—HSE fails to connect to the CSS over a secure (443) port. Workaround: Reboot the CSS.
- **CSCek69084**—Backend SSL client authentication fails with certain servers.
- **CSCsi27336**—The CSS stops functioning when doing a rcmd task. This issue occurs when you use **rcmd** to display a large file.
- **CSCsi43711**—The CSS may insert an HTTP header at the end of a packet instead of at the headers.
- **CSCsi44835**—The CSS G-arp issue occurs after a failover when you configure the **ip redundancy master** command.

- **CSCsi62218**—The CSS performs a core dump during codenomicon testing for SNMPv2.
- **CSCsi66718**—The CSS stops functioning when you run an snmpwalk.

Software Version 8.10.3.01 Resolved Caveats

The following caveats were resolved in software version 8.10.3.01:

- **CSCek57080**—When the CSS1 was configured for Session Level Redundancy (SLR/ASR) and a content rule with **redundant-index** and **advanced-balance arrowpoint-cookie** configured, it was observed that at a certain flow rate, the redundant CSS would start logging errors about the Flow Manager (FM) Peer Message Queue receiver task being overwhelmed. This was due to the ASR ARPT Cookie Sequence Number Update messages that were sent to the redundant peer.One message for each HTTP method was received on the persistent HTTP connection that did not contain a ARPT cookie, so for certain types of traffic (proxy for example), the number of these messages may overwhelm the CSS ASR peer.
- **CSCek55371**—When the CSS was configured with a custom scripted keepalive that did File Input/Output (IO) using the redirect characters) ">" and/or ">>" within the keepalive script and the script was configured using the keepalive type script command on multiple services, over time the CSS would leak file descriptors and after 32 descriptors were lost that pointed to files on the CSS disk (of the form c:/sg0610420s/log/sys.log), the CSS disk would unmount and a reboot was needed to clear. Do not use this type of custom keepalive script. Even with this DDTS fix, this type of custom script may not always work properly as a keepalive script. However the CSS disk will no longer unmount unexpectedly.
- **CSCek56722**—When the CSS was configured for UDP fragment support and UDP flow-disable using the **udp-ip-fragment** and **flow-state udp flow-disable nat-enable** commands, any UDP fragments whose UDP port also matched the entry in the flow-state table would fail to be forwarded through the CSS properly.
- **CSCek57074**—This DDTS is included in Cisco Security Response "Multiple vulnerabilities in OpenSSL library" published at: http://www.cisco.com/warp/public/707/cisco-sr-20061108-openssl.shtml.

  Summary-This is the Cisco PSIRT response to the multiple security advisories published by The OpenSSL Project. The vulnerabilities are as follows:

  - RSA Signature Forgery (CVE-2006-4339), described in http://www.openssl.org/news/secadv_20060905.txt
  - ASN.1 Denial of Service Attacks (CVE-2006-2937, CVE-2006-2940), described in http://www.openssl.org/news/secadv_20060928.txt
  - SSL_get_shared_ciphers() buffer overflow (CVE-2006-3738), also in http://www.openssl.org/news/secadv_20060928.txt
  - SSLv2 Client Crash (CVE-2006-4343) also in http://www.openssl.org/news/secadv_20060928.txt

  As of this publication there are no workarounds available for any of these vulnerabilities but it may be possible to mitigate some of the exposure. This Security Response lists the status of each product or application when considered individually. However, in cases where multiple applications are running on the same computer, a vulnerability in one application or component can compromise the entire system. This compromise can then be leveraged against applications that would otherwise be unaffected. Therefore, users must consider all applications when determining their exposure to these

vulnerabilities. Cisco strongly recommends that customers update all vulnerable applications and components to provide the greatest protection from the listed vulnerabilities. Cisco will update this document in the event of any changes.

- **CSCek58150**—When an ssl-proxy-list is activated, all the configured certificates are verified. The CSS rebooted because a configured certificate name could not be found on the disk. This condition will now generate an error.
- **CSCek58275**—When the CSS was configured for SSL termination and was processing SSL fragments within a SSL packet, a buffer pointer could be freed twice and the CSS would reboot.
- **CSCek54104**—When the CSS was configured for SSL termination, overtime SSL flow processing would cease. It was observed in the **show ssl resources** command that all SSL session chunks were in use. The session chunks were in the TCP FIN_WAIT_2 state waiting for their inactivity timer to elapsed. It was also noted that the inactivity timer would never elapsed, and was corrupted due to the wrap of an internal 32 bit VxWorks counter (tickGet()). It was determined that the math calculation needed to be changed to 64 bit arithmetic. This problem occurred because the servers were either very slow or would never respond to the CSS FIN and thus the sessions would only be cleaned up by the inactivity timer, which was running much longer than normal.
- **CSCek57234**—A new ssl-proxy-list command was added (**ssl-server tcp server mss**) to ensure that the SSL Module would respect the MSS from the backend server. This command was required because on the backend, the CSS handles the clear-text connection to and from the server and the server's original TCP SYN MSS is never seen by the SSL module.
- **CSCek59284**—Similar SSL CA certificate names (for example, *root* and *root2*), may incorrectly be flagged as duplicates depending on the order in which they were configured. The CSS was not using the length of the CA certificate when checking for duplicates.
- **CSCek60355**—This DDTS addresses an unintended side effect of fixing CSCeh77401. The original changes were added to not allow a re-loadbalance of the HTTP method CONNECT (or a custom authority URI) during a persistent connection. However, in this case, the HTTP CONNECT method was spanned over multiple packets, and due to CSCeh77401, the CSS failed to send the delayed TCP ACK to the client and eventually the connection was reset.
- **CSCek61065**—The fix for CSCed60068 caused the double free of an internal SSL buffer in certain edge case error conditions, which caused the CSS to reboot.
- **CSCek62401**—When the CSS was configured for SSL Termination and TCP packets arrived out of order to the SSL module and one of those packets had the TCP FIN bit set, the SSL module would lose the TCP FIN. This would result in a retransmission loop with the client/server.
- **CSCek62999**—When the CSS was configured for compression and was using an older style SSL Module, web pages were being truncated. The problem did not occur when a SSL-C (compression) was used.
- **CSCek63603**—When the APP **rcmd** command was issued to display a very large file, it caused a buffer leak on the SCM in Buffer Pool 2. Over time, this buffer leak would cause the CSS to reboot with an ONDM Life tick failure.
- **CSCek64233**—The TACACS commands **tacacs key** and **tacacs-server** cause the CSS to reboot if the quoted clear text authentication key was greater than 48 characters. The range of the authentication key in the MIB is 100 characters for the encrypted key. The unencrypted or clear text authentication key is limited to a maximum of 48 characters.
- **CSCek66496**—When using the **commit_vip_redundancy** command on a config containing many IP addresses applied to a circuit configuration, the sync may fail.
- **CSCek60679**—When using the **config_sync** command with the **restrict user-database** command already configured, the sync script may fail.

- **CSCek64713**— When the CSS is configured for FTP load balancing and a PORT or 227 FTP command is received, the TCP sequence space may be adjusted due to NAT'ing of the FTP payload. If a FTP TCP packet was retransmitted that was previous to the last FTP command, the CSS would incorrectly adjust (that is, corrupt) the TCP sequence space. Now an FTP TCP packet that was retransmitted before or after the last FTP command will have the correct TCP sequence space. If the FTP TCP retransmission occurs prior to two or more FTP commands, the TCP sequence space will still be incorrect.
- **CSCek65744**—When the CSS is configured for Layer 5 content load balancing and the **flow persist-span-ooo** command is configured, if the data portion of a HTTP POST spanned multiple ethernet packets, a portion of the subsequent POST data packet would not be forwarded to the server. This occurred with an SSL module and the server was the SSL module when doing SSL termination, however it may occur with any Layer 5 content rule.
- **CSCek66403**—When the CSS was configured with a named keepalive that was added to multiple configured services and one of the services was subsequently removed, the internal flag within the named keepalive was zeroed out. If you dynamically modified the named keepalive, the CSS would reboot. This issue was related to fixes for DDTS CSCeh65429 and CSCek46451.
- **CSCek67425**—When the CSS was configured for UDP fragment load balancing support using the **udp-ip-fragment enabled** command, if an incoming UDP fragment had a UDP checksum of zero (which is allowed), the CSS would incorrectly try to recalculate the UDP checksum of the packet, if the packet was NAT''d. This resulted in the outgoing UDP fragment having a corrupted UDP checksum. If the incoming UDP fragment has a zero checksum, it should remain zero on the outbound side.
- **CSCek67196**—When the CSS was configured with Layer 5 domain content rules and an incoming HTTP method had the domain starting with "http://" but did not have a "Host:" tag, an earlier fix for DDTS CSCeh72177 incorrectly caused this scenario to fail and the CSS would send a TCP RST back to the client as if no content rule matched.
- **CSCek65224**—When the CSS is configured with an FTP content rule with a source group and the **flow tcp-reset-vip-unavailable** command and the **passive FTP ls** commands are run repeatedly on the FTP connection, the CSS randomly sends a TCP RST to one of the FTP data channel requests. The CSS will now send a TCP RST for a properly setup FTP passive data channel connection.
- **CSCek69302**—When the CSS contains an SSL module and is configured for backend ssl or ssl-init and the ssl-proxy-list is configured with the backend-server type ssl command and a server is configured for client authentication, you must have a certificate configured on the ssl proxy list or the CSS reboots.
- **CSCek69849**—When the CSS is configured with the **flow-state flow-disable** command, the portmaps may not be cleared properly and this caused connections to be improperly NAT'd.
- **CSCsi06799**—Removed all references to the AcctRptErrorCheck debug variable, as it is no longer used.

