H3C 18-Session Management User Manual

18-Session Management
User Manual

Contents

Managing sessions

About session management
Session management is a common module, providing basic services for and attack detection and protection to implement their session-based services.
Session management defines packet exchanges at the transport layer as sessions. It updates session states and ages out sessions according to data flows from the initiators or responders. Session management allows multiple features to process the same service packet.
Session management operation
Session management tracks the session status by inspecting the transport layer protocol information.
It performs unified status maintenance and management of all connections based on session tables and related tables.

When a connection request passes through the device from a client to a server, the device creates a session entry. The entry can contain the request and response information, such as:

- Source IP address and port number.
- Destination IP address and port number.
- Transport layer protocol.
- Application layer protocol.
- Protocol state of the session.

A multichannel protocol requires that the client and the server negotiate a new connection based on an existing connection to implement an application. Session management enables the device to create a relation entry for each connection during the negotiation phase. The entry is used to associate the connection with the application. Relation entries will be removed after the associated connections are established.
If the destination IP address of a packet is a multicast IP address, the packet will be forwarded out of multiple ports. When a multicast connection request is received on an inbound interface, the device performs the following operations:

- Creates a multicast session entry on the inbound interface.
- Creates a corresponding multicast session entry for each outbound interface.

Unless otherwise stated, “session entry” in this chapter refers to both unicast and multicast session entries.
In actual applications, session management works with security modules to dynamically determine whether a packet can pass the firewall and enter the internal network according to connection status, thus preventing intrusion.
Session management only tracks connection status. It does not block potential attack packets.

Session management functions

Session management enables the device to provide the following functions:

- Creates sessions for protocol packets, updates session states, and sets the aging time for sessions in different protocol states.
- Sets the aging time for sessions based on application layer protocols.
- Supports ICMP/ICMPv6 error packet mapping, enabling the device to search for original sessions according to the payloads in the ICMP/ICMPv6 error packets.

  Because error packets are generated due to host errors, the mapping can help speed up the aging of the original sessions.
- Supports persistent sessions, which are kept alive for a long period of time.
- Supports session management for the control channels and dynamic data channels of application layer protocols, for example, FTP.
- Supports real-time synchronization for sessions and for dynamic entries of session-based services, such as.

Restrictions and guidelines: Session management configuration

For a TCP session in ESTABLISHED state, the priority order of the associated aging time is as follows:

- Aging time for persistent sessions.
- Aging time for sessions of application layer protocols.

- Aging time for sessions in different protocol states.

If the device has excessive sessions, do not set the aging time shorter than the default for a certain protocol state or an application layer protocol. Short aging time settings can make the device slow in response.

Setting the session aging time for different protocol states

About this task
If a session in a certain protocol state has no packet hit before the aging time expires, the device automatically removes the session.

Procedure

1. Enter system view.
   system-view
2. Set the session aging time for different protocol states.
   session aging-time state { **fin** | **icmp-reply** | **icmp-request** | **rawip-open** | **rawip-ready** | **syn** | **tcp-close** | **tcp-est** | **tcp-time-wait** | **udp-open** | **udp-ready** } ime-value
   The default aging time for sessions in different protocol states is as follows:
   o FIN_WAIT: 30 seconds.
   o ICMP-REPLY: 30 seconds.
   o ICMP-REQUEST: 60 seconds.
   o RAWIP-OPEN: 30 seconds.
   o RAWIP-READY: 60 seconds.
   o TCP SYN-SENT and SYN-RCV: 30 seconds.
   o TCP-CLOSE: 2 seconds.
   o TCP ESTABLISHED: 3600 seconds.
   o TCP-TIME-WAIT: 2 seconds.
   o UDP-OPEN: 30 seconds.
   o UDP-READY: 60 seconds.

Setting the session aging time for different application layer protocols or applications

About this task
The aging time for sessions of different application layer protocols or applications is valid for TCP sessions in the ESTABLISHED state or UDP sessions in a READY state. For sessions used by other application layer protocols, the aging time for sessions in different protocol states applies.
Procedure

1. Enter system view.
   system-view
2. Set the session aging time for different application layer protocols.
   session aging-time application application-name time-value
   By default, the aging time is 1200 seconds for sessions of application layer protocols or applications except for the following sessions:
   o BOOTPC sessions: 120 seconds.

