EMC²

TECHNICAL NOTES

EMC® VNX®

Version 8.1

Auditing in the VNX Control Station

P/N 300-015-127

REV A01

August, 2013

This technical note contains information on these topics:

Executive summary

In the 8.1 release family of VNX, the Control Station Linux-based operating system (OS) incorporates improvements to the auditing subsystem.

This technical note provides an introduction to the auditing infrastructure on the VNX Control Station as of release 8.x.

Introduction

Auditing is a specialized form of logging. The purpose of auditing is to record the “security-relevant events” that happen on a system and provide sufficient information about who initiated the event and the event’s effect on the system (for example, success or failure). The key, of course, is deciding on what the right set of security-relevant events happens to be for a particular environment.

Auditing differs from other kinds of logging in terms of the information that is captured. Of the various log files available on a VNX, many of them include informative messages about the overall status of the system. While these entries can be important for maintaining the VNX in peak operating condition, they are not necessarily security-relevant.

Audience

This technical note is aimed at VNX administrators and security officers responsible for monitoring administrative actions on a VNX.

Terminology

Audit Daemon

A process that manages audit log rotation and retention policies and is responsible for reading individual audit records from the kernel and writing them to the audit log.

Audit Log or Audit Trail

A file to which audit records are being (or have been) written by the audit daemon.

Audit Record

A portion of the information recorded about an audit event. An audit record has a type that indicates the fields that follow.

Audit Event

All the audit records associated with an individual activity on the system (for example, opening a file for read/write access). All audit records associated with a single event have the same serial number.

Auditing and audit logs

The Linux operating system for the VNX Control Station has a native auditing capability. This facility has the ability to audit system calls and access to file system objects. In addition, some trusted programs can write audit log entries as well (for example, the SSH daemon). An overview of the auditing subsystem is presented in Figure 1.