Software Version 8.10.3.01 Command Changes

Table 7 lists the commands and options that have been added or changed in software version 8.10.3.01.

Table 7 CLI Commands Added in Version 8.10.3.01

Command and Syntax	Description
ftp data-channel-timeout	The maximum range for this command was changed from 5 to 20 to 5 to 120.
show ip statistics *slot number*	This command enables you to display statistics for a specific slot number of a module. For a CSS 11503, enter an integer from 1 to 3. For a CSS 11506, enter an integer from 1 to 6. For a CSS 11501, enter 1. If you do not specify a slot number, this command displays the statistics for all slots in a CSS.
ssl-server "X" tcp server mss "Y"	This command was added to the **ssl-proxy-list** command to ensure the SSL module would acknowledge the MSS from the backend server.
flow-srvdown-reset **no flow-srvdown-reset**	This new command causes the CSS to send a TCP RST for a server failure.
clock summer-time TZ recurring	As of January 1, 2007 the beginning and ending of daylight savings time changed. Previously it started on the first Sunday of April and ended on the last Sunday in October. It now starts on the second Sunday in March and ends on the first Sunday in November. The **clock summer-time TZ recurring** command was modified to reflect these changes.
keepalive type http append-port-hosttag	This new command has been added to append the port to the Host: tag in the HTTP request.

Software Version 8.10.2.05 Open Caveats, Resolved Caveats, and Command Changes

The following sections contain the open caveats, resolved caveats, and command changes in software version 8.10.2.05:

- Software Version 8.10.2.05 Open Caveats
- Software Version 8.10.2.05 Resolved Caveats
- Software Version 8.10.2.05 Command Changes

Software Version 8.10.2.05 Open Caveats

The following caveats apply to software version 8.10.2.05:

- **CSCek54104**- If an SSL client does not respond to the CSS FIN with a FIN, the CSS waits for the inactivity timer to expire. Occasionally, the CSS has difficulty when the timer wraps. Thus, the CSS does not correctly clean up the SSL resources.
- **CSCek55371** - When a CSS custom scripted keepalive script includes the ">" and ">>" file redirection characters to perform file input/output (I/O), the script does not follow the CSS scripted keepalive guidelines. Over time, the CSS loses internal file descriptors that eventually cause the CSS file system to unmount. You must reboot the CSS to clear the condition. Do not include custom contain file redirection characters for file I/O in scripted keepalive scripts.
- **CSCek56722** - If the CSS 11500 is configured for UDP fragment support in addition to ports configured for flow-disable, the CSS does not properly forward any UDP fragments that match the UDP port number configured for flow-disable. For example:

  ```
  udp-ip-fragment enabled
  flow-state UDP_port_number flow-disable nat-enable
  ```

- **CSCek57234** - When the CSS spoofs a clear-text connection from the SSL module, it responds to the SYN with a SYN/ACK that has a maximum segment size (MSS) of 1,460. Eventually, the CSS also opens a connection to the server and the server informs the CSS that its MSS is a smaller value. When data arrives from the client, the CSS decrypts it. The CSS may combine data from multiple packets that it sends to the server, exceeding what the server can handle.
- **CSCek58150** - When an SSL-proxy list activation occurs, a verification of the certificate and key pairs occurs for all configured SSL servers. During that process, if the CSS finds a problem with the actual certificate and key pair files on the disk, it is not prepared to handle an error. The CSS references through a NULL pointer and fails.
- **CSCek58275** - The lifetick failure occurs due to a failed task on the SSL module when the module tries to free a buffer twice. If the module had properly cleaned out the address of the buffer after freeing it, it would not attempt to free it again.

Software Version 8.10.2.05 Resolved Caveats

The following caveats were resolved in software version 8.10.2.05:

- **CSCdu87494** - When a session to the CSS or scripted keepalive is closing down, it is possible for the CSS to reboot if the MORE buffer is in use.
- **CSCdx34275** - If you use a **show** command with the MORE option enabled, if the MORE buffer is full and you perform a forward search with the slash (/) character, the CSS reboots.
- **CSCej50977** - The CSS does not support compression services on content rules that do not have a VIP address configured. If a CSS with an SSL module is configured with compression services assigned to a content rule that does not a VIP address configured, flows are not established.
- **CSCej54451** - In a compression-only service, suspending the service while the compression is in progress causes the connection to be closed immediately rather than waiting for the current operation to complete.
- **CSCej87514** - The CSS fails to negotiate a TCP handshake successfully when it is proxying a connection to a server that returns a zero window size.
- **CSCek00530** - The CRL download fails if the HTTP header spans multiple packets. The CRL download occurs between the SSL module and the configured CRL server. The HTTP header is terminated by a CRLFCRLF, and the CRL download code expects that terminator to be in the first server data packet. The actual CRL data may span multiple packets. In testing with Linux, if the MTU is 278, the HTTP header splits and the CRL download fails.
- **CSCek10020** - When a CSS is configured with multiple SSL servers in a SSL proxy list, only the first SSL server records compression statistics.
- **CSCek15563** - The IPV4 critical message does not include adequate information to determine which traffic is causing the error message to be generated. For example, the following message should include the IP addresses or ports so that you can determine which traffic is generating the error condition:

  ```
  SEP 19 13:50:25 4/1 6307 IPV4-2: Ipv4SlaveForwBmanChk: no ingress LP in buffer
  ```

- **CSCek27227** - The CSS may reboot when receiving an SNMP get request for the MIB variable apCntStickyNoCookieString on a content rule.
- **CSCek29491** - When the CSS is configured with a service with **keepalive type http encrypt** (an encrypted keepalive) and the service IP address is not on the local subnet, but must be routed to it, the CSS fails to complete the SSL handshake and resets the connection. This causes the service to remain in the down state permanently.
- **CSCek32546** - When a CSS is configured with SSL header insertion, only the first GET contains the insertion.
- **CSCek32632** - The CSS reboots when it runs out of system application buffers and fails to check for a non-existent buffer return code.
- **CSCek33838** - When you suspend, modify, and reactivate an SSL-proxy list, the CSS updates the modifications on the SSL module in the CSS 11500 chassis. Though the CSS updates these changes on SSL services with different SSL slot numbers, it does not update SSL services of the same ssl-accel type. Thus, the CSS updates only the first configured SSL service of this type with the SSL-proxy list modifications. In addition, due to IP tuple collisions, the CSS may not download the same CRL when the CRL is configured on multiple SSL modules.
- **CSCek34035** - When the CSS is configured for DHCP and it receives a DHCP BOOTREQUEST to its circuit address, it incorrectly sends an ARP request out for itself causing the circuit to become unusable. The CSS should drop the DHCP packet because the CSS is a DHCP relay agent only.

- **CSCek34314** - On a CSS with a configured SSL module, when you enter the **no ssl associate cert** command to remove a certificate that is configured on an SSL-proxy list, the CSS removes the certificate globally, but it has no effect on the configured SSL-proxy list, SSL server, and traffic on the SSL module. The CSS should not allow the use of the **no ssl associate cert** command when a certificate is configured on an active SSL-proxy list.
- **CSCek34973** - When you use the WebManagement GUI to configure the CSS, it fails to allocate a socket through a call into the VxWorks kernel. All the sockets were in use due to a major network event and a large number of keepalives were configured. The GUI did not check whether the socket allocation returned an error. It used a NULL or zero socket pointer and the CSS rebooted.
- **CSCek35141** - When running the commit_vip_redundancy script in partial mode (that is, without the **-a** option), the script automatically checks that all VIP addresses on active local content rules and source groups are redundant on the remote CSS. The **-norvip** option has been added so that this checking is completely bypassed at the script execution time.
- **CSCek35783** - A SNMP GET or GET NEXT request for any OID in the rip2PeerEntry table suspends the SNMP engine on the CSS and no further SNMP actions can take place. The CSS sends the "%%Error - cannot obtain SNMP lock" error message, does not respond to SNMP requests, and appears to hang.
- **CSCek36511** - When CSS is configured with an ACL clause that preferred certain clients to a source group allowing the CSS to send the packet out with a NAT'd source IP address, several servers did not respond to the initial TCP SYN (a TCP SYN/ACK in all cases), causing the client to retransmit the TCP SYN repeatedly. Occasionally, an intermediate firewall logs an error due to unexpected IP addresses because the CSS eventually forwards some the retransmitted TCP SYNs unNATed.
- **CSCek37183** - When the CSS is configured for Session Level Redundancy (SLR) with content rules of the sticky advanced-balance arrowpoint-cookie method, if the arrowpoint-cookie content rule is not configured with the **redundant-index** command, the rule should not participate in the SLR peer-to-peer Flow Control Block (FCB) sharing. However, the CSS sends SLR flow-modify arrowpoint-cookie sequence-number updates to the SLR peer and, under an extremely heavy load, overwhelms it. The slots in the CSS 11500 chassis may display as bad in the **show chassis** command output or become unresponsive to different **show** commands, or unexpected behavior can occur.
- **CSCek37489** - The VxWorks timerLib, accessed through timerGet() or timerSet(), is a 32-bit value that wraps every 828 days (0xffffffff -> 0x0). This wrapping causes the following two issues:
  - If the CSS is running redundancy, the backup would also become master and create duplicate IP addresses in the network.
  - If the CSS is configured for service keepalives and the keepalive went down legitimately, the CSS may still mark it as alive.

