show statistics icmp

To display SB Internet Control Message Protocol (ICMP) statistics, use the **show statistics icmp** command in EXEC configuration mode.

show statistics icmp

Syntax Description

This command has no arguments or keywords.

Defaults

None

Command Modes

EXEC configuration mode.

Usage Guidelines

ICMP messages are sent in several situations, such as when a datagram cannot reach its destination, when the gateway does not have the buffering capacity to forward a datagram, and when the gateway can direct the host to send traffic on a shorter route. The purpose of these control messages is to provide feedback about problems in the communication environment, not to make IP reliable. There is still no guarantee that a datagram is delivered or a control message is returned. Some datagrams may still be undelivered without any report of their loss.

The ICMP messages typically report errors in the processing of datagrams. To avoid the infinite regress of messages about messages, no ICMP messages are sent about ICMP messages. Also, ICMP messages are only sent about errors in handling fragment zero of fragmented datagrams.

ICMP messages are sent using the basic IP header. The first octet of the data portion of the datagram is on a ICMP type field; the value of this field determines the format of the remaining data.

Many of the type fields contain more specific information about the error condition identified by a code value. ICMP messages have two types of codes:

- Query
- Error

Queries contain no additional information because they ask for information and show a value of 0 in the code field. ICMP uses the queries as shown in Table 4-1.

Table 4-1 Queries

Query	Type Field Value
Echo Reply	0
Echo Request	8
Router Advertisement	9
Router Solicitation	10
Time-stamp Request	13
Time-stamp Reply	14
Information Request (obsolete)	15

Table 4-1 Queries (continued)

Query	Type Field Value
Information Reply (obsolete)	16
Address Mask Request	17
Address Mask Reply	18

Error messages give specific information and have varying values that further describe conditions. Error messages always include a copy of the offending IP header and up to 8 bytes of the data that caused the host or gateway to send the error message. The source host uses this information to identify and fix the problem reported by the ICMP error message. ICMP uses the error messages as shown in Table 4-2.

Table 4-2 Errors

Error	Type Field Value
Destination Unreachable	3
Source Quench	4
Redirect	5
Time Exceeded	11
Parameter Problems	12

Table 4-3 describes the fields shown in the **show statistics icmp** display.

Table 4-3 show statistics icmp Field Descriptions

Field	Description
ICMP messages received	Total number of ICMP messages received by the SB.
ICMP messages receive failed	Total number of ICMP messages that were not received by the SB.
Destination unreachable	Number of destination-unreachable ICMP packets received by the SB. A destination-unreachable message (Type 1) is generated in response to a packet that cannot be delivered to its destination address for reasons other than congestion. The reason for the nondelivery of a packet is described by the code field value. Destination-unreachable packets use the code field values to further describe the function of the ICMP message being sent.

Table 4-3 show statistics icmp Field Descriptions (continued)

Field	Description
Timeout in transit	Number of ICMP time-exceeded packets received by the SB. The time-exceeded message occurs when a router receives a datagram with a TTL of 0 or 1. IP uses the TTL field to prevent infinite routing loops. A router cannot forward a datagram that has a TTL of 0 or 1. Instead, it trashes the datagram and sends a time-exceeded message. Two different time-exceeded error codes can occur, as follows: • 0 = Time-To-Live Equals 0 During Transit • 1 = Time-To-Live Equals 0 During Reassembly A router cannot forward a datagram with a TTL of 0 or 1 both during transit or reassembly. The TTL timer is measured, in seconds, and originally was used before the existence of routers to guarantee that a datagram did not live on the Internet forever. Each gateway processing a datagram reduces this value by at least one if it takes longer to process and forward the datagram. When this value expires, the gateway trashes the datagram and sends a message back to the sender notifying the host of the situation.
Wrong parameters	Number of ICMP packets with parameter problems received by the SB. An IP datagram that has been received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 12 denote a parameter problem on a datagram. ICMP parameter-problem datagrams are issued when a router has had to drop a malformed datagram. This condition is a normal and necessary type of network traffic; however, large numbers of this datagram type on the network can indicate network difficulties or hostile actions. A host or gateway can send this message when no other ICMP message covering the problem can be used to alert the sending host.
Source quenches	Number of ICMP source-quench packets received by the SB. A receiving host generates a source-quench message when it cannot process datagrams at the speed requested because of a lack of memory or internal resources. This message serves as a simple flow control mechanism that a receiving host can use to alert a sender to slow down its data transmission. When the source host receives this message, it must pass this information on to the upper-layer process, such as TCP, which then must control the flow of the application's data stream. A router generates this message when, in the process of forwarding datagrams, it has run low on buffers and cannot queue the datagram for delivery.

Table 4-3 show statistics icmp Field Descriptions (continued)

Field	Description
Redirects	Number of ICMP redirect packets received by the SB. A router sends a redirect error to the sender of an IP datagram when the sender should have sent the datagram to a different router or directly to an end host (if the end host is local). The message assists the sending host to direct a misdirected datagram to a gateway or host. This alert does not guarantee proper delivery; the sending host has to correct the problem if possible. Only gateways generate redirect messages to inform source hosts of misguided datagrams. A gateway receiving a misdirected frame does not trash the offending datagram if it can forward it.
Echo requests	Number of echo ICMP packets received by the SB. An echo request is an IP datagram that has been received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 8. The ICMP echo request is issued by the source to determine if the destination is alive. When the destination receives the request, it replies with an ICMP echo reply. This request and reply pair is most commonly implemented using the ping utility. Many network management tools use this utility or some derivative of it, and this condition is common as a part of network traffic. **Note** You should be suspicious when a large number of these packets are found on the network.
Echo replies	Number of echo-reply ICMP packets received by the SB. An echo reply is the message that is generated in response to an echo request message. An echo reply is an IP datagram that has been received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 0. This condition is common as a part of network traffic. **Note** You should be suspicious when a large number of these packets are found on the network.
Timestamp requests	Number of ICMP time stamp request packets received by the SB. An ICMP time stamp request is an IP datagram that has been received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 13. The ICMP time stamp request and reply pair can be used to synchronize system clocks on the network. The requesting system issues the time stamp request bound for a destination, and the destination system responds with a time stamp reply message. This condition is normal as a part of network traffic but is uncommon on most networks. **Note** You should be suspicious when a large number of these packets are found on the network.

Table 4-3 show statistics icmp Field Descriptions (continued)

Field	Description
Timestamp replies	Number of ICMP time stamp reply packets received by the SB. time stamp request and reply messages work in tandem. You have the option of using time stamps. When used, a time stamp request permits a system to query another for the current time. It expects a recommended value returned to be the number of milliseconds since midnight, UTC. This message provides millisecond resolution. The two systems compare the three time stamps and use a round-trip time to adjust the sender's or receiver's time if necessary. Most systems set the transmit and receive time as the same value.
Address mask requests	Number of ICMP address mask request packets received by the SB. An ICMP address mask request is an IP datagram that has been received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 17. ICMP address mask requests could be used to perform reconnaissance sweeps of networks. The ICMP address mask request and reply pair can be used to determine the subnet mask used on the network. When the requesting system issues the address mask request bound for a destination, the destination system responds with an address mask reply message. This condition can be a part of normal network traffic but is uncommon on most networks. **Note** You should be suspicious when a large number of these packets are found on the network.
Address mask replies	Number of ICMP address mask reply packets received by the SB. An address mask ICMP reply is an IP datagram that has been received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 18. No known exploits incorporate this option. The ICMP address mask request and reply pair can be used to determine the subnet mask used on the network. When the requesting system issues the address mask request bound for a destination, the destination system responds with an address mask reply message. This condition can be a part of normal network traffic but is uncommon on most networks. **Note** You should be suspicious when a large number of these packets are found on the network.
ICMP messages sent	Total number of ICMP messages sent by the SB.
ICMP messages send failed	Total number of ICMP messages that failed to be sent by the SB.
Destination unreachable	Number of destination-unreachable ICMP packets sent by the SB.
Timeout in transit	Number of ICMP time-exceeded packets sent by the SB.
Wrong parameters	Number of ICMP packets with parameter problems sent by the SB.
Source quenches	Number of ICMP source-quench packets sent by the SB.
Redirects	Number of ICMP redirect packets sent by the SB.
Echo requests	Number of echo ICMP packets sent by the SB.

Table 4-3 show statistics icmp Field Descriptions (continued)

Field	Description
Echo replies	Number of echo-reply ICMP packets sent by the SB.
Timestamp requests	Number of ICMP time stamp request packets sent by the SB.
Timestamp replies	Number of ICMP time stamp reply packets sent by the SB.
Address mask requests	Number of ICMP address mask requests sent by the SB.
Address mask replies	Number of ICMP address mask replies sent by the SB.

Related Commands

Command	Description
clear statistics	Clears the statistics settings.

show statistics ip

To display the IP statistics, use the **show statistics ip** command in user EXEC configuration mode.

show statistics ip

Syntax Description

This command has no arguments or keywords.

Defaults

None

Command Modes

User EXEC configuration mode.

Examples

The following is sample output from the **show statistics ip** command:

```
ServiceRouter# show statistics ip

IP statistics
-------------
Total packets in                = 1408126
 with invalid header            = 0
 with invalid address           = 0
 forwarded                      = 0
 unknown protocol               = 0
 discarded                      = 0
 delivered                      = 1408126
Total packets out               = 1500110
 dropped                        = 0
 dropped (no route)             = 0
Fragments dropped after timeout  = 0
Reassemblies required           = 0
Packets reassembled             = 0
Packets reassemble failed       = 0
Fragments received              = 0
Fragments failed                = 0
Fragments created               = 0

ServiceRouter#
```

Table 4-4 describes the fields shown in the **show statistics ip** display.

Table 4-4 show statistics ip Field Descriptions

Field	Description
Total packets in	Total number of input datagrams received from interfaces, including those received in error.
with invalid header	Number of input datagrams discarded because of errors in their IP headers, including bad checksums, version number mismatch, other format errors, Time To Live exceeded, errors discovered in processing their IP options, and so on.

Table 4-4 show statistics ip Field Descriptions (continued)

Field	Description
with invalid address	Number of input datagrams discarded because the IP address in the IP header's destination field was not a valid address to be received at this entity. This count includes invalid addresses (for example, 0.0.0.0) and addresses of unsupported classes (for example, Class E). For entities that are not IP routers and do not forward datagrams, this counter includes datagrams discarded because the destination address was not a local address.
forwarded	Number of input datagrams for which this entity was not the final IP destination, but the SB attempted to find a route to forward them to that final destination. In entities that do not act as IP routers, this counter includes only those packets that were source-routed through this entity, and the source-route option processing was successful.
unknown protocol	Number of locally addressed datagrams received successfully but discarded because of an unknown or unsupported protocol.
discarded	Number of input IP datagrams that were discarded even though the datagrams encountered no problems to prevent their continued processing. This counter does not include any datagrams discarded while awaiting reassembly.
delivered	Total number of input datagrams successfully delivered to IP user protocols (including ICMP).
Total packets out	Total number of IP datagrams that local IP user protocols (including ICMP) supplied to IP in requests for transmission. This counter does not include any datagrams counted in the forwarded field.
dropped	Number of output IP datagrams that were discarded even though the datagrams encountered no problems that would prevent their transmission to their destination. This counter would include datagrams counted in the forwarded field if any such packets met this (discretionary) discard criterion.
dropped (no route)	Number of IP datagrams that were discarded because the SB found no route to send them to their destination. This counter includes any packets counted in the forwarded field that meet this no-route criterion including any datagrams that a host cannot route because all its default routers are down.
Fragments dropped after timeout	Number of received fragments at this entity that are dropped after being held for the maximum number of seconds while awaiting reassembly at this entity.
Reassemblies required	Number of IP fragments received that needed to be reassembled at this entity.
Packets reassembled	Number of IP datagrams successfully reassembled.

Table 4-4 show statistics ip Field Descriptions (continued)

Field	Description
Packets reassemble failed	Number of failures detected by the IP reassembly algorithm (because of reasons such as timed out and errors.) This counter is not necessarily a count of discarded IP fragments because some algorithms (notably the algorithm in RFC 815) can lose track of the number of fragments by combining them as they are received.
Fragments received	Number of IP datagrams that have been successfully fragmented at this entity.
Fragments failed	Number of IP datagrams that have been discarded because they needed to be fragmented at this entity but could not be fragmented for reasons such as the Don't Fragment flag was set.
Fragments created	Number of IP datagram fragments that have been generated because of fragmentation at this entity.

Related Commands

Command	Description
clear statistics ip	Clears IP statistics counters.
ip	Configures the IP.
show ip routes	Displays the IP routing table.

show statistics lsof

To display the List of Open File (lsof) descriptors, use the **show statistics lsof** command in EXEC configuration mode.

show statistics lsof

Syntax Description

This command has no arguments or keywords.

Defaults

None

Command Modes

EXEC configuration mode.

Examples

The following example shows to display the lsof descriptors:

```
ServiceEingine# show statistics lsof
COMMAND     PID     USER   FD      TYPE             DEVICE     SIZE       NODE
NAME
init          1    admin  cwd       DIR                1,0     1024          2
/
init          1    admin  rtd       DIR                1,0     1024          2
/
init          1    admin  txt       REG                1,0    45436       7488
/sbin/init
init          1    admin  mem       REG                1,0  1852502       6566
/lib/libc-2.13.so
init          1    admin  mem       REG                1,0   154528       2006
/lib/ld-2.13.so
init          1    admin   10u     FIFO               0,13                4069
/dev/initctl
kthreadd      2    admin  cwd       DIR                1,0     1024          2
/
kthreadd      2    admin  rtd       DIR                1,0     1024          2
/
kthreadd      2    admin  txt   unknown
/proc/2/exe
migration     3    admin  cwd       DIR                1,0     1024          2
/
migration     3    admin  rtd       DIR                1,0     1024          2
/

<Output truncated>
```

show statistics netstat

To display SB Internet socket connection statistics, use the **show statistics netstat** command in EXEC configuration mode.

show statistics netstat

Syntax Description

This command has no arguments or keywords.

Defaults

None

Command Modes

EXEC configuration mode.

Usage Guidelines

Table 4-5 describes the fields shown in the **show statistics netstat** display.

Table 4-5 show statistics netstat Field Descriptions

Field	Description
Proto	Layer 4 protocol used on the Internet connection, such as TCP, UDP, and so forth.
Recv-Q	Amount of data buffered by the Layer 4 protocol stack in the receive direction on a connection.
Send-Q	Amount of data buffered by the Layer 4 protocol stack in the send direction on a connection.
Local Address	IP address and Layer 4 port used at the device end point of a connection.
Foreign Address	IP address and Layer 4 port used at the remote end point of a connection.
State	Layer 4 state of a connection. TCP states include the following: ESTABLISHED, TIME-WAIT, LAST-ACK, CLOSED, CLOSED-WAIT, SYN-SENT, SYN-RCVD, SYN-SENT, SYN-ACK-SENT, and LISTEN.

show statistics radius

To display SB RADIUS authentication statistics, use the **show statistics radius** command in EXEC configuration mode.

show statistics radius

Syntax Description

This command has no arguments or keywords.

Defaults

None

Command Modes

EXEC configuration mode.

Usage Guidelines

The fields in the **show statistics radius** display are as follows:

- Number of access requests
- Number of access deny responses
- Number of access allow responses
- Number of authorization requests
- Number of authorization failure responses
- Number of authorization success responses

Related Commands

Command	Description
clear statistics	Clears the statistics settings.
radius-server	Configures the RADIUS authentication.
show radius-server	Displays the RADIUS server information.

show statistics services

To display SB services statistics, use the **show statistics services** command in EXEC configuration mode.

show statistics services

Syntax Description

This command has no arguments or keywords.

Defaults

None

Command Modes

EXEC configuration mode.

Usage Guidelines

Table 4-6 describes the fields shown in the **show statistics services** display.

Table 4-6 show statistics services Field Descriptions

Field	Description
Port Statistics	Service-related statistics for each port on the WAAS[1] device.
Port	Port number.
Total Connections	Number of total connections.

1. WAAS = Wide Area Application Service

Related Commands

Command	Description
show services	Displays the services-related information.

show statistics snmp

To display SB Simple Network Management Protocol (SNMP) statistics, use the **show statistics snmp** command in EXEC configuration mode.

show statistics snmp

Syntax Description

This command has no arguments or keywords.

Defaults

None

Command Modes

EXEC configuration mode.

Usage Guidelines

Table 4-7 describes the fields shown in the **show statistics snmp** display.

Table 4-7 show statistics snmp Field Descriptions

Field	Description
SNMP packets input	Total number of SNMP packets input.
Bad SNMP version errors	Number of packets with an invalid SNMP version.
Unknown community name	Number of SNMP packets with an unknown community name.
Illegal operation for community name supplied	Number of packets requesting an operation not allowed for that community.
Encoding errors	Number of SNMP packets that were improperly encoded.
Number of requested variables	Number of variables requested by SNMP managers.
Number of altered variables	Number of variables altered by SNMP managers.
Get-request PDUs	Number of GET requests received.
Get-next PDUs	Number of GET-NEXT requests received.
Set-request PDUs	Number of SET requests received.
SNMP packets output	Total number of SNMP packets sent by the router.
Too big errors	Number of SNMP packets that were larger than the maximum packet size.
Maximum packet size	Maximum size of SNMP packets.
No such name errors	Number of SNMP requests that specified a MIB object that does not exist.
Bad values errors	Number of SNMP SET requests that specified an invalid value for a MIB object.

Table 4-7 show statistics snmp Field Descriptions (continued)

Field	Description
General errors	Number of SNMP SET requests that failed because of some other error. (It was not a No such name error, Bad values error, or any of the other specific errors.)
Response PDUs	Number of responses sent in reply to requests.
Trap PDUs	Number of SNMP traps sent.

Related Commands

Command	Description
show snmp	Displays the SNMP parameters.
snmp-server community	Configures the community access string to permit access to the SNMP.
snmp-server contact	Sets the system server contact string.
snmp-server enable	Enables the SB to send SNMP traps.
snmp-server group	Defines a user security model group.
snmp-server host	Specifies the hosts to receive SNMP traps.
snmp-server location	Sets the SNMP system location string.
snmp-server notify inform	Configures the SNMP notify inform request.
snmp-server user	Defines a user who can access the SNMP engine.

show statistics tacacs

To display Service Broker TACACS+ authentication and authorization statistics, use the **show statistics tacacs** command in user EXEC configuration mode.

show statistics tacacs

Syntax Description

This command has no arguments or keywords.

Defaults

None

Command Modes

User EXEC configuration mode.

Usage Guidelines

The fields shown in the **show statistics tacacs** display for the Service Broker are as follows:

- Number of access requests
- Number of access deny responses
- Number of access allow responses
- Number of authorization requests
- Number of authorization failure responses
- Number of authorization success responses
- Number of accounting requests
- Number of accounting failure responses
- Number of accounting success responses

Related Commands

Command	Description
clear tacacs	Clears the TACACS+ settings.
show tacacs	Displays TACACS+ authentication protocol configuration information.
tacacs	Configures TACACS+ server parameters.

show statistics tcp

To display SB Transmission Control Protocol (TCP) statistics, use the **show statistics tcp** command in EXEC configuration mode.

show statistics tcp

Syntax Description

This command has no arguments or keywords.

Defaults

None

Command Modes

EXEC configuration mode.

Usage Guidelines

Table 4-8 describes the fields shown in the **show statistics tcp** display.

Table 4-8 show statistics tcp Field Descriptions

Field	Description
Server connection openings	Number of connections opened from the SB to the server.
Client connection openings	Number of connections opened from the client to the SB.
Failed connection attempts	Number of incoming SYN connections rejected because of rate limiting or resource shortage.
Connections established	Number of incoming connections that have been set up.
Connections resets received	Number of RSTs[1] received by the SB.
Connection resets sent	Number of RSTs sent by the SB.
Segments received	Number of TCP segments received from the client and the server. The value of this field is almost equal to the sum of the values of the Server segments received and the Client segments received fields.
Segments sent	Number of TCP segments sent by the client and the server. The value of this field is almost equal to the sum of the values of the Server segments sent and the Client segments sent fields.
Bad segments received	Number of incoming segments dropped because of checksum or being outside the TCP window.
Segments retransmitted	Number of TCP segments retransmitted by the client and the server. The value of this field is almost equal to the sum of the values of the Server segments retransmitted and the Client segments retransmitted fields.

Table 4-8 show statistics tcp Field Descriptions (continued)

Field	Description
Retransmit timer expirations	Number of times that the TCP retransmit timer expires. The TCP sender uses a timer to measure the time that has elapsed between sending a data segment and receiving the corresponding ACK from the receiving side of the TCP transmission. When this retransmit timer expires, the sender (according to the RFC standards for TCP congestion control) must reduce its sending rate.
Server segments received	Number of TCP segments received by the SB from the server.
Server segments sent	Number of TCP segments sent by the SB to the server.
Server segments retransmitted	Number of TCP segments retransmitted by the SB from the server.
Client segments received	Number of TCP segments received by the SB from the client.
Client segments sent	Number of TCP segments sent by the SB to the server.
Client segments retransmitted	Number of TCP segments retransmitted by the SB to the client.
Sync cookies sent	Number of SYN^2 cookies sent by the SB. TCP requires unacknowledged data to be retransmitted. The server is supposed to retransmit the SYN.ACK packet before giving up and dropping the connection. When SYN.ACK arrives at the client but the ACK gets lost, there is a disparity about the establishment state between the client and server. Typically, this problem can be solved by the server's retransmission. But in the case of a SYN cookie, there is no state kept on the server and retransmission is impossible.
Sync cookies received	Number of SYN cookies received by the SB. The entire process of establishing the connection is performed by the ACK packet sent by the client, making the connection process independent of the preceding SYN and SYN.ACK packets. This type of connection establishment opens the possibility of ACK flooding, in the hope that the client has the correct value to establish a connection. This method also allows you to bypass firewalls that normally only filter packets with SYN bit set.
Sync cookies failed	Number of SYN cookies rejected by the SB. The SYN cookies feature attempts to protect a socket from a SYN flood attack. This feature is a violation of TCP and conflicts with other areas of TCP such as TCP extensions. It can cause problems for clients and relays. We do not recommend that you use this feature as a tuning mechanism for heavily loaded servers to help with overloaded or misconfigured conditions.
Embryonic connection resets	Number of TCP connections that have been reset before the SB accepted the connection.
Prune message called	Number of calls that the SB makes to the function that tries to reduce the number of received but not acknowledged packets.
Packets pruned from receive queue	Number of packets that the TCP drops from the receive queue (usually because of low memory).

Table 4-8 show statistics tcp Field Descriptions (continued)

Field	Description
Out-of-order-queue pruned	Number of times that the packet was dropped from the out-of-order queue.
Out-of-window Icmp messages	Number of ICMP packets that were outside the TCP window and dropped.
Lock dropped Icmp messages	Number of ICMP packets that hit a locked (busy) socket and were dropped.
Arp filter	Number of ARPs[3] not sent because they were meant for the SB.
Time-wait sockets	Number of current sockets in the TIME-WAIT state. The TIME-WAIT state removes old duplicates for fast or long connections. The clock-driven ISN selection is unable to prevent the overlap of the old and new sequence spaces. The TIME-WAIT delay allows enough time for all old duplicate segments to die in the Internet before the connection is reopened.
Time-wait sockets recycled	Number of TIME-WAIT sockets that were recycled (the address or port was reused before the waiting period was over). In TCP, the TIME-WAIT state is used as protection against old duplicate segments
Time-wait sockets killed	Number of TIME-WAIT sockets that were terminated to reclaim memory.
PAWS passive	Number of passive connections that were made with PAWS[4] numbers enabled. PAWS operates within a single TCP connection using a state that is saved in the connection control block.
PAWS active	Number of active connections that were made with PAWS enabled. PAWS uses the same TCP time stamps as the round-trip time measurement mechanism and assumes that every received TCP segment (including the data and ACK segments) contains a time stamp SEG.TSval that has values that are monotone and nondecreasing in time. A segment can be discarded as an old duplicate if it is received with a time stamp SEG.TSval less than some time stamp recently received on this connection.
PAWS established	Number of current connections that were made with PAWS enabled.
Delayed acks sent	Number of delayed ACK counters sent by the SB.
Delayed acks blocked by socket lock	Number of delayed ACK counters that were blocked because the socket was in use.
Delayed acks lost	Number of delayed ACK counters lost during transmission.
Listen queue overflows	Number of times that the three-way TCP handshake was completed, but enough space was not available in the listen queue.
Connections dropped by listen queue	Number of TCP connections dropped because of a resource shortage.

Table 4-8 show statistics tcp Field Descriptions (continued)

Field	Description
TCP packets queued to prequeue	Number of TCP packets queued to the prequeue.
TCP packets directly copied from backlog	Number of TCP packets delivered to the client from the backlog queue. Packets are queued in the backlog when the TCP receive routine runs and notices that the socket was locked.
TCP packets directly copied from prequeue	Number of TCP packets delivered to the client from the prequeue.
TCP prequeue dropped packets	Number of TCP packets dropped from the prequeue. The prequeue is where the TCP receives routine runs. It notes that the current running process as the TCP target process and queues it directly for copy after the TCP software interrupt is completed.
TCP header predicted packets	Number of incoming packets that successfully matched the TCP header prediction.
Packets header predicted and queued to user	Number of TCP packets copied directly to the user space.
TCP pure ack packets	Number of ACK[5] packets that contain no data.
TCP header predicted acks	Number of incoming ACKs that successfully matched the TCP header prediction.
TCP Reno recoveries	Number of times that the TCP fast recovery algorithm recovered a packet loss. TCP Reno induces packet losses to estimate the available bandwidth in the network. When there are no packet losses, TCP Reno continues to increase its window size by one during each round trip. When it experiences a packet loss, it reduces its window size to one half of the current window size. This feature is called *additive increase and multiplicative decrease*. TCP Reno, however, does not fairly allocate bandwidth because TCP is not a synchronized rate-based control scheme, which is necessary for the convergence.
TCP SACK recoveries	Number of times that the SB recovered from a SACK packet loss. If the data receiver has received a SACK-permitted option on the SYN for this connection, the data receiver may choose to generate SACK options. If the data receiver generates SACK options under any circumstance, it should generate them under all permitted circumstances. If the data receiver has not received a SACK-permitted option for a given connection, it must not send SACK options on that connection.

