NAME

auditd.conf − audit daemon configuration file

DESCRIPTION

The file */etc/audit/auditd.conf* contains configuration information specific to the audit daemon. Each line should contain one configuration keyword, an equal sign, and then followed by appropriate configuration information. All option names and values are case insensitive. The keywords recognized are listed and described below. Each line should be limited to 160 characters or the line will be skipped. You may add comments to the file by starting the line with a '#' character.

log_file This keyword specifies the full path name to the log file where audit records will be stored. It must be a regular file.

log_format
The log format describes how the information should be stored on disk. There are 2 options: raw and nolog. If set to *RAW*, the audit records will be stored in a format exactly as the kernel sends it. If this option is set to *NOLOG* then all audit information is discarded instead of writing to disk. This mode does not affect data sent to the audit event dispatcher.

log_group
This keyword specifies the group that is applied to the log file's permissions. The default is root. The group name can be either numeric or spelled out.

priority_boost
This is a non-negative number that tells the audit daemon how much of a priority boost it should take. The default is 4. No change is 0.

flush Valid values are *none*, *incremental*, *data*, and *sync*. If set to *none*, no special effort is made to flush the audit records to disk. If set to *incremental*, Then the *freq* parameter is used to determine how often an explicit flush to disk is issued. The *data* parameter tells the audit daemon to keep the data portion of the disk file sync'd at all times. The *sync* option tells the audit daemon to keep both the data and meta-data fully sync'd with every write to disk.

freq This is a non-negative number that tells the audit daemon how many records to write before issuing an explicit flush to disk command. This value is only valid when the *flush* keyword is set to *incremental*.

num_logs
This keyword specifies the number of log files to keep if rotate is given as the *max_log_file_action.* If the number is < 2, logs are not rotated. This number must be 99 or less. The default is 0 - which means no rotation. As you increase the number of log files being rotated, you may need to adjust the kernel backlog setting upwards since it takes more time to rotate the files. This is typically done in /etc/audit/audit.rules. If log rotation is configured to occur, the daemon will check for excess logs and remove them in effort to keep disk space available. The excess log check is only done on startup and when a reconfigure results in a space check.

disp_qos
This option controls whether you want blocking/lossless or non-blocking/lossy communication between the audit daemon and the dispatcher. There is a 128k buffer between the audit daemon and dispatcher. This is good enogh for most uses. If lossy is chosen, incoming events going to the dispatcher are discarded when this queue is full. (Events are still written to disk if log_format is not nolog.) Otherwise the auditd daemon will wait for the queue to have an empty spot before logging to disk. The risk is that while the daemon is waiting for network IO, an event is not being recorded to disk. Valid values are: lossy and lossless. Lossy is the default value.

dispatcher
The dispatcher is a program that is started by the audit daemon when it starts up. It will pass a copy of all audit events to that application's stdin. Make sure you trust the application that you add to this line since it runs with root privileges.

name_format

This option controls how computer node names are inserted into the audit event stream. It has the following choices: *none*, *hostname*, *fqd*, *numeric*, and *user*. *None* means that no computer name is inserted into the audit event. *hostname* is the name returned by the gethostname syscall. The *fqd* means that it takes the hostname and resolves it with dns for a fully qualified domain name of that machine. *Numeric* is similar to fqd except it resolves the IP address of the machine. In order to use this option, you might want to test that 'hostname −i' or 'domainname −i' returns a numeric address. Also, this option is not recommended if dhcp is used because you could have different addresses over time for the same machine. *User* is an admin defined string from the name option. The default value is *none*.

name

This is the admin defined string that identifies the machine if *user* is given as the *name_format* option.

max_log_file

This keyword specifies the maximum file size in megabytes. When this limit is reached, it will trigger a configurable action. The value given must be numeric.