  The only way to recover from either of these issues is to reboot the CSS.
- **CSCek38578** - On rare occasions, when two users log into the CSS and dynamically configure the same content rule and one user issues the **remove service ?** command and hints for the services on the content rule while the other user issues the **no content** *name* command and removes the content rule and all the associated services. The CSS reboots because the CSS removed the services as it collected the hints.
- **CSCek39096** - When the CSS is configured for SSL termination and an application is running over a 14.4 baud modem, a large HTTP POST data is divided over multiple packets. As the SSL module collects these packets and an internal hardware limit of 50 is reached, the module discards the HTTP POST data. The SSL module tries to compact the smaller buffers into larger buffers, decreasing the block chain side, and ensuring that the internal limit is not reached.

- **CSCek39894** - When the CSS has two DNS A records configured and dynamically reconfigures the weight from "0 to 1" and "1 to 0," the remote CSS peer incorrectly load balances between the two DNS A records with different weights (one a 0 and one at 1).
- **CSCek40630** - When multiple users log into the CSS and issue CSS configuration commands, the CSS SNMP application hangs and stops processing further commands. It is possible for two users to each take one of the necessary SNMP locks (internal name SNMP semaphore) and thus neither are able to complete the configuration commands.
- **CSCek40768** - The fix to DDTS CSCeh18228 attempted to ensure that the publishing of the VIP state is done when the reporter is fully up so that dormant flows are not activated too early. However, the VIP state was not updated when the VRID-PEERING router goes down, which may cause the state of the content rule to go down or become inconsistent.
- **CSCek41097** - If you configure a global named keepalive, left it suspended, and then add it to a service in the active state, the CSS does not configure the keepalive on the service. If you change the keepalive type on the service itself, the global keepalive becomes a ghost keepalive. It appears in the running-config file but the CSS deletes it internally so that you cannot delete it.
- **CSCek41354** - When a CSS is configured for SSL termination and Session Level Redundancy (SLR), a redundant index is configured on the clear-text rule used by the SSL module for decrypted SSL traffic and two physical ports are in the server VLAN network, if you establish a long-lived client session using the SSL rule with the session in progress and the active port fails to the server, the session uses the other port but the session does not recover. However, a long-lived non-SSL connection to the server through the clear-text rule recovers as well as an SSL connection using a clear-text rule that does not have a configured redundant index.
- **CSCek42526** - When the CSS is configured for SSL termination, it experiences a problem very similar to CSCek39096 (receiving a large SSL record split across many small packets due to the TCP MTU size and dial-up over a slow modem). The DDTS CSCek39096 fix handles this problem up to approximately 16,000 bytes. But when this number is exceeded, the CSS would drop a portion of the HTTP POST data and the SSL module would exceed the number of data blocks allowed. A coding error in the calculation of the number of data blocks occurred after compactions. The SSL modules now handles the compaction correctly up to the largest SSL record of 16,384 bytes.
- **CSCek42725** - The fix for DDTS CSCei03219 relaxed certain restrictions when processing an SSL PKCS12 file. However, this fix leaked SCM memory in the size of the PKCS12 file, occupying a large chunk of memory over a period of time and thus causing the CSS to reboot due to the unavailability of SCM memory.
- **CSCek43439** - The CSS reboots due to an ONDM Lifetick failure because the SSL module is out of buffers. When the CSS polls the flowMgrExt.mib/apFlowMgrExtSlotFlowStatsTable SNMP OIDs, the CSS incorrectly sends these SNMP requests to the SSL module and a buffer leak occurs.
- **CSCek43975** - The CSS may drop bridged IP packets with the IPV4 Type of Service (TOS) bits set when **set dscp af21** (DiffServ feature) is configured on a Catalyst policy and the IPV4 header TOS bits are sent (0x48 in this example).
- **CSCek44225** - An SNMP GET or NEXT of the apIpv4VrrpStateDuration apIpv4.mib leaks a small amount of memory. Over a period of time, this leak may cause the CSS 11500 SCM to reboot.
- **CSCek44615** - When the CSS is configured for a Global Server Load Balancing (GLSB) dns-record encrypted-KAL keepalive, a misconfiguration on the peer device corrupts the data in the AP-KAL message. When the CSS processes this corrupted data, it may reboot.
- **CSCek44734** - Per DDTS CSCei86650, the HTTP "Connection: closed" tag is added instead of the "Connection: close" tag.
- **CSCek44888** - The **passive sync** command returns a Busy error message for a period of many weeks. A CSS reboot clears the issue.

- **CSCek45031** - When the CSS is configured for compression, the client sends an HTTP POST request. The server responds with an HTTP 100 Continue response and then later responds with an HTTP 200 OK response. CSS compression did not recognize the scenario of a Client Request/Server Response/Server Response and sends a TCP RST in response to the HTTP 200 OK server response.
- **CSCek46451** - If you attempt to modify a configured service or global keepalive, you may incorrectly receive the message "%% Maximum keepalives of this type have been exceeded. Cannot activate" when the maximum number has not been exceeded. This message may occur when you configure a global keepalive and add the global keepalive to a service. Later, you change the global keepalive type to type tcp. After you activate the keepalive or it is modified dynamically when the global keepalive is already active, the internal keepalive count is corrupted. This problem may cause any further service or global keepalive modifications to fail with the previously-described error message.
- **CSCek46686** - When you log into the CSS with a username that has embedded control characters, the login is invalid. When the CSS generates the subsequent SNMP login trap, the trap contains the embedded control characters, which is incorrect. The RFC specifies the removal of control characters before the SNMP login trap is generated.
- **CSCek47850** - The CSS can leave unreachable host entries in the route table causing the table to exceed the 5,120 entry limit. The CSS can not learn any additional route entries. These entries accumulate when an ARP resolution fails for a host that has already been marked unreachable.
- **CSCek48356** - DDTS CSCdx09860 fixed a long standing advanced-balance arrowpoint-cookie issue that a server retransmission of the HTTP 200 OK response (usually the first server data packet) would not have the ARPT cookie reinserted. DDTS CSCee55759 fixed a problem that the TCP sequence number was wrong in the retransmitted server data. However, the fix failed to redo the TCP packet checksum when the TCP sequence number is adjusted in the server retransmission and the client sees a TCP checksum error. Now, a server retransmission of what is usually the first server data packet (HTTP 200 OK) has the inserted ARPT cookie, the correct TCP sequence number, and the correct TCP checksum. CSCek48833 also had the same TCP checksum issue and that problem is corrected.
- **CSCek48429** - RFC 1155 states that SYNTAX Counter is a non-negative integer that monotonically increases. The CSS 11500 MIBs have cases where a MIB OID is defined as a COUNTER or Counter32 when it is really a value that varies. An example of this is from the svcExt.mib - apSvcCurrentLocalConnections. The current connection counter on a configured service does not reflect RFC 1155. Instead, it should be defined as GAUGE or GAUGE32, which indicates an unsigned integer value that will not consistently increase until it wraps. All the CSS 11500 are updated to properly define MIB OIDs as Gauge where appropriate.
- **CSCek48831** - When you run a script manually on the CSS and the script exits unexpectedly, the EXIT_MSG defined in the script should appear at the CLI prompt. This functionality was broken by the DDTS CSCei41874 fix.
- **CSCek48833** - A long lived CSS 11500 flow may incorrectly be made eligible for garbage collection every 49 days, 17 hours, and approximately 6 minutes. The flow appears inactive for longer than any configurable flow-timeout-multiplier period because an internal CSS unsigned 32-bit variable overflowed wraps because it contains milliseconds since the CSS boot. The flow would be eligible for garbage collection until the next packet (activity) occurs and then the flow is again safe for the next approximately 50 days.
- **CSCek49389** - When the CSS contains an SSL module, the module should send an ACK for every other packet instead of every single packet.