Table 4-8 show statistics tcp Field Descriptions (continued)

Field	Description
TCP SACK reneging	Number of times that the SB refused to accept packets that have not been acknowledged to the data sender, even if the data has already been reported in a SACK option. Such discarding of SACK packets is discouraged but may be used if the receiver runs out of buffer space. The data receiver may choose not to keep data that it has reported in a SACK option. Because the data receiver may later discard data reported in a SACK option, the sender must not discard data before it is acknowledged by the Acknowledgment Number field in the TCP header.
TCP FACK reorders	Number of FACK[6] packets that were out of sequence order. The FACK algorithm makes it possible to treat congestion control during recovery in the same manner as during other parts of the TCP state space. The FACK algorithm is based on first principles of congestion control and is designed to be used with the proposed TCP SACK option. By decoupling congestion control from other algorithms, such as data recovery, it attains more precise control over the data flow in the network. FACK takes advantage of the SACK option; it takes into account which segments have been SACKed. It also uses the receipt of a SACK that leaves at least 3*MSS bytes unacknowledged as a trigger for Fast Retransmit.
TCP SACK reorders	Number of SACK[7] packets that were out of sequence order.
TCP Reno reorders	Number of TCP Renos that were out of sequence order.
TCP TimeStamp reorders	Number of segments received with out-of-order time stamps.
TCP full undos	Number of times that the congestion window (cwnd) was fully recovered.
TCP partial undos	Number of times that the congestion window (cwnd) was partially recovered.
TCP DSACK undos	Number of times that the D-SACK[8] packets were recovered.
TCP loss undos	Number of times that the congestion window (cwnd) recovered from a packet loss.
TCP losses	Number of times that data was lost and the size of the congestion window (cwnd) decreased.
TCP lost retransmit	Number of times that a retransmitted packet was lost.

Table 4-8 show statistics tcp Field Descriptions (continued)

Field	Description
TCP Reno failures	Number of times that the congestion window (cwnd) failed because the TCP fast recovery algorithm failed to recover from a packet loss. The congestion avoidance mechanism, which is adopted by TCP Reno, causes the window size to vary. This situation causes a change in the round-trip delay of the packets, larger delay jitter, and an inefficient use of the available bandwidth because of many retransmissions of the same packets after the packet drops occur. The rate at which each connection updates its window size depends on the round-trip delay of the connection. The connections with shorter delays can update their window sizes faster than other connections with longer delays.
TCP SACK failures	Number of times that the cwnd[9] shrunk because the SB failed to recover from a SACK packet loss. The selective acknowledgment extension uses two TCP options. The first is an enabling option, SACK-permitted, which may be sent in a SYN segment to indicate that the SACK option can be used once the connection is established. The other is the SACK option, which may be sent over an established connection once permission has been given by the SACK-permitted option.
TCP loss failures	Number of times that the TCP timeout occurred and data recovery failed.
TCP fast retransmissions	Number of TCP fast retransmission counters. TCP may generate an immediate acknowledgment (a duplicate ACK) when an out-of-order segment is received. The duplicate ACK lets the other end know that a segment was received out of order and tells it what sequence number is expected. Because TCP does not know whether a duplicate ACK is caused by a lost segment or just a reordering of segments, it waits for a small number of duplicate ACKs to be received. If there is just a reordering of the segments, there is only one or two duplicate ACKs before the reordered segment is processed, which then generates a new ACK. If three or more duplicate ACKs are received in a row, it is a strong indication that a segment has been lost. TCP then retransmits what appears to be the missing segment without waiting for a retransmission timer to expire.

Table 4-8 show statistics tcp Field Descriptions (continued)

Field	Description
TCP forward retransmissions	Number of TCP forward retransmission counters. This field applies only to SACK-negotiated connections; this field is the counter for FACK segments. The value of this field is for segments that were retransmitted even though there is no indication that they were actually lost. Retransmission is stopped when either one of the following occurs: • Maximum time to wait for a remote response is reached. This timeout occurs when the total time of all retransmission intervals exceeds the maximum time to wait for a remote response. • Number of retransmissions configured in maximum retransmissions per packet is reached.
TCP slowstart retransmissions	Number of TCP slow-start retransmission counters. The slow-start algorithm begins by sending packets at a rate that is determined by the congestion window. The algorithm continues to increase the sending rate until it reaches the limit set by the slow-start threshold (ssthresh) variable. (Initially, the value of the ssthresh variable is adjusted to the receiver's maximum window size [RMSS]. However, when congestion occurs, the ssthresh variable is set to half the current value of the cwnd variable, marking the point of the onset of network congestion for future reference.)
TCP Timeouts	Number of times that a TCP timeout occurred.
TCP Reno recovery fail	Number of times that the TCP fast recovery algorithm failed to recover from a packet loss. In TCP Reno, the maximum number of recoverable packet losses in a congestion window without timeout is limited to one or two packets. No more than six losses can be recovered with a maximum window size of 128 packets. This failure of recovery is because TCP Reno cuts the congestion window by half for each recovered loss.
TCP Sack recovery fail	Number of times that the SB failed to recover from a SACK packet loss. When receiving an ACK containing a SACK option, the data sender should record the selective acknowledgment for future reference. The data sender is assumed to have a retransmission queue that contains the segments that have been sent but not yet acknowledged in sequence number order. If the data sender performs repacketization before retransmission, the block boundaries in a SACK option that it receives may not fall within the boundaries of segments in the retransmission queue.
TCP scheduler failed	Number of times that the TCP scheduler failed.
TCP receiver collapsed	Number of times that the data in an out-of-order queue collapsed.

Table 4-8 show statistics tcp Field Descriptions (continued)

Field	Description
TCP DSACK old packets sent	Number of D-SACKs sent by the SB. The use of D-SACK does not require a separate negotiation between a TCP sender and receiver that have already negotiated SACK. The absence of a separate negotiation for D-SACK means that the TCP receiver could send D-SACK blocks when the TCP sender does not understand this extension to SACK. In this case, the TCP sender discards any D-SACK blocks and processes the other SACK blocks in the SACK option field as it normally would.
TCP DSACK out-of-order packets sent	Number of out-of-order D-SACK packets sent by the SB. A D-SACK block is used only to report a duplicate contiguous sequence of data received by the receiver in the most recent packet. Each duplicate contiguous sequence of data received is reported in at most one D-SACK block. (The receiver sends two identical D-SACK blocks in subsequent packets only if the receiver receives two duplicate segments.) If the D-SACK block reports a duplicate contiguous sequence from a (possibly larger) block of data in the receiver's data queue above the cumulative acknowledgement, then the second SACK block in that SACK option should specify that (possibly larger) block of data.
TCP DSACK packets received	Number of D-SACK packets received by the SB. TCP senders receiving D-SACK blocks should be aware that a segment reported as a duplicate segment could possibly have been from a prior cycle through the sequence number space. This awareness of the TCP senders is independent of the use of PAWS by the TCP data receiver.
TCP DSACK out-of-order packets received	Number of out-of-order D-SACK packets received by the SB. Following a lost data packet, the receiver receives an out-of-order data segment, which triggers the SACK option as specified in RFC 2018. Because of several lost ACK packets, the sender then retransmits a data packet. The receiver receives the duplicate packet and reports it in the first D-SACK block.
TCP connections abort on sync	Number of times that a valid SYN segment was sent in the TCP window and the connection was reset.
TCP connections abort on data	Number of times that the connection closed after reading the data.
TCP connections abort on close	Number of times that the connection aborted with pending data.
TCP connections abort on memory	Number of times that memory was not available for graceful closing of the connection resulting in the connection being aborted immediately.
TCP connections abort on timeout	Number of times that the connection timed out.
TCP connections abort on linger	Number of times that the linger timeout expired resulting in the data being discarded and closing of the connection.

Table 4-8 show statistics tcp Field Descriptions (continued)

Field	Description
TCP connections abort failed	Number of times that the TCP connection ran out of memory, transmits failed, or peer TCP Reset (RST) could not be sent.
TCP memory pressures	Number of times that the TCP subsystem encounters memory constraints.

1. RST = reset
2. SYN = synchronized
3. ARP = Address Resolution Protocol
4. PAWS = Protection Against Wrapped Sequence
5. ACK = acknowledgment
6. FACK = Forward Acknowledgment
7. SACK = Selective Acknowledgment
8. D-SACK = Duplicate Selective Acknowledgment
9. cwnd = congestion window

Related Commands

Command	Description
clear statistics	Clears the statistics settings.

show statistics transaction-logs

To display SB transaction log export statistics, use the **show statistics transaction-logs** command in EXEC configuration mode.

show statistics transaction-logs

Syntax Description

This command has no arguments or keywords.

Defaults

None

Command Modes

EXEC configuration mode.

Usage Guidelines

To display the transaction log export statistics, you must first configure the FTP server.

Table 4-9 describes the fields shown in the **show statistics transaction-logs** display.

Table 4-9 show statistics transaction-logs Field Descriptions

Field	Description
Initial Attempts	Initial attempts made to contact the external server at the configured export intervals.
Initial Successes	Number of times that an initial attempt made to contact the external server succeeded.
Initial Open Failures	Number of times that the SB failed to open a connection to the FTP export server.
Initial Put Failures	Number of times that the SB failed to transfer a file to the FTP export server.
Retry Attempts	Number of retries made to contact the external server at the configured export intervals.
Retry Successes	Number of times that a retry made to contact the external server succeeded.
Retry Open Failures	Number of times that the SB failed to open a connection to the FTP export server on a retry.
Retry Put Failures	Number of times that the SB failed to transfer a file to the FTP export server on a retry.
Authentication Failures	Number of times that the SB failed to authenticate with the FTP export server. This situation might occur if the SB is misconfigured with the wrong password for the FTP server or the password on the FTP server has been changed since the SB was configured.
Invalid Server Directory Failures	Number of times the SB failed to direct traffic to the correct server directory.

Related Commands

Command	Description
clear transaction-log	Clears the working transaction logs settings.
show transaction-logging	Displays the transaction log configuration settings and a list of archived transaction log files.
transaction-log force	Forces the archive or export of the transaction log.

show statistics udp

To display SB User Datagram Protocol (UDP) statistics, use the **show statistics udp** command in EXEC configuration mode.

show statistics udp

Syntax Description

This command has no arguments or keywords.

Defaults

None

Command Modes

EXEC configuration mode.

Usage Guidelines

Table 4-10 describes the fields shown in the **show statistics udp** display.

Table 4-10 show statistics udp Field Descriptions

Field	Description
Packets received	Total number of UDP packets received.
Packets to unknown port received	Number of packets to unknown ports received.
Packet receive error	Number of packet receive errors.
Packet sent	Number of UDP packets sent.

show tacacs

To display TACACS+ authentication protocol configuration information, use the **show tacacs** command in EXEC configuration mode.

show tacacs

Syntax Description

This command has no arguments or keywords.

Defaults

None

Command Modes

EXEC configuration mode.

Usage Guidelines

The **show tacacs** command displays the TACACS+ configuration for the Service Broker.

Table 4-11 describes the fields shown in the **show tacacs** display.

Table 4-11 show tacacs Field Descriptions

Field	Description
Login Authentication for Console/Telnet Session	Status of whether TACACS+ server is enabled for login authentication.
Configuration Authentication for Console/Telnet Session	Status of whether TACACS+ server is enabled for authorization or configuration authentication.
Authentication scheme fail-over reason	Status of whether Service Brokers fails over to the secondary method of administrative login authentication whenever the primary administrative login authentication method is used.
TACACS+ Configuration	TACACS+ server parameters.
TACACS+ Authentication	Status of whether TACACS+ authentication is enabled on the Service Broker.
Key	Secret key that the Service Broker uses to communicate with the TACACS+ server. The maximum number of characters in the TACACS+ key should not exceed 99 printable ASCII characters (except tabs).
Timeout	Number of seconds that the Service Broker waits for a response from the specified TACACS+ Authentication Server before declaring a timeout.
Retransmit	Number of times that the Service Broker is to retransmit its connection to the TACACS+ server if the TACACS+ timeout interval is exceeded.
Password type	Mechanism for password authentication. By default, the PAP[1] is the mechanism for password authentication.

Table 4-11 show tacacs Field Descriptions (continued)

Field	Description
Server	Hostname or IP address of the TACACS+ server.
Status	Status of whether server is the primary or secondary host.

1. PAP = Password Authentication Protocol

Related Commands

Command	Description
clear tacacs	Clears the TACACS+ settings.
show statistics tacacs	Displays the SB TACACS+ authentication and authorization statistics.
tacacs	Configures TACACS+ server parameters.

show tech-support

To view information necessary for the Cisco Technical Assistance Center (TAC) to assist you, use the **show tech-support** command in EXEC configuration mode.

show tech-support [**list-files** *directory_name* [**recursive**] | **page** | **service** {**authentication** | **cms** | **kernel**] | **authentication**}]

Syntax Description

list-files	(Optional) Displays the list of files under a directory.
directory_name	Directory name (use absolute path, such as /local1/logs).
recursive	Specifies to include files in recursive sub-directories.
page	(Optional) Specifies the pages through the output.
service	(Optional) Displays technical support information specific to a service.
authentication	Displays technical support information related to HTTP authentication.
cms	Displays technical support information related to CMS.
kernel	Displays technical support information related to the kernel.

Defaults

None

Command Modes

EXEC configuration mode.

Usage Guidelines

Use this command to view system information necessary for TAC to assist you with your SB. We recommend that you log the output to a disk file. Use the streaming option to view information specific to the streaming feature.

You can access the following general information when you enter the **show tech-support** command:

- Version and hardware (**show version**)
- Running configuration (**show running-config**)
- Processes (**show processes**)
- Process memory (**show processes memory**)
- System memory
- File system information
- Interface information
- Media file system statistics
- Application and kernel core dump information
- Netstat

Examples

The following example shows the types of information available about the CDS software. Because the **show tech-support** command output is comprehensive and can be extensive, only excerpts are shown in the following example:

```
ServiceBroker# show tech-support

CPU Usage:
   cpu: 0.39% User,  0.42% System,  0.33% User(nice),  98.86% Idle
  cpu0: 0.39% User,  0.42% System,  0.33% User(nice),  98.86% Idle
------------------------------------------------------------------
 PID  STATE PRI User T  SYS T        COMMAND
----- ----- --- ------ ------ --------------------
    1     S   0   4386   1706 (init)
    2     S   0      0      0 (keventd)
    3     S  19      0      0 (ksoftirqd_CPU0)
    4     S   0      0      0 (kswapd)
    5     S   0      0      0 (bdflush)
    6     S   0      0      0 (kupdated)
    7     S   0      0      0 (scsi_eh_0)
   45     S   0   4733   4114 (nodemgr)
   46     S   0      0      0 (syslogd)
   47     R   0     83     65 (dataserver)
  920     S   0      0      0 (login)
 1207     S   0      0      0 (parser_server)
 1208     S   0      0      0 (eval_timer_mana)
 1211     S   0     46      1 (parser_server)
 1443     S   0      0      0 (overload)
 1444     S   0      0      0 (standby)
 1445     S   0     13     29 (cache)
 1446     S   0      0      0 (proxy_poll)
 1447     S   0      0      0 (snmpced)
 1448     S   0      0      0 (http_authmod)
 1458     S   0      0      0 (http_authmod)
 1465     S   0      0      0 (http_authmod)
 1466     S   0      0      0 (http_authmod)
 1467     S   0      0      0 (http_authmod)
 1537     S   0      0      0 (cache)
 1538     S   0      0      0 (unified_log)
 1540     S   0      0      1 (webserver)
 1541     S   0      2      2  (mcm)
 1542     S   0      0      0 (cache)
 1543     S   0      0      0 (cache)
 1550     S   0      0      0 (cache)
 1551     S   0      0      0 (cache)
 1556     S   0      0      0 (cache)
 1567     S   0      0      0  (mcm)
 1568     S   0      0      0  (mcm)
 1629     S   0  18982   4140 (crond)
 1936     S   0   1669    611 (bootnet)
 1937     S  10      0      0 (tracknet)
 1938     S  10  33545   5556 (checkup)
 1983     S   0      0      0 (srcpd)
 2023     S   0      1      0 (admin-shell)
 2024     S   0      0      0 (parser_server)
 2150     S   0      0      0 (rsvpd)
 2152     S   0      0      0 (rtspd)
 2153     S   0   1635   1067 (httpsd)
 2164     S   0      0      0 (librarian)
 2167     S   0   1667   2105 (libaux)
 2170     S   0      0      0 (mapper)
 2178     S   0     32     37 (cache)
 2179     S   0      0      0 (router)
 2180     S   0      0      0 (fill)
```

```
 2183    S   0      0      0 (remotereq)
 2185    S -20      0      0 (videosvr)
 2188    S   0      9      4 (contentsvr)
 2189    S   0      0      0 (routeraux)
 2190    S   0      0      1 (dfcontrolsvr)
 2226    S   0      0      0 (smbd)
 2228    S   0      0      0 (nmbd)
 2973    Z   0      0      0 (cache)
 8446    S   0      0      0 (httpsd)
 8447    S   0      0      0 (gcache)
18173    S   0      0      0 (in.telnetd)
18174    S   0      0      0 (login)
18175    S   0      2      2 (admin-shell)
18176    S   0      0      0 (parser_server)
19426    S   0      0      0 (httpsd)
19427    S   0      0      0 (httpsd)
19456    Z   0      0      0 (cache)
19503    Z   0     30      3 (crond)
19515    S   0      0      0 (more)
19516    S   0      6     18 (exec_show_tech-)
19553    R   0      0      0 (exec_show_proce)

------------------ process memory --------------------

     Total       Used       Free     Shared    Buffers     Cached
1050943488  564785152  486158336          0    5222400  475176960



   PID State    TTY  %MEM    VM Size RSS (pages) Name
------ ----- ------ ----- ---------- ----------- ----
     1     S      0  0.0    1146880         119 (init)
     2     S      0  0.0          0           0 (keventd)
     3     S      0  0.0          0           0 (ksoftirqd_CPU0)
     4     S      0  0.0          0           0 (kswapd)
     5     S      0  0.0          0           0 (bdflush)
     6     S      0  0.0          0           0 (kupdated)
     7     S      0  0.0          0           0 (scsi_eh_0)
    45     S      0  0.0    1208320         143 (nodemgr)
    46     S      0  0.0    1630208         194 (syslogd)
    47     R      0  0.0    1974272         238 (dataserver)
   920     S   1088  0.0    1728512         236 (login)
  1207     S      0  0.3    4980736         847 (parser_server)
  1208     S      0  0.0    1933312         151 (eval_timer_mana)
  1211     S      0  0.3    4980736         847 (parser_server)
  1443     S      0  0.0    1548288         154 (overload)
  1444     S      0  0.0    1724416         161 (standby)
  1445     S      0  5.9   65646592       15266 (cache)
  1446     S      0  0.0    1957888         173 (proxy_poll)
  1447     S      0  0.1    2097152         290 (snmpced)
  1448     S      0  0.0    1757184         205 (http_authmod)
  1458     S      0  0.0    1757184         205 (http_authmod)
  1465     S      0  0.0    1757184         205 (http_authmod)
  1466     S      0  0.0    1757184         205 (http_authmod)
  1467     S      0  0.0    1757184         205 (http_authmod)
  1537     S      0  5.9   65646592       15266 (cache)
  1538     S      0  0.0    1789952         169 (unified_log)
  1540     S      0  0.4   10817536        1164 (webserver)
  1541     S      0  0.0    2150400         251 (mcm)
  1542     S      0  5.9   65646592       15266 (cache)
  1543     S      0  5.9   65646592       15266 (cache)
  1550     S      0  5.9   65646592       15266 (cache)
  1551     S      0  5.9   65646592       15266 (cache)
```

```
  1556     S      0  5.9   65646592       15266 (cache)
  1567     S      0  0.0    2150400         251 (mcm)
  1568     S      0  0.0    2150400         251 (mcm)
  1629     S      0  0.0    1187840         137 (crond)
  1936     S      0  0.6    7532544        1605 (bootnet)
2189     S      0  0.3    6103040         953 (routeraux)
  2190     S      0  0.4   10272768        1075 (dfcontrolsvr)
  2226     S      0  0.1    3559424         504 (smbd)
  2228     S      0  0.0    2084864         247 (nmbd)
  2973     Z      0  0.0          0           0 (cache)
  8446     S      0  0.1    2506752         327 (httpsd)
  8447     S      0  0.0    1421312         116 (gcache)
 18173     S      0  0.0    1220608         132 (in.telnetd)
 18174     S  34816  0.0    1736704         238 (login)
 18175     S  34816  0.0    2162688         184 (admin-shell)
 18176     S      0  0.3    4980736         847 (parser_server)
 19426     S      0  0.1    2551808         350 (httpsd)
 19427     S      0  0.1    2576384         354 (httpsd)
 19456     Z      0  0.0          0           0 (cache)
 19503     Z      0  0.0          0           0 (crond)
 19515     S  34816  0.0    1163264         109 (more)
 19516     S  34816  0.0    1941504         168 (exec_show_tech-)
 19554     R  34816  0.1    2277376         266 (exec_show_proce)

------------------ system memory --------------------

Total physical memory    :    1026312 KB
Total free memory        :     474692 KB
Total memory shared      :          0 KB
Total buffer memory      :       5100 KB
Total cached memory      :     464040 KB

------------------ interfaces --------------------

Interface type: GigabitEthernet Slot: 0  Port: 0
Type:Ethernet
Ethernet address:00:05:32:02:DD:74
Internet address:172.16.5.234
Netmask:255.255.255.0
Maximum Transfer Unit Size:1500
Metric:1
Packets Received: 513241
Input Errors: 0
Input Packets Dropped: 0
Input Packets Overruns: 0
Input Packets Frames: 0
Packet Sent: 153970
Output Errors: 0
Output Packets Dropped: 0
Output Packets Overruns: 0
Output Packets Carrier: 0
Output Queue Length:100
Collisions: 0
Interrupts:9
MULTICASTMode:autoselect, 100baseTX
```

show telnet

To display the Telnet services configuration, use the **show telnet** command in EXEC configuration mode.

show telnet

Syntax Description

This command has no arguments or keywords.

Defaults

Enabled.

Command Modes

EXEC configuration mode.

Examples

The following example shows how to display the Telnet service details:

```
ServiceBroker# show telnet
telnet service is enabled
```

Related Commands

Command	Description
exec-timeout	Configures the length of time that an inactive Telnet or SSH session remains open.
telnet enable	Enables the Telnet services.

show transaction-logging

To display the transaction log configuration settings and a list of archived transaction log files, use the **show transaction-logging** command in EXEC configuration mode.

show transaction-logging

Syntax Description

This command has no arguments or keywords.

Defaults

None

Command Modes

EXEC configuration mode.

Usage Guidelines

To display information about the current configuration of transaction logging on an SB, use the **show transaction-logging** command. Transaction log file information is displayed for HTTP and WMT caching proxy transactions and TFTP and ICAP transactions.

Examples

The following example shows how to display information about the current configuration of transaction logging on an SB:

```
ServiceBroker# show transaction-logging
Transaction log configuration:
--------------------------------------
Logging is enabled.
Archive interval: 1800 seconds
Maximum size of archive file: 2000000 KB
Maximum number of archive files: 50 files
Log File format is apache.
Windows domain is not logged with the authenticated username

Exporting files to ftp servers is enabled.
File compression is disabled.
Export interval: 30 minutes

server              type  username        directory
10.77.153.110       ftp   root            /var/ftp/test



WMT MMS Caching Proxy/Server Transaction Log File Info
  Working Log file - size : 556
                     age: 483497
  Archive Log file - mms_export_3.1.18.8_20090522_074807        size: 556

WMT MMS Caching Proxy/Server Transaction Log File Info (WMS-90 format)
  Working Log file - size : 665
                     age: 483497
  Archive Log file - mms_export_wms_90_3.1.18.8_20090522_074807 size: 665

WMT MMS Caching Proxy/Server Transaction Log File Info (Ext. WMS-90 format)
  Working Log file - size : 702
```

```
                    age: 483497
  Archive Log file - mms_export_e_wms_90_3.1.18.8_20090522_074807       size: 70
2

WMT MMS Caching Proxy/Server Transaction Log File Info (Ext. WMS-41 format)
  Working Log file - size : 584
                    age: 483497
  Archive Log file - mms_export_e_wms_41_3.1.18.8_20090522_074807       size: 58
4

A&D Transaction Log File Info
  Working Log file - size : 138
                    age: 483497
  Archive Log file - acqdist_3.1.18.8_20090522_074807   size: 138
Movie Streamer Transaction Log File Info
  Working Log file - size : 488
                    age: 482196
  Archive Log file - movie-streamer_3.1.18.8_20090522_062602    size: 648
  Archive Log file - movie-streamer_3.1.18.8_20090522_064309    size: 805
  Archive Log file - movie-streamer_3.1.18.8_20090522_065857    size: 645
  Archive Log file - movie-streamer_3.1.18.8_20090522_070038    size: 648
  Archive Log file - movie-streamer_3.1.18.8_20090522_074807    size: 645
  Archive Log file - movie-streamer_3.1.18.8_20090522_080016    size: 648
  Archive Log file - movie-streamer_3.1.18.8_20090523_030829    size: 645
ICAP Transaction Log File Info
  Working Log file - size : 61
                    age: 483496
  Archive Log file - icap_3.1.18.8_20090522_074807      size: 61

Web Engine Transaction Log File Info - Apache format
  Working Log file - size : 86
                    age: 483497
  Archive Log file - we_accesslog_apache_3.1.18.8_20090522_074807       size: 82

Web Engine Transaction Log File Info - CLF format
  Working Log file - size : 3
                    age: 483497
  Archive Log file - we_accesslog_clf_3.1.18.8_20090522_074807  size: 3

Web Engine Transaction Log File Info - Extended Squid format
  Working Log file - size : 102
                    age: 483497
  Archive Log file - we_accesslog_extsqu_3.1.18.8_20090522_074807       size: 10
2

Cached Content Log File Info
  Working Log file - size : 41
                    age: 483496
  Archive Log file - cache_content_3.1.18.8_20090522_074807     size: 41

Flash Media Streaming Access Transaction Log File Info
  Working Log file - size : 36
                    age: 482196
  Archive Log file - fms_access_3.1.18.8_20090522_062602        size: 650
  Archive Log file - fms_access_3.1.18.8_20090522_064309        size: 509
  Archive Log file - fms_access_3.1.18.8_20090522_065857        size: 650
  Archive Log file - fms_access_3.1.18.8_20090522_074807        size: 509
  Archive Log file - fms_access_3.1.18.8_20090522_080016        size: 509
  Archive Log file - fms_access_3.1.18.8_20090523_030830        size: 650

Flash Media Streaming Authorization Transaction Log File Info
  Working Log file - size : 43
                    age: 482196
  Archive Log file - fms_auth_3.1.18.8_20090522_062602  size: 4826
```

```
  Archive Log file - fms_auth_3.1.18.8_20090522_063036  size: 281
  Archive Log file - fms_auth_3.1.18.8_20090522_064309  size: 596
  Archive Log file - fms_auth_3.1.18.8_20090522_065857  size: 4789
  Archive Log file - fms_auth_3.1.18.8_20090522_070038  size: 277
  Archive Log file - fms_auth_3.1.18.8_20090522_074807  size: 596
  Archive Log file - fms_auth_3.1.18.8_20090523_030830  size: 4790

Authserver Transaction Log File Info
  Working Log file - size : 108
                     age: 483496
  Archive Log file - authsvr_3.1.18.8_20090522_065857   size: 108
ServiceBroker#
```

The following example shows how to display information about the current configuration of transaction logging on an SB:

```
ServiceBroker# show transaction-logging
Transaction log configuration:
--------------------------------------
Logging is enabled.
Archive interval: 120 seconds
Maximum size of archive file: 2000000 KB
Maximum number of archive files: 50 files

Exporting files to ftp servers is enabled.
File compression is disabled.
Export interval: 1 minute

server               type  username        directory
10.74.115.12         sftp  xinwwang        /workspace/xinwwang/test
10.74.124.156        sftp  root            /root/test
10.74.124.157        sftp  root            /root/test
171.71.50.162        sftp  root            /test



Service Broker Log File Info
  Working Log file - size : 96
                     age: 169813
  Archive Log file - service_broker_3.1.14.70_20090421_222006   size: 256
  Archive Log file - service_broker_3.1.14.70_20090422_020038   size: 223
  Archive Log file - service_broker_3.1.14.70_20090422_210022   size: 351
  Archive Log file - service_broker_3.1.14.70_20090423_020006   size: 1248
  Archive Log file - service_broker_3.1.14.70_20090423_210021   size: 456
  Archive Log file - service_broker_3.1.14.70_20090521_000218   size: 402
  Archive Log file - service_broker_3.1.14.70_20090521_014815   size: 243
  Archive Log file - service_broker_3.1.14.70_20090521_015020   size: 225
  Archive Log file - service_broker_3.1.14.70_20090521_015227   size: 243
  Archive Log file - service_broker_3.1.14.70_20090521_015417   size: 272
  Archive Log file - service_broker_3.1.14.70_20090521_015601   size: 390
  Archive Log file - service_broker_3.1.14.70_20090521_015816   size: 243
  Archive Log file - service_broker_3.1.14.70_20090521_020033   size: 243
  Archive Log file - service_broker_3.1.14.70_20090521_020249   size: 143
  Archive Log file - service_broker_3.1.14.70_20090521_032633   size: 168
  Archive Log file - service_broker_3.1.14.70_20090526_025027   size: 143
  Archive Log file - service_broker_3.1.14.70_20090526_030002   size: 176
  Archive Log file - service_broker_3.1.14.70_20090526_030226   size: 250
  Archive Log file - service_broker_3.1.14.70_20090526_052206   size: 250
  Archive Log file - service_broker_3.1.14.70_20090526_052413   size: 143
  Archive Log file - service_broker_3.1.14.70_20090526_200213   size: 168
  Archive Log file - service_broker_3.1.14.70_20090526_200413   size: 481
  Archive Log file - service_broker_3.1.14.70_20090526_200645   size: 173
  Archive Log file - service_broker_3.1.14.70_20090526_201010   size: 250
```

Related Commands

Command	Description
clear transaction-log	Clears the working transaction log settings.
show statistics transaction-logs	Displays the SB transaction log export statistics.
transaction-log force	Forces the archive or export of the transaction log.

show url-signature

To display the URL signature information, use the **show url-signature** command in EXEC configuration mode.

show url-signature

Syntax Description

This command has no arguments or keywords.