- o BOOTPS sessions: 120 seconds.
- o DNS sessions: 1 second.
- o FTP sessions: 3600 seconds.
- o FTP-DATA sessions: 240 seconds.
- o GPRS-DATA sessions: 60 seconds.
- o GPRS-SIG sessions: 60 seconds.
- o GTP-CONTROL sessions: 60 seconds.
- o GTP-USER sessions: 60 seconds.
- o H.225 sessions: 3600 seconds.
- o H.245 sessions: 3600 seconds.
- o HTTPS sessions: 600 seconds.
- o ILS sessions: 3600 seconds.
- o L2TP sessions: 120 seconds.
- o MGCP-CALLAGENT sessions: 60 seconds.
- o MGCP-GATEWAY sessions: 60 seconds.
- o NETBIOS-DGM sessions: 3600 seconds.
- o NETBIOS-NS sessions: 3600 seconds.
- o NETBIOS-SSN sessions: 3600 seconds.
- o NTP sessions: 120 seconds.
- o PPTP sessions: 3600 seconds.
- o QQ sessions: 120 seconds.
- o RAS sessions: 300 seconds.
- o RIP sessions: 120 seconds.
- o RSH sessions: 60 seconds.
- o RTSP session: 3600 seconds.
- o SCCP sessions: 3600 seconds.
- o SIP sessions: 300 seconds.
- o SNMP sessions: 120 seconds.
- o SNMPTRAP sessions: 120 seconds.
- o SQLNET sessions: 600 seconds.
- o STUN sessions: 600 seconds.
- o SYSLOG sessions: 120 seconds.
- o TACACS-DS sessions: 120 seconds.
- o TFTP sessions: 60 seconds.
- o WHO sessions: 120 seconds.
- o XDMCP sessions: 3600 seconds.

Specifying persistent sessions

About this task

This task is only for TCP sessions in an ESTABLISHED state. You can specify TCP sessions that match the permit statements in the specified ACL as persistent sessions, and set longer lifetime or never-age-out persistent sessions.

A persistent session is not removed until one of the following events occurs:

- The session entry ages out.
- The device receives a connection close request from the initiator or responder.
- You manually clear the session entries.

Procedure

1. Enter system view.
   system-view
2. Specify persistent sessions.
   session persistent acl [ipv6] acl-number [**aging-time** time-value]

Enabling session statistics collection software fast-forwarding

About this task
This feature enables the device to collect session-based outbound and inbound packets and bytes. You can display session statistics based on different criteria.

- To display statistics per unicast session, use the **display session table** command.
- To display statistics per unicast packet type, use the **display session statistics** command.
- To display statistics per multicast session, use the **display session table multicast** command.
- To display statistics per multicast packet type, use the **display session statistics multicast** command.

Procedure

1. Enter system view.
   system-view
2. Enable session statistics collection for software fast-forwarding.
   session statistics enable
   By default, session statistics collection is disabled for software fast-forwarding.

Configuring session logging

About this task
Session logs provide information about user access, IP address translation, and network traffic for security auditing. These logs are sent to the log server or the information center.
The device supports time-based or traffic-based logging:

- **Time-based logging**—The device outputs session logs regularly.
- **Traffic-based logging**—The device outputs a session log when the traffic amount of a session reaches a threshold only when the session statistics collection for software fast-forwarding feature is enabled. After outputting a session log, the device resets the traffic counter for the session. The traffic-based thresholds can be byte-based and packet-based. If you set both thresholds, the last configuration takes effect.

If you set both time-based and traffic-based logging, the device outputs a session log when whichever is reached. After outputting a session log, the device resets the traffic counter and restarts the interval for the session.
If you enable session logging but do not enable logging for session creation or deletion, the device does not output a session log when a session entry is created or removed.