- **CSCek49708** - When the CSS is configured for VIP/IF redundancy with spanning tree disabled and the **no enable** command is configured on the virtual-router IP interface, if you run the commit_vip_redundancy script or the **copy startup-config running-config** command, the state of the virtual router becomes Master or Backup instead of Down.
- **CSCek50736** - When the CSS is configured with an Layer 5 VIP and the client sends a SYN to the VIP, the CSS responds with a SYN/ACK to the client. The SYN/ACK is returned as an ICMP unreachable to the VIP by a router unable to locate the client. This action may cause the CSS to forward the ICMP unreachable with a source and destination IP address of 0.0.0.0.
- **CSCek51806** - The chassis backplane part number is 16 characters long. This length may cause the CSS to reboot when you issue the show chassis inventory command or run the diagnostic showtech script.
- **CSCek52385** - The following two commands have been added to allow you to modify the default TCP window size (12288) to a larger value between 12288 and 40960 for both the server and client side independently: **ssl-server** *number* **tcp server window** *bytes* and **ssl-server** *number* **tcp virtual window** *bytes.*
- **CSCek52881** - When the CSS is configured with the **advanced-balance arrrowpoint-cookie** command, during a backend remap condition, a subsequent method comes into the arrowpoint cookie without the cookie being set. This action may cause the CSS to send a RST to the client and the server.
- **CSCek53172** - On a CSS that contains an SSL module and an SNMP WALK of the sslExt mib occurs, the CSS may return the keys and certificates in the wrong order.
- **CSCek53697** - When the CSS is configured with VIP/IF redundancy, running the **show running-config service** *service_name* command may cause a redundancy failover.
- **CSCek55754** - The CSS allows the configuration of an IP address as an IP management route that overlapped with the IP address configured on a service. This configuration is not valid and causes the commit scripts to fail.
- **CSCek57865** - CSS compression fails on large (approximately 100K bytes) files using a web browser.
- **CSCin99962** - When the SNMP configuration tool performs a byte-by-byte comparison of the startup-config and the running-config files obtained from the CSS, it did not perform a comparison for extra Carriage Returns (CRs) in the startup-config output.

Software Version 8.10.2.05 Command Changes

Table 8 lists the commands and options that have been added in software version 8.10.2.05.

Table 8 CLI Commands Added in Version 8.10.2.05

Mode	Command and Syntax	Description

ACL	**exclude ssl circuit-(VLAN***number***)** {*acl_clause*}
	
	no exclude ssl circuit-(VLAN*number***)**
		This new command allows you to exclude all clauses or specific clauses within an ACL to outbound traffic from the SSL module. By default, the CSS applies all clauses within the ACL to outbound traffic from the SSL module. The variables for this command are:
		• *number* - Number of the circuit on which to exclude the ACL clauses.
		• *acl_clause* - (Optional) The number of the clause to exclude. You can configure one or more clauses, or a range of clauses. To enter more than one clause, separate each number by a comma with no spaces. To enter a range of clauses, separate the first and last number in the range by a dash (-) with no spaces.
		If you do not specify a clause, all clauses are excluded.
		For example, to exclude clauses 1, 5, and 10 through 20 on ACL 7 for VLAN1, enter:
		`(config-acl[7])# exclude ssl circuit-(VLAN1) 1,5,10-20`
		Consider the following requirements when using the **exclude** command:
		• The CSS must contain an SSL module for use with the **exclude** command.
		• Before reconfiguring the **exclude** command on an ACL, you must use the **no** form of the **exclude** command. Otherwise, the CSS displays an error.
		`Must issue <no exclude ssl circuit-(VLAN#)> command first`
		• You can configure only one **exclude** command per ACL. This rule includes use of the **no exclude** command for a different VLAN other than the configured VLAN. Otherwise, the following error message appears:
		`Only one <exclude ssl circuit-(VLAN#)> command per-ACL`

Table 8 CLI Commands Added in Version 8.10.2.05 (continued)

Mode	Command and Syntax	Description

ACL (continued)	**exclude ssl circuit-(VLAN***number***)** {*acl_clause*}
	
	no exclude ssl circuit-(VLAN*number***)**
	
	(continued)
		• The **exclude** command cannot be on different ACLs for the same VLANs. Otherwise, the following error message appears:
		
		`Command <exclude ssl circuit-(VLAN#)> command found on different ACL`
		
		• When you configure the **exclude** command on an ACL, you can configure only one **apply** command on that ACL. Otherwise, the following error message appears:
		
		`Only one <apply circuit-(VLAN#)> command allowed with exclude configured`
		
		If you have multiple **apply** commands configured on an ACL, you cannot configure the **exclude** command.
		
		You can configure the **exclude** command without the **apply** command but it does not take effect until the **apply** command is configured.
		
		• When you configure the **exclude** and **apply** commands on an ACL, the circuit VLAN number must match in these commands. Otherwise, the following error message appears:
		
		`No circuit apply command or exclude ssl circuit mismatch`
		
		• The **exclude** and **apply** commands for the same circuit must be on the same ACLs. Otherwise, the following error message appears:
		
		`Command <exclude ssl circuit-(VLAN#)> command on different ACL than apply`
		
		• If you configure the **apply** command and then configure the **exclude** command or its **no** form, the CSS internally reissues the **apply** command to reapply the ACL to the circuit. Reissuing this command allows the SSL setting to be updated on the remote session processors.
		
		• The following command set negates the **exclude** command if the circuit VLAN is removed:
		
		interface *slot*/*subslot* command
		
		no bridge vlan command
		
		Use the **no** form of the **exclude** command to apply all ACL clauses to the outbound traffic from the SSL module.

Table 8 CLI Commands Added in Version 8.10.2.05 (continued)

Mode	Command and Syntax	Description
Global	**flow-state flow-disable timeout** *seconds* **no flow-state flow-disable timeout**	This new command sets the wait time for any response from a server for a configured flow-disable port. By default, the CSS times out a flow-disable (no flow) connection in 5 seconds if it does not receive a response from the server. In the case of DNS responses, they may take longer than 5 seconds, causing the connection to fail. By using the **flow-state flow-disable timeout** command to set a longer wait time for server responses, these connections are less likely to fail. The *seconds* variable is the time in seconds. Enter an integer from 5 to 20. The default value is 5. Use the **no** form of the command to reset the default flow-disable timeout to 5 seconds.
	snmp trap-type enterprise disk-quota {**interval** *minutes* {**quota-threshold** *percentage*}} **no snmp trap-type enterprise disk-quota**	The new **disk-quota** option enables SNMP enterprise traps when the space used on the CSS disk is greater than or equal to the configured threshold. By default, the time interval in minutes is 720 minutes (12 hours) and the percentage threshold is 90. The options are: • **interval** *minutes* - Configures the disk quota interval in minutes. This interval determines how often to check the percentage of space used on the CSS disk. For the *minutes* variable, enter an integer from 1 to 1440. The default is 720. • **quota-threshold** *percentage* - Configures the disk quota threshold. This threshold is the percentage of bytes used on the CSS disk. For the *percentage* variable, enter an integer from 10 to 99. The default is 90. Use the **no** form of the command to disable this trap.
SSL-proxy	**backend-server**|**ssl-server** *server-num* **tcp virtual**|**server window** *bytes* **no** [**backend-server** | **ssl-server**] *server-num* **tcp virtual**|**server window**	As per CSCek52385, the new **window** keyword allows you to increase the client-side or server-side SSL TCP window size for your specific environment to improve performance. The *bytes* variable is the window size in bytes. Enter a number from 12288 to 40960. By default, the CSS sends a window size of 12,288 bytes. Use the **no** form of the command to reset the default value for the window size.

Table 9 lists the commands and options that have changed in software version 8.10.2.05.

Table 9 CLI Commands Changed in Version 8.10.2.05

Mode	Command and Syntax	Description
Content	**application realaudio-control**	The **realaudio-control** keyword has been removed from the CLI. If configured, this keyword would cause startup errors per CSCek32262.
Global	**compress tcp server comp-queue-delay** *ms* **no compress tcp server comp-queue-delay**	The default maximum delay in milliseconds (ms) has changed from 750 to 1000.
Service	**domain "***string***"**	The *string* variable is now a quoted string. The quoted string allows the use of the % character. For backward compatibility, you can enter an unquoted text string with no spaces and a maximum length of 64 characters. However, the CSS does not allow you to enter the % character. Also, the CSS changes the string in the running-config to a quoted string.