Defaults

None

Command Modes

EXEC configuration mode.

Examples

The following example shows how to display the URL signature information:

```
key-id-owner  key-id-number  key  public-key  private-key  symmetric-key
-------------------------------------------------------------------
     1             1             ****        ****         ****
```

show user

To display the user identification number and username information for a particular user, use the **show** command in EXEC configuration mode.

show user {**uid** *num* | **username** *name*}

Syntax Description

uid	Displays the user's identification number.
num	Identification number. The range is from 0 to 65535.
username	Displays the name of user.
name	Name of the user.

Defaults

None

Command Modes

EXEC configuration mode.

Usage Guidelines

Table 4-12 describes the fields shown in the **show user** display.

Table 4-12 show user Field Descriptions

Field	Description
Uid	User ID number.
Username	Username.
Password	Login password. This field does not display the actual password.
Privilege	Privilege level of the user.
Configured in	Database in which the login authentication is configured.

Related Commands

Command	Description
clear user	Clears the user settings.
show users	Displays the specified users.
username	Establishes the username authentication.

show users

To display users, use the **show users** command in EXEC configuration mode.

show users administrative

Syntax Description

administrative	Lists users with administrative privileges.

Defaults

None

Command Modes

EXEC configuration mode.

Examples

The following example shows how to display the list of users with administrative privileges:

```
ServiceBroker# show users administrative
           UID USERNAME
           0 admin
```

Related Commands

Command	Description
clear user	Clears the user settings.
show user	Displays the user identification number and username information for a particular user.
username	Establishes the username authentication.

show version

To display version information about the software, use the **show version** command in EXEC configuration mode.

show version pending

Syntax Description

pending	Displays the version for pending upgraded image.

Defaults

None

Command Modes

EXEC configuration mode.

Usage Guidelines

Table 4-13 describes the fields shown in the **show version** display.

Table 4-13 show version Field Descriptions

Field	Description
Version	VDS-SB software version.
Compiled hour:minute:second month day year by cnbuild	Compile information for the software build.
System was restarted on day of week month day hour:minute:second year	Date and time that the system was last restarted.
The system has been up for X hours, X minutes, X seconds	Length of time the system has been running since the last reboot.

Note If you update the VDS-SB software on an SB, the new version displays in the **show version pending** command output, but it says, "Pending version will take effect after reload." You must reboot the device for the software update to take affect.

Examples

The follow example shows how to display the software version:

```
ServiceBroker# show version
VDS Service Broker Software
Copyright (c) 1999-2011 by Cisco Systems, Inc.
Content Delivery System Software Release 3.0.0 (build b460 Aug 28 2011)
Version: cde220-2g2-DEVELOPMENT[vcn-build1:/auto/vcn-u1/vosis_release_builds/vos
is_3.0.0-b460/spcdn]

Compiled 05:55:01 Aug 28 2011 by ipvbuild
Compile Time Options: KQ SS

System was restarted on Mon Aug 29 11:56:58 2011.
The system has been up for 1 day, 23 hours, 32 minutes, 15 seconds.
```

```
ServiceBroker#
```

The following example shows how to display the pending software version:

```
ServiceBroker# show version pending
Pending version is VDS-SB 3.0.0-b360, built on 05:17:52 Jun 19 2011 by ipvbuild
It will take effect after reload
ServiceBroker#
```

Related Commands

Command	Description
show flash	Displays the flash memory version and usage information.

shutdown (Interface configuration)

To shut down a specific hardware interface, use the **shutdown** command in interface configuration mode. To restore an interface to operation, use the **no** form of this command.

shutdown

no shutdown

Syntax Description

This command has no arguments or keywords.

Defaults

None

Command Modes

Interface configuration (config-if) mode.

Usage Guidelines

See the "interface" section on page 2-85 for alternative mechanism.

Examples

The following example shows how to shut down an interface configured on an SB:

```
ServiceBroker(config-if)# shutdown
```

Related Commands

Command	Description
interface	Configures a Gigabit Ethernet or port channel interface.
show interface	Displays the hardware interface information.
show running-config	Displays the current operating configuration.
show startup-config	Displays the startup configuration.

shutdown (EXEC Configuration)

To shut down the SB or VDSM, use the **shutdown** command in EXEC configuration mode.

shutdown [**poweroff**]

Syntax Description

poweroff	(Optional) Turns off the power after closing all applications and the operating system.

Defaults

None

Command Modes

EXEC configuration mode.

Usage Guidelines

A controlled shutdown refers to the process of properly shutting down an SB without turning off the power on the device. With a controlled shutdown, all the application activities and the operating system are properly stopped on an SB but the power is still on. Controlled shutdowns of an SB can help you minimize the downtime when the SB is being serviced.

The **shutdown** command enables you to shut down and optionally power off an SB:

- *Shutdown* means that all application activities (applications and operating system) are stopped, but the power is still on. This shutdown is similar to the Linux **halt** command.
- *Shutdown poweroff* means that the SB is powered down by the VDS-SB software after being shut down. This operation is also referred to as a software poweroff. The implementation of the shutdown poweroff feature uses the Advanced Configuration and Power Interface (ACPI) power management interface.

Caution If you do not perform a controlled shutdown, the SB file system can be corrupted. It also takes longer to reboot the SB if the SB is not properly shut down.

Note You cannot power on SBs again through software after a software poweroff operation. You must press the power button once on these SBs to bring these SBs back online.

The **shutdown** command facilitates a proper shutdown for SBs, or VDSMs. Where the **shutdown** command is supported on all content networking hardware models, the **shutdown poweroff** command is supported only on those models that support ACPI.

The **shutdown** command closes all applications and stops all system activities but keeps the power on. The fans continue to run and the power LED is on, indicating that the device is still powered on. When you enter the **shutdown** command, you are prompted to save your configuration changes, if any. The device console displays a menu after the shutdown process is completed. You need to log in to the SB using a console to display the following menu:

```
ServiceBroker# shutdown
System configuration has been modified. Save? [ yes ] :yes
Device can not be powered on again through software after shutdown.
Proceed with shutdown? [ confirm ] yes
Shutting down all services, will timeout in 15 minutes.
shutdown in progress ..Halt requested by CLI@ttyS0.
..........
Shutdown success

Cisco Service Broker Console

Username: admin
Password:

================= SHUTDOWN SHELL =================
   System has been shut down.

   You can either
      Power down system by pressing and holding power button
   or
   1. Reload system through software
   2. Power down system through software
   Please select [ 1-2 ] :
```

The **shutdown poweroff** command closes all applications and the operating system, stops all system activities, and turns off the power. The fans stop running and the power LED starts flashing, indicating that the device has been powered off.

Note If you use the **shutdown** or **shutdown poweroff** commands, the device does not perform a file system check when you power on and boot the device the next time.

Table 4-14 describes the shutdown and shutdown power-off operations for SBs.

Table 4-14 Shutting Down Content Engines Through CLI Commands

Activity	All Content Engine Models	Content Engines with Power Management Capability
User performs a shutdown operation on the SB	`ServiceBroker#` **`shutdown`**	`ServiceBroker#` **`shutdown poweroff`**
User intervention to bring SB back online	To bring an SB that has an on/off switch on the back online after a shutdown operation, flip the on/off switch twice. To bring an SB that has a power button (instead of an on/off switch on the back) back online after a shutdown operation, first press and hold the power button for several seconds to power off these models, and then press the power button once again.	After a shutdown poweroff, press the power button once to bring the SB back online.
File system check	Is not performed after you turn the power on again and reboot the SB.	Is not performed after you turn the power on again and reboot the SB.

You can enter the **shutdown** command from a console session or from a remote session (Telnet or SSH Version 1 or SSH Version 2) to perform a shutdown on an SB.

To perform a shutdown on an SB, enter the **shutdown** command as follows:

```
ServiceBroker# shutdown
```

When you are asked if you want to save the system configuration, enter **yes** as follows:

```
System configuration has been modified. Save? [ yes ] :yes
```

When you are asked if you want to proceed with the shutdown, press **Enter** to proceed with the shutdown operation as follows:

```
Device can not be powered on again through software after shutdown.
Proceed with shutdown? [ confirm ]
```

The following message appears, reporting that all services are being shut down on this SB:

```
Shutting down all services, will timeout in 15 minutes.
shutdown in progress ..System halted.
```

After the system is shut down (the system has halted), an VDS-SB software shutdown shell displays the current state of the system (for example, `System has been shut down`) on the console. You are asked whether you want to perform a software power off (the Power down system by software option), or if you want to reload the system through the software.

```
================= SHUTDOWN SHELL =================
System has been shut down.
```

Table 4-15 show statistics wmt all Field Descriptions

Field	Description
Unicast Requests Statistics	
Total unicast requests received	Total number of unicast requests received. Display shows the number of requests in each category and calculates the percentage of the total for each category.
Streaming Requests served	Number of streaming requests received.
Multicast nsc file Request	Number of multicast NSC file requests received.
Authenticate Requests	Number of authenticated requests received.
Requests error	Number of request errors received.
By Type of Content	
Live content	Number of live content requests received.
On-Demand Content	Number of on-demand content requests received.
By Transport Protocol	
HTTP	Number of HTTP requests received.
RTSPT	Number of RTSPT requests received.
RTSPU	Number of RTSPU requests received.
Unicast Savings Statistics	
Total bytes saved	Total number of bytes saved.
By Source of Content	
Local	Number of local bytes saved.
Remote HTTP	Number of remote HTTP bytes saved.
Remote RTSP	Number of remote RTSP bytes saved.
Multicast	Number of multicast bytes saved.
CDN-Related WMT Requests	
CDN Content Hits	Number of CDN content request hits.
CDN Content Misses	Number of CDN content request misses.
CDN Content Live	Number of CDN live content requests.

Table 4-15 show statistics wmt all Field Descriptions (continued)

Field	Description
CDN Content Errors	Number of CDN content request errors.
Fast Streaming-related WMT Requests	
Normal Speed	Number of normal-speed Fast Streaming-related WMT requests.
Fast Start Only	Number of Fast Start WMT requests.
Fast Cache Only	Number of Fast Cache WMT requests.
Fast Start and Fast Cache	Number of Fast Start and Fast Cache WMT requests.
Authenticated Requests	
By Type of Authentication	
Negotiate	Number of negotiated authentication authenticated requests.
Digest	Number of digest authentication authenticated requests.
Basic	Number of basic authentication authenticated requests.
Unicast Bytes Statistics	
Total unicast incoming bytes	Total number of bytes incoming as unicast streams.
By Type of Content	
Live content	Number of bytes incoming as unicast streams for live content.
On-Demand Content	Number of bytes incoming as unicast streams for on-demand content.
By Transport Protocol	
HTTP	Number of bytes incoming as unicast streams using the HTTP transport protocol.
RTSPT	Number of bytes incoming as unicast streams using the RTSPT transport protocol.
Total unicast outgoing bytes	Total number of bytes outgoing as unicast streams.
Unicast Savings Statistics	
Total bytes saved	Total number of bytes saved.
By prepositioned content	Number of bytes saved for prepositioned content.
By live-splitting	Number of bytes saved for live-splitting content.
By cache-hit	Number of bytes saved for cached content.
Live Splitting	
Incoming bytes	Number of bytes incoming as live-split streams.
Outgoing bytes	Number of bytes outgoing as live-split streams.
Bytes saved	Number of bytes saved.

Table 4-15 show statistics wmt all Field Descriptions (continued)

Field	Description
Caching	
Bytes cache incoming	Number of bytes incoming for the cache.
Bytes cache outgoing	Number of bytes outgoing from the cache.
Bytes cache total	Total number of bytes cached.
Bytes cache-bypassed	Number of bytes that bypassed the cache.
Cacheable requests	Number of cacheable requests.
Req cache-miss	Number of cacheable requests that were cache misses.
Req cache-hit	Number of cacheable requests that were cache hits.
Req cache-partial-hit	Number of cacheable requests that were partial cache hits.
Req cache-total	Total number of requests that were cached.
Objects not cached	Number of objects that were not cached.
Cache bypassed	Number of objects that were not cached because they bypassed the cache.
Exceed max-size	Number of objects that were not cached because they exceeded the maximum cacheable size limit.
Usage Summary	
Concurrent Unicast Client Sessions	Total number of concurrent unicast client sessions.
Current	Number of concurrent unicast client sessions currently running.
Max	Maximum number of concurrent unicast client sessions recorded.
Concurrent Remote Server Sessions	Total number of concurrent remote server sessions.
Concurrent Active Multicast Sessions	Total number of concurrent active multicast sessions.
Concurrent Unicast Bandwidth (Kbps)	Total amount of bandwidth being used (in kilobits per second) for concurrent unicast sessions.
Concurrent Bandwidth to Remote Servers (Kbps)	Total amount of bandwidth being used (in kilobits per second) for concurrent remote server sessions.
Concurrent Multicast Out Bandwidth (Kbps)	Total amount of bandwidth being used (in kilobits per second) for concurrent multicast out sessions.
Error Statistics	
Total request errors	Total number of request errors.
Errors generated by this box	Number of request errors generated by this device.

Table 4-15 show statistics wmt all Field Descriptions (continued)

Field	Description
Errors generated by remote servers	Number of request errors generated by remote servers.
Other Statistics	
Authentication Retries from Clients	Number of authentication retries from clients.
WMT Rule Template Statistics	
URL Rewrite	Number of URL rewrites.
URL Redirect	Number of URL redirects.
URL Block	Number of blocked URLs.
No-Cache	Number of no-cache matches.
Allow	Number of allow matches.
Multicast Statistics	
Total Multicast Outgoing Bytes	Total number of bytes outgoing as multicast-out streams.
Total Multicast Logging Requests	Total number of multicast logging requests.
Aggregate Multicast Out Bandwidth (Kbps)	Aggregated amount of bandwidth being used (in kilobits per second) for multicast out sessions.
Current	Number of concurrent multicast out sessions currently running.
Max	Maximum number of multicast out sessions recorded.
Number of Concurrent Active Multicast Sessions	Number of concurrent active multicast sessions.

```
You can either
   Power down system by pressing and holding power button
or
1. Reload system through software
2. Power down system through software
```

To power down the SB, press and hold the power button on the SB, or use one of the following methods to perform a shutdown poweroff:

- From the console command line, enter **2** when prompted as follows:

```
================= SHUTDOWN SHELL =================
System has been shut down.
You can either
   Power down system by pressing and holding power button
or
1. Reload system through software
2. Power down system through software
```

- From the SB CLI, enter the **shutdown poweroff** command as follows:

```
ServiceBroker#  shutdown poweroff
```

When you are asked if you want to save the system configuration, enter **yes** as follows:

```
System configuration has been modified. Save? [ yes ] :yes
```

When you are asked to confirm your decision, press **Enter**.

```
Device can not be powered on again through software after poweroff.
Proceed with poweroff? [ confirm ]
Shutting down all services, will timeout in 15 minutes.
poweroff in progress ..Power down.
```

Examples

The following example shows that the **shutdown** command is used to close all applications and stop all system activities:

```
ServiceBroker1# shutdown
System configuration has been modified. Save? [ yes ] :yes
Device can not be powered on again through software after shutdown.
Proceed with shutdown? [ confirm ]
Shutting down all services, will timeout in 15 minutes.
shutdown in progress ..System halted.
```

The following example shows that the **shutdown poweroff** command is used to close all applications, stop all system activities, and then turn off power to the SB:

```
ServiceBroker2# shutdown poweroff
System configuration has been modified. Save? [ yes ] :yes
Device can not be powered on again through software after poweroff.
Proceed with poweroff? [ confirm ]
Shutting down all services, will timeout in 15 minutes.
poweroff in progress ..Power down.
```

snmp-server community

To configure the community access string to permit access to the Simple Network Management Protocol (SNMP), use the **snmp-server community** command in Global configuration mode. To remove the specified community string, use the **no** form of this command.

snmp-server community *community_string* [**group** *group_name* | **rw**]

no snmp-server community *community_string* [**group** *group_name* | **rw**]

Syntax Description

community_string	Community string that acts like a password and permits access to SNMP.
group	(Optional) Specifies the group to which this community name belongs.
group_name	(Optional) Name of the group.
rw	(Optional) Specifies read-write access with this community string.

Defaults

An SNMP community string permits read-only access to all MIB objects.

A community string is assigned to the Secure Domain Router (SDR) owner.

Command Modes

Global configuration (config) mode.

Usage Guidelines

To use this command, you must be in a user group associated with a task group that includes the proper task IDs. Use the **snmp-server community** command to configure the community access string to permit access to SNMP. To remove the specified community string, use the **no** form of this command.

Note In a non-owner SDR, a community name provides access only to the object instances that belong to that SDR, regardless of the access privilege assigned to the community name. Access to the owner SDR and system-wide access privileges are available only from the owner SDR.

Examples

The following example shows how to add the community comaccess:

```
ServiceBroker(config)# snmp-server community comaccess rw
```

The following example shows how to remove the community comaccess:

```
ServiceBroker(config)# no snmp-server community comaccess
```

Related Commands

Command	Description
snmp-server view	Defines a Version 2 SNMP (SNMPv2) MIB view.

snmp-server contact

To set the system server contact (sysContact) string, use the **snmp-server contact** command in Global configuration mode. To remove the system contact information, use the **no** form of this command.

snmp-server contact *line*

no snmp-server contact

Syntax Description

line	Identification of the contact person for this managed node.

Defaults

No system contact string is set.

Command Modes

Global configuration (config) mode.

Usage Guidelines

The system contact string is the value stored in the MIB-II system group sysContact object.

Examples

The following example shows how to configure a system contact string:

```
ServiceBroker(config)# snmp-server contact Dial System Operator at beeper # 27345
```

The following example shows how to reset the system contact string:

```
ServiceBroker(config)# no snmp-server contact
```

Related Commands

Command	Description
show snmp	Displays the SNMP parameters.
snmp-server community	Configures the community access string to permit access to the SNMP.
snmp-server enable traps	Enables the SB to send SNMP traps.
snmp-server group	Defines a user security model group.
snmp-server host	Specifies the hosts to receive SNMP traps.
snmp-server location	Sets the SNMP system location string.
snmp-server notify inform	Configures the SNMP notify inform request.
snmp-server user	Defines a user who can access the SNMP engine.
snmp-server view	Defines a SNMPv2 MIB view.

snmp-server enable traps

To enable the SB to send SNMP traps, use the **snmp-server enable traps** command in Global configuration mode. To disable all SNMP traps or only SNMP authentication traps, use the **no** form of this command.

snmp-server enable traps [**alarm** [**clear-critical | clear-major | clear-minor | raise-critical | raise-major | raise-minor**] | **config | entity | event | service-broker** [**disk-fail | disk-read | disk-write | transaction-log**] | **snmp** [**authentication | cold-start**]]

no snmp-server enable traps [**alarm** [**clear-critical | clear-major | clear-minor | raise-critical | raise-major | raise-minor**] | **config | entity | event | service-broker** [**disk-fail | disk-read | disk-write | transaction-log**] | **snmp** [**authentication | cold-start**]]

Syntax Description

alarm	(Optional) Enables SB alarm traps.
clear-critical	(Optional) Enables the clear-critical alarm trap.
clear-major	(Optional) Enables the clear-major alarm trap.
clear-minor	(Optional) Enables the clear-minor alarm trap.
raise-critical	(Optional) Enables the raise-critical alarm trap.
raise-major	(Optional) Enables the raise-major alarm trap.
raise-minor	(Optional) Enables the raise-minor alarm trap.
config	(Optional) Enables CiscoConfigManEvent traps.
entity	(Optional) Enables SNMP entity traps.
event	(Optional) Enables Event MIB traps.
service-broker	(Optional) Enables SNMP SB traps.
disk-fail	(Optional) Enables the disk failure error trap.
disk-read	(Optional) Enables the disk read error trap.
disk-write	(Optional) Enables the disk write error trap.
transaction-log	(Optional) Enables the transaction log write error trap.
snmp	(Optional) Enables SNMP-specific traps.
authentication	(Optional) Enables the authentication trap.
cold-start	(Optional) Enables the cold-start trap.

Defaults

This command is disabled by default. No traps are enabled.

Command Modes

Global configuration (config) mode.

Usage Guidelines

You can configure an SB to generate an SNMP trap for a specific alarm condition. You can configure the generation of SNMP alarm traps on SBs based on the following:

- Severity of the alarm (critical, major, or minor)
- Action (the alarm is raised or cleared)

Cisco VDS Service Broker software supports six generic alarm traps. These six generic alarm traps provide SNMP and Node Health Manager integration. Each trap can be enabled or disabled through the SB CLI.

Note Some SNMP traps are different between v1 and v2 and v3 when configure the trap.

SNMP notifications can be sent as traps or inform requests. The **snmp-server enable traps** command enables both traps and inform requests for the specified notification types.

To configure traps, enter the **snmp-server enable traps** command. If you do not enter the **snmp-server enable traps** command, no traps are sent.

If you do not enter an **snmp-server enable traps** command, no notifications controlled by this command are sent. To configure the SB to send these SNMP notifications, enter at least one **snmp-server enable traps** command. If you enter the command with no keywords, all notification types are enabled. If you enter the command with a keyword, only the notification type related to that keyword is enabled. To enable multiple types of notifications, enter a separate **snmp-server enable traps** command for each notification type and notification option.

The **snmp-server enable traps** command is used with the **snmp-server host** command. Use the **snmp-server host** command to specify which host or hosts receive SNMP traps. To send traps, configure at least one host using the **snmp-server host** command.

For a host to receive a trap, enable both the **snmp-server enable traps** command and the **snmp-server host** command for that host.

In addition, enable SNMP with the **snmp-server community** command.

To disable the sending of the MIB-II SNMP authentication trap, enter the **no snmp-server enable traps snmp authentication** command.