Restrictions and guidelines

The session logging feature must work with the flow log or fast log output feature to generate session logs. Session logs can be output in flow log or fast log output format. By default, they are output in flow log format. For information about flow log and fast log output, see Network Management and Monitoring.

Procedure

1. Enter system view.

   system-view
2. (Optional.) Set the threshold for time-based session logging.

   session log time-active time-value

   By default, no threshold is set for time-based session logging.
3. (Optional.) Set a threshold for traffic-based logging.

   session log { bytes-active bytes-value | **packets-active packets-value** }

   By default, no threshold is set for traffic-based logging.
4. (Optional.) Enable logging for session creation.

   session log flow-begin

   By default, logging for session creation is disabled.
5. (Optional.) Enable logging for session deletion.

   session log flow-end

   By default, logging for session deletion is disabled.
6. Enter interface view.

   interface interface-type interface-number
7. Enable session logging.

   session log enable { **ipv4** | **ipv6 } [acl acl-number] { inbound** |**outbound** }

   By default, session logging is disabled.

Display and maintenance commands for session management

IMPORTANT:

The WX1800H series, WX2500H series, and WX3000H series access controllers do not support parameters or commands that are available only in IRF mode.

Execute display commands in any view and reset commands in user view.

Task	Command
Display the aging time for sessions of dif ferent application layer protocols.	display session aging-time application
Display the aging time for sessions in dif ferent protocol states.	display session aging-time state
Display relation table entries.	In standalone mode: **display session relation-table { ipv4** | **ipv6** } In IRF mode: **display session relation-table** { **ipv4** | **ipv6** } [slot slot-number]

Task	Command
Display unicast session statistics.	In standalone mode: display session statistics [history-max |summary] In IRF mode: display session statistics [history-max |summary] [slot slot-number]

Display IPv4 unicast session statistics.	In standalone mode: display session statistics ipv4{ destination-ip destination-ip |destination-port destination-port |protocol { dccp | dns | ftp | gtp | h323 | http |icmp | ils | mgcp | nbt | pptp | raw-ip | rsh |rtsp | sccp | sctp | sip | smtp | sqlnet | ssh |tcp | telnet | tftp | udp | udp-lite | xdmcp } |source-ip source-ip | source-port source-port } * In IRF mode: display session statistics ipv4 { destination-ip destination-ip |destination-port destination-port |protocol { dccp | dns | ftp | gtp | h323 | http |icmp | ils | mgcp | nbt | pptp | raw-ip | rsh |rtsp | sccp | sctp | sip | smtp | sqlnet | ssh |tcp | telnet | tftp | udp | udp-lite | xdmcp } |source-ip source-ip | source-portsource-port } * [slot slot-number]
Display IPv6 unicast session statistics.	In standalone mode: display session statistics ipv6 { destination-ip destination-ip |destination-port destination-port |protocol { dccp | dns | ftp | gtp | h323 | http |icmpv6 | ils | mgcp | nbt | pptp | raw-ip | rsh |rtsp | sccp | sctp | sip | smtp | sqlnet | ssh | tcp | telnet | tftp | udp | udp-lite | xdmcp } |source-ip source-ip | source-port source-port } * In IRF mode: display session statistics ipv6 { destination-ip destination-ip |destination-port estination-port |protocol { dccp | dns | ftp | gtp | h323 | http |icmpv6 | ils | mgcp | nbt | pptp | raw-ip | rsh |rtsp | sccp | sctp | sip | smtp | sqlnet | ssh |tcp | telnet | tftp | udp | udp-lite | xdmcp } |source-ip source-ip | source-port source-port } * [slot slot-number]
Display multicast session statistics.	In standalone mode: display session statistics multicast In IRF mode: display session statistics multicast [slot