Software Version 8.10.1.06 Open Caveats, Resolved Caveats, and Command Changes

The following sections contain the open caveats, resolved caveats, and command changes in software version 8.10.1.06:

- Software Version 8.10.1.06 Open Caveats
- Software Version 8.10.1.06 Resolved Caveats
- Software Version 8.10.1.06 Command Changes

Software Version 8.10.1.06 Open Caveats

The following caveats apply to software version 8.10.1.06:

- **CSCej50977** - The CSS does not support compression services on content rules that do not have a VIP address configured. If a CSS with an SSL module is configured with compression services assigned to a content rule that does not a VIP address configured, flows are not established.
- **CSCej54451** - In a compression-only service, suspending the service while the compression is in progress causes the connection to be closed immediately rather than waiting for the current operation to complete.
- **CSCej87514** - The CSS fails to negotiate a TCP handshake successfully when it is proxying a connection to a server that returns a zero window size.
- **CSCek00530** - The CRL download fails if the HTTP header spans multiple packets. The CRL download occurs between the SSL module and the configured CRL server. The HTTP header is terminated by a CRLFCRLF, and the CRL download code expects that terminator to be in the first server data packet. The actual CRL data may span multiple packets. In testing with Linux, if the MTU was 278, the HTTP header splits and the CRL download fails.

- **CSCek10020** - When a CSS is configured with multiple SSL servers in a SSL proxy list, only the first SSL server records compression statistics.
- **CSCek15563** - The IPV4 critical message does not include adequate information to determine which traffic is causing the error message to be generated. For example, the following message should include the IP addresses or ports so you can determine which traffic is generating the error condition.
  SEP 19 13:50:25 4/1 6307 IPV4-2: Ipv4SlaveForwBmanChk: no ingress LP in buffer
- **CSCek27227** - The CSS may reboot when receiving an SNMP get request for the MIB variable apCntStickyNoCookieString on a content rule.
- **CSCek29491** - When the CSS is configured with a service with **keepalive type http encrypt** (an encrypted keepalive) and the service IP address is not on the local subnet, but must be routed to, the CSS fails to complete the SSL handshake and resets the connection. This causes the service to remain in the down state permanently.
- **CSCek32546** - When a CSS is configured with SSL header insertion, only the first GET contains the insertion.
- **CSCek32632** - The CSS reboots when it runs out of system application buffers and fails to check for a non-existent buffer return code.
- **CSCek32637** - The CSS reboots when it runs out of file descriptors and is configured with scripted keepalives and the command scheduler.

Software Version 8.10.1.06 Resolved Caveats

The following caveats were resolved in software version 8.10.1.06:

- **CSCei55869** - The CSS ignores the header-insert information in the ssl-proxy-list after you suspend and then reactivate the ssl-proxy-list. The configuration appears in the running configuration.
- **CSCej70513** - When the CSS is configured with services and content rules and you activate a content rule, the remote SP does not check the validity on the service index and the CSS may reboot. The CSS will now log a message to warn about this issue.
- **CSCej76133** - The global configuration **flow reserve-clean** command is being removed and the associated MIB object deprecated. This command has been replaced with the **flow permanent** and the **flow-timeout-multiplier** commands.
- **CSCej76835** - The CSS SSL module may hang in a Down state and then attempt to reboot because it was unable to create a core file. During this time, all traffic to the SSL module is dropped. When this condition exists, the **show task** command in debug mode displays suspended tasks on the SSL module.
- **CSCej83237** - Using the **ssl genscr** command to generate a new certificate with an existing filename causes the CSS to reboot.
- **CSCej88415** - On a CSS configured with SSL header insertion, when the CSS processes an application data frame that contains a GET, it attempts to insert session information into the clear text request header, but the cipher is NULL, causing the SSL module to reboot.
- **CSCek00656** - In some instances, an ap-kal-dns scripted keepalive stops being sent from CSS to server.
- **CSCek04270** - The CSS reboots when you add a DNS entry to a content rule.
- **CSCek04631** - The **ip route originated-packets** command did not work consistently when configured on the CSS and the results were undefined.

- **CSCek06031** - An FTP test tool was run against the CSS to perform vulnerability testing and the CSS experienced many core dumps. The tool would send FTP commands with very long file and path names and the CSS would corrupt internal memory and reboot.
- **CSCek12106** - The CSS allows you to add a primary or a secondary sorry server (whose service does not contain a redundant-index) to a content rule that contains a redundant-index when that content rule is active. This should not be allowed and may cause the **config-sync** command to fail and Adaptive Session Redundancy to not work properly.
- **CSCek22918** - When accessing the CSS GUI, you are prompted with a SSL certificate from the CSS. The SSL certificate was configured to expire on 5/29/2006. Although the expired certificate can continue to be used to access the GUI, a new certificate has been provided.
- **CSCek24806** - If a TACACS server responds to the three way TCP handshake but then fails to fully respond to the actual TACACS request, the CSS authentication ability may fail to respond and no further login attempts will be authenticated.
- **CSCek24921** - A connection that is being authenticated is closed before the authentication process is completed causing the CSS to reboot.
- **CSCek25025** - When the CSS is configured with SSL initiation and SSL backend, the CSS terminates the cleartext connection but does not create the corresponding SSL connection.
- **CSCek25247** - The CSS reboots when it is configured for XML and receives a HTTP content request with a large number of tags that uses all the available HTTP daemon memory, which leaves zero memory when it is time to process the MIME authorization.
- **CSCek26020** - The CSS reboots if you enter the **no ssl-server** xx **cipher ?** command and "xx" is not a configured ssl-server.
- **CSCek26454** - A CSS with an SSL module configured with compression services may reboot it if receives IP fragmented packets.
- **CSCek26792** - The CSS did not send a TCP RST for a "Mid Spoof Reject" as it did for a "Mid Nat Reject". These errors occur when the Flow Control Blocks (FCBs) for a connection have been deleted and reused for new incoming connections. If the configured content rule configured is a Layer 3 rule or a Layer 4 rule, then the error is "Mid Nat Reject". If the configured content rule is a Layer 5 rule, then the error is "Mid Spoof Reject".
- **CSCek34363** - On a CSS with an SSL module with client authentication and session id reuse (which is enabled by default) configured, when IE browser connections are made, the connections hangs. Once the HTTP GET is received, the CSS does not forward that GET to the server. The client browser hangs until the connection times out.

Software Version 8.10.1.06 Command Changes

Table 10 lists the commands and options that have been added in software version 8.10.1.06.

Table 10 CLI Commands Added in Version 8.10.1.06

Mode	Command and Syntax	Description
Service	**compress tcp server comp-queue-delay** *ms* **no compress tcp server comp-queue-delay**	The new **comp-queue-delay** keyword allows you to configure the maximum delay in milliseconds (ms) before sending data into the SSL module for compression. For the *ms* variable, enter a number from 50 to 10000. The default delay is 750 ms. Use the **no** form of this command to reset the default maximum delay.
SSL-Proxy	**ssl-server** *number* **crl** *crl_record_name* **expiration-enable** {**verification-enable**}	The new **expiration-enabled** keyword allows the SSL module to determine whether a reloaded CRL file has expired by checking the Next Update field in the file. By default, when the CSS successfully loads the CRL initially and then reloads a new copy of the CRL file at the configured hourly refresh interval, it does not check the Next Update field in the file to determine if the CRL has expired, and subsequently downloads an expired file from the configured server. When you configure this keyword and the CSS tries to load a new copy of the CRL, the SSL module checks the Next Update field in the file. If the field indicates that the CRL has expired, the module clears it from each associated SSL server and rejects all resulting client connections. The SSL module checks the Next Update field when the CSS loads the CRL file. A load occurs when: • You activate an ssl-accel type service. • An SSL-server VIP address associated with a CRL goes to the master state (for example when a content rule is activated). • The CRL hourly refresh interval is reached. • You enter the **ssl force-crl** command. The new **verification-enabled** option allows the SSL module to clear the CRL from each associated SSL server and rejects all resulting client connections when any of the following failures occurs when downloading a CRL file: • Host Timeout • Host TCP Reset • Host HTTP “File not Found” return code • CRL File Format Bad • CRL Signature Bad • CRL Next Update Field Invalid • CRL Next Update Expired • Internal CRL memory allocation failure

Table 10 CLI Commands Added in Version 8.10.1.06 (continued)

Mode	Command and Syntax	Description
SuperUser	**clear ssl crl statistics** {*crl_name*}	Clears all SSL certificate revocation list (CRL) records statistics displayed through the **show ssl crl-record** command. You can optionally clear the statistics for a specified CRL record name, *crl_name*.
	ssl clear-crl {*crl_name*}	Clears the CRL file from all associated SSL servers. You can optionally clear a specified CRL record name, *crl_name*. **Caution** Use this command with caution. If client authentication is configured and you clear the CRL, all resulting client connections are reset.

Table 11 lists the commands and options that have changed in software version 8.10.1.06.

Table 11 CLI Commands Changed in Version 8.10.1.06

Mode	Command and Syntax	Description
All	**show service**	Now displays compression TCP configuration settings.
Global	**flow reserve-clean** **no flow reserve-clean**	This command has been removed from the CLI.