Examples

The following example shows how to enable the SB to send all traps to the host 172.31.2.160 using the community string public:

```
ServiceBroker(config)# snmp-server enable traps
ServiceBroker(config)# snmp-server host 172.31.2.160 public
```

The following example disables all traps:

```
ServiceBroker(config)# no snmp-server enable traps
```

Related Commands

Command	Description
show snmp	Displays the SNMP parameters.
snmp-server community	Configures the community access string to permit access to the SNMP.
snmp-server contact	Sets the system server contact string.
snmp-server group	Defines a user security model group.
snmp-server host	Specifies the hosts to receive SNMP traps.
snmp-server location	Sets the SNMP system location string.
snmp-server notify inform	Configures the SNMP notify inform request.
snmp-server user	Defines a user who can access the SNMP engine.
snmp-server view	Defines a SNMPv2 MIB view.

snmp-server group

To define a user security model group, use the **snmp-server group** command in Global configuration mode. To remove the specified group, use the **no** form of this command.

snmp-server group *name* {**v1** [**notify** *name*] [**read** *name*] [**write** *name*] | **v2c** [**notify** *name*] [**read** *name*] [**write** *name*] | **v3** {**auth** [**notify** *name*] [**read** *name*] [**write** *name*] | **noauth** [**notify** *name*] [**read** *name*] [**write** *name*] | **priv** [**notify** *name*] [**read** *name*] [**write** *name*]}}

no snmp-server group *name* {**v1** [**notify** *name*] [**read** *name*] [**write** *name*] | **v2c** [**notify** *name*] [**read** *name*] [**write** *name*] | **v3** {**auth** [**notify** *name*] [**read** *name*] [**write** *name*] | **noauth** [**notify** *name*] [**read** *name*] [**write** *name*] | **priv** [**notify** *name*] [**read** *name*] [**write** *name*]}}

Syntax Description

name	Name of the SNMP group. Supports up to a maximum of 64 characters.
v1	Specifies the group using the Version 1 Security Model.
notify	(Optional) Specifies a notify view for the group that enables you to specify a notify, inform, or trap.
name	Notify view name. Supports up to a maximum of 64 characters.
read	(Optional) Specifies a read view for the group that enables you only to view the contents of the agent.
name	Read view name. Supports up to a maximum of 64 characters.
write	(Optional) Specifies a write view for the group that enables you to enter data and configure the contents of the agent.
name	Write view name. Supports up to a maximum of 64 characters.
v2c	Specifies the group using the Version 2c Security Model.
v3	Specifies the group using the User Security Model (SNMPv3).
auth	Specifies the group using the AuthNoPriv Security Level.
noauth	Specifies the group using the noAuthNoPriv Security Level.
priv	Specifies the group using the AuthPriv Security Level.

Defaults

The default is that no user security model group is defined.

Command Modes

Global configuration (config) mode.

Usage Guidelines

The maximum number of SNMP groups that can be created is 10.

Select one of three SNMP security model groups: Version 1 (**v1**) Security Model, Version 2c (**v2c**) Security Model, or the User Security Model (**v3** or SNMPv3). Optionally, you then specify a notify, read, or write view for the group for the particular security model chosen. The **v3** option allows you to specify the group using one of three security levels: **auth** (AuthNoPriv Security Level), **noauth** (noAuthNoPriv Security Level), or **priv** (AuthPriv Security Level).

Note Each community is associated with a group. Each group has a view and users are assigned to a group. If the group does not have a view associated with it, then users associated that group cannot access any MIB entry.

The Cisco VDS Service Broker software supports the following versions of SNMP:

- Version 1 (SNMPv1)—This version is the initial implementation of SNMP. See RFC 1157 for a full description of its functionality.
- Version 2 (SNMPv2c)—This version is the second release of SNMP, described in RFC 1902. It provides additions to data types, counter size, and protocol operations.
- Version 3 (SNMPv3)—This version is the most recent SNMP version, defined in RFC 2271 through RFC 2275.

SNMP Security Models and Security Levels

SNMPv1 and SNMPv2c do not have any security (authentication or privacy) mechanisms to keep SNMP packet traffic on the wire confidential. As a result, packets on the wire can be detected and SNMP community strings can be compromised.

To solve the security shortcomings of SNMPv1 and SNMPv2c, SNMPv3 provides secure access to SBs by authenticating and encrypting packets over the network. The SNMP agent supports SNMPv3, SNMPv1, and SNMPv2c.

Using SNMPv3, users can securely collect management information from their SNMP agents. Also, confidential information, such as SNMP set packets that change an SB's configuration, can be encrypted to prevent their contents from being exposed on the wire. Also, the group-based administrative model allows different users to access the same SNMP agent with varying access privileges.

Examples

The following example shows how to configure the SNMP group name, security model, and notify view on the SB:

```
ServiceBroker(config)# snmp-server group acme v1 notify mymib
```

Related Commands

Command	Description
show snmp	Displays the SNMP parameters.
snmp-server community	Configures the community access string to permit access to the SNMP.
snmp-server contact	Sets the system server contact string.
snmp-server enable traps	Enables the SB to send SNMP traps.
snmp-server host	Specifies the hosts to receive SNMP traps.
snmp-server location	Sets the SNMP system location string.
snmp-server notify inform	Configures the SNMP notify inform request.
snmp-server user	Defines a user who can access the SNMP engine.
snmp-server view	Defines a SNMPv2 MIB view.

snmp-server host

To specify the recipient of a host SNMP trap operation, use the **snmp-server host** command in Global configuration mode. To remove the specified host, use the **no** form of this command.

snmp-server host {*hostname* | *ip_address*} *communitystring* [**v2c** [**retry** *number*] [**timeout** *seconds*] | [**v3** {**auth** [**retry** *number*] [**timeout** *seconds*] | **noauth** [**retry** *number*] [**timeout** *seconds*] | **priv** [**retry** *number*] [**timeout** *seconds*]}]

no snmp-server host {*hostname* | *ip_address*} [**v2c** [**retry** *number*] [**timeout** *seconds*] | [**v3** {**auth** [**retry** *number*] [**timeout** *seconds*] | **noauth** [**retry** *number*] [**timeout** *seconds*] | **priv** [**retry** *number*] [**timeout** *seconds*]} | *communitystring*]

Syntax Description

hostname	Hostname of the SNMP trap host that is sent in the SNMP trap messages from the SB.
ip_address	IP address of the SNMP trap host that is sent in the SNMP trap messages from the SB.
communitystring	Password-like community string sent in the SNMP trap messages from the SB. You can enter a maximum of 64 characters.
v2c	(Optional) Specifies the Version 2c Security Model.
retry	(Optional) Sets the count for the number of retries for the inform request. (The default is 2 tries.)
number	Number of retries for the inform request. The range is from 1 to 10.
timeout	(Optional) Sets the timeout for the inform request The default is 15 seconds.
seconds	Timeout value, in seconds. The range is from 1 to 1000.
v3	(Optional) Specifies the User Security Model (SNMPv3).
auth	Sends notification using the AuthNoPriv Security Level.
noauth	Sends notification using the noAuthNoPriv Security Level.
priv	Sends notification using the AuthPriv Security Level.

Defaults

This command is disabled by default. No traps are sent. The version of the SNMP protocol used to send the traps is SNMP Version 1.

retry *number*: 2

timeout *seconds*: 15

Command Modes

Global configuration (config) mode.

Usage Guidelines

SNMP notifications can be sent as traps or inform requests. Traps are unreliable because the receiver does not send acknowledgments when it receives traps. The sender cannot determine if the traps were received. However, an SNMP entity that receives an inform request acknowledges the message with an SNMP response protocol data unit (PDU). If the sender never receives the response, the inform request can be sent again. Informs are more likely to reach their intended destination.

However, informs consume more resources in the agent and in the network. Unlike a trap, which is discarded as soon as it is sent, an inform request must be held in the memory until a response is received or the request times out. Also, traps are sent only once, while an inform may be retried several times. The retries increase traffic and contribute to a higher overhead on the network.

If you do not enter an **snmp-server host** command, no notifications are sent. To configure the SB to send SNMP notifications, enter at least one **snmp-server host** command. To enable multiple hosts, enter a separate **snmp-server host** command for each host. You can specify multiple notification types in the command for each host.

When multiple **snmp-server host** commands are given for the same host and kind of security model, each succeeding command overwrites the previous command. Only the last **snmp-server host** command is in effect. For example, if you enter an **snmp-server host v2c** command for a host and then enter another **snmp-server host v3** command for the same host, the second command replaces the first.

The maximum number of SNMP hosts that can be created by entering the **snmp-server host** commands is eight.

When multiple **snmp-server host** commands are given for the same host, the community string in the last command is used.

The **snmp-server host** command is used with the **snmp-server enable traps** command. Use the **snmp-server enable traps** command to specify which SNMP notifications are sent globally. For a host to receive most notifications, at least one **snmp-server enable traps** command and the **snmp-server host** command for that host must be enabled.

Note You must enable SNMP with the **snmp-server community** command.

Examples

The following example sends the SNMP traps defined in RFC 1157 to the host specified by the IP address 172.16.2.160. The community string is comaccess:

```
ServiceBroker(config)# snmp-server enable traps
ServiceBroker(config)# snmp-server host 172.16.2.160 comaccess
```

The following example shows how to remove the host 172.16.2.160 from the SNMP trap recipient list:

```
ServiceBroker(config)# no snmp-server host 172.16.2.160
```

Related Commands

Command	Description
show snmp	Displays the SNMP parameters.
snmp-server community	Configures the community access string to permit access to the SNMP.
snmp-server contact	Sets the system server contact string.
snmp-server enable traps	Enables the SB to send SNMP traps.
snmp-server group	Defines a user security model group.
snmp-server location	Sets the SNMP system location string
snmp-server notify inform	Configures the SNMP notify inform request.
snmp-server user	Defines a user who can access the SNMP engine.
snmp-server view	Defines a SNMPv2 MIB view.

snmp-server location

To set the SNMP system location string, use the **snmp-server location** command in Global configuration mode. To remove the location string, use the **no** form of this command.

snmp-server location *line*

no snmp-server location

Syntax Description

line	String that describes the physical location of this node.

Defaults

No system location string is set.

Command Modes

Global configuration (config) mode.

Usage Guidelines

The system location string is the value stored in the MIB-II system group system location object. You can see the system location string with the **show snmp** command.

Examples

The following example shows how to configure a system location string:

```
ServiceBroker(config)# snmp-server location Building 3/Room 214
```

Related Commands

Command	Description
show snmp	Displays the SNMP parameters.
snmp-server community	Configures the community access string to permit access to the SNMP.
snmp-server contact	Sets the system server contact string.
snmp-server enable traps	Enables the SB to send SNMP traps.
snmp-server group	Defines a user security model group.
snmp-server host	Specifies the hosts to receive SNMP traps.
snmp-server notify inform	Configures the SNMP notify inform request.
snmp-server user	Defines a user who can access the SNMP engine.
snmp-server view	Defines a SNMPv2 MIB view.

snmp-server notify inform

To configure the SNMP notify inform request, use the **snmp-server notify inform** command in Global configuration mode. To return the setting to the default value, use the **no** form of this command.

snmp-server notify inform

no snmp-server notify inform

Syntax Description

This command has no arguments or keywords.

Defaults

If you do not enter the **snmp-server notify inform** command, the default is an SNMP trap request.

Command Modes

Global configuration (config) mode.

Usage Guidelines

The **snmp-server host** command specifies which hosts receive informs. The **snmp-server enable traps** command globally enables the production mechanism for the specified notifications (traps and informs).

For a host to receive an inform, enable the inform globally by entering the **snmp-server notify inform** command.

The SNMP inform requests feature allows SBs to send inform requests to SNMP managers. SBs can send notifications to SNMP managers when particular events occur. For example, an agent SB might send a message to a manager when the agent SB experiences an error condition.

SNMP notifications can be sent as traps or inform requests. Traps are unreliable because the receiver does not send any acknowledgment when it receives a trap. The sender cannot determine if the trap was received. However, an SNMP manager that receives an inform request acknowledges the message with an SNMP response protocol data unit (PDU). If the manager does not receive an inform request, it does not send a response. If the sender never receives a response, the inform request can be sent again. Informs are more likely to reach their intended destination.

Because they are more reliable, informs consume more resources in the SB and in the network. Unlike a trap, which is discarded as soon as it is sent, an inform request must be held in the memory until a response is received or the request times out. Also, traps are sent only once, while an inform may be retried several times. The retries increase traffic and contribute to a higher overhead on the network. Traps and inform requests provide a trade-off between reliability and resources.

Tip If it is important that the SNMP manager receives every notification, then you should use inform requests in your network. If you are concerned about traffic on your network or about the memory in the SB and you do not need to receive every notification, then you should use traps in your network.

Examples

The following example shows how to configure the SNMP notify inform request on the SB:

```
ServiceBroker(config)# snmp-server notify inform
```

Related Commands

Command	Description
show snmp	Displays the SNMP parameters.
snmp-server community	Configures the community access string to permit access to the SNMP.
snmp-server contact	Sets the system server contact string.
snmp-server enable traps	Enables the SB to send SNMP traps.
snmp-server group	Defines a user security model group.
snmp-server host	Specifies the hosts to receive SNMP traps.
snmp-server location	Sets the SNMP system location string.
snmp-server user	Defines a user who can access the SNMP engine.
snmp-server view	Defines a SNMPv2 MIB view.

snmp-server user

To define a user who can access the SNMP server, use the **snmp-server user** command in Global configuration mode. To remove access, use the **no** form of this command.

snmp-server user *name group* [**auth** {**md5** *password* [**priv** *password*] | **sha** *password* [**priv** *password*]} | **remote** *octet_string* [**auth** {**md5** *password* [**priv** *password*] | **sha** *password* [**priv** *password*]}]]

no snmp-server user *name group* [**auth** {**md5** *password* | **sha** *password*} [**priv** *password*] | **remote** *octetstring* [**auth** {**md5** *password* | **sha** *password*} [**priv** *password*]]]

Syntax Description

name	Name of the SNMP user. Use letters, numbers, dashes, and underscores, but no blanks. This is the name of the user on the SNMP host who wants to communicate with the SNMP agent on the SB. You can enter a maximum of 64 characters.
group	Name of the group to which the SNMP user belongs. You can enter a maximum of 64 characters.
auth	(Optional) Configures user authentication parameters.
md5	Configures the Hashed-Based Message Authentication Code Message Digest 5 (HMAC MD5) authentication algorithm.
password	HMAC MD5 user authentication password.
priv	(Optional) Configures authentication parameters for the packet.
password	HMAC MD5 user private password. You can enter a maximum of 256 characters.
sha	Configures the HMAC Secure Hash Algorithm (SHA) authentication algorithm.
password	HMAC SHA authentication password. You can enter a maximum of 256 characters.
remote	(Optional) Specifies the engine identity of the remote SNMP entity to which the user belongs.
octet_string	Globally unique identifier for a remote SNMP entity (for example, the SNMP network management station) for at least one of the SNMP users.

Defaults

None

Command Modes

Global configuration (config) mode.

Usage Guidelines

The maximum number of SNMP users that can be created is 10. Follow these guidelines when defining SNMP users for SBs:

- If SNMPv3 is going to be used for SNMP requests, define at least one SNMPv3 user account on the SB for the SB to be accessed through SNMP.
- Group defined with the SNMPv1 or SNMPv2c security model should not be associated with SNMP users; they should only be associated with the community strings.

Tip To send an SNMPv3 inform message, you must configure at least one SNMPv3 user with a remote SNMP ID option on the SB. The SNMP ID is entered in octet string form. For example, if the IP address of a remote SNMP entity is 192.147.142.129, then the octet string would be 00:00:63:00:00:00:a1:c0:93:8e:81.

Examples

The following example shows that an SNMPv3 user account is created on the SB. The SNMPv3 user is named acme and belongs to the group named admin. Because this SNMP user account has been set up with no authentication password, the SNMP agent on the SB does not perform authentication on SNMP requests from this user.

```
ServiceBroker(config)#  snmp-server user acme admin
```

Related Commands

Command	Description
show snmp	Displays the SNMP parameters.
snmp-server community	Configures the community access string to permit access to the SNMP.
snmp-server contact	Sets the system server contact string.
snmp-server enable traps	Enables the SB to send SNMP traps.
snmp-server group	Defines a user security model group.
snmp-server host	Specifies the hosts to receive SNMP traps.
snmp-server location	Sets the SNMP system location string.
snmp-server notify inform	Configures the SNMP notify inform request.
snmp-server view	Defines a SNMPv2 MIB view.

snmp-server view

To define a SNMP Version 2 (SNMPv2) MIB view, use the **snmp-server view** command in Global configuration mode. To undefine the MIB view, use the **no** form of this command.

snmp-server view *view_name MIB_family* {**excluded** | **included**}

no snmp-server view *view_name MIB_family* {**excluded** | **included**}

Syntax Description

view_name	Name of this family of view subtrees. You can enter a maximum of 64 characters.
MIB_family	An object identifier that identifies a subtree of the MIB. You can enter a maximum of 64 characters.
excluded	Excludes the MIB family from the view.
included	Includes the MIB family from the view.

Defaults

None

Command Modes

Global configuration (config) mode.

Usage Guidelines

An *SNMP view* is a mapping between SNMP objects and the access rights available for those objects. An object can have different access rights in each view. Access rights indicate whether the object is accessible by either a community string or a user. The **snmp-server view** command is used with the **snmp-server group** to limit the read-write access of MIB trees based on the group. Because the group can be associated with the SNMP community string or users, using the **snmp-server view** command extends the limit to users and community strings. If the view is not configured, read-write access to the community string applies to the MIB tree and all users (SNMPv3).

The maximum number of views that can be created is 10. You can configure the SNMP view settings only if you have previously configured the SNMP server settings.

To remove a view record, use the **no snmp-server view** command.

You can enter the **snmp-server view** command multiple times for the same view record. Later lines take precedence when an object identifier is included in two or more lines.

Note When configuring an SNMP View with Excluded, the specified MIB that is excluded is not accessible for the community associated with the group that has that view.

Examples

The following example shows how to configure the view name, family name, and view type:

```
ServiceBroker(config)# snmp-server view contentview ciscoServiceBrokerMIB included
```

The following example creates a view that includes all objects in the MIB-II system group and all objects in the Cisco enterprise MIB:

```
ServiceBroker(config)# snmp-server view phred system included
ServiceBroker(config)# snmp-server view phred cisco included
```

The following example shows how to create a view that includes all objects in the MIB-II system group except for sysServices (System 7) in the MIB-II interfaces group:

```
ServiceBroker(config)# snmp-server view agon system included
ServiceBroker(config)# snmp-server view agon system.7 excluded
```

Related Commands

Command	Description
show snmp	Displays the SNMP parameters.
snmp-server community	Configures the community access string to permit access to the SNMP.
snmp-server contact	Sets the system server contact string.
snmp-server enable traps	Enables the SB to send SNMP traps.
snmp-server group	Defines a user security model group.
snmp-server host	Specifies the hosts to receive SNMP traps.
snmp-server location	Sets the SNMP system location string.
snmp-server notify inform	Configures the SNMP notify inform request.
snmp-server user	Defines a user who can access the SNMP engine.

SS

To dump socket statistics, use the **ss** command in EXEC configuration mode.

ss *line*

Syntax Description

line	ss connection information, -h to get help.

Command Defaults

None

Command Modes

EXEC configuration.

Usage Guidelines

The **ss** utility is used to dump socket statistics. It shows information similar to the **netstat** command and displays more TCP information than other tools.

When specifying the options and filters, you can use the short form of the option (a single dash followed by a character) or the long form of the option (two dashes followed by the whole word). To view the list of options and filters, enter **ss -h** (or **ss --help**) and the list of options and filters are displayed along with descriptions.

```
ServiceBroker# ss -h
Usage: ss [OPTIONS]
       ss [OPTIONS] [FILTER]
   -h, --help           this message
   -V, --version        output version information
   -n, --numeric        does not resolve service names
   -r, --resolve       resolve host names
   -a, --all            display all sockets
   -l, --listening      display listening sockets
   -o, --options       show timer information
   -e, --extended      show detailed socket information
   -m, --memory        show socket memory usage
   -p, --processes      show process using socket
   -i, --info           show internal TCP information
   -s, --summary        show socket usage summary

   -4, --ipv4          display only IP version 4 sockets
   -0, --packet         display PACKET sockets
   -t, --tcp            display only TCP sockets
   -u, --udp            display only UDP sockets
   -d, --dccp           display only DCCP sockets
   -w, --raw            display only RAW sockets
   -x, --unix           display only Unix domain sockets
   -7, --filter display when tcp rqueue threshold meet
   -8, --filter display when tcp wqueue threshold meet
   -9, --filter display when tcp retransmit threshold meet
   -W, --filter display only window scale disable
   -B, --background display output in new format
   -L, --no_loop_back   display without loopback interface
   -S, --basic_output  display basic information
   -f, --family=FAMILY display sockets of type FAMILY

   -A, --query=QUERY
```

```
        QUERY := {all | inet | tcp | udp | raw | unix | packet | netlink}[,QUERY]

   -F, --filter=FILE   read filter information from FILE
       FILTER := [state TCP-STATE] [EXPRESSION]
```

With the **-A** query option, you list the identifiers (all, inet, tcp, udp, and so on) of the socket tables you want displayed, separated by commas.

With the -F filter option, you can filter by TCP state, or using a boolean expression you can filter by IP addresses and ports.

The default output does not resolve host addresses (IP addresses) and does resolve service names (usually stored in local files). To resolve host addresses, use the **-r** option. To suppress resolution of service names, use the **-n** option.

Examples

The following command shows how to display all TCP sockets:

```
ServiceBroker# ss -t -a
```

The following command shows how to display all UDP sockets:

```
ServiceBroker# ss -u -a
```

The following command shows how to display all established SSH connections and display the timer information:

```
ServiceBroker# ss -o state established '(dport = :ssh or sport = :ssh)'
```

The following command shows how to display all established HTTP connections and display the timer information:

```
ServiceBroker# ss -o state established '(dport = :http or sport = :http)'
```

Related Commands

Command	Description
gulp	Captures lossless gigabit packets and writes them to disk.
netmon	Displays the transmit and receive activity on an interface.
netstatr	Displays the rate of change of netstat statistics.
tcpmon	Searches all TCP connections.

ssh-key-generate

To generate the SSH host key, use the **ssh-key-generate** command in Global configuration mode. To disable the SSH key, use the **no** form of this command.

ssh-key-generate [**key-length** *num*]

no ssh-key-generate [**key-length** *num*]

Syntax Description

key-length	Configures the length of SSH key.
num	Specifies the number of bits in the SSH key to create.

Defaults

key-length *bits*: 2048

Command Modes

Global configuration (config) mode.

Usage Guidelines

SSH enables login access to the SB through a secure and encrypted channel. SSH consists of a server and a client program. Like Telnet, you can use the client program to remotely log on to a machine that is running the SSH server, but unlike Telnet, messages transported between the client and the server are encrypted. The functionality of SSH includes user authentication, message encryption, and message authentication.

When you enable the SSH server, the Secure File Transfer Protocol (SFTP) server is also enabled. The SFTP is a file transfer program that provides a secure and authenticated method for transferring files between VDS-SB devices and other workstations or clients.

Note SFTP is the standard file transfer protocol introduced in SSH Version 2. The SFTP client functionality is provided as part of the SSH component. If you use SSH Version 1 on the SB, SFTP support is not available.

Examples

The following example shows how to generate an SSH host key on an SB:

```
ServiceBroker(config)# ssh-key-generate key-length 2048
```

The following example disables the ssh host key:

```
ServiceBroker(config)# no ssh-key-generate key-length 2048
```

Related Commands

Command	Description
show ssh	Displays the SSH status and configuration.

sshd

To enable the Secure Shell (SSH) daemon, use the **sshd** command in Global configuration mode. To disable SSH, use the **no** form of this command.

sshd {**enable** | **timeout** *seconds* | **version** {**1** | **2**}}

no sshd {**enable** | **password-guesses** | **timeout** | **version** {**1** | **2**}}

Syntax Description

enable	Enables the SSH feature.
timeout	Configures the number of seconds for which an SSH session is active during the negotiation (authentication) phase between the client and the server before it times out. **Note** If you have established an SSH connection to the SB but have not entered the username when prompted at the login prompt, the connection is terminated by the SB even after successful login if the grace period expires.
seconds	SSH login grace time value, in seconds. The range is from 1 to 99999. The default is 300.
version	Configures the SSH version to be supported on the SB.
1	Specifies that SSH Version 1 is supported on the SB.
2	Specifies that SSH Version 2 is supported on the SB.

Defaults

timeout *seconds*: 300

version: Both SSH Version 1 and 2 are enabled.

Command Modes

Global configuration (config) mode.

Usage Guidelines

SSH enables login access to the SB through a secure and encrypted channel. SSH consists of a server and a client program. Like Telnet, you can use the client program to remotely log on to a machine that is running the SSH server, but unlike Telnet, messages transported between the client and the server are encrypted. The functionality of SSH includes user authentication, message encryption, and message authentication.

When you enable the SSH server, the Secure File Transfer Protocol (SFTP) server is also enabled. The SFTP is a file transfer program that provides a secure and authenticated method for transferring files between VDS-SB devices and other workstations or clients.

Note SFTP is the standard file transfer protocol introduced in SSH Version 2. The SFTP client functionality is provided as part of the SSH component. If you use SSH Version 1 on the SB, SFTP support is not available.

The **sshd version** command in Global configuration mode allows you to enable support for either SSH Version 1 or SSH Version 2. When you enable SSH using the **sshd enable** command in Global configuration mode, the VDS-SB software enables support for both SSH Version 1 and SSH Version 2 on the SB. If you want the SB to support only one version of SSH (for example SSH Version 2), disable the other version (in this example, SSH Version 1) by using the **no sshd version 1** command.

When support for both SSH Version 1 and SSH Version 2 are enabled in the SB, the **show running-config** command output does not display any sshd configuration. If you have disabled the support for one version of SSH, the **show running-config** command output contains the following line:

```
no sshd version version_number
```

Note You cannot disable both SSH versions in an SB. Use the **no sshd enable** command in Global configuration mode to disable SSH on the SB.

Examples

The following example shows how to enable the SSH daemon and configure the number of allowable password guesses and timeout for the SB:

```
ServiceBroker(config)# sshd enable
ServiceBroker(config)# sshd password-guesses 4
ServiceBroker(config)# sshd timeout 20
```

The following example disables the support for SSH Version 1 in the SB:

```
ServiceBroker(config)# no sshd version 1
```

Related Commands

Command	Description
show ssh	Displays the SSH status and configuration.

sysreport

To save the sysreport to a user-specified file, use the **sysreport** privilege command in EXEC configuration mode.

sysreport {**authentication** [**date-range** *start_date end_date* | *filename*] | **cms** [**date-range** *start_date end_date* | *filename*] | **dns** | **ftp** | **http** | **icap**}

Syntax Description

authentication	Generates sysreport information related to http authentication.
cms	Generates sysreport information related to Centralized Management System (CMS).
dns	Generates sysreport information related to Domain Name Server (DNS).
ftp	Generates sysreport information related to FTP.
http	Generates sysreport information related to HTTP.
icap	Generates sysreport information related to ICAP

Defaults

None

Command Modes

Privilege EXEC configuration mode.

