NAME

audisp-remote.conf − the audisp-remote configuration file

DESCRIPTION

audisp-remote.conf is the file that controls the configuration of the audit remote logging subsystem. The options that are available are as follows:

remote_server
This is a one word character string that is the remote server hostname or address that this plugin will send log information to. This can be the numeric address or a resolvable hostname.

port This option is an unsigned integer that indicates what port to connect to on the remote machine.

local_port
This option is an unsigned integer that indicates what local port to connect from on the local machine. If unspecified (the default) or set to the word *any* then any available unprivileged port is used. This is a security mechanism to prevent untrusted user space apps from injecting events into the audit daemon. You should set it to an unused port < 1024 to ensure that only privileged users can bind to that port. Then also set the tcp_client_ports in the aggregating auditd.conf file to match the ports that clients are sending from.

transport
This parameter tells the remote logging app how to send events to the remote system. The valid options are*TCP*, and *KRB5*. If set to*TCP*, the remote logging app will just mak e a normal clear text connection to the remote system. If its set to *KRB5*, then Kerberos 5 will be used for authentication and encryption. The default value is TCP.

mode This parameter tells the remote logging app what strategy to use getting records to the remote system. Valid values are *immediate*, and *forwar d* . If set to *immediate*, the remote logging app will attempt to send events immediately after getting them. *forwar d* means that it will store the events to disk and then attempt to send the records. If the connection cannot be made, it will queue records until it can connect to the remote system. The depth of the queue is controlled by the *queue_depth* option.

queue_file
Path of a file used for the event queue if *mode* is set to *forward*. The default is **/var/spool/audit/remote.log**.

queue_depth
This option is an unsigned integer that determines how many records can be buffered to disk or in memory before considering it to be a failure sending. This parameter affects the *forwar d* mode of the *mode* option and internal queueing for temporary network outages. The default depth is 2048.

format This parameter tells the remote logging app what data format will be used for the messages sent over the network. The default is *managed* which adds some overhead to ensure each message is properly handled on the remote end, and to receive status messages from the remote server. If *ascii* is given instead, each message is a simple ASCII text line with no overhead at all. The *ascii* format is a very simplistic protocol. If there are any network problems, it will cause audisp-remote to exit. Auditd may or may not restart it on next event. If something more robust is needed, use the *managed* format. If *mode* is set to *forward*, *format* must be *managed*.

network_retry_time
The time, in seconds, between retries when a network error is detected. Note that this pause applies starting after the second attempt, so as to avoid unneeded delays if a reconnect is sufficient to fix the problem. The default is 1 second.

max_tries_per_record
The maximum number of times an attempt is made to deliver each message. The minimum value is one, as even a completely successful delivery requires at least one try. If too many attempts are made, the network_failure_action action is performed. The default is 3.

max_time_per_record
The maximum amount of time, in seconds, spent attempting to deliver each message. Note that both this and *max_tries_per_record* should be set, as each try may take a long time to time out. The default value is 5 seconds. If too much time is used on a message, the network_failure_action action is performed.

heartbeat_timeout
This parameter determines how often in seconds the client should send a heartbeat event to the remote server. This is used to let both the client and server know that each end is alive and has not terminated in a way that it did not shutdown the connection uncleanly. This value must be coordinated with the server's *tcp_client_max_idle* setting. The default value is 0 which disables sending a heartbeat.

network_failure_action
This parameter tells the system what action to take whenever there is an error detected when sending audit events to the remote system. Valid values are *ignore*, *syslog*, *exec*, *warn_once*, *suspend*, *single*, *halt*, and *stop*. If set to*ignor e*, the remote logging app does nothing. If an event was sent, its dequeued. *Syslo g* means that it will issue a warning to syslog. If an event was sent, its dequeued. This is the default. *exec* /path-to-script will execute the script. You cannot pass parameters to the script. If an event was sent, its dequeued. *warn_once_continue* is like syslog except that only one message is put in syslog until an event is successfully transferred. *warn_once* is lik e warn_once_continue except that the event is not dequeued. *Suspend* will cause the remote logging app to stop sending records to the remote system. The logging app will still be alive. If an event was sent, it is not dequeued. The *single* option will cause the remote logging app to put the computer system in single user mode. If an event was sent, it is not dequeued. The *stop* option will cause the remote logging app to exit, but leave other plugins running. If an event was sent, it is not dequeued. The *halt* option will cause the remote logging app to shutdown the computer system. If an event was sent, it is not dequeued. The default is to stop.