Software Version 8.10.0.02 Open Caveats, Resolved Caveats, and Command Changes

The following sections contain the open caveats, resolved caveats, and command changes in software version 8.10.0.02:

- Software Version 8.10.0.02 Open Caveats
- Software Version 8.10.0.02 Resolved Caveats
- Software Version 8.10.0.02 Command Changes

Software Version 8.10.0.02 Open Caveats

The following caveats apply to the CSS 11501, CSS 11503 and the CSS 11506:

- **CSCeh65783** - When a critical service becomes active, the CSS does not apply the VRRP hold down timer. Immediately after the critical services becomes active, VRRP transitions to a master state.
- **CSCej22808** - On a CSS with an SSL module, the CSS may reboot while processing a TCP RST due to the CSS attempts to free memory that had already been freed.

- **CSCej34271** - CSSs that are in a box-to-box redundancy configuration may get into a condition in which both redundant CSSs can hang. This condition may cause a network outage. During the event, the CSSs start repeatedly generating the following messages in the sys.log:

```
 2/1 262 FP_DRV-4: PrismImmFastPath::Send: Could not allocate an MCID. Remote message
send aborted.
 2/1 263 FP_DRV-4: PrismImmFastPath::Send: Could not allocate an MCID. Remote message
send aborted.
```

- **CSCej38383**- The CSS SSL module may fail to forward data to the client of backend SSL connections and sends RSTs to both client and server. This condition has been observed with the MS IIS5 SSL server.
- **CSCej45447** - In a CSS with an SSL module using SSL session ID reuse, if SSL sessions are reused with the same session ID, VIP, and port, some SSL sessions may be leaked causing the SSL module to refuse new SSL connections.
- **CSCej54451** - When you suspend a compression-only service, the CSS closes all active connections immediately.
- **CSCej57067** - Performing the **admin-shutdown** command of the server-side VLAN causes the critical reporter to go into a Down state. The reporter correctly brings all virtual routers into a Down state as well. After approximately four seconds, the CSS begins to log duplicate IP messages for all redundant VIPs. If the virtual routers are down, the CSS should no longer believe it should respond to the redundant VIPs, therefore no duplicate IP log messages should be logged. The duplicate IP log messages stops after the VRRP-4 log message appears indicating that the CSS is now the backup.
- **CSCej60160** - A CSS under minimum load may send many traplog messages that display extremely high DoS attack numbers and display the numbers as negative.
- **CSCej61680** - When you configure the CSS with an unsupported wildcard domain name and the content rule is hit, the domain name wildcard URL may cause the CSS to reboot.
- **CSCej70513** - The CSS reboots after you modify an SSL configuration and then run the commit_vip_redundancy script.
- **CSCej76133** - The global configuration **flow reserve-clean** command and the associated MIB object are deprecated. The command has been replaced in the CSS 11500 software with the **flow permanent** or **flow-timeout-multiplier** commands.
- **CSCej76835** - The CSS SSL module may hang in a Down state because it was unable to create a core file and then attempts to reboot. During this time, all traffic to the SSL module is dropped. When this condition exists, the **show task** command in debug mode displays suspended tasks on the SSL module. Workaround: Reboot the SSL module.
- **CSCej81355** - If the CSS successfully loads the CRL but then the CRL expires and the CSS is unable to obtain an updated version, the CSS may continue to allow new client connections.
- **CSCej83082** - On a CSS with an SSL-C module configured with custom methods and compression, if the CSS receives any connection with a request type that does not match methods in RFC 2616 or RFC 2518, the CSS closes the connection with a reset.

Software Version 8.10.0.02 Resolved Caveats

The following caveats were resolved in software version 8.10.0.02:

- **CSCeg64394** - In an ASR redundancy configuration, the sticky tables may not synchronize completely after the backup CSS is rebooted.
- **CSCeg69358** - When you configure the expiration time and date for a location cookie using the **location-cookie expiration** command, or the **arrowpoint-cookie expiration** command and the **advanced-balance arrowpoint-cookie command**, the CSS CPU may spike and the CSS may experience a degradation in its performance. Configure the expiration option with the location-cookie or the **arrowpoint-cookie expiration** command only when necessary.
- **CSCeh00709** - When you configure the CSS using the **IP advanced-route-remap** command, the command does not take effect on services that are local to the CSS.
- **CSCeh18228** - When you configure the CSS virtual router with a critical reporter that is in a Backup state, this places the virtual router into the Master(ReportBkup) state, which causes the CSS to incorrectly bring the dormant flows to an active state. The CSS should keep these flows in a dormant state until the reporter is master again.
- **CSCeh34493** - A backup CSS may reboot during a VIP redundancy config synch operation.
- **CSCeh34858** - A CSS running 7.40.1.07s with an SSL module and URL rewrite activated may not rewrite the URLs in 302 redirect answers from the servers if the "Location" word in the HTTP header spans two different TCP packets.
- **CSCeh35317** - In a Content Replication configuration using a UNIX directory structure on the publisher, if the publisher FTP server uses UserID/GroupID instead of UserName/GroupName in the directory listing, the CSS fails to detect the files for replication on the Publisher.
- **CSCeh35328** - In a Content Replication configuration, it was possible for the CSS to improperly send numerous test files to the Subscriber. In some cases, the Subscriber FTP server would detect this as an attack and deny FTP access from the CSS. This was changed so that the CSS will send no more than four test files per minute.
- **CSCeh38202** - Client authentication fails when the client certificate spans multiple packets.
- **CSCeh38676** - When ASR is configured, the ISC link will not come up unless the SCM is in slot 1.
- **CSCeh38890** - On a CSS 11503 or CSS 11506, the CSS may inject incorrect arrowpoint cookie expiration values.
- **CSCeh39182** - On networks that experience frequent packet losses and long transaction times, a configuration parameter is needed to deal with SSL transactions terminated on the CSS so the user can tune the retransmission timers to account for these delays.
- **CSCeh39266** - Running VIP/interface redundancy with a pair of CSSs connected to a Catalyst 6509/Supervisor 720, the GB ports on the backup CSS may fail unless the interfaces connected to the Catalyst are explicitly shut down using **admin-shutdown** command.
- **CSCeh44262** - For a CSS in a VIP/Interface redundant configuration, when a critical service transitioned from DOWN to BACKUP, the CSS would improperly GARP causing devices to update their ARP tables with incorrect information.
- **CSCeh45167** - On a CSS with an SSL module and URL rewrite activated, if non-standard ports are configured to be rewritten as well as the "https://", and the 3XX response from the server spans across multiple packets, only the "https" may be rewritten, but not the "port".
- **CSCeh45575** - When ASR is configured, the CSS may reboot during a VRRP transition.

- **CSCeh48648** - When the CSS was configured for backend remapping, the TCP RST ACK number sent to the backend server to close the connection was incorrect.
- **CSCeh49741** - When the CSS is configured for SSL termination, if a SSL handshake message contained multiple SSL messages inside a single record and the record size was greater then 1520 bytes, the resulting CSS behavior was incorrect. The CSS sent an SSL alert, rebooted, or failed to verify the SSL client certificate.
- **CSCeh49861** - When a CSS was configured with a DNS entry that was added to a content rule as well as configured as a proximity record, the CSS improperly freed some of the associated memory and rebooted.
- **CSCeh51008** - If a new client authentication certificate was placed on the CSS and you entered the **no ssl associate** command followed by the **ssl associate** command that contained a name that already existed in the ssl-proxy-list, and then you suspended and activated the server that was using the ssl-proxy-list, the CSS would reboot.
- **CSCeh53894** - On a CSS with an SSL module, the TCP acknowledge timer may become corrupt, causing the CSS to reboot.
- **CSCeh54012** - When a CSS was configured with a service type redirect and a long URL was requested, resulting in a redirect response from the CSS, the redirect was being logged. When the redirect string was logged, it was long enough to corrupt memory and caused the CSS to reboot.
- **CSCeh54652** - When configuring location cookie, the service types of ssl-accel-backend and ssl-init need to be permitted. Previously only **local** and **redirect** were allowed to be configured.
- **CSCeh56281** - The CSS may reboot when suspending a content rule due to internal rule tree corruption using Layer 5 rules containing a wildcardURL "/hraward*" and a header tag rule using the url "/home*". This is because both URLs begin with the same letter.
- **CSCeh57760** - The CSS may not NAT all ICMP error packets. The IP packet within the ICMP error is translated, but the encompassing ICMP error packet may not be NAT-translated before being sent out of the CSS.
- **CSCeh64254** - When typing the **show group** command on a group name that is not configured using specific arguments and you use the question mark (?) to get the list of available options, the CSS may reboot.
- **CSCeh65429** - When configuring the CSS to add an HTTP keepalive, you may see the following error message:

  ```
  Error %% Maximum keepalives of this type have been exceeded. Cannot activate when
  trying to add a new HTTP head keepalive.
  ```

- **CSCeh65531** - The debug mode flowmgr reset logging may cause the port number in the log message to be incorrect.
- **CSCeh68829** - When using advanced balance arrowpoint or location cookies, if the server packets are out of order and HTTP data arrives before the HTTP header, the CSS will not correctly adjust the tcp sequence number, resulting in corrupted data received on the client.
- **CSCeh70529** - With the CSS configured with an SSL module and URL rewrite activated, if the HTTP 3XX response from the server contained the tag "Content-Location:", the URL rewrite failed because the HTTP tag in the packet was modified. The CSS should modify the \r\nLocation: <>\r\n" tag only instead of any HTTP tag that contains the word "Location:".
- **CSCeh70874** - When using the commit_vip_redundancy script to sync a config that includes ACLs and has authChallenge configured on the APP session, if the session secret ends with the string "app", the commit may fail.