Examples

The following example saves the sysreport for WMT to a user-specified file:

```
ServiceBroker# sysreport wmt date-range 2009/05/07 2009/05/11 xxx.tar.gz
The sysreport has been saved onto file xxx.tar.gz in local1
```

tacacs

To configure TACACS+ server parameters, use the **tacacs** command in Global configuration mode. To disable individual options, use the **no** form of this command.

tacacs {**host** {*hostname* | *ip_address*} [**primary**] | **key** *keyword* | **password ascii** | **retransmit** *retries* | **timeout** *seconds*}

no tacacs {**host** {*hostname* | *ip_address*} [**primary**] | **key** | **password ascii** | **retransmit** | **timeout**}

Syntax Description

host	Sets a server address.
hostname	Hostname of the TACACS+ server.
ip_address	IP address of the TACACS+ server.
primary	(Optional) Sets the server as the primary server.
key	Sets the security word.
keyword	Keyword. An empty string is the default.
password ascii	Specifies ASCII as the TACACS+ password type.
retransmit	Sets the number of times that requests are retransmitted to a server.
retries	Number of retry attempts allowed. The range is from 1 to 3. The default is 2.
timeout	Sets the number of seconds to wait before a request to a server is timed out.
seconds	Timeout, in seconds. The range is from 1 to 20. The default is 5.

Defaults

keyword: none (empty string)

timeout *seconds*: 5

retransmit *retries*: 2

password ascii: PAP

Command Modes

Global configuration (config) mode.

Usage Guidelines

Using the **tacacs** command, configure the TACACS+ key, the number of retransmits, the server hostname or IP address, and the timeout.

Execute the following two commands to enable user authentication with a TACACS+ server:

```
ServiceBroker(config)# authentication login tacacs enable
ServiceBroker(config)# authentication configuration tacacs enable
```

HTTP request authentication is independent of user authentication options and must be disabled with the following separate commands:

```
ServiceBroker(config)# no authentication login tacacs enable
ServiceBroker(config)# no authentication configuration tacacs enable
```

The Users GUI page or the **username** command in Global configuration provide a way to add, delete, or modify usernames, passwords, and access privileges in the local database. The TACACS+ remote database can also be used to maintain login and configuration privileges for administrative users. The **tacacs host** command or the TACACS+ Service Broker GUI page allows you to configure the network parameters required to access the remote database.

One primary and two backup TACACS+ servers can be configured; authentication is attempted on the primary server first and then on the others in the order in which they were configured. The primary server is the first server configured unless another server is explicitly specified as primary with the **tacacs host** *hostname* **primary** command.

Use the **tacacs key** command to specify the TACACS+ key that is used to encrypt the packets sent to the server. This key must be the same as the one specified on the server daemon. The maximum number of characters in the key should not exceed 99 printable ASCII characters (except tabs). An empty key string is the default. All leading spaces are ignored; spaces within and at the end of the key string are not ignored. Double quotes are not required even if there are spaces in the key, unless the quotes themselves are part of the key.

The **tacacs timeout** is the number of seconds that the Service Broker waits before declaring a timeout on a request to a particular TACACS+ server. The range is from 1 to 20 seconds with 5 seconds as the default. The number of times that the Service Broker repeats a retry-timeout cycle before trying the next TACACS+ server is specified by the **tacacs retransmit** command. The default is two retry attempts.

Three unsuccessful login attempts are permitted. TACACS+ logins may appear to take more time than local logins depending on the number of TACACS+ servers and the configured timeout and retry values.

Use the **tacacs password ascii** command to specify the TACACS+ password type as ASCII. The default password type is Password Authentication Protocol (PAP). In earlier releases, the password type was not configurable. When users needed to log in to a Service Broker, a TACACS+ client sent the password information in PAP format to a TACACS+ server. However, TACACS+ servers that were configured for router management required the passwords to be in ASCII cleartext format instead of PAP format to authenticate users logging in to the Service Broker. The password type to authenticate user information to ASCII was configurable from the CLI.

Note When the **no tacacs password ascii** command is used to disable the ASCII password type, the password type is once again reset to PAP.

The TACACS+ client can send different requests to the server for user authentication. The client can send a TACACS+ request with the PAP password type. In this scenario, the authentication packet includes both the username and the user's password. The server must have an appropriately configured user's account.

Alternatively, the client can send a TACACS+ request with the ASCII password type as another option. In this scenario, the authentication packet includes the username only and waits for the server response. Once the server confirms that the user's account exists, the client sends another Continue request with the user's password. The Authentication Server must have an appropriately configured user's account to support either type of password.

Examples

The following example shows how to configure the key used in encrypting packets:

```
ServiceBroker(config)# tacacs key human789
```

The following example shows how to configure the host named spearhead as the primary TACACS+ server:

```
ServiceBroker(config)# tacacs host spearhead primary
```

The following example shows how to set the timeout interval for the TACACS+ server:

```
ServiceBroker(config)# tacacs timeout 10
```

The following example shows how to set the number of times that authentication requests are retried (retransmitted) after a timeout:

```
ServiceBroker(config)# tacacs retransmit 5
```

The following example shows the password type to be PAP by default:

```
ServiceBroker# show tacacs
    Login Authentication for Console/Telnet Session: enabled (secondary)
    Configuration Authentication for Console/Telnet Session: enabled (secondary)

    TACACS+ Configuration:
    ---------------------
    TACACS+ Authentication is off
    Key        = *****
    Timeout    = 5
    Retransmit = 2
    Password type: pap

    Server                          Status
    ----------------------------   ------
    10.107.192.148                primary
    10.107.192.168
    10.77.140.77
ServiceBroker#
```

However, you can configure the password type to be ASCII using the **tacacs password ascii** command. You can then verify the changes using the **show tacacs** command as follows:

```
ServiceBroker(config)# tacacs password ascii
ServiceBroker(config)# exit
ServiceBroker# show tacacs
    Login Authentication for Console/Telnet Session: enabled (secondary)
    Configuration Authentication for Console/Telnet Session: enabled (secondary)

    TACACS+ Configuration:
    ---------------------
    TACACS+ Authentication is off
    Key        = *****
    Timeout    = 5
    Retransmit = 2
    Password type: ascii

    Server                          Status
    ----------------------------   ------
    10.107.192.148                primary
    10.107.192.168
    10.77.140.77
```

Related Commands

Command	Description
show authentication	Displays the authentication configuration.
show statistics tacacs	Displays the Service Broker TACACS+ authentication and authorization statistics.
show tacacs	Displays TACACS+ authentication protocol configuration information.

tcpdump

To dump the network traffic, use the **tcpdump** command in EXEC configuration mode.

tcpdump [*LINE*]

Syntax Description

LINE	(Optional) Dump options.

Defaults

None

Command Modes

EXEC configuration mode.

Usage Guidelines

Use the **tcpdump** command to gather a sniffer trace on the SB, or VDSM for troubleshooting when asked to gather the data by the Cisco Technical Support. This utility is very similar to the Linux or UNIX **tcpdump** command.

The **tcpdump** command allows an administrator (must be an admin user) to capture packets from the Ethernet. On the SB 500 series, the interface names are GigabitEthernet 1/0 and GigabitEthernet 2/0. On all VDS-SB platforms, we recommend that you specify a path/filename in the local1 directory.

You can do a straight packet header dump to the screen by entering the **tcpdump** command. Press **Ctrl-C** to stop the dump.

The **tcpdump** command has the following options:

- **-w** <*filename*>—Writes the raw packet capture output to a file.
- **-s** <*count*>—Captures the first <*count*> bytes of each packet.
- **-i** <*interface*>—Allows you to specify a specific interface to use for capturing the packets.
- **-c** <*count*>—Limits the capture to <*count*> packets.

The following example captures the first 1500 bytes of the next 10,000 packets from interface Ethernet 0 and puts the output in a file named dump.pcap in the local1 directory on the SB:

```
ServiceBroker# tcpdump -w /local1/dump.pcap -i GigabitEthernet 1/0 -s 1500 -c 10000
```

When you specify the **-s** option, it sets the packet snap length. The default value captures only 64 bytes, and this default setting saves only packet headers into the capture file. For troubleshooting of redirected packets or higher level traffic (HTTP, authentication, and so on), copy the complete packets.

After the TCP dump has been collected, you need to move the file from the SB to a PC so that the file can be viewed by a sniffer decoder.

```
ftp <ip address of the SB>

!--- Log in using the admin username and password.

    cd local1
    bin
    hash
```

```
     get <name of the file>

!--- Using the above example, it would be dump.pcap.

     bye
```

We recommend that you use Ethereal as the software application for reading the TCP dump. With Ethereal, you can decode packets that are encapsulated into a GRE tunnel. See the Ethereal website for further information.

Note In most cases, redirected packets captured by the tcpdump facility with the VDS-SB CLI differ from the data received on the interface. The destination IP address and TCP port number are modified to reflect the device IP address and the port number 8999.

Examples

The following example shows how to dump the TCP network traffic:

```
ServiceBroker# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on GigabitEthernet 1/0, link-type EN10MB (Ethernet), capture size 68 bytes
12:45:43.017677 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P
3342832089:3342832201(112) ack 1248615673 win 15232
12:45:43.018950 IP 172.19.226.63 > ServiceBroker.cisco.com: icmp 36: 172.19.226.63 udp
port 2048 unreachable
12:45:43.019327 IP ServiceBroker.cisco.com.10015 > dns-sj2.cisco.com.domain:  49828+ [  |
domain ]
12:45:43.021158 IP dns-sj2.cisco.com.domain > ServiceBroker.cisco.com.10015:  49828
NXDomain* [  | domain ]
12:45:43.021942 IP ServiceBroker.cisco.com.10015 > dns-sj2.cisco.com.domain:  49829+ [  |
domain ]
12:45:43.023799 IP dns-sj2.cisco.com.domain > ServiceBroker.cisco.com.10015:  49829
NXDomain* [  | domain ]
12:45:43.024240 IP ServiceBroker.cisco.com.10015 > dns-sj2.cisco.com.domain:  49830+ [  |
domain ]
12:45:43.026164 IP dns-sj2.cisco.com.domain > ServiceBroker.cisco.com.10015:  49830* [  |
domain ]
12:45:42.702891 802.1d config TOP_CHANGE 8000.00:03:9f:f1:10:63.8042 root
8000.00:01:43:9a:c8:63 pathcost 26 age 3 max 20 hello 2 fdelay 15
12:45:42.831404 IP 10.77.140.97.4314 > ServiceBroker.cisco.com.ssh: . ack 112 win 64351
12:45:42.831490 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: . 112:1444(1332) ack 1
win 15232
12:45:42.831504 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 1444:1568(124) ack 1
win 15232
12:45:42.831741 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 1568:1696(128) ack 1
win 15232
12:45:43.046176 IP 10.77.140.97.4314 > ServiceBroker.cisco.com.ssh: . ack 1568 win 65535
12:45:43.046248 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 1696:2128(432) ack 1
win 15232
12:45:43.046469 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 2128:2256(128) ack 1
win 15232
12:45:43.046616 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 2256:2400(144) ack 1
win 15232
12:45:43.107700 802.1d config TOP_CHANGE 8000.00:03:9f:f1:10:63.8042 root
8000.00:01:43:9a:c8:63 pathcost 26 age 3 max 20 hello 2 fdelay 15
12:45:43.199710 IP 10.77.140.97.4314 > ServiceBroker.cisco.com.ssh: . ack 1696 win 65407
12:45:43.199784 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 2400:2864(464) ack 1
win 15232
12:45:43.199998 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 2864:2992(128) ack 1
win 15232
```

```
12:45:43.259968 IP 10.77.140.97.4314 > ServiceBroker.cisco.com.ssh: . ack 2400 win 64703
12:45:43.260064 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 2992:3280(288) ack 1
win 15232
12:45:43.260335 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 3280:3408(128) ack 1
win 15232
12:45:43.260482 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 3408:3552(144) ack 1
win 15232
12:45:43.260621 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 3552:3696(144) ack 1
win 15232
12:45:43.413320 IP 10.77.140.97.4314 > ServiceBroker.cisco.com.ssh: . ack 2992 win 65535
12:45:43.413389 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 3696:3984(288) ack 1
win 15232
12:45:43.413597 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 3984:4112(128) ack 1
win 15232
12:45:43.413741 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 4112:4256(144) ack 1
win 15232
12:45:43.473601 IP 10.77.140.97.4314 > ServiceBroker.cisco.com.ssh: . ack 3552 win 64975
12:45:43.473659 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 4256:4544(288) ack 1
win 15232
12:45:43.473853 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 4544:4672(128) ack 1
win 15232
12:45:43.473994 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 4672:4816(144) ack 1
win 15232
12:45:43.474132 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 4816:4960(144) ack 1
win 15232
12:45:43.484117 IP 10.77.140.97.4314 > ServiceBroker.cisco.com.ssh: P 1:81(80) ack 3696
win 64831
12:45:43.484167 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 4960:5248(288) ack
81 win 15232
12:45:43.484424 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 5248:5392(144) ack
81 win 15232
12:45:43.627125 IP 10.77.140.97.4314 > ServiceBroker.cisco.com.ssh: . ack 4112 win 64415
12:45:43.627204 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 5392:5680(288) ack
81 win 15232
12:45:43.627439 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 5680:5808(128) ack
81 win 15232
12:45:43.627586 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 5808:5952(144) ack
81 win 15232
12:45:43.688261 IP 10.77.140.97.4314 > ServiceBroker.cisco.com.ssh: . ack 4544 win 65535
12:45:43.688316 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 5952:6240(288) ack
81 win 15232
12:45:43.688495 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 6240:6368(128) ack
81 win 15232
12:45:43.688638 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 6368:6512(144) ack
81 win 15232
12:45:43.689012 IP 10.77.140.97.4314 > ServiceBroker.cisco.com.ssh: . ack 4960 win 65119
12:45:43.689046 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 6512:6800(288) ack
81 win 15232
12:45:43.689170 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 6800:6928(128) ack
81 win 15232
12:45:43.689309 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 6928:7072(144) ack
81 win 15232
12:45:43.689447 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 7072:7216(144) ack
81 win 15232
12:45:43.698391 IP 10.77.140.97.4314 > ServiceBroker.cisco.com.ssh: . ack 5392 win 64687
12:45:43.698437 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 7216:7504(288) ack
81 win 15232
12:45:43.698599 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 7504:7632(128) ack
81 win 15232
12:45:43.698740 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 7632:7776(144) ack
81 win 15232
12:45:43.840558 IP 10.77.140.97.4314 > ServiceBroker.cisco.com.ssh: . ack 5808 win 64271
12:45:43.840622 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 7776:8064(288) ack
81 win 15232
```

```
12:45:43.840819 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 8064:8192(128) ack
81 win 15232
12:45:43.840962 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 8192:8336(144) ack
81 win 15232
12:45:43.901868 IP 10.77.140.97.4314 > ServiceBroker.cisco.com.ssh: . ack 6368 win 65535
12:45:43.901938 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 8336:8624(288) ack
81 win 15232
12:45:43.901887 IP 10.77.140.97.4314 > ServiceBroker.cisco.com.ssh: . ack 6928 win 64975
12:45:43.901910 IP 10.77.140.97.4314 > ServiceBroker.cisco.com.ssh: . ack 7216 win 64687
12:45:43.902137 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 8624:8752(128) ack
81 win 15232
12:45:43.902281 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 8752:8896(144) ack
81 win 15232
12:45:43.902414 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 8896:9024(128) ack
81 win 15232
12:45:43.902547 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 9024:9152(128) ack
81 win 15232
12:45:43.902687 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 9152:9296(144) ack
81 win 15232
12:45:43.902826 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 9296:9440(144) ack
81 win 15232
12:45:43.902965 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 9440:9584(144) ack
81 win 15232
12:45:43.903104 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 9584:9728(144) ack
81 win 15232
12:45:43.922413 IP 10.77.140.97.4314 > ServiceBroker.cisco.com.ssh: . ack 7632 win 64271
12:45:43.922459 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 9728:10304(576) ack
81 win 15232
12:45:43.922622 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 10304:10432(128) ack
81 win 15232
12:45:43.922764 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 10432:10576(144) ack
81 win 15232
12:45:44.053872 IP 10.77.140.97.4314 > ServiceBroker.cisco.com.ssh: . ack 8192 win 65535
12:45:44.053972 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 10576:10864(288) ack
81 win 15232
12:45:44.054308 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 10864:11104(240) ack
81 win 15232
12:45:44.054453 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 11104:11248(144) ack
81 win 15232
12:45:44.054596 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 11248:11392(144) ack
81 win 15232
12:45:44.111702 802.1d config TOP_CHANGE 8000.00:03:9f:f1:10:63.8042 root
8000.00:01:43:9a:c8:63 pathcost 26 age 3 max 20 hello 2 fdelay 15
12:45:44.114626 IP 10.77.140.97.4314 > ServiceBroker.cisco.com.ssh: . ack 8752 win 64975
12:45:44.114712 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 11392:11712(320) ack
81 win 15232
12:45:44.115219 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 11712:11952(240) ack
81 win 15232
12:45:44.115381 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 11952:12096(144) ack
81 win 15232
12:45:44.115426 IP 10.77.140.97.4314 > ServiceBroker.cisco.com.ssh: . ack 9152 win 64575
12:45:44.115617 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 12096:12336(240) ack
81 win 15232
12:45:44.115760 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 12336:12480(144) ack
81 win 15232
12:45:44.115904 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 12480:12624(144) ack
81 win 15232
12:45:44.116045 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 12624:12768(144) ack
81 win 15232
12:45:44.116094 IP 10.77.140.97.4314 > ServiceBroker.cisco.com.ssh: . ack 9440 win 64287
12:45:44.116114 IP 10.77.140.97.4314 > ServiceBroker.cisco.com.ssh: . ack 9728 win 65535
12:45:44.116332 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 12768:13088(320) ack
81 win 15232
```

```
12:45:44.116473 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 13088:13232(144) ack
81 win 15232
12:45:44.116614 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 13232:13376(144) ack
81 win 15232
12:45:44.116755 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 13376:13520(144) ack
81 win 15232
12:45:44.116895 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 13520:13664(144) ack
81 win 15232
12:45:44.135947 IP 10.77.140.97.4314 > ServiceBroker.cisco.com.ssh: . ack 10432 win 64831
12:45:44.135996 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 13664:13808(144) ack
81 win 15232
12:45:44.136223 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 13808:14048(240) ack
81 win 15232
12:45:44.136366 IP ServiceBroker.cisco.com.ssh > 10.77.140.97.4314: P 14048:14192(144) ack
81 win 15232
12:45:44.144104 IP 10.77.140.97.4314 > ServiceBroker.cisco.com.ssh: P 81:161(80) ack 10576
win 64687

102 packets captured
105 packets received by filter
0 packets dropped by kernel
```

The following example shows how to dump the TCP network traffic and redirect it to a file named test:

```
ServiceBroker# tcpdump port 8080 -w test
tcpdump: listening on GigabitEthernet 1/0, link-type EN10MB (Ethernet), capture size 68
bytes
216 packets captured
216 packets received by filter
0 packets dropped by kernel
```

tcpdumpx

To dump the network traffic with the tcpdump extension for a multi-interface capture, use the **tcpdumpx** command in EXEC configuration mode.

tcpdumpx [*LINE*]

Syntax Description

LINE	(Optional) Dump options, -h to get help.

Defaults

None

Command Modes

EXEC configuration mode.

Usage Guidelines

The **tcpdumpx** command enables tcpdump to capture multiple interfaces in separate files. Each member interface of a PortChannel can be captured in a separate file. For example, if eth2, eth3, eth4 and eth5 are members of PortChannel 1 (bond0), they can be captured in different files.

Current: issue "tcpdump -i" for each PortChannel member in a different shell at the same time.

Implemented: New flag (-j), not used by tcpdump, under tcpdumpx handles this:

```
tcpdumpx -j PortChannel 1 -w filename.cap
```

This command internally expands to capture each physical interface's dump in an individual file:

```
tcpdump -i eth2 -w filename.eth2.cap
tcpdump -i eth3 -w filename.eth3.cap
tcpdump -i eth4 -w filename.eth4.cap
tcpdump -i eth5 -w filename.eth5.cap
```

If eth2 and eth3 need to be captured, use "--" as a command separator to separate the two tcpdump instances:

```
tcpdumpx -i eth2 -w filename.cap -k -m -- -i eth3 -w filename2.cap -c -k -- … --
```

This command internally expands to:

```
tcpdump -i eth2 -w filename.cap
tcpdump -i eth3 -w filename.cap
```

Other examples:

```
tcpdumpx -j PortChannel 1 -w filename.cap -- -j PortChannel 2 -w filename2.cap
tcpdumpx -i eth2 -w filename.cap -- -i eth3 -w filename2.cap -- j PortChannel 1  -w
filename3.cap
```

This is documented in tcpdumpx help "tcpdumpx -h":

```
tcpdump                   Dump traffic on a network
tcpdumpx                 tcpdump extension for multi-interface capture
tcpdumpx -h
tcpdumpx - tcpdump extension for multiple interface capture
[WARNING] This program consumes HIGH CPU & memory and impacts system performance

Usage: tcpdumpx [-w filename] [-j PortChannel X] [--] [all tcpdump options]
```

```
[-w filename]          Required. Write tcpdump output to filename
[-j PortChannel X]     Capture each PortChannel slave to file:
                       "filename" --> "filenameslavename"
                       "filename.xxx"   --> "filename.slavename.xxx"
[--]                   Interface seperator. Capture Multiple Interfaces by:
                       tcpdumpx -i eth0 -w eth0 -- -i eth2 -w eth2 -- ... -- ..
                       tcpdumpx -i eth0 -w eth0 -- -j PortChannel 1 -w pc
                       tcpdumpx -j PortChannel 1 -w pc1 -- -j PortChannel 2
                       -w pc2
[all tcpdump options] Specify any tcpdump options
                       Please use "tcpdump -h" to get tcpdump help options
[-h(elp)]              Print this help
```

Examples

The following example shows how to dump the TCP network traffic with a tcpdump extension for multi-interface capture:

```
ServiceBroker# tcpdumpx
```

tcpmon

To search all TCP connections, use the **tcpmon** command in EXEC configuration mode.

tcpmon *line*

Syntax Description

line	Shows TCP connection information, -h to get help.

Command Defaults

None

Command Modes

EXEC configuration.

Usage Guidelines

The **tcpmon** utility is a script that constantly calls the ss utility at specified intervals. The **tcpmon** utility searches all TCP connections every 30 seconds and displays information about any socket that meets the search criteria. To view the list of options, enter **tcpmon -h**.

Table 4-16 describes the tcpmon output fields.

Table 4-16 tcpmon Output Fields

Field	Description
State	One of the following TCP connection states: ESTAB, SYN-SENT, SYN-RECV, FIN-WAIT-1, FIN-WAIT-2, TIME-WAIT, CLOSE-WAIT, LAST-ACK, LISTEN, and CLOSING.
Recv-Q	Number of bytes in the receiving queue.
Send-Q	Number of bytes in the sending queue.
Local Address: Port	Source address and port.
Peer Address: Port	Destination address and port.
Rtt/var	Average round-trip time (in seconds) and the deviation.
Send	Current sending rate (in Mbps).
Retrans	Number of retransmit timeouts.

Examples

The following command sets the polling cycle to 30 seconds and the receive-queue threshold to 100:

```
ServiceBroker# tcpmon -R 100 30
```

The following command sets the polling cycle to 30 seconds and displays only the sockets with window scaling disabled:

```
ServiceBroker# tcpmon -N 30
```

The following example shows the output for the **tcpmon** utility:

```
State   Recv-Q   Send-Q   Local Address:Port    Peer Address:Port  Rtt/var   Swnd  Retrans
ESTAB     0      257744      10.3.5.2:80         10.3.5.137:32963   530/15    13      0
```

```
ESTAB      0       861560      10.3.5.2:80          10.3.5.137:32849   545/24      4       0
ESTAB      0       234576      10.3.5.2:80          10.3.5.122:32979   547/22.2    6       0
ESTAB      0       254848      10.3.5.2:80          10.3.5.103:32909   531/14.8   10       0
ESTAB      0       231680      10.3.5.2:80          10.3.5.135:32925   532/11.5    9       0
ESTAB      0       224440      10.3.5.2:80          10.3.5.133:33057   550/32      7       0
ESTAB      0       267880      10.3.5.2:80          10.3.5.135:32985   530/18.2    7       0
ESTAB      0       291048      10.3.5.2:80          10.3.5.113:32909   539/12.2    6       0
ESTAB      0       249056      10.3.5.2:80          10.3.5.103:32903   520/23.2    8       0
ESTAB      0       218648      10.3.5.2:80          10.3.5.132:33069   522/14.5   16       0
ESTAB      0       702280      10.3.5.2:80          10.3.5.100:32829   539/24.5    5       0
ESTAB      0       412680      10.3.5.2:80          10.3.5.110:32992   546/22.8    7       0
ESTAB      0       254848      10.3.5.2:80          10.3.5.115:33136   552/37.2    5       0
```

Related Commands

Command	Description
gulp	Captures lossless gigabit packets and writes them to disk.
netmon	Displays the transmit and receive activity on an interface.
netstatr	Displays the rate of change of netstat statistics.
ss	Dumps socket statistics.

tcp

To configure TCP-related parameters, use the **tcp timestamp** command in Global configuration mode. To disable the TCP timestamp, use the **no** form of this command.

tcp timestamp

no tcp timestamp

Syntax Description

timetstamp	Enables TCP timestamps.

Defaults

TCP timestamp is enabled by default.

Command Modes

Global configuration (config) mode.

Examples

The following example shows how to disable the TCP timestamp:

```
ServiceBroker# no tcp timestamp
ServiceBroker#
```

telnet (EXEC Configuration)

To log in to a network device using the Telnet client, use the **telnet** command in EXEC configuration mode.

telnet {*hostname* | *ip_address*} [*port_num*]

Syntax Description

hostname	Hostname of the network device.
ip_address	IP address of the network device.
port_num	(Optional) Port number. The range is from 1 to 65535. Default port number is 23.

Defaults

The default port number is 23.

Command Modes

EXEC configuration mode.

Usage Guidelines

Some UNIX shell functions, such as escape and the **suspend** command, are not available in the Telnet client. In addition, multiple Telnet sessions are also not supported.

The Telnet client allows you to specify a destination port. By entering the **telnet** command, you can test websites by attempting to open a Telnet session to the website from the SB CLI.

Examples

The following example shows how to open a Telnet session to a network device using the hostname:

```
ServiceBroker# telnet cisco-ce
```

The following example shows how to open a Telnet session to a network device using the IP address:

```
ServiceBroker# telnet 172.16.155.224
```

The following example shows how to open a Telnet session to a network device on port 8443 using the hostname:

```
ServiceBroker# telnet cisco-ce 8443
```

The following example shows how to open a Telnet session to a network device on port 80 using the hostname:

```
ServiceBroker# telnet www.yahoo.com 80
```

telnet (Global Configuration)

To enable Telnet service, use the **telnet enable** command in Global configuration mode. To disable Telnet, use the **no** form of this command.

telnet

no telnet

Syntax Description

enable	Enables Telnet service.