max_log_file_action

This parameter tells the system what action to take when the system has detected that the max file size limit has been reached. Valid values are *ignore*, *syslog*, *suspend*, *rotate* and *keep_logs.* If set to *ignore*, the audit daemon does nothing. *syslog* means that it will issue a warning to syslog. *suspend* will cause the audit daemon to stop writing records to the disk. The daemon will still be alive. The *rotate* option will cause the audit daemon to rotate the logs. It should be noted that logs with higher numbers are older than logs with lower numbers. This is the same convention used by the logrotate utility. The *keep_logs* option is similar to rotate except it does not use the num_logs setting. This prevents audit logs from being overwritten. The effect is that logs accumulate and are not deleted − which will trigger the *space_left_action* if the volume fills up. This is best used in combination with an external script used to archive logs on a periodic basis.

action_mail_acct

This option should contain a valid email address or alias. The default address is root. If the email address is not local to the machine, you must make sure you have email properly configured on your machine and network. Also, this option requires that /usr/lib/sendmail exists on the machine.

space_left

This is a numeric value in megabytes that tells the audit daemon when to perform a configurable action because the system is starting to run low on disk space.

space_left_action

This parameter tells the system what action to take when the system has detected that it is starting to get low on disk space. Valid values are *ignore*, *syslog*, *email*, *exec*, *suspend*, *single*, and *halt*. If set to *ignore*, the audit daemon does nothing. *syslog* means that it will issue a warning to syslog. *Email* means that it will send a warning to the email account specified in *action_mail_acct* as well as sending the message to syslog. *exec* /path-to-script will execute the script. You cannot pass parameters to the script. *suspend* will cause the audit daemon to stop writing records to the disk. The daemon will still be alive. The *single* option will cause the audit daemon to put the computer system in single user mode. The *halt* option will cause the audit daemon to shutdown the computer system.

admin_space_left

This is a numeric value in megabytes that tells the audit daemon when to perform a configurable action because the system **is running low** on disk space. This should be considered the last chance to do something before running out of disk space. The numeric value for this parameter should be lower than the number for space_left.

admin_space_left_action

This parameter tells the system what action to take when the system has detected that it **is low on disk space.** Valid values are *ignore, syslog*, *email*, *exec*, *suspend*, *single,* and *halt*. If set to *ignore*, the audit daemon does nothing. *Syslog* means that it will issue a warning to syslog. *Email* means that it will send a warning to the email account specified in *action_mail_acct* as well as sending the message to syslog. *exec* /path-to-script will execute the script. You cannot pass parameters to the script. The script is also responsible for telling the auditd daemon to resume logging once its completed its action. This can be done by adding service auditd resume to the script. *Suspend* will cause the audit daemon to stop writing records to the disk. The daemon will still be alive. The *single* option will cause the audit daemon to put the computer system in single user mode. The *halt* option will cause the audit daemon to shutdown the computer system.

disk_full_action

This parameter tells the system what action to take when the system has detected that the partition to which log files are written has become full. Valid values are *ignore*, *syslog*, *exec*, *suspend*, *single*, and *halt*. If set to *ignore*, the audit daemon will issue a syslog message but no other action is taken. *Syslog* means that it will issue a warning to syslog. *exec* /path-to-script will execute the script. You cannot pass parameters to the script. The script is also responsible for telling the auditd daemon to resume loggin g once its completed its action. This can be done by adding service auditd resume to the script. *Suspend* will cause the audit daemon to stop writing records to the disk. The daemon will still be alive. The *single* option will cause the audit daemon to put the computer system in single user mode. *halt* option will cause the audit daemon to shutdown the computer system.

disk_error_action

This parameter tells the system what action to take whenever there is an error detected when writing audit events to disk or rotating logs. Valid values are *ignore*, *syslog*, *exec*, *suspend*, *single*, and *halt*. If set to *ignore*, the audit daemon will not take any action. *Syslog* means that it will issue no more than 5 consecutive warnings to syslog. *exec* /path-to-script will execute the script. You cannot pass parameters to the script. *Suspend* will cause the audit daemon to stop writing records to the disk. The daemon will still be alive. The *single* option will cause the audit daemon to put the computer system in single user mode. *halt* option will cause the audit daemon to shutdown the computer system.