- **CSCeh71185** - On a CSS configured with a Layer 5 rule, when receiving a POST with multiple data packets, if one packet starts with the content "HEAD", it will be blocked by the CSS.
- **CSCeh75114** - When a POST is processed by the CSS, if the data that follows the POST begins with a CONNECT or GET, the CSS would erroneously interpret that to be an HTTP method. The CSS will now fully qualify all HTTP Methods to ensure that the POST data is not incorrectly processed as a valid HTTP method.
- **CSCeh76035** - The CSS may reboot. when configuring an RMON alarm, if you suspend, activate, suspend, and then enter the **no rmon-alarm** command.
- **CSCeh72177** - The CSS incorrectly rejects HTTP methods that have an authority form (for example, CONNECT) if the authority string has one '.' or more then three '.'.
- **CSCeh83740** - On a CSS with an SSL module configured with an SSL proxy list using a CRL and VIP/interface redundancy, the backup CSS does not download the CRL, causing DoS attacks.
- **CSCeh83762** - If the CSS was configured with services with encrypted http keepalives of type ssl-backend or ssl-initiation, memory may be leaked on the SSL module until eventually all memory blocks were depleted and user SSL traffic would cease.
- **CSCeh86543** - If the CSS is configured for SSL Termination using a CRL list and the SSL module was in the process of retrieving the CRL when the global CRL record was deleted on the SCM, the SSL module may reboot. This may also occur when you issue the **clear running-config** command.
- **CSCeh86555** - The CSS may reboot when enabling OSPF due to an OSPF LSA update that contained the maximum Ethernet packet size.
- **CSCeh87082** - If the CSS was configured for logging to an SMTP server, when the CSS opened an SMTP connection to the mail host, the CSS was incorrectly detecting the "continue" character of "-". This caused the CSS and the SMTP mail host to get out of sync in the SMTP protocol and the sendmail connection would be terminated by the CSS prematurely, causing the sendmail to fail.
- **CSCeh89126** - If the CSS is configured for client authentication, SSL handshake failures may occur after the CSS has been rebooted if the client authentication certificate spanned multiple packets.
- **CSCeh89468** - A CSS running 7.50.0.5s will not match the HTTP method CONNECT on a Layer 5 content rule. You must configure a Layer 3 or Layer 4 content rule or the CSS will reset the HTTP CONNECT method. The HTTP method CONNECT will now properly match on a configured Layer 5 content rule and be load balanced to the appropriate server.
- **CSCeh89398** - When trying to set and enable the SNTP server through the GUI on the CSS running 7.4.1.11s, the following error may occur:

  ```
  "An error occurred while processing your request. The request was not completed."
  ```

- **CSCeh97409** - If the CSS was configured with a protocol-only content rule (that is, "protocol tcp" but no "port") and the VIP range on the content rule was changed, a reboot was required for the configuration change to take effect even after suspending and activating the content rule.
- **CSCei00309** - The CSS may reboot if the ARP timing list has duplicate entries.
- **CSCei00983** - On a CSS with an SSL module, the available memory on the SSL module may drop significantly on a daily basis until all available memory was lost, severely impacting SSL traffic and requiring a reboot to recover the memory.
- **CSCei02447** - When an SSL module was configured for header insertion, the SSL header insertion was not occurring for all POSTs and potentially GETs if the HTTP header terminator spanned multiple packets.

- **CSCei04797** - The CSS was allowing a scripted keepalive under a service to be configured, even if the script did not exist. Once the service was activated, the following error message appeared in the show service command display:

  ```
  Script Error: Script failed to load. Is script present on disk?
  ```

- **CSCei08501** - The backup CSS does not download the CRL information in a box-to-box redundancy setup because the interfaces are not active. When the CSS moves from backup to master, the SSL module does not attempt to download the CRL after the interfaces become active. This prevents the backup CSS from having the correct CRL information until the first update is sent after it becomes the master CSS. Because of this condition, the backup CSS will not have the correct CRL information when it becomes the master CSS.
- **CSCei15420** - When a CSS is configured with VIP/Interface redundancy, critical reporters, and SNMP redundancy-transition traps enabled, it reboots when a reporter transitioned to down due to a string over-run on the trap text.
- **CSCei21776** - If the CSS receives a RST packet while a connection is already in the process of being shut down, the SSL module may reboot.
- **CSCei27622** - Invalid “SSL FINISHED” messages may cause the CSS SSL module to reset, thus causing the CSS to deny any SSL connections. When the offending packet is no longer sent to the CSS and the timer expiration causes the SSL module to reset, the CSS will start accepting new connections.
- **CSCei31328** - When you configure client authentication on an SSL module, the SSL module may incorrectly reuse the session ID with different VIPs.
- **CSCei31463** - VRRP traps may no longer be sent by the backup CSS when the commit_redundancy script is run.
- **CSCei33610** - When an SSL module is configured with the global configuration mode **http-method parse RFC2518-methods** command and the SSL-proxy-list configuration mode **ssl-server 20 http-header static "WL-Proxy-SSL: true"** command, the custom header is not seen when the RFC2518 PROPFIND header is present.
- **CSCei35940** - The following new log message was added for a source group mis-configuration where 'index' is the internal source group index value. However the log message is only logged if an internal source group debug flag "FwPortMapLogging" is enabled, which can only be done using symbols in debug mode. This may cause confusion when tracking log messages because the log message should be at warning, info, or debug level logging.
- **CSCei40272** - When using an SSL module, there may be packets that are being seen on the client-side connection that are believed to be destined to the SSL module.
- **CSCei47195** - After rebooting the CSS, the isc-port reports LifeTick failures that may not cause session replication to occur correctly because the peers are not passing messages across the isc-port. Workaround: To enable messages to be passed correctly, remove and re-add the isc-port that is experiencing the issue.
- **CSCei49115** - Creating a service using the CSS GUI may result in the following error message:

  ```
  "An error has occurred while processing your service configuration request."
  ```

  Though the service may get created, the keepalive parameters may not be displayed in the CLI and the service will not be activated. Workaround: Add the service using the CLI.
- **CSCei55203** - The CSS does not receive CRLs when booting even though it is able to resolve DNS requests. Workaround: Use an IP address instead of a hostname in the CRL record to avoid this issue.

- **CSCei55727** - Software version 7.40.2.02 enables you to suspend and activate the ssl-proxy list without making changes to the type ssl-init, ssl-accel, and ssl-accel-backend services. If you make changes to your backend-server configuration and suspend and activate the ssl-proxy list, the CSS will not process traffic correctly.
- **CSCei81533** - The CSS leaks a TCPFAST application source port when the CSS received a TCP FIN and the CSS was in the process of closing the connection. The leaking of source ports caused services to remain in the DOWN state.
- **CSCei88708** - A CSS that contains an SSL module may reboot due to improperly handling an error condition when it closes a connection.
- **CSCei91293** - When the CSS is configured for SSL termination using the HTTP-header insertion feature and the **ssl-server** *number* **http-header insert-per-request** command, and it received HTTP POST requests that spanned multiple packets, the SSL module incorrectly inserted the static HTTP header into multiple packets in the spanned POST requests. This incorrect insertion caused the connection to fail.
- **CSCej01719** - When you configure the CSS with an ACL preferred service clause and a source group that both matched an incoming ICMP ECHO request, the CSS properly performs source NAT on the ICMP request but does not properly forward the request to the preferred service in the matched ACL clause.
- **CSCej12554** - The CSS may provide the wrong MAC address for the VIP address or not properly handle VIP load-balanced traffic if the CSS VIP address is inserted into the internal CSS ARP or routing tables.
- **CSCej12745** - If you configure a service with the ap-kal-pinglist scripted keepalive, the service would be in the wrong service state if one of the script arguments is a local VIP address on this CSS.
- **CSCej14453** - The CSS may reboot when trying to import or export an SSL file using SFTP.
- **CSCej17291** - When you configure the CSS for SSL termination, it may fail to complete an SSL connection and issue an alert when the server combines multiple SSL messages into a single record layer message.
- **CSCej30229**- In some cases, the SSL module inserts an extra byte into the SSL record causing all of the subsequent bytes in the record to decode incorrectly. Thus, the client cannot find the next SSL record header and the session falls apart with "short record" errors.
- **CSCej34375** - The CSS SSL backend-server IP and server addresses, and their port values must be unique. If they are not unique, the following error message appears: `%% Backend-server ip/server address and port values must form unique tuples`.
- **CSCej35592** - If you configure the number of hours before you update the CRL to 0, the CSS may reboot.
- **CSCej46421** - The CSS may reboot when the CSS SNMP agent receives an SNMP bulk NEXT request and one of the SNMP OID requests returns an error.
- **CSCej64552** - During an FTP session, if you enter a list (**ls**) command with a pathname greater than 256 characters, the CSS reboots.
- **CSCej72467** - Occasionally, the CSS SSL module may leak chunks of memory causing the CSS to run out of sessions and to be unable to accept new incoming connections.
- **CSCej72718** - On a CSS configured with URL rewrite, if the CSS cannot find the http:// value in the expected Location: field, the CSS may perform the URL rewrite incorrectly and reboot.