Defaults

Telnet is enabled by default.

Command Modes

Global configuration (config) mode.

Usage Guidelines

Use this Terminal Emulation protocol for a remote terminal connection. The **telnet enable** command allows users to log in to other devices using a Telnet session.

Examples

The following example shows how to enable Telnet on the SB:

```
ServiceBroker(config)# telnet enable
```

Related Commands

Command	Description
show telnet	Displays the Telnet services configuration.

terminal

To set the number of lines displayed in the console window, or to display the current console **debug** command output, use the **terminal** command in EXEC configuration mode.

terminal {**length** *length* | **monitor** [**disable**]}

Syntax Description

length	Sets the length of the display on the terminal.
length	Length of the display on the terminal (the range is 0 to 512). Setting the length to 0 means that there is no pausing.
monitor	Copies the debug output to the current terminal.
disable	(Optional) Disables monitoring at this specified terminal.

Defaults

The default length is 24 lines.

Command Modes

EXEC configuration mode.

Usage Guidelines

When 0 is entered as the *length* parameter, the output to the screen does not pause. For all nonzero values of *length*, the -More- prompt is displayed when the number of output lines matches the specified *length* number. The -More- prompt is considered a line of output. To view the next screen, press the **Spacebar**. To view one line at a time, press the **Enter** key.

The **terminal monitor** command allows a Telnet session to display the output of the **debug** commands that appear on the console. Monitoring continues until the Telnet session is terminated.

Examples

The following example shows how to set the number of lines to display to 20:

```
ServiceBroker# terminal length 20
```

The following example shows how to configure the terminal for no pausing:

```
ServiceBroker# terminal length 0
```

Related Commands

All **show** commands.

test-url

To test the accessibility of a URL using FTP, HTTP, or HTTPS, use the **test-url** command in EXEC configuration mode.

test-url {**ftp** *url* [**use-ftp-proxy** *proxy_url*] | **http** *url* [**custom-header** *header* [**head-only**] [**use-http-proxy** *proxy_url*] | **head-only** [**custom-header** *header*] [**use-http-proxy** *proxy_url*] | **use-http-proxy** *proxy_url* [**custom-header** *header*] [**head-only**]]}

Syntax Description

ftp	Specifies the FTP URL to be tested.
url	FTP URL to be tested. Use one of the following formats to specify the FTP URL: • ftp://*domainname*/*path* • ftp://*user*:*password*@*domainname*/*path*
use-ftp-proxy	(Optional) Specifies the FTP proxy that is used to test the URL.
proxy_url	FTP proxy URL. Use one of the following formats to specify the proxy URL: • *proxy IP Address*:*proxy Port* • *proxy Username*:*proxy Password*@*proxy IP Address*:*proxy Port*
http	Specifies the HTTP URL to be tested.
url	HTTP URL to be tested. Use one of the following formats to specify the HTTP URL: • http://*domainname*/*path* • http://*user*:*password*@*domainname*/*path*
custom-header	(Optional) Specifies the custom header information to be sent to the server.
header	Custom header information to be sent to the server. Use the format *header*:*line* to specify the custom header.
head-only	(Optional) Specifies that only the HTTP header information must be retrieved.
use-http-proxy	(Optional) Specifies the HTTP proxy that is used to test the URL.
proxy_url	HTTP proxy URL. Use one of the following formats to specify the HTTP proxy URL: http://*proxyIp*:*proxyPort* http://*proxyUser*:*proxypasswd*@*proxyIp*:*proxyPort*
head-only	(Optional) Specifies that only the HTTPS header information must be retrieved.

Defaults

None

Command Modes

EXEC configuration mode.

Usage Guidelines

The HTTP CLI client allows you to test connectivity and debug caching issues. The **test-url** command allows you to test whether a URL is accessible over the FTP, HTTP, and HTTPS protocols. When you test the connectivity using the **test-url** command, the SB sends a request using the protocol that you have specified to the server and fetches the requested contents. The actual content is dumped into the path /dev/null, and the server response with the header information is displayed to the user.

You can use the **test-url ftp** command to test the following for the specified URL:

- Connectivity to the URL
- Connectivity to the URL through the FTP proxy (using the **use-ftp-proxy** option)
- Authentication
- FTP proxy authentication

You can use the **test-url http** command to test the following for the specified URL:

- Test the connectivity to the URL
- Test the connectivity to the URL through the HTTP proxy (using the **use-http-proxy** option)
- Authentication
- HTTP proxy authentication
- Header information only for the specified page (using the **head-only** option) or additional header information (using the **custom-header** option)

Examples

The following example tests the accessibility to the URL http://192.168.171.22 using HTTP:

```
ServiceBroker# test-url http http://ce1.server.com
--02:27:20-- http://ce1.server.com/
           => `/dev/null'
Len - 22 , Restval - 0 , contlen - 0 , Res - 134728056Resolving ce1.server.com..

done.
Connecting to ce1.server.com [ 192.168.171.22 ] :80... connected.
HTTP request sent, awaiting response...
 1 HTTP/1.1 200 OK
 2 Date: Mon, 26 Jul 2004 08:41:34 GMT
 3 Server: Apache/1.2b8
 4 Last-Modified: Fri, 25 Apr 2003 12:23:04 GMT
 5 ETag: "1aee29-663-3ea928a8"
 6 Content-Length: 1635
 7 Content-Type: text/html
 8 Via: 1.1 Content Delivery System Software 5.2
 9 Connection: Keep-Alive
 (1635 to go)
 0% [                                    ] 0             --.--K/s    ETA --:--L
en - 0   ELen - 1635     Keepalive - 1
100% [ ====================================> ] 1,635          1.56M/s    ETA 00:00

02:27:20 (1.56 MB/s) - `/dev/null' saved [ 1635/1635 ]
```

The following example tests the accessibility to the URL http://192.168.171.22 through the HTTP proxy 10.107.192.148:

```
ServiceBroker# test-url http http://192.168.171.22 use-http-proxy 10.107.192.148:8090
--15:22:51--  http://10.77.155.246/
           => `/dev/null'
Len - 1393 , Restval - 0 , contlen - 0 , Res - 134728344Connecting to
10.107.192.148:8090... connected.
Proxy request sent, awaiting response...
```

```
 1 HTTP/1.1 401 Authorization Required
 2 Date: Mon, 27 Sep 2004 15:29:18 GMT
 3 Server: Apache/1.3.27 (Unix) tomcat/1.0
 4 WWW-Authenticate: Basic realm="IP/TV Restricted Zone"
 5 Content-Type: text/html; charset=iso-8859-1
 6 Via: 1.1 Content Delivery System Software 5.2.1
 7 Connection: Close
Len - 0 , Restval - 0 , contlen - -1 , Res - -1Connecting to 10.107.192.148:8090...
connected.
Proxy request sent, awaiting response...
 1 HTTP/1.1 401 Authorization Required
 2 Date: Mon, 27 Sep 2004 15:29:19 GMT
 3 Server: Apache/1.3.27 (Unix) tomcat/1.0
 4 WWW-Authenticate: Basic realm="IP/TV Restricted Zone"
 5 Content-Type: text/html; charset=iso-8859-1
 6 Via: 1.1 Content Delivery System Software 5.2.1
 7 Connection: Keep-Alive
 (1635 to go)
 0% [                                    ] 0             --.--K/s    ETA --:--L
en - 0   ELen - 1635     Keepalive - 1
100% [ ====================================> ] 1,635          1.56M/s    ETA 00:00

02:27:20 (1.56 MB/s) - `/dev/null' saved [ 1635/1635 ]
```

The following example tests the accessibility to the URL ftp://ssivakum:ssivakum@10.77.157.148 using FTP:

```
ServiceBroker# test-url ftp ftp://ssivakum:ssivakum@10.77.157.148/antinat-0.90.tar
Mar 30 14:33:44 nramaraj-ce admin-shell: %SB-PARSER-6-350232: CLI_LOG shell_parser_log:
test-url ftp ftp://ssivakum:ssivakum@10.77.157.148/antinat-0.90.tar
--14:33:44--  ftp://ssivakum:*password*@10.77.157.148/antinat-0.90.tar
           => `/dev/null'
Connecting to 10.77.157.148:21... connected.
Logging in as ssivakum ...
220 (vsFTPd 1.1.3)
--> USER ssivakum

331 Please specify the password.
--> PASS Turtle Power!
230 Login successful. Have fun.
--> SYST

215 UNIX Type: L8
--> PWD

257 "/home/ssivakum"
--> TYPE I

200 Switching to Binary mode.
==> CWD not needed.
--> PORT 10,1,1,52,82,16

200 PORT command successful. Consider using PASV.
--> RETR antinat-0.90.tar

150 Opening BINARY mode data connection for antinat-0.90.tar (1771520 bytes).
Length: 1,771,520 (unauthoritative)

 0% [
] 0             --.--K/s    ETA --:--Len - 0   ELen - 1771520          Keepalive - 0
100% [
================================================================================> ]
1,771,520    241.22K/s    ETA 00:00
```

```
226 File send OK.
14:33:53 (241.22 KB/s) - `/dev/null' saved [ 1771520 ]

ServiceBroker#
```

Related Commands

Command	Description
acquirer (EXEC)	Starts or stops content acquisition on a specified acquirer delivery service.

top

To see a dynamic real-time view of a running VDS-SB, use the **top** command in EXEC configuration mode.

top {*line*}

Syntax Description

line	Specifies top options, enter **-h** to get Help. Press **q** to quit from the output.

Defaults

No default behavior values

Command Modes

EXEC configuration mode.

Examples

The following example shows sample output from the **top** command on an SB:

```
ServiceBroker# top
top - 01:08:45 up 8 days, 23:39, 3 users, load average: 1244.22, 1246.32, 1243.66
Tasks: 1789 total,  4 running, 1785 sleeping,  0 stopped,  0 zombie
Cpu(s): 0.0%us, 13.2%sy, 18.1%ni, 57.8%id, 1.1%wa, 0.7%hi, 9.2%si, 0.0%st
Mem: 32825728k total, 32671416k used,  154312k free,  137164k buffers
Swap:       0k total,        0k used,        0k free, 21289468k cached
```

traceroute

To trace the route to a remote host, use the **traceroute** command in EXEC configuration mode.

traceroute {*hostname* | *ip_address*}

Syntax Description

hostname	Name of the remote host.
ip_address	IP address of the remote host.

Defaults

None

Command Modes

EXEC configuration mode.

Usage Guidelines

Traceroute is a widely available utility on most operating systems. Similar to ping, traceroute is a valuable tool for determining connectivity in a network. Ping allows the user to find out if there is a connection between the two end systems. Traceroute does this as well, but additionally lists the intermediate routers between the two systems. Users can see the routes that packets can take from one system to another. Use the **traceroute** command to find the route to a remote host when either the hostname or the IP address is known.

The **traceroute** command uses the TTL field in the IP header to cause routers and servers to generate specific return messages. Traceroute starts by sending a UDP datagram to the destination host with the TTL field set to 1. If a router finds a TTL value of 1 or 0, it drops the datagram and sends back an ICMP time-exceeded message to the sender. The traceroute facility determines the address of the first hop by examining the source address field of the ICMP time-exceeded message.

To identify the next hop, traceroute sends a UDP packet with a TTL value of 2. The first router decrements the TTL field by 1 and sends the datagram to the next router. The second router sees a TTL value of 1, discards the datagram, and returns the time-exceeded message to the source. This process continues until the TTL is incremented to a value large enough for the datagram to reach the destination host (or until the maximum TTL is reached).

To determine when a datagram has reached its destination, traceroute sets the UDP destination port in the datagram to a very large value that the destination host is unlikely to be using. When a host receives a datagram with an unrecognized port number, it sends an ICMP “port unreachable” error to the source. This message indicates to the traceroute facility that it has reached the destination.

Examples

The following example shows how to trace the route to a remote host from the SB:

```
ServiceBroker# traceroute 10.77.157.43
traceroute to 10.77.157.43 (10.77.157.43), 30 hops max, 38 byte packets
 1  10.1.1.50 (10.1.1.50)  2.024 ms  2.086 ms  2.219 ms
 2  sblab2-rtr.cisco.com (192.168.10.1)  3.718 ms 172.19.231.249 (172.19.231.249)  0.653
ms  0.606 ms
 3  sjc22-00lab-gw1.cisco.com (172.24.115.65)  0.666 ms  0.624 ms  0.597 ms
 4  sjc20-lab-gw2.cisco.com (172.24.115.109)  0.709 ms  0.695 ms  0.616 ms
 5  sjc20-sbb5-gw2.cisco.com (128.107.180.97)  0.910 ms  0.702 ms  0.674 ms
 6  sjc20-rbb-gw5.cisco.com (128.107.180.9)  0.762 ms  0.702 ms  0.664 ms
 7  sjc12-rbb-gw4.cisco.com (128.107.180.2)  0.731 ms  0.731 ms  0.686 ms
```

```
 8  sjc5-gb3-f1-0.cisco.com (10.112.2.158)  1.229 ms  1.186 ms  0.753 ms
 9  capnet-hkidc-sjc5-oc3.cisco.com (10.112.2.238)  146.784 ms  147.016 ms  147.051 ms
10  hkidc-capnet-gw1-g3-1.cisco.com (10.112.1.250)  147.163 ms  147.319 ms  148.050 ms
11  hkidc-gb3-g0-1.cisco.com (10.112.1.233)  148.137 ms  148.332 ms  148.361 ms
12  capnet-singapore-hkidc-oc3.cisco.com (10.112.2.233)  178.137 ms  178.273 ms  178.005
ms
13  singapore-capnet2-fa4-0.cisco.com (10.112.2.217)  179.236 ms  179.606 ms  178.714 ms
14  singapore-gb1-fa2-0.cisco.com (10.112.2.226)  179.499 ms  179.914 ms  179.873 ms
15  capnet-chennai-singapore-ds3.cisco.com (10.112.2.246)  211.858 ms  212.167 ms  212.854
ms
16  hclodc1-rbb-gw2-g3-8.cisco.com (10.112.1.213)  213.639 ms  212.580 ms  211.211 ms
17  10.77.130.18 (10.77.130.18)  212.248 ms  212.478 ms  212.545 ms
18  codc-tbd.cisco.com (10.77.130.34)  212.315 ms  213.088 ms  213.063 ms
19  10.77.130.38 (10.77.130.38)  212.955 ms  214.353 ms  218.169 ms
20  10.77.157.9 (10.77.157.9)  217.217 ms  213.424 ms  222.023 ms
21  10.77.157.43 (10.77.157.43)  212.750 ms  217.260 ms  214.610 ms
```

The following example shows how the **traceroute** command fails to trace the route to a remote host from the SB:

```
ServiceBroker# traceroute 10.0.0.1
traceroute to 10.0.0.1 (10.0.0.1), 30 hops max, 38 byte packets
 1  10.1.1.50 (10.1.1.50)  2.022 ms  1.970 ms  2.156 ms
 2  sblab2-rtr.cisco.com (192.168.10.1)  3.955 ms 172.19.231.249 (172.19.231.249)  0.654
ms  0.607 ms
 3  sjc22-00lab-gw1.cisco.com (172.24.115.65)  0.704 ms  0.625 ms  0.596 ms
 4  sjc20-lab-gw1.cisco.com (172.24.115.105)  0.736 ms  0.686 ms  0.615 ms
 5  sjc20-sbb5-gw1.cisco.com (128.107.180.85)  0.703 ms  0.696 ms  0.646 ms
 6  sjc20-rbb-gw5.cisco.com (128.107.180.22)  0.736 ms  0.782 ms  0.750 ms
 7  sjce-rbb-gw1.cisco.com (171.69.7.249)  1.291 ms  1.314 ms  1.218 ms
 8  sjce-corp-gw1.cisco.com (171.69.7.170)  1.477 ms  1.257 ms  1.221 ms
 9  * * *
10  * * *
.
.
.
29  * * *
30  * * *
```

Table 4-17 describes the fields in the **traceroute** command output.

Table 4-17 traceroute Command Output Fields

Field	Description
30 hops max, 38 byte packets	Maximum TTL value and the size of the ICMP datagrams being sent.
2.022 ms 1.970 ms 2.156 ms	Total time (in milliseconds) for each ICMP datagram to reach the router or host plus the time it took for the ICMP time-exceeded message to return to the host. An exclamation point following any of these values (for example, 20 ms) indicates that the port-unreachable message returned by the destination had a TTL of 0 or 1. Typically, this situation occurs when the destination uses the TTL value from the arriving datagram as the TTL in its ICMP reply. The reply does not arrive at the source until the destination receives a traceroute datagram with a TTL equal to the number of hops between the source and destination.
*	An asterisk (*) indicates that the timeout period (default of 5 seconds) expired before an ICMP time-exceeded message was received for the datagram.

Related Commands

Command	Description
ping	Sends echo packets for diagnosing basic network connectivity on networks.

transaction-log force

Command	Description
ipv6	Specifies the IPv6 address of the default gateway.

To force the archive or export of the transaction log, use the **transaction-log force** command in EXEC configuration mode.

transaction-log force {**archive** | **export**}

Syntax Description

archive	Forces the archive of the *working.log* file.
export	Forces the archived files to be exported to the server.

Defaults

None

Command Modes

EXEC configuration mode.

Usage Guidelines

The **transaction-log force archive** command causes the transaction log *working.log* file to be archived to the SB hard disk following the next transaction. This command has the same effect as the **clear transaction-log** command.

The **transaction-log force export** command causes the transaction log to be exported to an FTP server designated by the **transaction-logs export ftp-server** command.

The **transaction-log force** command does not change the configured or default schedule for archive or export of transaction log files. If the archive interval is configured, in seconds, or the export interval is configured in minutes, the forced archive or export interval period is restarted after the forced operation.

If a scheduled archive or export job is in progress when a corresponding **transaction-log force** command is entered, the command has no effect. If a **transaction-log force** command is in progress when an archive or export job is scheduled to run, the forced operation is completed and the archive or export is rescheduled for the next configured interval.

Examples

The following example shows how to archive the transaction log file to the SB hard disk:

```
ServiceBroker# transaction-log force archive
```

The following example shows that the SB is configured to export its transaction logs to two FTP servers:

```
ServiceBroker(config)#  transaction-logs export ftp-server 10.1.1.1 mylogin mypasswd
/ftpdirectory
ServiceBroker(config)#  transaction-logs export ftp-server myhostname mylogin mypasswd
/ftpdirectory
```

The following example shows how to export the transaction log file from the SB hard disk to an FTP server designated by the **transaction-logs export ftp-server** command:

```
ServiceBroker# transaction-log force export
```

Related Commands

Command	Description
clear transaction logs	Clears the working transaction log settings.
show statistics transaction-logs	Displays the SB transaction log export statistics.
show transaction-logging	Displays the transaction log configuration settings and a list of archived transaction log files.
transaction-logs	Configures and enables the transaction logging parameters.

transaction-logs

To configure and enable transaction logs, use the **transaction-logs** command in Global configuration mode. To disable transaction logs, use the **no** form of this command.

transaction-logs {**archive** {**interval** {*seconds* | **every-day** {**at** *hour:minute* | **every** *hours*} | **every-hour** {**at** *minute* | **every** *minutes*} | **every-week** [**on** *weekdays* **at** *hour:minute*]} | **max-file-number** *file_number* | **max-file-size** *file_size*} | **ds-snapshot-counter enable** | **enable** | **export** {**compress** | **enable** | **ftp-server** {*hostname* | *serv_ip_addrs*} *login passw directory* | **interval** {*minutes* | **every-day** {**at** *hour:minute* | **every** *hours*} | **every-hour** {**at** *minute* | **every** *minutes*} | **every-week** [**on** *weekdays* **at** *hour:minute*] | **sftp-server** {*hostname* | *serv_ip_addrs*} *login passw directory* | **format** {**apache** | **custom** *string* | **extended-squid**} | **log-windows-domain**}

no transaction-logs {**archive** {**interval** {*seconds* | **every-day** {**at** *hour:minute* | **every** *hours*} | **every-hour** {**at** *minute* | **every** *minutes*} | **every-week** [**on** *weekdays* **at** *hour:minute*]} | **max-file-number** *file_number* | **max-file-size** *file_size*} | **ds-snapshot-counter enable** | **enable** | **export** {**compress** | **enable** | **ftp-server** {*hostname* | *serv_ip_addrs*} *login passw directory* | **interval** {*minutes* | **every-day** {**at** *hour:minute* | **every** *hours*} | **every-hour** {**at** *minute* | **every** *minutes*} | **every-week** [**on** *weekdays* **at** *hour:minute*] | **sftp-server** {*hostname* | *serv_ip_addrs*} *login passw directory* | **format** {**apache** | **custom** *string* | **extended-squid**} | **log-windows-domain**}

Syntax Description

archive	Configures archive parameters.
interval	Determines how frequently the archive file is to be saved.
seconds	Frequency of archiving, in seconds. The range is from120 to 604800.
every-day	Archives using intervals of 1 day or less.
at	Specifies the local time at which to archive each day.
hour:minute	Time of day at which to archive in local time (hh:mm).
every	Specifies the interval in hours. Interval aligns with midnight.
hours	Number of hours for daily file archive. 1—Hourly 12—Every 12 hours 2—Every 2 hours 24—Every 24 hours 3—Every 3 hours 4—Every 4 hours 6—Every 6 hours 8—Every 8 hours
every-hour	Specifies the archives using intervals of 1 hour or less.
at	Sets the time to archive at each hour.
minute	Minute alignment for the hourly archive. The range is from 0 to 59.
every	Specifies the interval in minutes for hourly archive that aligns with the top of the hour.

minutes	Number of minutes for hourly archive. 10—Every 10 minutes 15—Every 15 minutes 2—Every 2 minutes 20—Every 20 minutes 30—Every 30 minutes 5—Every 5 minutes
every-week	Archives using intervals of 1 or more times a week.
on	(Optional) Sets the day of the week on which to archive.
weekdays	Weekdays on which to archive. One or more weekdays can be specified. Fri—Every Friday Mon—Every Monday Sat—Every Saturday Sun—Every Sunday Thu—Every Thursday Tue—Every Tuesday Wed—Every Wednesday
at	(Optional) Sets the local time at which to archive each day.
hour:minute	Time of day at which to archive in local time (hh:mm).
max-file-number	Sets the maximum number of the archived log file.
file_number	Maximum number of the archived log file. The range is from 1 to 10000.
max-file-size	Sets the maximum archive file size.
filesize	Maximum archive file size in kilobytes. The range is from 1000 to 2000000.
ds-snapsot-counter enable	Enables the per delivery service snapshot counter.
enable	Enables the transaction log.
export	Configures file export parameters.
compress	Compresses the archived files in the gzip format before exporting.
enable	Enables the exporting of log files at the specified interval.
ftp-server	Sets the FTP server to receive exported archived files.
hostname	Hostname of the target FTP server.
serv_ip_addrs	IP address of the target FTP server.
login	User login to target FTP server.
passw	User password to target FTP server.
directory	Target directory path for exported files on FTP server.
interval	Determines how frequently the file is to be exported.
minutes	Number of minutes in the interval at which to export a file. The range is from 1 to 10080.
every-day	Specifies the exports using intervals of 1 day or less.
at	Specifies the local time at which to export each day.
hour:minute	Time of day at which to export in local time (hh:mm).
every	Specifies the interval in hours for the daily export.

hours	Number of hours for the daily export. 1—Hourly 12—Every 12 hours 2— Every 2 hours 24—Every 24 hours 3— Every 3 hours 4—Every 4 hours 6—Every 6 hours 8—Every 8 hours
every-hour	Specifies the exports using intervals of 1 hour or less.
at	Specifies the time at which to export each hour.
minute	Minute alignment for the hourly export. The range is from 0 to 59.
every	Specifies the interval in minutes that align with the top of the hour.
minutes	Number of minutes for the hourly export. 10—Every 10 minutes 15—Every 15 minutes 2—Every 2 minutes 20—Every 20 minutes 30—Every 30 minutes 5—Every 5 minutes
every-week	Specifies the exports using intervals of 1 of more times a week.
on	(Optional) Specifies the days of the week for the export.
weekdays	Weekdays on which to export. One or more weekdays can be specified. Fri—Every Friday Mon—Every Monday Sat—Every Saturday Sun—Every Sunday Thu—Every Thursday Tue—Every Tuesday Wed—Every Wednesday
at	(Optional) Specifies the time of day at which to perform the weekly export.
hour:minute	Time of day at which to export in the local time (hh:mm).
sftp-server	Sets the SFTP[1] server to receive exported archived files.
hostname	Hostname of the target SFTP server.
serv_ip_addrs	IP address of the target SFTP server.
login	User login to the target SFTP server (less than 40 characters).
passw	User password to the target SFTP server (less than 40 characters).
directory	Target directory path for exported files on the SFTP server.
format	Sets the format to use for the HTTP transaction log entries in the working.log file.
apache	Configures the HTTP transaction logs output to the Apache CLF[2].
custom	Configures the HTTP transaction logs output to the custom log format.
string	Quoted log format string containing the custom log format.
extended-squid	Configures the HTTP transaction logs output to the Extended Squid log format.

log-windows-domain	Logs the Windows domain with an authenticated username if available in HTTP transaction log entries.
enable	Enables the remote transaction logging.
entry-type	Specifies the type of transaction log entry.
all	Sets the SB to send all transaction log messages to the remote syslog server.
request-auth-failures	Sets the SB to log to the remote syslog server only those transactions that the SB failed to authenticate with the Authentication Server. **Note** Only those authentication failures that are associated with an end user who is attempting to contact the Authentication Server are logged. The transactions in pending state (that have contacted the Authentication Server, but waiting for a response from the Authentication Server) are not logged.
facility	Configures a unique facility to create a separate log on the remote syslog host for real-time transaction log entries.
parameter	Specifies one of the following facilities: auth—Authorization system daemon—System daemons kern—Kernel local0—Local use local1—Local use local2—Local use local3—Local use local4—Local use local5—Local use local6—Local use local7—Local use mail—Mail system news—USENET news syslog—Syslog itself user—User process uucp—UUCP system
host	Configures the remote syslog server.
hostname	Hostname of the remote syslog server.
ip-address	IP address of the remote syslog server.
port	Configures the port to use when sending transaction log messages to the syslog server.
port-num	Port number to use when sending transaction log messages to the syslog server. The default is 514.
rate-limit	Configures the rate at which the transaction logger is allowed to send messages to the remote syslog server.
rate	Rate (number of messages per second) at which the transaction logger is allowed to send messages to the remote syslog server.