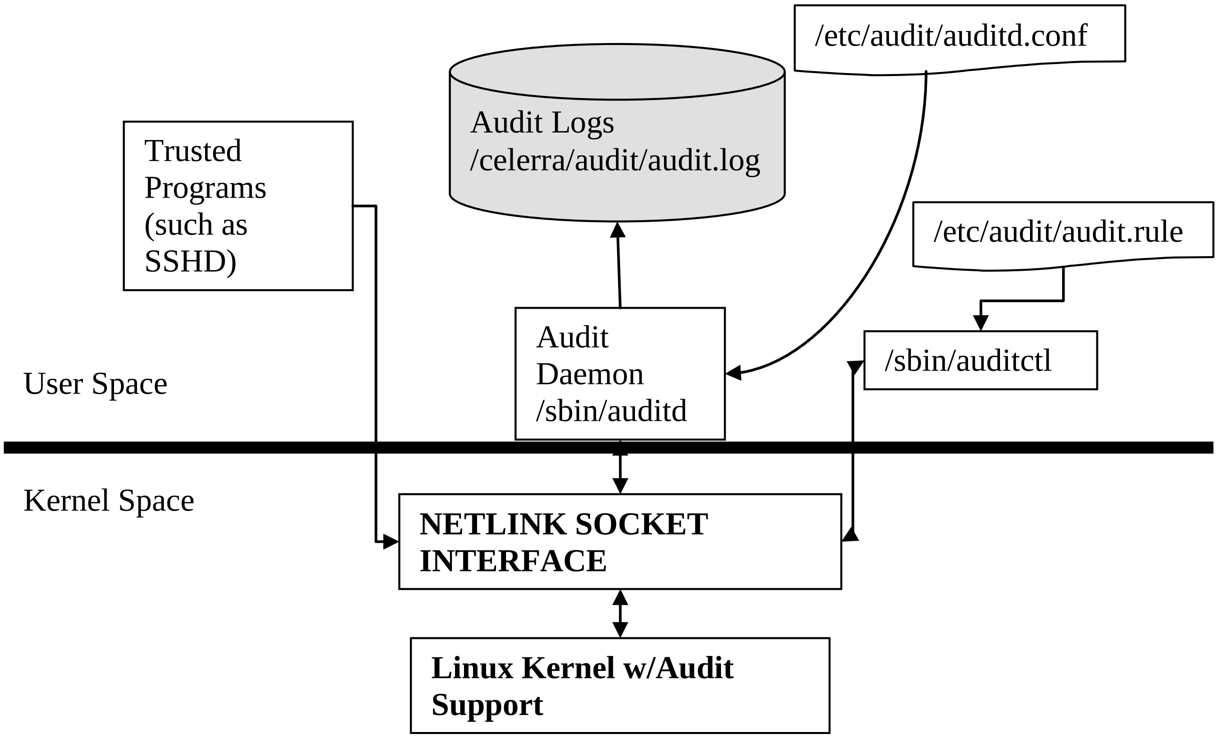

Figure 1 Diagram of the auditing system

Linux auditing is controlled by a combination of user and kernel space components. Configuration information is typically read from files and passed to the kernel via the `/sbin/auditctl` command. The kernel decides what is to be audited (or rather, which audit records are to be recorded) and writes them to the NETLINK interface. The job of the audit daemon is to pull records off of the link as fast as possible and write them to the audit log. While the audit log is located at `/var/log/audit` by default, it is located at `/celerra/audit/audit.log`[1] on the Control Station. Trusted programs (such as SSHD, /bin/login, and other trusted applications) write audit records to the NETLINK interface and the kernel, in turn, passes these back out for the audit daemon to record. The daemon also handles log rotation tasks (to ensure that a valid log is always available for recording audit records).

The `/sbin/auditctl` command also provides information about operational characteristics (maximum message rate, backlog size, and so on) and the audit rules (an indication of what is to be audited). While audit rules can be added via the command line individually, strictly controlling the order of audit rules requires entering them into the audit rules file.

Audit configuration

The behavior of the audit subsystem is mainly controlled by two separate files:

- /etc/audit/auditd.conf

  Default location of the file that controls the configuration of the audit daemon. The details on what this file controls and its format are in the auditd.conf(8) man page. The suggested initial version of this file for the VNX Control Station is delivered in /nas/sys/auditd.conf as well as being installed in the default location.

- /etc/audit/audit.rules

  Default location of the file with the audit rules to load when auditing is started. Details on what the file controls (basically a description of what can be audited) and the format are in the audit.rules(8) man page. The suggested initial version of this file for the VNX Control Station is delivered in `/nas/sys/audit.rules` as well as being installed in the default location.

By default, the VNX Control Station has been configured to have a separate file system to hold audit trails. This file system is roughly 120 MB in size and is mounted at `/celerra/audit`. The auditd.conf configuration file is used to tell the audit subsystem to use this file system.

[1] In the 5.6.x releases, the audit logs were located at `/nas/var/auditing`. By moving the location to `/celerra/audit`, the audit logs are stored on the Control Station's internal drive. This simplifies starting auditing automatically at boot time.

The audit daemon also controls log rotation and actions to take if the audit subsystem is unable to record audit events for some reason (for example, when the file system is full). The default values for controlling the audit daemon, as reflected in `/etc/audit/auditd.conf,` are:

Table 1 Audit Rule Default Values

Keyword	Value	Meaning
log_file	/celerra/audit/audit.log	Directory where the audit logs should be written. This is a separate file system on the Control Station.
log_format	RAW	Records are stored exactly as the kernel writes them. There are no other choices, presently.
priority_boost	0	No boost is provided to the audit daemon.
flush	Incremental	This value means that the “freq” parameter (below) controls how often records are flushed to disk.
freq	20	The audit daemon writes 20 records before issuing an explicit flush to disk command.
Dispatcher	/nasmcd/sbin/mt_mon	Monitoring process to faciltate auditing access to a Data Mover’s root file system
num_logs	6	The number of log files to keep.
max_log_file	15	Log files can be up to 15 MB in size.
max_log_file_action	ROTATE	Once the current log file reaches 15 MB, the audit daemon will rotate to a new log file.
space_left	15	When 15 MB of space are left in /nasmcd/var/auditing, a message is sent to the syslog.
space_left_action	SYSLOG	
admin_space_left	12	When 12 MB of space are left in /celerra/audit, auditing is suspended.
admin_space_left_action	SUSPEND	
disk_full_action	SUSPEND	When /nasmcd/var/auditing is full, auditing is suspended.
disk_error_action	SUSPEND	If there is a problem writing the audit log, suspend the auditing.

These values were chosen as defaults to suit the needs of a security-conscious VNX user with a bias to maintaining service if there are any difficulties with auditing. While these values may be changed to a stronger security/monitoring bias, the administrator is urged to consider the potential consequences of such actions very carefully. For example, if the “disk_error_action” is set to “halt”, then the Control Station will be shut down if there is a problem writing the audit log. This may have severe repercussions on the overall functionality of your VNX. In addition, the log rotation values are designed to keep the file system from becoming full. If you change these values or add additional information to the file system, you should monitor the file system to ensure that free space for recording audit logs is maintained.