Software Version 8.10.0.02 Command Changes

Table 12 lists the commands that have been added in software version 8.10.0.02. Table 13 lists the commands that have changed in software version 8.10.0.02.

Table 12 CLI Commands Added in Version 8.10.0.02

Mode	Command and Syntax	Description
All modes	**show chassis coprocessors**	The new **coprocessors** option displays compression coprocessor information in the SSL compression module including the chassis slot and subslot, the type, software version, and state of the flash location for the image. For information about the fields in the command output, refer to the *Cisco Content Services Switch SSL Configuration Guide*.
	show inventory	Displays the Unique Device Identifier (UDI) information for the chassis and the modules in the CSS. The UDI information includes the product identifier (PID), version number (VID), and serial number for the chassis and the modules. For information about the fields in the **show inventory** command output, refer to the *Cisco Content Services Switch Administration Guide*.
	zero service compression	The new **compression** keyword sets the compression statistics displayed by the **show service** command to zero.
SuperUser	**copy ftp** *ftp_record filename* **gui-image**	The new **gui-image** keyword copies the CiscoView Device Manager (CVDM) zip file onto the CSS hard drive. For more information on this command, see the *Cisco Content Services Switch Getting Started Guide*.
Boot	**remove-gui**	Removes CiscoView Device Manager (CVDM) from the CSS. For more information on this command, see the *Cisco Content Services Switch Getting Started Guide*.
	unpack-gui *filename*	Unpacks the CiscoView Device Manager (CVDM) on the CSS. For more information on this command, see the *Cisco Content Services Switch Getting Started Guide*.

Table 12 CLI Commands Added in Version 8.10.0.02 (continued)

Mode	Command and Syntax	Description
Service	**compress accept-omit deflate|gzip|identity**	Specifies the compression encoding type for HTTP requests that do not include the Accept-Encoding field. By default, the compression encoding type is set to **identity**, bypassing compression. The keywords are: • **deflate** - The deflate compression encoding type • **gzip** - The gzip compression encoding type • **identity** - The cleartext identity encoding type that bypasses the flow for compression For information on this command, refer to the *Cisco Content Services Switch SSL Configuration Guide*.
	compress disable	Disables HTTP compression on a service. By default, compression is not enabled on a service. For information on this command, refer to the *Cisco Content Services Switch SSL Configuration Guide*.
	compress encode auto|deflate|gzip|force-deflate|force-gzip	Configures the CSS to prefer a compression encoding type provided by the Accept-Encode field from the client or the **compress accept-omit** command. By default, the preferred compression encoding type is set to **auto**. The keywords are: • **auto** - The CSS uses the compression algorithm token in the Accept-Encoding field in the HTTP request from the client. From a list of tokens in the Accept-Encoding field, the CSS has the following preference for applying the compression algorithm: deflate, gzip, and then identity. If the field does not contain a compression type or does not exist, the CSS uses the setting in the **compress accept-omit** command. • **deflate** - The CSS prefers the deflate algorithm for compression and bypasses the flow for compression for the identity or gzip compression encoding type. • **gzip** - The CSS prefers the gzip algorithm and bypasses the flow for compression for the identity or deflate compression encoding type. • **force-deflate** - The CSS always uses the deflate algorithm for encoding, except if the Accept-Encoding field contains an identity entry. • **force-gzip** - The CSS always uses the gzip algorithm for encoding, except if the Accept-Encoding field contains an identity entry. For information on this command, refer to the *Cisco Content Services Switch SSL Configuration Guide*.
	compress enable	Enables compression on a service to allow the compression of HTTP response data to the client. By default, compression is not enabled on a service. For information on this command, refer to the *Cisco Content Services Switch SSL Configuration Guide*.

Table 12 CLI Commands Added in Version 8.10.0.02 (continued)

Mode	Command and Syntax	Description
Service (cont.)	**compress tcp buffer-share** [**rx**|**tx**] *bytes* **no compress tcp buffer-share** [**rx**|**tx**]	Sets the TCP buffering from the client or server on a given connection for HTTP compression. • To set the amount of data in bytes that a given connection can buffer from the client traffic, use the **rx** *bytes* keyword and variable. • To set the amount of data in bytes that a given connection can buffer from the server to the client, use the **tx** *bytes* keyword and variable. By default, the receive (RX) buffer size is 32768. The default transmit (TX) buffer size is 65536. The range of the buffer size for RX and TX is from 16400 to 262144. Use the **no** form of this command to reset the default values. For information on this command, refer to the *Cisco Content Services Switch SSL Configuration Guide*.
	compress tcp [**server**|**virtual**] **inactivity-timeout** *seconds*|**nagle** [**enable**|**disable**]|**syn-timeout** *seconds2*|**retrans** *ms*|**ack-delay** *value* **no compress tcp** [**server**|**virtual**] **inactivity-timeout** |**syn-timeout**|**retrans**|**ack-delay**	Configures a client or server TCP connections for HTTP compression. You can specify: • **inactivity-timeout** *seconds* - The timeout value that the CSS waits to receive flows before terminating the TCP connection. Enter a TCP inactivity timeout value in seconds, from 0 disabling the TCP inactivity timeout to 3600 (1 hour). The default is 240 seconds. • **nagle** [**enable**|**disable**] - The Nagle algorithm for the TCP connection. By default, the Nagle algorithm is enabled for each TCP connection. – Use the **disable** keyword to disable the Nagle algorithm when you observe a delay on the TCP connection. – Use the **enable** keyword to reenable the Nagle algorithm. • **syn-timeout** *seconds2* - A timeout value that the CSS uses to terminate a TCP connection for inactivity or an unsuccessful TCP three-way handshake. Enter a TCP SYN timeout value in seconds, from 1 to 3600 (1 hour). The default is 30 seconds. • **retrans** *ms* - The retransmission time for TCP transactions. The *ms* variable is the minimum time in milliseconds for retransmitting TCP transactions. Enter a number from 50 to 500. The default value is 500. • **ack-delay** *value* - The time length for delayed acknowledgements. The *value* variable is the timer length in milliseconds (ms) for delayed acknowledgements. The default value is 200. Enter a value from 0 to 10000. A value of 0 disables the acknowledgement delay in receiving traffic from the client or server. Use the **no** form of this command to reset the default values. For information on this command, refer to the *Cisco Content Services Switch SSL Configuration Guide*.

Table 12 CLI Commands Added in Version 8.10.0.02 (continued)

Mode	Command and Syntax	Description
Service (cont.)	**compress type ascii | default | numeric**	Configures the Huffman code type to optimize the compression for different traffic types. By default, the Huffman code is set to **ascii**. The keywords are • **ascii** - ASCII alpha-numeric optimized Huffman code for traffic that is mostly text (for example, HTML, XML, TXT) • **default** - Default Huffman code defined by the deflate algorithm for mixed traffic • **numeric** - ASCII numeric optimized Huffman code for traffic that is mostly ASCII numbers (for example, 0 through 9) For information on this command, refer to the *Cisco Content Services Switch SSL Configuration Guide*.
SSL-proxy-list	**no** [**backend-server | ssl-server**] *number* **tcp** [**virtual | server**] **ack-delay**	Resets the timer length for delayed acknowledgements to 200 milliseconds (ms).
	no ssl-server *number* **tcp virtual retrans**	Resets the retransmission timer for TCP transactions to 500 milliseconds.

Table 13 lists the commands that changed in software version 8.10.0.02.

Table 13 CLI Commands Changed in Version 8.10.0.02

Mode	Command and Syntax	Description
SSL-proxy-list	**backend-server | ssl-server number tcp server | virtual syn-timeout**	The SSL client and server TCP SYN timeout range changed from 0 to 3600 to 1 to 3600.

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Subscribe to the *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS Version 2.0.