1. SFTP = Secure File Transfer Protocol
2. CLF = common log format

Defaults

archive: disabled

enable: disabled

export compress: disabled

export: disabled

file-marker: disabled

archive interval: every day, every one hour

archive max-file-size: 2,000,000 KB

export interval: every day, every one hour

format: apache

logging port *port_num*: 514

Command Modes

Global configuration (config) mode.

Usage Guidelines

SBs can record all errors and access activities. Each content service module on the SB provides logs of the requests that were serviced. These logs are referred to as transaction logs.

Typical fields in the transaction log are the date and time when a request was made, the URL that was requested, whether it was a cache hit or a cache miss, the type of request, the number of bytes transferred, and the source IP address. Transaction logs are used for problem identification and solving, load monitoring, billing, statistical analysis, security problems, and cost analysis and provisioning.

The translog module on the SB handles transaction logging and supports the Apache CLF, Extended Squid format, and the World Wide Web Consortium (W3C) customizable logging format.

Note For RTSP, when you choose the **Repeat** option from the Play menu in the Windows Media player to play media files continuously in a loop, an extra entry is logged in the transaction logs for each playback of the file. This situation occurs mostly with the WMT RTSPU protocol because of the behavior of the player.

Enable transaction log recording with the **transaction-logs enable** command. The transactions that are logged include HTTP and FTP. In addition, Extensible Markup Language (XML) logging for MMS-over-HTTP and MMS-over-RTSP (RTSP over Windows Media Services 9) is also supported.

When enabled, daemons create a *working.log* file in /local1/logs/ on the sysfs volume for HTTP and FTP transactions and a separate *working.log* file in /local1/logs/export for Windows Media transactions. The posted XML log file from the Windows Media Player to the SB (Windows Media server) can be parsed and saved to the normal WMT transaction logs that are stored on the SB.

The *working.log* file is a link to the actual log file with the timestamp embedded in its filename. When you configure the **transaction-logs archive interval** command, the first transaction that arrives after the interval elapses is logged to the *working.log* file as usual, and then actual log file is archived and a new log file is created. Only transactions subsequent to the archiving event are recorded in the new log file. The *working.log* file is then updated to point to the newly created log file. The transaction log archive file naming conventions are shown in Table 4-18. The SB default archive interval is once an hour every day.

Note The time stamp in the transaction log filename is in UTC and is irrespective of the time zone configured on the SB. The time stamp in the transaction log filename is the time when the file was created. The logs entries in the transaction logs are in the time zone configured on the SB.

Use the **transaction-logs ds-snapshot-counter enable** command to enable or disable snapshot counter transaction logs. This command is available for SB. On SB, the snapshot counter transaction log records per delivery service Storage Usage. On the SB, the snapshot counter transaction log records per delivery service Session and Bandwidth Usage.

Use the **transaction-logs archive max-file-size** command to specify the maximum size of an archive file. The *working.log* file is archived when it attains the maximum file size if this size is reached before the configured archive interval time.

Use the **transaction-logs file-marker** option to mark the beginning and end of the HTTP, HTTPS, and FTP proxy logs. By examining the file markers of an exported archive file, you can determine whether the FTP process transferred the entire file. The file markers are in the form of dummy transaction entries that are written in the configured log format.

The following example shows the start and end dummy transactions in the default native Squid log format.

- 970599034.130 0 0.0.0.0 TCP_MISS/000 0 NONE TRANSLOG_FILE_START - NONE/- -
- 970599440.130 0 0.0.0.0 TCP_MISS/000 0 NONE TRANSLOG_FILE_END - NONE/- -

Use the **format** option to format the HTTP, HTTPS, and FTP proxy log files for custom format, native Squid or Extended Squid formats, or Apache CLF.

The **transaction-logs format custom** command allows you to use a *log format string* to log additional fields that are not included in the predefined native Squid or Extended Squid formats or the Apache CLF. The *log format string* is a string that contains the tokens listed in Table 4-18 and mimics the Apache log format string. The log format string can contain literal characters that are copied into the log file. Two backslashes (\\) can be used to represent a literal backslash, and a backslash followed by a single quotation mark (\') can be used to represent a literal single quotation mark. A literal double quotation mark cannot be represented as part of the log format string. The control characters \t and \n can be used to represent a tab and a new line character, respectively.

Table 4-18 lists the acceptable format tokens for the log format string. The ellipsis (...) portion of the format tokens shown in this table represent an optional condition. This portion of the format token can be left blank, as in %a. If an optional condition is included in the format token and the condition is met, then what is shown in the Value column of Table 4-18 is included in the transaction log output. If an optional condition is included in the format token but the condition is not met, the resulting transaction log output is replaced with a hyphen (-). The form of the condition is a list of HTTP status codes, which may or may not be preceded by an exclamation point (!). The exclamation point is used to negate all the status codes that follow it, which means that the value associated with the format token is logged if none of the status codes listed after the exclamation point (!) match the HTTP status code of the request. If any of the status codes listed after the exclamation point (!) match the HTTP status code of the request, then a hyphen (-) is logged.

For example, %400,501 { User-Agent } i logs the User-Agent header value on 400 errors and 501 errors (Bad Request, Not Implemented) only, and %!200,304,302 { Referer } i logs the Referer header value on all requests that did not return a normal status.

The custom format currently supports the following request headers:

- User-Agent
- Referer

- Host
- Cookie

The output of each of the following Request, Referer, and User-Agent format tokens specified in the custom *log format string* is always enclosed in double quotation marks in the transaction log entry:

%r

% { Referer } i

% { User-Agent } i

The % { Cookie } i format token is generated without the surrounding double quotation marks, because the Cookie value can contain double quotes. The Cookie value can contain multiple attribute-value pairs that are separated by spaces. We recommend that when you use the Cookie format token in a custom format string, you should position it as the last field in the format string so that it can be easily parsed by the transaction log reporting tools. By using the format token string \'% { Cookie } i\' the Cookie header can be surrounded by single quotes (').

Note Each transaction log includes a header line that provides the Cisco VDS Service Broker software version and a summary line as the last line in the transaction log, which includes a summary of all the requests that appear in the transaction log.

The following command can generate the well-known Apache Combined Log Format:

transaction-log format custom " [**% { %d } t/% { %b } t/% { %Y } t:% { %H } t:% { %M } t:% { %S } t % { %z } t] %r %s %b % { Referer } i % { User-Agent } i"**

The following transaction log entry example in the Apache Combined Format is configured using the preceding custom format string:

```
 [ 11/Jan/2003:02:12:44 -0800 ] "GET http://www.cisco.com/swa/i/site_tour_link.gif
HTTP/1.1" 200 3436 "http://www.cisco.com/" "Mozilla/4.0 (compatible; MSIE 5.5; Windows NT
5.0)"
```

Table 4-18 Custom Format Log Format String Values

Format Token	Value
%a	IP address of the requesting client.
%A	IP address of the SB.
%b	Bytes sent, excluding HTTP headers.
%c	Log Entry Generation Time.
%C	Records AuthLOOKupTime|CALLOOKuptime|CacheRouterTime|OSDownload Time in microseconds.
%D	Time consumed to serve the request in microseconds.
%g	Storage URL when URL Resolve rule action is configured in Service Rule file.
%G	Source URL when URL Resolve rule action is configured in Service Rule file.
%h	Remote host (IP address of the requesting client is logged).
%H	Request protocol.
%I	Bytes received from the client.
%J	Gives the average RTT (Round trip time) for that transaction.
%K	Gives the congestion window flickers for the transaction.

Table 4-18 Custom Format Log Format String Values (continued)

Format Token	Value
%L	Prints the asset size, irrespective of the bytes transferred.
%m	Request method.
%M	MIME type of the requested asset.
%N	The network interface and bytes transferred in that interface.
%O	Bytes sent to client, including the headers.
%p	The client who set up the transport session for the request.
%q	Query string (which is preceded by a question mark (?) if a query string exists; otherwise, it is an empty string).
%r	First line of the request. The space in the first line of the request is replaced with a vertical bar (|) delimiter (for example, Get|/index.html|HTTP/1.1)
%R	Request description (Squid description codes).
%s	Status. The translog code always returns the HTTP response code for the request.
%t	Time in common log time format (or standard English format).
%T	Time consumed to serve the request in seconds (a floating point number with 3 decimal places).
%u	URL path requested, including query strings.
%U	URL path requested, not including query strings.
%V	Value of the host request header field reported if the host appeared in the request. If the host did not appear in the host request header, the IP address of the server specified in the URL is reported.
%X	Connection status when the response is completed. The %X field has the following possible values: X-Connection aborted before the response completed. + -Connection may be kept alive after the response is sent. - -Connection is closed after the response is sent.
%Z	Print the request received time stamp in milliseconds; otherwise, the request received time stamp is in seconds.
%{Header-Field}i	Any request header. Replace the Header-Field with the actual header field you want to log; for example, %{Cache-Control}i. **Note** All client request headers are only logged on the edge SB.

Sanitizing Transaction Logs

Use the **sanitized** option to disguise the IP address of clients in the transaction log file. The default is that transaction logs are not sanitized. A sanitized transaction log disguises the network identity of a client by changing the IP address in the transaction logs to 0.0.0.0.

The **no** form of this command disables the sanitize feature. The **transaction-logs sanitize** command does not affect the client IP (%a) value associated with a custom log format string that is configured with the CLI (configured with the **transaction-logs format custom** *string* command in Global configuration

mode in which the string is the quoted log format string that contains the custom log format). To hide the identity of the client IP in the custom log format, either hard code 0.0.0.0 in the custom log format string or exclude the %a token, which represents the client IP, from the format string.

Exporting Transaction Log Files

To facilitate the postprocessing of cache log files, you could export transaction logs to an external host.

This feature allows log files to be exported automatically by FTP to an external host at configurable intervals. The username and password used for FTP are configurable. The directory to which the log files are uploaded is also configurable.

The log files automatically have the following naming convention:

- Module name
- Host IP address
- Date
- Time
- File generation number

For example, the filename for a Web Engine access log would be the following:

```
we_accesslog_apache_192.0.2.22_20091207_065624_00001
```

where we_accesslog_apache is the module name, 192.0.2.22 is the IP address of the device, 20091207 is the date of the log file (December 7, 2009), and 065624_00001 is the file generation number. The file generation number ranges from 00001 to 99999.

Note WMT logs have no .txt extension in the filename.

Exporting and Archiving Intervals

The transaction log archive and export functions are configured with the following commands:

- The **transaction-logs archive interval** command in Global configuration mode allows the administrator to specify when the *working.log* file is archived.
- The **transaction-logs export interval** command in Global configuration mode allows the administrator to specify when the archived transaction logs are exported.

The following limitations apply:

- When the interval is scheduled in units of hours, the value must divide evenly into 24. For example, the interval can be every 4 hours, but not every 5 hours.
- When the interval is scheduled in units of minutes, the value must divide evenly into 60.
- Only the more common choices of minutes are supported. For example, the interval can be 5 minutes or 10 minutes, but not 6 minutes.
- Selection of interval alignment is limited. If an interval is configured for every 4 hours, it aligns with midnight. It cannot align with 12:30 or with 7 a.m.
- Feature does not support different intervals within a 24-hour period. For example, it does not support an interval that is hourly during regular business hours and then every 4 hours during the night.

Transaction Log Archive Filenaming Convention

The archive transaction log file is named as follows for HTTP and WMT caching:

```
celog_10.1.118.5_20001228_235959.txt
```

```
mms_export_10.1.118.5_20001228_235959
```

If the **export compress** feature is enabled when the file is exported, then the file extension is *.gz* after the file is compressed for the export operation, as shown in the following example:

```
celog_10.1.118.5_20001228_235959.txt.gz

mms_export_10.1.118.5_20001228_235959.gz
```

Table 4-19 describes the name elements.

Table 4-19 Archive Log Name Element Descriptions

Sample of Element	Description
acqdist_	Acquisition and distribution archive log file.
cseaccess	Cisco Streaming Engine archive file.
tftp_server_	TFTP server archive file.
webengine_apache	Web Engine Apache transaction logging format log file.
webengine_clf	Web Engine custom transaction logging format log file.
webengine_extsquid	WebEngine extended-squid transaction logging format log file.
fms_access	Flash Media Streaming transaction log file.
fms_authorization	Flash Media Streaming transaction log for authorization and diagnostic logs.
fms_wsl	Flash Media Streaming transaction log for wholesale licensing.
movie-streamer	Movie Streamer transaction log file.
cache_content	Content Access Layer transaction log file.
authsvr	VDS-SB Authorization Server transaction log file.
mms_export_	Standard Windows Media Services 4.1 caching proxy server archive file.
mms_export_e_wms_41_	Extended Windows Media Services 4.1 caching proxy server archive file.
mms_export_wms_90_	Standard Windows Media Services 9.0 caching proxy server archive file.
mms_export_e_wms_90_	Extended Windows Media Services 9.0 caching proxy server archive file.
10.1.118.5_	IP address of the SB creating the archive file.
20001228_	Date on which the archive file was created (yyyy/mm/dd).
235959	Time when the archive file was created (hh/mm/ss).

Table 4-20 lists the directory names and the corresponding examples of the archive filenames.

Table 4-20 Archive Filename Examples and Directories

Directory	Archive Filename
logs/acqdist	acqdist_10.1.94.4_20050315_001545
logs/cisco-streaming-engine	cseaccess10.1.94.4__050315000.log
logs/tftp_server	tftp_server_10.1.94.4_20050315_001545
logs/webengine_apache	we_accesslog_apache_114.0.92.27_20110322_213143_00001

Table 4-20 Archive Filename Examples and Directories (continued)

Directory	Archive Filename
logs/webengine_clf	we_accesslog_clf_114.0.92.27_20110322_213143_00004
logs/webengine_extsquid	we_accesslog_extsqu_114.0.92.27_20110322_213143_00072
logs/fms_access	fms_access_10.1.94.4_20110323_210446_00001
logs/fms_authorization	fms_auth_10.1.94.4_20110323_210446_00001
logs/fms_wsl	fms_wsl_10.1.94.4_20110323_210446_00001
logs/movie-streamer	movie-streamer_10.1.94.4_20110323_210446_00001
logs/cache_content	cache_content_10.1.94.4_20110323_210446_00001
logs/authsvr	authsvr_10.1.94.4_20110323_210446_00001
logs/export	mms_export_18.0.101.116_20110318_121111_00120
logs/export/extended-wms-41	mms_export_e_wms_41_18.0.101.116_20110318_012847_00001
logs/wms-90	mms_export_wms_90_18.0.101.116_20110318_012847_00001
logs/export/extended-wms-90	mms_export_e_wms_90_18.0.101.116_20110318_012847_00001

Compressing Archive Files

The **transaction-logs export compress** option compresses an archive into a gzip file format before exporting it. Compressing the archive file uses less disk space on both the SB and the FTP export server. The compressed file uses less bandwidth when transferred. The archive filename of the compressed file has the extension *.gz*.

Exporting Transaction Logs to External FTP Servers

The **transaction-logs export ftp-server** option can support up to four FTP servers. To export transaction logs, first enable the feature and configure the FTP server parameters. The following information is required for each target FTP server:

- FTP server IP address or the hostname

  The SB translates the hostname with a DNS lookup and then stores the IP address in the configuration.

- FTP user login and user password
- Path of the directory where transferred files are written

  Use a fully qualified path or a relative path for the user login. The user must have write permission to the directory.

Use the **no** form of the **transaction-logs export enable** command to disable the entire transaction logs feature while retaining the rest of the configuration.

Exporting Transaction Logs to External SFTP Servers

Use the **transaction-logs export sftp-server** option to export transaction logs. First enable the feature and configure the Secure File Transfer Protocol (SFTP) server parameters. The following information is required for each target SFTP server:

- SFTP server IP address or the hostname

  The SB translates the hostname with a DNS lookup and then stores the IP address in the configuration.

- SFTP user login and user password
- Path of the directory where transferred files are written

  Use a fully qualified path or a relative path for the user login. The user must have write permission to the directory.

Use the **no** form of the **transaction-logs export enable** command to disable the entire transaction logs feature while retaining the rest of the configuration.

Receiving a Permanent Error from the External FTP Server

A permanent error (Permanent Negative Completion Reply, RFC 959) occurs when the FTP command to the server cannot be accepted, and the action does not take place. Permanent errors can be caused by invalid user logins, invalid user passwords, and attempts to access directories with insufficient permissions.

When an FTP server returns a permanent error to the SB, the export is retried at 10-minute intervals or sooner if the configured export interval is sooner. If the error is a result of a misconfiguration of the **transaction-logs export ftp server** command, then re-enter the SB parameters to clear the error condition. The **show statistics transaction-logs** command displays the status of logging attempts to export servers.

The **show statistics transaction-logs** command shows that the SB failed to export archive files.

The **transaction-logs format** command has three options: **extended-squid**, **apache**, and **custom**.

Use the **no** form of the **transaction-logs export enable** command to disable the entire transaction logs feature while retaining the rest of the configuration.

Configuring Intervals Between 1 Hour and 1 Day

The archive or export interval can be set for once a day with a specific time stamp. It can also be set for hour frequencies that align with midnight. For example, every 4 hours means archiving occurs at 0000, 0400, 0800, 1200, and 1600. It is not possible to archive at half-hour intervals such as 0030, 0430, or 0830. The following intervals are acceptable: 1, 2, 3, 4, 6, 8, 12, and 24.

Configuring Intervals of 1 Hour or Less

The interval can be set for once an hour with a minute alignment. It can also be set for frequencies of less than an hour; these frequencies align with the top of the hour. Every 5 minutes means that archiving occurs at 1700, 1705, and 1710.

Configuring Export Interval on Specific Days

The export interval can be set for specific days of the week at a specific time. One or more days can be specified. The default time is midnight.

Archived logs are automatically deleted when free disk space is low. It is important to select an export interval that exports files frequently enough so that files are not automatically removed before export.

Monitoring HTTP Request Authentication Failures in Real Time

HTTP transaction log messages are sent to a remote syslog server so that you can monitor the remote syslog server for HTTP request authentication failures in real time. This real-time transaction log allows you to monitor transaction logs in real time for particular errors such as HTTP request authentication errors. The existing transaction logging to the local file system remains unchanged.

Note Because system logging (syslog) occurs through UDP, the message transport to the remote syslog host is not reliable.

Summary Line

The transaction logs include a summary line as the last line in the transaction log, which includes a summary of all the requests that appear in the transaction log.

Examples

The following example shows how to configure an FTP server:

```
ServiceBroker(config)# transaction-logs export ftp-server 10.1.1.1 mylogin mypasswd
/ftpdirectory

ServiceBroker(config)# transaction-logs export ftp-server myhostname mylogin mypasswd
/ftpdirectory
```

The following example shows how to delete an FTP server:

```
ServiceBroker(config)# no transaction-logs export ftp-server 10.1.1.1
ServiceBroker(config)# no transaction-logs export ftp-server myhostname
```

Use the **no** form of the command to disable the entire transaction log export feature while retaining the rest of the configuration:

```
ServiceBroker(config)# no transaction-logs export enable
```

The following example shows how to change a username, password, or directory:

```
ServiceBroker(config)# transaction-logs export ftp-server 10.1.1.1 mynewname mynewpass
/newftpdirectory
```

Note For security reasons, passwords are never displayed.

The following example shows how to restart the export of archive transaction logs:

```
ServiceBroker(config)# transaction-logs export ftp-server 172.16.10.5 goodlogin pass
/ftpdirectory
```

The following example shows how to delete an SFTP server from the current configuration:

```
ServiceBroker(config)# no transaction-logs export sftp-server sftphostname
```

The following examples show how to configure the archiving intervals:

```
ServiceBroker(config)# transaction-logs archive interval every-day
  at        Specify the time at which to archive each day
  every     Specify the interval in hours. It will align with midnight

ServiceBroker(config)# transaction-logs archive interval every-day at
<0-23>:  Time of day at which to archive (hh:mm)

ServiceBroker(config)# transaction-logs archive interval every-day every
<1-24>  Interval in hours: { 1, 2, 3, 4, 6, 8, 12 or 24 }
```

The following example shows that the SB has failed to export archive files:

```
ServiceBroker# show statistics transaction-logs
Transaction Log Export Statistics:

Server:172.16.10.5
      Initial Attempts:1
      Initial Successes:0
      Initial Open Failures:0
      Initial Put Failures:0
      Retry Attempts:0
      Retry Successes:0
```

```
Retry Open Failures:0
Retry Put Failures:0
Authentication Failures:1
Invalid Server Directory Failures:0
```

The following example shows how to correct a misconfiguration:

```
ServiceBroker(config)# transaction-logs export ftp-server 10.1.1.1 goodlogin pass
/ftpdirectory
```

The working.log file and archived log files are listed for HTTP and WMT.

The following example shows how to export transaction logs to an SFTP server:

```
ServiceBroker(config)# transaction-logs export sftp-server 10.1.1.100 mylogin mypasswd
/mydir
```

The following example shows how to archive every 4 hours and align with the midnight local time (0000, 0400, 0800, 1200, 1600, and 2000):

```
ServiceBroker(config)# transaction-logs archive interval every-day every 4
```

The following example shows how to export once a day at midnight local time:

```
ServiceBroker(config)# transaction-logs export interval every-day every 24
```

The following example shows how to configure export intervals:

```
ServiceBroker(config)# transaction-logs archive interval every-hour ?
  at        Specify the time at which to archive each day
  every     Specify interval in minutes. It will align with top of the hour

ServiceBroker(config)# transaction-logs archive interval every-hour at ?
  <0-59>  Specify the minute alignment for the hourly archive
ServiceBroker(config)# transaction-logs archive interval every-hour every ?
  <2-30>  Interval in minutes: { 2, 5, 10, 15, 20, 30 }
```

Related Commands

Command	Description
clear transaction-log	Clears the working transaction log settings.
show statistics transaction-logs	Displays the SB transaction log export statistics.
show transaction-logging	Displays the transaction log configuration settings and a list of archived transaction log files.
transaction-log force	Forces the archive or export of the transaction log.

type

To display the contents of a file, use the **type** command in EXEC configuration mode.

type *filename*

Syntax Description

filename	Name of file.

Defaults

None

Command Modes

EXEC configuration mode.

Usage Guidelines

Use this command to display the contents of a file within any SB file directory. This command may be used to monitor features such as transaction logging or system logging (syslog).

Examples

The following example shows how to display the syslog file on the SB:

```
ServiceBroker# type /local1/syslog.txt

Jan 10 22:02:46 (none) populate_ds: %SB-CLI-5-170050: Cisco VDS Service Broker Software
starts booting
Jan 10 22:02:47 (none) create_etc_hosts.sh: %SB-CLI-5-170051: HOSTPLUSDOMAIN: NO-HOSTNAME
Jan 10 22:02:47 NO-HOSTNAME : %SB-CLI-5-170053: Recreated etc_hosts (1, 0)
Jan 10 22:02:48 NO-HOSTNAME Nodemgr: %SB-NODEMGR-5-330082: [ CLI_VER_NTP ] requests stop
service ntpd
Jan 10 22:02:49 NO-HOSTNAME Nodemgr: %SB-NODEMGR-5-330082: [ ver_tvout ] requests stop
service tvoutsvr
Jan 10 22:02:50 NO-HOSTNAME Nodemgr: %SB-NODEMGR-5-330084: [ ver_rtspg ] requests restart
service rtspg
Jan 10 22:02:50 NO-HOSTNAME Nodemgr: %SB-NODEMGR-5-330082: [ ver_iptv ] requests stop
service sbss
Jan 10 22:02:51 NO-HOSTNAME Nodemgr: %SB-NODEMGR-5-330080: [ ver_telnetd ] requests start
service telnetd
Jan 10 22:02:52 NO-HOSTNAME Nodemgr: %SB-NODEMGR-5-330082: [ ver_wmt ] requests stop
service wmt_mms
Jan 10 22:02:53 NO-HOSTNAME Nodemgr: %SB-NODEMGR-5-330082: [ ver_wmt ] requests stop
service wmt_logd
Jan 10 22:02:55 NO-HOSTNAME Nodemgr: %SB-NODEMGR-5-330082: [ Unknown ] requests stop
service mcast_sender
Jan 10 22:02:55 NO-HOSTNAME Nodemgr: %SB-NODEMGR-5-330082: [ Unknown ] requests stop
service mcast_receiver
Jan 10 22:02:56 NO-HOSTNAME Nodemgr: %SB-NODEMGR-5-330024: Service 'populate_ds' exited
normally with code 0
Jan 10 22:02:56 NO-HOSTNAME Nodemgr: %SB-NODEMGR-5-330040: Start service 'parser_server'
using: '/ruby/bin/parser_server' with pid: 1753
Jan 10 22:02:56 NO-HOSTNAME Nodemgr: %SB-NODEMGR-5-330040: Start service
'syslog_bootup_msgs' using: '/ruby/bin/syslog_bootup_msgs' with pid:
1754
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SB-SYS-5-900001: <4>Linux version 2.4.16
(cnbuild@builder2.cisco.com) (gcc version 3.0.4) # 1
SMP Fri Jan 7 19:26:58 PST 2005
```

```
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SB-SYS-5-900001: <6>setup.c: handling
flash window at [ 15MB..16MB)
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SB-SYS-5-900001: <6>BIOS-provided
physical RAM map:
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SB-SYS-5-900001: <4> BIOS-e820:
0000000000000000 - 000000000009ec00 (usable)
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SB-SYS-5-900001: <4> BIOS-e820:
000000000009ec00 - 00000000000a0000 (reserved)
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SB-SYS-5-900001: <4> BIOS-e820:
00000000000e0800 - 0000000000100000 (reserved)
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SB-SYS-5-900001: <4> BIOS-e820:
0000000000100000 - 00000000000f00000 (usable)
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SB-SYS-5-900001: <4> BIOS-e820:
00000000000f00000 - 0000000001000000 (reserved)
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SB-SYS-5-900001: <4> BIOS-e820:
0000000001000000 - 0000000010000000 (usable)
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SB-SYS-5-900001: <4> BIOS-e820:
00000000fff00000 - 0000000100000000 (reserved)
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SB-SYS-5-900001: <6>setup.c: reserved
bootmem for INITRD_START = 0x6000000, INITRD_SIZE = 117
09348
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SB-SYS-5-900001: <4>On node 0 totalpages:
65536
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SB-SYS-5-900001: <4>zone(0): 4096 pages.
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SB-SYS-5-900001: <4>zone(1): 61440 pages.
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SB-SYS-5-900001: <4>zone(2): 0 pages.
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SB-SYS-5-900001: <4>Local APIC disabled
by BIOS -- reenabling.
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SB-SYS-5-900001: <4>Found and enabled
local APIC!
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SB-SYS-5-900001: <4>Kernel command line:
root=/dev/ram ramdisk_size=100000 ramdisk_start=0x60
00000 console=ttyS0,9600n8
Jan 10 22:02:56 NO-HOSTNAME syslog_bootup_msgs: %SB-SYS-5-900001: <6>Initializing CPU# 0
<output truncated>
```

Related Commands

Command	Description
cpfile	Copies a file.
dir	Displays the files in a directory in a long-list format.
lls	Displays a long list of directory names.
ls	Lists the files and subdirectories in a directory.
mkfile	Makes a file (for testing).

type-tail

To view a specified number of lines of the end of a log file or to view the end of the file continuously as new lines are added to the file, use the **type-tail** command in EXEC configuration mode.

type-tail *filename* [*line* | **follow**]

Syntax Description

filename	File to be examined.
line	(Optional) The number of lines from the end of the file to be displayed (the range is 1 to 65535).
follow	(Optional) Displays the end of the file continuously as new lines are added to the file.