Setting up the Control Station for auditing

In 5.6.x releases, auditing was not enabled on the Control Station by default. As of 6.0.x, auditing is enabled by default for new installations. In the case of upgrading from a 5.6.x system to a 6.0.x system, the auditing state (enabled or disabled) will be preserved on upgrade. EMC recommends that you enable auditing on your Control Station.

Starting the audit service

In some circumstances, auditing is started by default and nothing more needs to be done for it to start automatically with each Control Station reboot. If auditing is stopped for some reason, it may be started manually with the command `/sbin/service auditd start`. You are required to be root to control the audit subsystem. You may check that auditing is running with the command `/sbin/service auditd status.`

If auditing is not enabled by default on your system (e.g., you upgraded from 5.6.x and it was disabled there), you may use the `/sbin/chkconfig` command to indicated the run levels for which the init process should start the audit daemon. For example, the command:

```
/sbin/chkconfig –levels 2345 auditd
```

would indicate that the audit daemon should be started for run levels 2, 3, 4, and 5.

Audit commands

The commands are:

- /sbin/auditctl

  A utility to assist controlling the kernel's audit system. It is used to check the status of auditing, change the rules for what to audit, disable auditing, and so on. Details on its usage are in the auditctl(8) man page.

- /sbin/ausearch

A tool to query audit daemon logs. This is the tool for extracting information from the audit logs based upon some search criteria, such as a file or username. Details on its usage are in the ausearch(8) man page.

- /sbin/aureport

  A tool to produce summary reports of audit daemon logs. Details are in the aureport(8) man page.

- /sbin/auditd

  The actual audit daemon.

- /sbin/service auditd (start | stop | status | restart | reload | rotate | condrestart)

  The interface for starting, stopping, or otherwise controlling the audit subsystem.

Audits and actions

When auditing is enabled, you must specify exactly what should be audited. Too much detailed information can be as bad as no information at all. The VNX Control Station delivers a suggested initial version of `audit.rules` that is designed to capture an interesting set of security-relevant events but not overtax the Control Station resources or an administrators ability to monitor them.

The suggested initial version of audit.rules is designed to audit the following:

- Administrators accessing the root file system of a Data Mover
- Access to particularly sensitive system files on the Control Station
- Changes to the auditing infrastructure
- Users authenticating to the system

The last three sets of actions are easy to audit; however, auditing access to the root file system of a Data Mover is more difficult. Because these file systems are automounted, it is not possible to set a file system watch for the relevant pathnames at boot time. The file systems need to be mounted when the watch is set. Even if they were mounted, the watch is cleared when the file systems are unmounted (which happens automatically after they are not accessed for some period of time).

Enhancements in the underlying auditing subsystem have made it possible to provide a more elegant solution to this issue than was possible before. The dispatcher line in the configuration file `/etc/audit/auditd.conf` refers to a program that will monitor all audit records as they are generated. These records are analyzed and when the dispatcher detects that the root file system of a Data Mover has been mounted, it will set the appropriate file system watch. This has the desired behavior since the file system is mounted when the watch is issued. Direct access to the file system by any administrator or service provider will be recorded in the audit log unless auditing is disabled.

This approach means that fewer file open events are recorded for administrators than was previously the case. This will record a more security-relevant set of events in the audit log and make it easier to identify events of interest

Reading audit records

As mentioned previously, an *audit event* is a specific action that may have multiple *audit records* associated with it. One consequence of this is that the audit records associated with a particular audit event may not be grouped together in the raw audit trail. In addition, the raw audit trail has numerical values for values that have text-based representations (for example, names associated with UIDs and GIDs, system call names). This means that the best way to read audit events from an audit trail is with the `ausearch` command.

The complete list of record types that can show up in audit records are recorded in Appendix A: Audit Record Types.

The type of an audit record indicates the fields that are present in the record and, hence, the information that a record type might contain. While there are on the order of 100 record types, only a handful will normally show up in VNX Control Station audit logs (based upon the initial `/etc/audit/audit.rules` file). These types are:

- SYSCALL – Information associated with a system call invocation by a process or thread
- PATH – Information about a file being accessed
- CWD – The current working directory of the process when the event occurred
- USER_xxxx – Events associated with a user authenticating to the system
- FS_WATCH – Access to a file system object that has an explicit watch placed on it

Brief field descriptions are in Appendix B: Audit record field names and descriptions. Most field names are self-explanatory. The field names that are most audit-specific are:

- Auid – The Audit ID is set when a user authenticates to the system and follows them through all their actions, including actions via the sudo command or after su-ing to root.
- Items – In a SYSCALL type record, this indicates how many PATH records are associated with the call.