Task	**Command**

	slot-number]
Display IPv4 unicast session table entries.	In standalone mode: display session table ipv4 [[responder]{ application application-name |destination-ip start-destination-ip[end-destination-ip]	destination-portdestination-port	protocol { dccp	icmp |raw-ip	sctp	tcp	udp	udp-lite } |source-ip start-source-ip [end-source-ip]| source-port source-port	state { dccp-closereq	dccp-closing	dccp-open| dccp-partopen	dccp-request |dccp-respond	dccp-timewait	icmp-reply |icmp-request	rawip-open	rawip-ready |sctp-closed	sctp-cookie-echoed |sctp-cookie-wait	sctp-established |sctp-shutdown-ack-sent |sctp-shutdown-recd	sctp-shutdown-sent |tcp-close	tcp-lose-wait	tcp-est |tcp-fin-wait	tcp-last-ack	tcp-syn-recv| tcp-syn-sent	tcp-syn-sent2 |tcp-time-wait	udp-open	udp-ready |udplite-open	udplite-ready } } *][verbose] In IRF mode: display session table ipv4 [slotslot-number] [[responder] { applicationapplication-name	destination-ip start-destination-ip [end-destination-ip]| destination-port destination-port |protocol { dccp	icmp	raw-ip	sctp	tcp |udp	udp-lite }	source-ip start-source-ip[end-source-ip]	source-port source-port| state { dccp-closereq	dccp-closing |dccp-open	dccp-partopen	dccp-request |dccp-respond	dccp-timewait	icmp-reply |icmp-request	rawip-open	rawip-ready |sctp-closed	sctp-cookie-echoed |sctp-cookie-wait	sctp-established |sctp-shutdown-ack-sent |sctp-shutdown-recd	sctp-shutdown-sent	tcp-close	tcp-close-wait	tcp-est |tcp-fin-wait	tcp-last-ack	tcp-syn-recv| tcp-syn-sent	tcp-syn-sent2 |tcp-time-wait	udp-open	udp-ready |udplite-open	udplite-ready } } *][verbose]
Display IPv6 unicast session table entries.	In standalone mode: display session table ipv6 [[responder] { application application-name	destination-ip start-destination-ip[end-destination-ip]	destination-portdestination-port	protocol { dccp	icmpv6	

Task	Command
	raw-ip | sctp | tcp | udp | udp-lite } |source-ip start-source-ip [end-source-ip]| source-port source-port | state{ dccp-closereq | dccp-closing | dccp-open| dccp-partopen | dccp-request |dccp-respond | dccp-timewait | icmpv6-reply| icmpv6-request | rawip-open | rawip-ready| sctp-closed | sctp-cookie-echoed |sctp-cookie-wait | sctp-established |sctp-shutdown-ack-sent |sctp-shutdown-recd | sctp-hutdown-sent |tcp-close | tcp-close-wait | tcp-est |tcp-fin-wait | tcp-last-ack | tcp-syn-recv| tcp-syn-sent | tcp-syn-sent2 |tcp-time-wait | udp-open | udp-ready |udplite-open | udplite-ready } } *][verbose] In IRF mode: display session table ipv6 [slotslot-number] [[responder] { applicationapplication-name | destination-ipstart-destination-ip [end-destination-ip]| destination-port destination-port |protocol { dccp | icmpv6 | raw-ip | sctp | tcp |udp | udp-lite } | source-ip start-source-ip[end-source-ip] | source-port source-port| state { dccp-closereq | dccp-closing |dccp-open | dccp-partopen | dccp-request |dccp-respond | dccp-timewait | icmpv6-reply| icmpv6-request | rawip-open | rawip-ready| sctp-closed | sctp-cookie-echoed |sctp-cookie-wait | sctp-established |sctp-shutdown-ack-sent |sctp-shutdown-recd | sctp-shutdown-sent |tcp-close | tcp-close-wait | tcp-est |tcp-fin-wait | tcp-last-ack | tcp-syn-recv| tcp-syn-sent | tcp-syn-sent2 |tcp-time-wait | udp-open | udp-ready |udplite-open | udplite-ready } } *][verbose]
Display IPv4 multicast session table entries.	In standalone mode: display session table multicast ipv4 [[responder] { destination-ipstart-destination-ip [end-destination-ip]| destination-port destination-port |protocol { dccp | icmp | raw-ip | sctp | tcp |udp | udp-lite } | source-ip start-source-ip[end-source-ip] | source-portsource-port } *] [verbose] In IRF mode: display session table multicast ipv4 [slot slot-number] [[responder]{ destination-ip start-destination-ip