tcp_listen_port

This is a numeric value in the range 1..65535 which, if specified, causes auditd to listen on the corresponding TCP port for audit records from remote systems. The audit daemon may be linked with tcp_wrappers. You may want to control access with an entry in the hosts.allow and deny files.

tcp_listen_queue

This is a numeric value which indicates how many pending (requested but unaccepted) connections are allowed. The default is 5. Setting this too small may cause connections to be rejected if too many hosts start up at exactly the same time, such as after a power failure.

tcp_max_per_addr

This is a numeric value which indicates how many concurrent connections from one IP address is allowed. The default is 1 and the maximum is 1024. Setting this too large may allow for a Denial of Service attack on the logging server. Also note that the kernel has an internal maximum that will eventually prevent this even if auditd allows it by config. The default should be adequate in most cases unless a custom written recovery script runs to forward unsent events. In this case you would increase the number only large enough to let it in too.

use_libwrap

This setting determines whether or not to use tcp_wrappers to discern connection attempts that are from allowed machines. Legal values are either *yes*, or *no* The default value is yes.

tcp_client_ports

This parameter may be a single numeric value or two values separated by a dash (no spaces allowed). It indicates which client ports are allowed for incoming connections. If not specified, any port is allowed. Allowed values are 1..65535. For example, to require the client use a privledged port, specify *1−1023* for this parameter. You will also need to set the local_port option in the audisp-remote.conf file. Making sure that clients send from a privileged port is a security feature to prevent log injection attacks by untrusted users.

tcp_client_max_idle

This parameter indicates the number of seconds that a client may be idle (i.e. no data from them at all) before auditd complains. This is used to close inactive connections if the client machine has a problem where it cannot shutdown the connection cleanly. Note that this is a global setting, and must be higher than any individual client heartbeat_timeout setting, preferably by a factor of two. The default is zero, which disables this check.

enable_krb5

If set to "yes", Kerberos 5 will be used for authentication and encryption. The default is "no".

krb5_principal

This is the principal for this server. The default is "auditd". Given this default, the server will look for a key named like *auditd/hostname@EXAMPLE.COM* stored in */etc/audit/audit.key* to authenticate itself, where hostname is the canonical name for the server's host, as returned by a DNS lookup of its IP address.

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/audit/audit.key*

NOTES

In a CAPP environment, the audit trail is considered so important that access to system resources must be denied if an audit trail cannot be created. In this environment, it would be suggested that /var/log/audit be on its own partition. This is to ensure that space detection is accurate and that no other process comes along and consumes part of it.

The flush parameter should be set to sync or data.

Max_log_file and num_logs need to be adjusted so that you get complete use of your partition. It should be noted that the more files that have to be rotated, the longer it takes to get back to receiving audit events. Max_log_file_action should be set to keep_logs.

Space_left should be set to a number that gives the admin enough time to react to any alert message and perform some maintenance to free up disk space. This would typically involve running the **aureport −t** report and moving the oldest logs to an archive area. The value of space_left is site dependent since the rate at which events are generated varies with each deployment. The space_left_action is recommended to be set to email. If you need something like an snmp trap, you can use the exec option to send one.

Admin_space_left should be set to the amount of disk space on the audit partition needed for admin actions to be recorded. Admin_space_left_action would be set to single so that use of the machine is restricted to just the console.

The disk_full_action is triggered when no more room exists on the partition. All access should be terminated since no more audit capability exists. This can be set to either single or halt.

The disk_error_action should be set to syslog, single, or halt depending on your local policies regarding handling of hardware malfunctions.

Specifying a single allowed client port may make it difficult for the client to restart their audit subsystem, as it will be unable to recreate a connection with the same host addresses and ports until the connection closure TIME_WAIT state times out.

FILES

/etc/audit/auditd.conf

Audit daemon configuration file

SEE ALSO

auditd(8), **audisp−remote.conf**(5).

AUTHOR

Steve Grubb