Example: A user logging in via SSH

The following six audit events are associated with a single login event. This output was generated with the `ausearch` command, using the `-i` switch (to convert numeric values to a human readable form). This makes timestamps and system calls much more intelligible.

```
----

type=USER_AUTH msg=audit(07/11/2007 12:04:58.826:52810) : user
pid=19373 uid=root auid=unknown(4294967295) msg='PAM
authentication: user=taylot1 exe="/usr/sbin/sshd" (hostname=
sys7dhcp59, addr=192.168.7.59, terminal=ssh result=Success)'
----

type=USER_ACCT msg=audit(07/11/2007 12:04:58.831:52811) : user
pid=19373 uid=root auid=unknown(4294967295) msg='PAM accounting:
user=taylot1 exe="/usr/sbin/sshd" (hostname= sys7dhcp59,
addr=192.168.7.59, terminal=ssh result=Success)'
----

type=LOGIN msg=audit(07/11/2007 12:04:58.854:52812) : login
pid=19375 uid=root old auid=unknown(4294967295) new auid=taylot1
----

type=USER_START msg=audit(07/11/2007 12:04:58.856:52813) : user
pid=19375 uid=root auid=taylot1 msg='PAM session open:
user=taylot1 exe="/usr/sbin/sshd" (hostname= sys7dhcp59,
addr=192.168.7.59, terminal=ssh result=Success)'
----

type=CRED_REFR msg=audit(07/11/2007 12:04:58.858:52814) : user
pid=19375 uid=root auid=taylot1 msg='PAM setcred: user=taylot1
exe="/usr/sbin/sshd" (hostname= sys7dhcp59, addr=192.168.7.59,
terminal=ssh result=Success)'
----

type=USER_LOGIN msg=audit(07/11/2007 12:04:58.868:52815) : user
pid=19373 uid=root auid=unknown(4294967295) msg='uid=765961:
exe="/usr/sbin/sshd" (hostname= sys7dhcp59, addr=192.168.7.59,
terminal=/dev/pts/3 res=success)'
```

Here, we see the following taking place:

- User taylot1 has authenticated to the system via SSH. Notice that PAM authentication is enabled.
- PAM authentication is typically a three-step process, going through

an auth, account, and session stage (see the pam(8) man page for details). In the second audit event, we see that the PAM accounting step has been completed successfully.

- In the fourth audit event, we see that the PAM session step is completed successfully. An important part of this step is that the auid value has been set to “taylot1” from the previously “unknown” value. The value of 4294967295 represents an unset auid value.

Example: A failed login attempt via SSH

An example of what happens when a login attempt fails.

```
----

type=USER_LOGIN msg=audit(07/11/2007 12:05:14.047:52816) : user
pid=19397 uid=root auid=unknown(4294967295) msg='acct=foobar:
exe="/usr/sbin/sshd" (hostname=?, addr=?, terminal=sshd
res=failed)'
----

type=USER_AUTH msg=audit(07/11/2007 12:05:18.951:52817) : user
pid=19397 uid=root auid=unknown(4294967295) msg='PAM
authentication: user=foobar exe="/usr/sbin/sshd" (hostname=
sys7dhcp59, addr=192.168.7.59, terminal=ssh result=Authentication
failure)'
----

type=USER_LOGIN msg=audit(07/11/2007 12:05:18.952:52818) : user
pid=19397 uid=root auid=unknown(4294967295) msg='acct=foobar:
exe="/usr/sbin/sshd" (hostname=?, addr=?, terminal=sshd
res=failed)'
----
```

Example: A file open event

Here we see an example of a single system call to open a file. These three audit *records* make up a single audit *event*. This is determined by looking at the msg=audit(timestamp:event id) field. A unique audit event ID is associated with each record and occurs after the colon that follows the timestamp.

> The timestamps for the records associated with a single event may not always be identical.