Task	Command
	[end-destination-ip] | destination-port destination-port | protocol { dccp | icmp |raw-ip | sctp | tcp | udp | udp-lite } |source-ip start-source-ip [end-source-ip]| source-port source-port } *] [verbose]
Display IPv6 multicast session table entries.	In standalone mode: display session table multicast ipv6[[responder] { destination-ip start-destination-ip [end-destination-ip] | destination-port destination-port | protocol { dccp | icmpv6 | raw-ip | sctp | tcp |udp | udp-lite } | source-ip start-source-ip[end-source-ip] | source-port source-port } *] [verbose] In IRF mode: display session table multicast ipv6 [slot slot-number] [[responder] { destination-ip start-destination-ip [end-destination-ip] | destination-port destination-port | protocol { dccp | icmpv6 |raw-ip | sctp | tcp | udp | udp-lite } |source-ip start-source-ip [end-source-ip]| source-port source-port } *] [verbose]
Clear relation table entries.	In standalone mode: reset session relation-table [ipv4 | ipv6] In IRF mode: reset session relation-table [ipv4 | ipv6] [slot slot-number]
Clear unicast session statistics.	In standalone mode: reset session statistics In IRF mode: reset session statistics [slot slot-number]
Clear multicast session table entries.	In standalone mode: reset session statistics multicast In IRF mode: reset session statistics multicast [slot slot-number]

Clear IP unicast session table entries.	In standalone mode: reset session table In IRF mode: reset session table [slot slot-number]
Clear IPv4 unicast session table entries.	In standalone mode: reset session table ipv4 [source-ip source-ip] [destination-ip destination-ip] [protocol { dccp | icmp | raw-ip | sctp | tcp | udp | udp-lite }]

Task	**Command**
	[source-port source-port] [destination-port destination-port] In IRF mode: reset session table ipv4 [slot slot-number][source-ip source-ip] [destination-ip destination-ip] [protocol { dccp | icmp | raw-ip | sctp | tcp | udp | udp-lite }] [source-port source-port][destination-port destination-port]
Clear IPv6 unicast session table entries.	In standalone mode: reset session table ipv6 [source-ip source-ip] [destination-ip destination-ip] [rotocol { dccp | icmpv6 | raw-ip | sctp | tcp | udp | udp-lite }] [source-port source-port] [destination-port destination-port] In IRF mode: reset session table ipv6 [slot slot-number] [source-ip source-ip] [destination-ip destination-ip] [protocol { dccp | icmpv6 | raw-ip | sctp | tcp | udp | udp-lite }][source-port source-port] [destination-port destination-port]

Clear IP multicast session table entries.	In standalone mode: reset session table multicast I n IRF mode: reset session table multicast [slot slot-number]
Clear IPv4 multicast session table entries.	In standalone mode: reset session table multicast ipv4 [source-ip source-ip] [destination-ip destination-ip] [protocol { dccp | icmp | raw-ip | sctp | tcp | udp | udp-lite }] [source-port source-port] [destination-port destination-port] In IRF mode: reset session table multicast ipv4 [slot slot-number] [source-ip source-ip] [destination-ip destination-ip] [protocol { dccp | icmp | raw-ip | sctp | tcp | udp | udp-lite }] [source-port source-port] [destination-port destination-port]
Clear IPv6 multicast session table entries.	In standalone mode: reset session table multicast ipv6 [source-ip source-ip] [destination-ip destination-ip] [protocol { dccp | icmpv6 | raw-ip | sctp | tcp | udp | udp-lite }] [source-port source-port] [destination-port destination-port] In IRF mode: reset session table multicast ipv6 [slot slot-number] [source-ip source-ip] [destination-ip destination-ip] [protocol { dccp | icmpv6 | raw-ip | sctp | tcp | udp | udp-lite }] [source-port source-port] [destination-port destination-port]

Documents / Resources

	H3C 18-Session Management [pdf] User Manual 18-Session Management