disk_low_action
Likewise, this parameter tells the system what action to take if the remote end signals a disk low error. The default is ignore.

disk_full_action
Likewise, this parameter tells the system what action to take if the remote end signals a disk full error. The default is warn_once.

disk_error_action
Likewise, this parameter tells the system what action to take if the remote end signals a disk error. The default is warn_once.

remote_ending_action
Likewise, this parameter tells the system what action to take if the network connection is lost. This action has one additional option, *reconnect* which tells the remote plugin to attempt to reconnect to the server upon receipt of the next audit record. If an event was being sent when something triggered this action, it is not dequeued. If it is unsuccessful in reconnecting, the audit record could be lost. The default is to reconnect.

generic_error_action
Likewise, this parameter tells the system what action to take if the remote end signals an error we don't recognize. The default is to log it to syslog.

generic_warning_action
Likewise, this parameter tells the system what action to take if the remote end signals a warning we don't recognize. The default is to log it to syslog.

queue_error_action
Likewise, this parameter tells the system what action to take if there is a problem working with a local record queue. The default is stop.

overflow_action
This parameter tells the system what action to take if the internal event queue overflows. Valid values are *ignore*, *syslog*, *suspend*, *single*, and *halt* . If set to *ignore*, the remote logging app does nothing. *Syslog* means that it will issue a warning to syslog. This is the default. *Suspend* will cause the remote logging app to stop sending records to the remote system. The logging app will still be alive. The *single* option will cause the remote logging app to put the computer system in single user mode. The *halt* option will cause the remote logging app to shutdown the computer system.

startup_failure_action
This parameter tells the system what action to take whenever there is an error connecting to the remote system during startup. Typically, this is benign as the plugin's default behavior is to attempt reconnecting until it succeeds. But there may be times when you want to do something different. Valid values are *ignore*, *syslog*, *exec*, *warn_once*, and *warn_once_continue* . If set to *ignore*, the remote logging app does nothing. *Syslog* means that it will issue a warning to syslog. *e xec* /path-to-script will execute the script. You cannot pass parameters to the script. *warn_once* is lik e syslog except that only one message is put in syslog until an event is successfully transferred. *warn_once_continue* is like warn_once except it ignores the problem. This is the default.

enable_krb5
This option is deprecated. Use the *transport* option to enable Kerberos support. If this option follows the transport configuration option, it will override the transport setting. This would be the normal expected behavior for backwards compatibility. If set to *yes*, Kerberos 5 will be used for authentication and encryption. Default is *no*. Note that encryption can only be used with managed connections, not plain ASCII.

krb5_principal
If specified, This is the expected principal for the server. The client and server will use the specified principal to negotiate the encryption. The format for the *krb5_principal* is like somename/hostname, see the auditd.conf man page for details. If not specified, the krb5_client_name and remote_server values are used.

krb5_client_name
This specifies the name portion of the client's own principal. If unspecified, the default is "auditd". The remainder of the principal will consist of the host's fully qualified domain name and the default Kerberos realm, like this: *auditd/host14.example.com@EXAMPLE.COM* (assuming you gave "auditd" as the krb_client_name). Note that the client and server must have the same principal name and realm.

krb5_key_file
Location of the key for this client's principal. Note that the key file must be owned by root and mode 0400. The default is */etc/audisp/audisp-r emote.key*

NOTES

Specifying a local port may make it difficult to restart the audit subsystem due to the previous connection being in a TIME_WAIT state, if you're reconnecting to and from the same hosts and ports as before.

The network failure logic works as follows: The first attempt to deliver normally "just works". If it doesn't, a second attempt is immediately made, perhaps after reconnecting to the server. If the second attempt also fails, *audispd-remote* pauses for the configured time and tries again. It continues to pause and retry until either too many attempts have been made or the allowed time expires. Note that these times govern the maximum amount of time the remote server is allowed in order to reboot, if you want to maintain logging across a reboot.

It is recommended to set a large q_depth in auditd.conf if using this plugin. Also set an even bigger q_depth in audisp-remote.conf. Also set the heartbeat_timeout to something non-zero but coordinate it with the

server so that it’s half the size of the server’s tcp_client_max_idle setting. This is required to get retries in a reasonable time if the network has a problem.

SEE ALSO

audisp-remote(8), **auditd.conf**(5).

AUTHOR

Steve Grubb