A single audit event may have several audit records associated with it for a variety of reasons. They may be triggered by multiple rules or conditions. For example, the system may be auditing open system call invocations and watching the particular file being opened. One of the things an administrator can do with a file system watch is to associate an arbitrary test string (a filter key) with the record for easy searching later. Audit records may be generated separately, but they have the same audit event ID.

```
----
type=PATH msg=audit(07/11/2007 11:16:07.944:52585) :
name=/etc/audit/auditd.conf flags=open inode=3428178 dev=fd:00
mode=file,640 ouid=root ogid=root rdev=00:00
type=CWD msg=audit(07/11/2007 11:16:07.944:52585) :  cwd=/root

type=FS_INODE msg=audit(07/11/2007 11:16:07.944:52585) :
inode=3428178 inode_uid=root inode_gid=root inode_dev=fd:00
inode_rdev=00:00

type=FS_WATCH msg=audit(07/11/2007 11:16:07.944:52585) :
watch_inode=3428178 watch=auditd.conf filterkey=CFG_auditd.conf
perm=read,write,exec,append perm_mask=read

type=SYSCALL msg=audit(07/11/2007 11:16:07.944:52585) : arch=i386
syscall=open success=yes exit=3 a0=8052273 a1=20000 a2=5 a3=0
items=1 pid=19201 auid=taylot1 uid=root gid=root euid=root
suid=root fsuid=root egid=root sgid=root fsgid=root comm=ausearch
exe=/sbin/ausearch
```

The audit event above shows that the user taylot1, while su-ed to root, has executed the `/sbin/ausearch` command and this command issued an open system call against the `/etc/audit/auditd.conf` file. The FS_WATCH record corresponds to a watch placed on the `/etc/audit/auditd.conf` file and the SYSCALL record was generated because the open system call was being audited. The CWD record is reporting the current working directory in effect for the command invocation. Note that all the records have event ID 52585.

Additional sources of monitoring information

For now, the auditing subsystem is not intended to capture all the potentially security-relevant information that a security administrator might desire in the VNX Control Station environment. The auditing subsystem is designed to capture events that would be difficult to capture otherwise for those who require more extensive auditing.

There are additional sources of monitoring or log information for the Control Station, and, for completeness' sake, they are mentioned below. For additional information on these logs, please read the relevant documentation, such as the *VNX System Operations* technical module.

/nas/log/cmd_log

This file displays a list of most successfully-executed VNX-specific commands. Commands that only display information or commands that reside in the `/nas/sbin` directory are not logged. Log entries look like:

- 2007-02-21 16:35:55.532 server_2:0:21096:S: /nas/bin/server_mount server_2 -p root_fs_2 /
- 2007-02-21 16:35:56.526 server_2:0:21096:E: /nas/bin/server_mount server_2 -p root_fs_2 /

The fields contained in these log entries and their values in the example above are:

- Data & Time: 2007-02-21 16:35:55.532
- Source: server_2
- User ID: 0
- PID: 21096
- Start or End Marker: S or E
- The Actual Command: /nas/bin/server_mount server_2 -p root_fs_2 /

/nas/log/sys_log

This log file displays a cumulative list of system event and log messges from the most recent Control Station reboot.

/nas/log/osmlog (aka /var/log/messages)

This is the standard Linux “syslog.” Many of the non-VNX specific commands will record security-relevant messages to this file.

/nas/http/logs/access_log

This file records activity via Unisphere and includes the authenticated user identity of the issuer of HTTP requests. The fields most likely to be of interest are:

- Originating IP address
- User Identity (as a name)
- Date and Timestamp
- URL accessed
- HTTP Return Status (for example, 200 to indicate success)

References

There are man pages on the VNX Control Station for each of the audit commands and audit configuration files mentioned. In addition, the default audit configuration files include comments.

Appendix A: Audit Record Types

All audit records begin with a “type=<type name>” entry. This characterizes the audit record as having a particular set of information and provides a handle for searches (using `/sbin/ausearch –m`, for example).