Defaults

The default is ten lines shown.

Command Modes

EXEC configuration mode.

Usage Guidelines

This command allows you to monitor a log file by letting you view the end of the file. You can specify the number of lines at the end of the file that you want to view, or you can follow the last line of the file as it continues to log new information. To stop the last line from continuously scrolling, press **Ctrl-C**.

Examples

The following example shows the list of log files in the /local1 directory:

```
stream-ServiceBroker# ls /local1
WS441
Websense
WebsenseEnterprise
Websense_config_backup
WsInstallLog
badfile.txt
codecoverage
core.stunnel.5.3.0.b100.cnbuild.5381
core_dir
crash
crka.log
cse_live
cse_vod
dbdowngrade.log
dbupgrade.log
downgrade
errorlog
http_authmod.unstrip
index.html
logs
lost+found
netscape-401-proxy
netscape-401-proxy1
netscape-dump
newwebsense
oldWsInstallLog
preload_dir
```

```
proxy-basic1
proxy1
proxy2
proxy3
proxy4
proxy5
proxy6
proxy7
proxy8
proxyreply
proxyreply-407
real_vod
ruby.bin.cli_fix
ruby.bin.no_ws_fix
ruby.bin.ws_edir_fix
sa
service_logs
smartfilter
smfnaveen
superwebsense
syslog.txt
syslog.txt.1
syslog.txt.2
temp
two.txt
url.txt
urllist.txt
var
vpd.properties
websense.pre-200
webtarball44
webtarball520
wmt_vod
ws_upgrade.log
```

The following example shows how to display the last ten lines of the syslog.txt file. In this example, the number of lines to display is not specified; however, ten lines is the default.

```
stream-ServiceBroker# type-tail /local1/syslog.txt
Oct  8 21:49:15 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:15 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:15 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:17 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:17 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:17 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:19 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:19 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:19 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:21 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
```

The following example shows how to display the last 20 lines of the syslog.text file:

```
stream-ServiceBroker# type-tail /local1/syslog.txt 20
Oct  8 21:49:11 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:11 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:13 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
```

```
return 0, ready = 0
Oct  8 21:49:13 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:13 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:15 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:15 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:15 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:17 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:17 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:17 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:19 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:19 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:19 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:21 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:21 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:21 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:23 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:23 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:23 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
```

The following example follows the file as it grows:

```
stream-ServiceBroker# type-tail /local1/syslog.txt ?
  <1-65535>  The numbers of lines from end
  follow     Follow the file as it grows
  <cr>
stream-ServiceBroker# type-tail /local1/syslog.txt follow
Oct  8 21:49:39 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:41 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:41 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:41 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:43 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:43 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:43 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:45 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:45 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:45 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:47 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:47 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:47 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
Oct  8 21:49:49 stream-ce syslog:(26830)TRCE:input_serv.c:83-> select_with
return 0, ready = 0
Oct  8 21:49:49 stream-ce syslog:(26832)TRCE:al_master.c:246-> select_with
return 0, ready = 0
Oct  8 21:49:49 stream-ce syslog:(26832)TRCE:in_mms.c:1747-> tv = NULL
```

undebug

To disable debugging functions, use the **undebug** EXEC command.

undebug *option*

Syntax Description

This command has no arguments or keywords.

Defaults

None

Command Modes

EXEC configuration mode.

Usage Guidelines

We recommend that you use the **debug** and **undebug** commands only at the direction of Cisco TAC. See the "debug" section on page 2-47 for more information about debug functions.

Valid values for *command* are as follows:

Command	Description	Device Mode
access-lists	Access Control List debug commands.	SB
all	Disables all debugging.	All
authentication	Authentication debug commands.	All
cli	CLI debug commands.	SB
cms	Debugs the CMS[1].	All
dataserver	Dataserver debug commands.	All
dfs	DFS[2] debug commands.	SB
dhcp	DHCP[3] debug commands.	All
emdb	Embedded database debug commands.	All
logging	LOG debug commands.	All
malloc	Memory allocation debug commands.	All
ntp	NTP[4] debug commands.	All
rpc	Interbox RPC[5] debug commands.	All
service-broker	Service Broker debug commands.	SB
service-monitor	Service Monitor debug commands.	All
snmp	SNMP debug commands.	All
standby	Standby debug commands.	SB
stats	Statistics debug commands.	VDSM
translog	Transaction Log debug commands.	SB

1. CMS = centralized management system
2. DFS = distributed filesystem
3. DHCP = Dynamic Host Configuration Protocol

4. NTP = network time protocol
5. RPC = remote procedure call

Related Commands

Command	Description
debug	Configures the debugging options.
show debugging	Displays the state of each debugging option.

url-signature

The VDS-SB uses a combination of key owners, key ID numbers, and a word value to generate URL signature keys. To configure the url signature, use the **url-signature** command in Global configuration mode.

url-signature key-id-owner *num* **key-id-number** *id_num* {**key** *keyword* | **public key** *url* [**symmetric key** *word* | **private key** *url*]}

no url-signature key-id-owner *num* **key-id-number** *num*

Syntax Description

key-id-owner	Configures the owner ID for this key.
num	Specifies the ID for the owner of this key. The range is from 1 to 8.
key-id-number	Configures the number ID for this key.
id_num	Specifies the ID for the number of this key. The range is from 1 to 8.
key	Configures the encryption key for signing a URL.
keyword	Text of encryption key (maximum of 64 characters, no spaces). **Note** This field accepts only printable ASCII characters (alphabetic, numeric, and others) and does not support a space or the following special characters: pipe (|), question mark (?), double quotes (“), and apostrophe ('). The following special characters are allowed: {}!#$%&()*+,-./;:<=>@\~^[]_.
public-key	Configures the Public Key file location (PEM).
url	The URL from where the Public Key file can be downloaded (maximum of 54 characters).
symmetric-key	(Optional) Configure the Symmetric Key.
word	The Symmetric Key (Must be 16 characters, no spaces).
private-Key	(Optional) Configures the Private Key file location (PEM).
url	The URL from where the Private Key file can be downloaded (maximum of 54 characters).

Command Modes

Global configuration (config) mode.

Usage Guidelines

Service Rules for Directing Requests to a Policy Server

If your network is configured to work with Camiant PCMM-compliant third-party policy servers for servicing requests that require guaranteed bandwidth, you can use the following rule patterns and rule actions to filter the requests and to direct them to the policy server. The rule patterns and rule actions also enable you to generate URL signatures in the response for a valid request for a Windows Media metafile (.asx file extension), Movie Streamer file, or Flash Media Streaming file, and to validate the URL signature on incoming requests to the SB. URL signature key authentication is implemented by using the generate-url-signature and validate-url-signature rule actions that can be applied to specific rule patterns.

Note Movie Streamer and Flash Media Streaming support URL signing. Flash Media Streaming only supports the following actions: allow, block, and validate-url-signature.

The following table lists the rule patterns that support the use-icap-service rule action for directing requests that require guaranteed bandwidth to the third-party policy server:

Rule Patern	Description
url-regex	Filters the request based on any regular expression n the URL.
domain	Filters the request based on the domain name specified.
src-ip	Filters the request based on the IP address of the source.
header-field user-agent	Filters the request based on the user agent specified in the request header.
header-field referer	Filters the request based on the referer in the request header.
header-field request-line	Filters the request based on the request line in the request header.

You can set the use-icap-service rule action for any of the rule patterns above. If the request matches the parameters that you have set for the rule pattern, then the SB redirects the request to the third-party policy server using ICAP services. However, make sure that your network is configured to interoperate with the third-party policy server using ICAP services. You can set up the necessary ICAP configurations from the ICAP Services page. You can also use the rule pattern and rule action to generate URL signatures in the response for a valid request for a Windows Media metafile. You can use the following rule patterns to filter out requests for which you want to generate a URL signature key:

Rule Patern	Description
url-regex	Filters the request based on any regular expression in the URL.
domain	Filters the request based on the domain name specified.

For the rule patterns mentioned above, you can set the following rule actions:

Rule Action	Description
generate-url-signature	Generates the URL signatures in the Windows Media metafile response associated with prepositioned content, based on the SB configuration for the URL signature and this rule action.
validate-url-signature	Validates the URL signature for a request by using the configuration on your SB for the URL signature and allows the request processing to proceed for this request.

Note When configuring service rules, you must configure the same service rules on all SBs participating in a delivery service for the service rules to be fully implemented. The rule action must be common for all client requests because the SB may redirect a client request to any SB in a delivery service depending on threshold conditions.

URL Signing Components

However, because any of these strings in the URL could potentially be edited manually and circumvented by any knowledgeable user, it is important to generate and attach a signature to the URL. This can be achieved by attaching a keyed hash to the URL, using a secret key shared only between the signer (the portal) and the validating component (VDS-SB).

The URL signing script offers three different versions:

- MD5 hash algorithm
- SHA-1 hash algorithm
- SHA-1 hash algorithm with the protocol removed from the beginning of the URL

When a URL is signed for RTSP and a player does a fallback to HTTP for the same URL, the validation fails because the URL signature includes RTSP. If the URL signature does not include the protocol, the fallback URL is validated correctly even though the protocol is HTTP.

If you do not specify a version for the script, MD5 is used and the SIGV string in the script is not added.

At the portal, URLs can be signed for a particular user (client IP address) and expiry time using a URL signing script. The URL signing script example included in this section requires Python 2.3.4 or higher.

Following is an example of the URL signing script using the MD5 security hash algorithm:

```
python vos-ims-urlsign.py http://www.cisco.com/index.html 8.1.0.4 200000 1 2 cisco
```

An example of the resulting signed URL follows:

```
http://www.cisco.com/index.html?IS=0&ET=1241194518&CIP=8.1.0.4&KO=1&KN=2&US=deebacde45bf71
6071c8b2fecaa755b9
```

If you specify Version 1 for the script, SHA-1 is used and the SIGV=1 string is added.

Following is an example of the URL signing script using the SHA-1 security hash algorithm:

```
python vos-ims-urlsign.py http://www.cisco.com/index.html 8.1.0.4 200000 1 2 cisco 1
```

An example of the resulting signed URL follows:

```
http://www.cisco.com/index.html?SIGV=1&IS=0&ET=1241194679&CIP=8.1.0.4&KO=1&KN=2&US=8349348
ffac7987d11203122a98e7e64e410fa18
```

If you specify Version 2 for the script, SHA-1 is used. The protocol from the beginning of the URL is also removed before the signature is generated, and the SIGV=2 string is added. The protocol is RTSP, HTTP, or RTMP. The URL is signed without the protocol, but the final signed URL is printed with the protocol.

Following is an example of the URL signing script using the SHA-1 security hash algorithm with Version 2 specified:

```
python vos-ims-urlsign.py http://www.cisco.com/index.html 8.1.0.4 200000 1 2 cisco 2
```

An example of the resulting signed URL follows:

```
http://www.cisco.com/index.html?SIGV=2&IS=0&ET=1241194783&CIP=8.1.0.4&KO=1&KN=2&US=68b5f5e
d97d1255a0ec42a42a4f779e794df679c
```

Note The URL signature key field accepts only printable ASCII characters (alphabetic, numeric, and others) and does not support a space or the following special characters: pipe (|), question mark (?), double quotes (“), and apostrophe ('). The following special characters are allowed: {}!#$%&()*+,-./;:<=>@\~^[]_

Examples

Following is an example of generating and encrypting the public key and private key using the **url-signature** command:

```
ServiceBroker(config)# url-signature key-id-owner 1 key-id-number 10 public-key
http://1.1.1.1/ec_pub_key private-key http://1.1.1.1/ec_pub_key symmetric-key
```

Following is an example of the URL signing script using the MD5 security hash algorithm:

```
python vos-ims-urlsign.py http://www.cisco.com/index.html 8.1.0.4 200000 1 2 cisco
```

An example of the resulting signed URL follows:

```
http://www.cisco.com/index.html?IS=0&ET=1241194518&CIP=8.1.0.4&KO=1&KN=2&US=deebacde45bf71
6071c8b2fecaa755b9
```

If you specify Version 1 for the script, SHA-1 is used and the SIGV=1 string is added.

Following is an example of the URL signing script using the SHA-1 security hash algorithm:

```
python vos-ims-urlsign.py http://www.cisco.com/index.html 8.1.0.4 200000 1 2 cisco 1
```

An example of the resulting signed URL follows:

```
http://www.cisco.com/index.html?SIGV=1&IS=0&ET=1241194679&CIP=8.1.0.4&KO=1&KN=2&US=8349348
ffac7987d11203122a98e7e64e410fa18
```

If you specify Version 2 for the script, SHA-1 is used. The protocol from the beginning of the URL is also removed before the signature is generated, and the SIGV=2 string is added. The protocol is RTSP, HTTP, or RTMP. The URL is signed without the protocol, but the final signed URL is printed with the protocol.

Following is an example of the URL signing script using the SHA-1 security hash algorithm with Version 2 specified:

```
python vos-ims-urlsign.py http://www.cisco.com/index.html 8.1.0.4 200000 1 2 cisco 2
```

An example of the resulting signed URL follows:

```
http://www.cisco.com/index.html?SIGV=2&IS=0&ET=1241194783&CIP=8.1.0.4&KO=1&KN=2&US=68b5f5e
d97d1255a0ec42a42a4f779e794df679c
```

username

To establish username authentication, use the **username** command in Global configuration mode.

username *name* {**cifs-password** | **samba-password**} {**0** *plain_word* | **1** *lan_crypto nt_crypto* | *clear_text*} | **password** {**0** *plain_word* | **1** *crypto_word* | *clear_text*} [**uid** *u_id*] | **privilege** {**0** | **15**}}

no username *name*

Syntax Description

name	Username.
cifs-password	Sets the Windows user password.
samba-password	Deprecated, same as **cifs-password**.
0	Specifies a clear-text password. This is the default password setting.
plain_word	Clear-text user password.
1	Specifies a type 1 encrypted password.
lan_crypto	Encrypted password for LAN Manager networks.
nt_crypto	Encrypted password for Windows NT networks.
clear_text	Unencrypted (clear-text) password for Windows NT networks.
password	Sets the user password.
crypto_word	Encrypted user password.
uid	Sets the user ID for a clear-text password or an encrypted password.
u_id	Encrypted password user ID (the range is 2001 to 65535).
privilege	Sets the user privilege level.
0	Sets the user privilege level for a normal user.
15	Sets the user privilege level for a superuser.

Defaults

The **password** value is set to 0 (cleartext) by default.

Default administrator account:

- **Uid**: 0
- **Username**: admin
- **Password**: default
- **Privilege**: superuser (15)

Command Modes

Global configuration (config) mode.

Usage Guidelines

The **username** command changes the password and privilege level for existing user accounts.

Note The following characters are not permitted in a username or password: ? . / ; [ ] { } “ @ = |.

User Authentication

User access is controlled at the authentication level. For every HTTP or HTTPS request that applies to the administrative interface, including every CLI and API request that arrives at the VDS-SB network devices, the authentication level has visibility into the supplied username and password. Based on CLI-configured parameters, a decision is then made to either accept or reject the request. This decision is made either by checking local authentication or by performing a query against a remote Authentication Server. The authentication level is decoupled from the authorization level, and there is no concept of role or domain at the authentication level.

When local CLI authentication is used, all configured users can be displayed by entering the **show running-config** command. Normally, only administrative users need to have username authentication configured.

Note Every VDS-SB network device should have an administrative password that can override the default password.

User Authorization

Domains and roles are applied by the VDSM at the authorization level. Requests must be accepted by the authentication level before they are considered by the authorization level. The authorization level regulates the access to resources based on the VDSM GUI role and domain configuration.

Regardless of the authentication mechanism, all user authorization configuration is visible in the GUI.

Examples

When you first connect an VDS-SB device to an VDS-SB network, you should immediately change the password for the username *admin*, which has the password *default*, and the privilege-level superuser.

The following example shows how to change the password:

```
ServiceBroker(config)# username admin password yoursecret
```

The following example shows how passwords and privilege levels are reconfigured:

```
ServiceBroker# show user username abeddoe
Uid                 : 2003
Username            : abeddoe
Password            : ghQ.GyGhP96K6
Privilege           : normal user
ServiceBroker# show user username bwhidney
Uid                 : 2002
Username            : bwhidney
Password            : bhlohlbIwAMOk
Privilege           : normal user
ServiceBroker(config)# username bwhidney password 1 victoria
ServiceBroker(config)# username abeddoe privilege 15
User's privilege changed to super user (=15)
ServiceBroker# show user username abeddoe
Uid                 : 2003
Username            : abeddoe
Password            : ghQ.GyGhP96K6
Privilege           : super user

ServiceBroker# show user username bwhidney
Uid                 : 2002
Username            : bwhidney
Password            : mhYWYw.7P1Ld6
Privilege           : normal user
```

Related Commands

Command	Description
show user	Displays the user identification number and username information for a particular user.
show users	Displays the specified users.

vdsm

To configure the VDS-SB IP address to be used for the SBs, or to configure the role and GUI parameters on a VDSM device, use the **VDSM** command in Global configuration mode. To negate these actions, use the **no** form of this command.

vdsm {**ip** {*hostname* | *ip-address* | **role** {**primary** | **standby**} | **ui port** *port-num*}}

no vdsm {**ip** | **role** {**primary** | **standby**} | **ui port**}

Syntax Description

ip	Configures the VDSM hostname or IP address.
hostname	Hostname of the VDSM.
ip-address	IP address of the VDSM.
role	Configures the VDSM role to either primary or standby (available only from the VDSM CLI).
primary	Configures the VDSM to be the primary VDSM.
standby	Configures the VDSM to be the standby VDSM.
ui	Configures the VDSM GUI port address (available only from the VDSM CLI).
port	Configures the VDSM GUI port.
port-num	Port number. The range is from 1 to 65535.

Defaults

None

Command Modes

Global configuration (config) mode.

Usage Guidelines

You can use the **VDSM ui port** command to change the VDSM GUI port from the standard number 8443 as follows:

```
VDSM(config)# VDSM ui port 35535
```

Note The **role** and **ui** options are only available on VDSM devices. Changing the VDSM GUI port number automatically restarts the Centralized Management System (CMS) service if this has been enabled.

The **VDSM ip** command associates the device with the VDSM so that the device can be approved as a part of the network.

After the device is configured with the VDSM IP address, it presents a self-signed security certificate and other essential information, such as its IP address or hostname, disk space allocation, and so forth, to the VDSM.

Configuring Devices Inside a NAT

In an VDS-SB network, there are two methods for a device registered with the VDSM (SBs, or standby VDSM) to obtain configuration information from the primary VDSM. The primary method is for the device to periodically poll the primary VDSM on port 443 to request a configuration update. You cannot configure this port number. The backup method is when the VDSM pushes configuration updates to a registered device as soon as possible by issuing a notification to the registered device on port 443. This method allows changes to take effect in a timelier manner. You cannot configure this port number even when the backup method is being used. VDS-SB networks do not work reliably if devices registered with the VDSM are unable to poll the VDSM for configuration updates. Similarly, when a receiver SB requests content and content metadata from a forwarder SB, it contacts the forwarder SB on port 443.

All the above methods become complex in the presence of Network Address Translation (NAT) firewalls. When a device (SBs at the edge of the network, SBs, and primary or standby VDSMs) is inside a NAT firewall, those devices that are inside the same NAT use one IP address (the inside local IP address) to access the device and those devices that are outside the NAT use a different IP address (the inside global IP address) to access the device. A centrally managed device advertises only its inside local IP address to the VDSM. All other devices inside the NAT use the inside local IP address to contact the centrally managed device that resides inside the NAT. A device that is not inside the same NAT as the centrally managed device is not able to contact it without special configuration.

If the primary VDSM is inside a NAT, you can allow a device outside the NAT to poll it for getUpdate requests by configuring a *static translation* (inside global IP address) for the VDSM's inside local IP address on its NAT, and using this address, rather than the VDSM's inside local IP address, in the **VDSM ip** *ip-address* command when you register the device to the VDSM. If the SB is inside a NAT and the VDSM is outside the NAT, you can allow the SB to poll for getUpdate requests by configuring a static translation (inside global IP address) for the SB or SIR's inside local address on its NAT and specifying this address in the Use IP Address field under the NAT Configuration heading in the Device Activation window.

Note Static translation establishes a one-to-one mapping between your inside local address and an inside global address. Static translation is useful when a host on the inside must be accessible by a fixed address from the outside.

Standby VDSMs

The Cisco VDS Service Broker software implements a standby VDSM. This process allows you to maintain a copy of the VDS-SB network configuration. If the primary VDSM fails, the standby can be used to replace the primary.

For interoperability, when a standby VDSM is used, it must be at the same software version as the primary VDSM to maintain the full VDSM configuration. Otherwise, the standby VDSM detects this status and does not process any configuration updates that it receives from the primary VDSM until the problem is corrected.

Note We recommend that you upgrade your standby VDSM first and then upgrade your primary VDSM. We also recommend that you create a database backup on your primary VDSM and copy the database backup file to a safe place before you upgrade the software.

Switching a VDSM from Warm Standby to Primary

If your primary VDSM becomes inoperable for some reason, you can manually reconfigure one of your warm standby VDSMs to be the primary VDSM. Configure the new role by using the Global configuration **VDSM role primary** command as follows:

```
ServiceBroker# configure
ServiceBroker(config)# VDSM role primary
```

This command changes the role from standby to primary and restarts the management service to recognize the change.

Note Check the status of recent updates from the primary VDSM. Use the **show cms info** command in EXEC configuration mode and check the time of the last update. To be current, the update time should be between 1 and 5 minutes old. You are verifying that the standby VDSM has fully replicated the primary VDSM configuration. If the update time is not current, check whether there is a connectivity problem or if the primary VDSM is down. Fix the problem, if necessary, and wait until the configuration has replicated as indicated by the time of the last update. Make sure that both VDSMs have the same Coordinated Universal Time (UTC) configured.

If you switch a warm standby VDSM to primary while your primary VDSM is still online and active, both VDSMs detect each other, automatically shut themselves down, and disable management services. The VDSMs are switched to halted, which is automatically saved in flash memory.

Examples

The following example shows how to configure an IP address and a primary role for a VDSM:

```
VDSM(config)# VDSM ip 10.1.1.1
VDSM(config)# VDSM role primary
```

The following example shows how to configure a new GUI port to access the VDSM GUI:

```
VDSM(config)# VDSM ui port 8550
```

The following example shows how to configure the VDSM as the standby VDSM:

```
VDSM(config)# VDSM role standby
Switching VDSM to standby will cause all configuration settings made on this VDSM
 to be lost.
Please confirm you want to continue [ no ] ?yes
Restarting CMS services
```

The following example shows how to configure the standby VDSM with the IP address of the primary VDSM by using the **VDSM ip** *ip-address* command. This command associates the device with the primary VDSM so that it can be approved as a part of the network.

```
VDSM# VDSM ip 10.1.1.1
```

whoami

To display the username of the current user, use the **whoami** command in EXEC configuration mode.

whoami

Syntax Description

This command has no arguments or keywords.

Defaults

None

Command Modes

EXEC configuration mode.

Usage Guidelines

Use this command to display the username of the current user.

Examples

The following example shows how to display the username of the user who has logged in to the SB:

```
ServiceBroker# whoami
admin
```

Related Commands

Command	Description
pwd	Displays the present working directory.

write

To save startup configurations, use the **write** command in EXEC configuration mode.

write [**erase | memory | terminal**]

Syntax Description

erase	(Optional) Erases the startup configuration from NVRAM.
memory	(Optional) Writes the configuration to NVRAM. This setting is the default.
terminal	(Optional) Writes the configuration to a terminal session.

Defaults

The configuration is written to NVRAM by default.

Command Modes

EXEC configuration mode.

Usage Guidelines

Use this command to either save running configurations to NVRAM or erase memory configurations. Following a **write erase** command, no configuration is held in memory, and a prompt for configuration specifics occurs after you reboot the SB.

Use the **write terminal** command to display the current running configuration in the terminal session window. The equivalent command is **show running-config**.

The **write memory** command saves modified Websense configuration files (the eimserver.ini, config.xml, and websense.ini files and the Blockpages directory) across disk reconfiguration and VDS-SB software release upgrades.

Note Clicking the **Save Changes** button from the Websense Enterprise Manager window does not save the Websense configuration modifications across device reboots. You need to use the **write memory** command to save the Websense configuration changes across reboots.

Execute the **write memory** command to save the most recent configuration modifications, including websense.ini file modifications and Websense URL filtering configuration changes. The **write memory** command enables the changes made from the external Websense Manager GUI to be saved across disk reconfiguration and upgrades (which might erase disk content).

The Websense configurations from the last use of the **write memory** command are retained under the following situations:

- If the **write memory** command is not used before a reboot but after a disk reconfiguration or an VDS-SB software upgrade that erases disk content.
- If you are using the CLI and did not answer **Yes** when asked if you wanted to save the configurations at the reload prompt.

However, if the **write memory** command has never been used before, then default configurations are applied when the content in the /local1/WebsenseEnterprise/EIM directory on the SB is erased.

Examples

The following command saves the running configuration to NVRAM:

```
ServiceBroker# write memory
```

Related Commands

Command	Description
copy	Copies the configuration or image files to and from the CD-ROM, flash memory, disk, or remote hosts.
show running-config	Displays the current operating configuration.