ADD_GROUP
ADD_USER
ANOM_ACCESS_FS
ANOM_ADD_ACCT
ANOM_AMTU_FAIL
ANOM_CRYPTO_FAIL
ANOM_DEL_ACCT
ANOM_EXEC
ANOM_LOGIN_ACCT
ANOM_LOGIN_FAILURES
ANOM_LOGIN_LOCATION
ANOM_LOGIN_SESSIONS
ANOM_LOGIN_TIME
ANOM_MAX_DAC
ANOM_MAX_MAC
ANOM_MK_EXEC
ANOM_MOD_ACCT
ANOM_PROMISCUOUS
ANOM_RBAC_FAIL
ANOM_RBAC_INTEGRITY_FAIL
ANOM_SEGFAULT
AVC
AVC_PATH
CHGRP_ID
CONFIG_CHANGE
CRED_ACQ
CRED_DISP
CRED_REFR
CWD
DAC_CHECK
DAEMON_ABORT
DAEMON_CONFIG
DAEMON_END
DAEMON_ROTATE
DAEMON_START
DEL_GROUP
DEL_USER
EXECVE
FD_PAIR
FS_RELABEL
IPC
IPC_SET_PERM
KERNEL
KERNEL_OTHER
LABEL_LEVEL_CHANGE
LABEL_OVERRIDE
LOGIN
MAC_CIPSOV4_ADD
MAC_CIPSOV4_DEL
MAC_CONFIG_CHANGE
MAC_IPSEC_ADDSA
MAC_IPSEC_ADDSPD
MAC_IPSEC_DELSA
MAC_IPSEC_DELSPD
MAC_MAP_ADD
MAC_MAP_DEL
MAC_POLICY_LOAD
MAC_STATUS
SOCKADDR
TEST
TRUSTED_APP
USER
USER_ACCT
USER_AUTH
USER_AVC
USER_CHAUTHTOK
USER_CMD
USER_END
USER_ERR
USER_LABELED_EXPORT
USER_LOGIN
USER_LOGOUT
USER_MGMT
USER_ROLE_CHANGE
USER_SELINUX_ERR
USER_START
USER_UNLABELED_EXPORT
USYS_CONFIG
MQ_GETSETATTR
MQ_NOTIFY
MQ_OPEN

MQ_SENDRECV
OBJ_PID
PATH
RESP_ACCT_LOCK
RESP_ACCT_LOCK_TIMED
RESP_ACCT_REMOTE
RESP_ACCT_UNLOCK_TIMED
RESP_ALERT
RESP_ANOMALY
RESP_EXEC
RESP_HALT
RESP_KILL_PROC
RESP_SEBOOL
RESP_SINGLE
RESP_TERM_ACCESS
RESP_TERM_LOCK
ROLE_ASSIGN
ROLE_REMOVE
SELINUX_ERR
SOCKADDR
SYSCALL
TEST
TRUSTED_APP
USER
USER_ACCT
USER_AUTH
USER_AVC
USER_CHAUTHTOK
USER_CMD
USER_END
USER_ERR
USER_LABELED_EXPORT
USER_LOGIN
USER_LOGOUT
USER_MGMT
USER_ROLE_CHANGE
USER_SELINUX_ERR
USER_START
USER_UNLABELED_EXPORT
USYS_CONFIG

Appendix B: Audit record field names and descriptions

Field names in audit records capture the various pieces of information that are associated with the record. Of course, not all audit record types have the same fields.

Possible field names and their meaning are:

Table 2 Field names and meanings

Name	Meaning
ao, a1, a2, or a3	alphanumeric, the first four arguments to a syscall
acct	alphanumeric, a user's account name
addr	the remote address that the user is connecting from
arch	numeric, the elf architecture flags
audit_backlog_limit	numeric, audit system's backlog queue size
audit_enabled	numeric, audit system’s enable/disable status
audit_failure	numeric, audit system's failure mode
auid	numeric, login user id, once set this should be immutable. The idea is that this value should always represent the user’s true identity.
comm	alphanumeric, command line program name
cwd	path name, the current working directory
dev	numeric, in path records, major and minor for device
dev	in avc records, device name as found in /dev
egid	numeric, effective group id
euid	numeric, effective user id
exe	path name, executable name
exit	numeric, syscall exit code
file	file name
flags	numeric, file system namei flags
format	alphanumeric, audit log's format
fsgid	numeric, file system group id
fsuid	numeric, file system user id
gid	numeric, group id
hostname	alphanumeric, the hostname that the user is connecting from
id	numeric, during account changes, the user id of the account
inode	numeric, inode number
inode_gid	numeric, group id of the inode's owner

inode_uid	numeric, user id of the inode's owner
item	numeric, which item is being recorded
items	numeric, the number of path records in the event
list	numeric, the audit system's filter list number
mode	numeric, mode flags on a file
msg	alphanumeric, the payload of the audit record
nargs	numeric, the number of arguments to a socket call
name	file name in avcs
obj	alphanumeric, lspp object context string
ogid	numeric, object owner's group id
old	numeric, old audit_enabled, audit_backlog, or audit_failure value
old_prom	numeric, network promiscuity flag
op	alphanumeric, the operation being performed that is audited
ouid	numeric, object owner's user id
parent	numeric, the inode number of the parent file
path	file system path name
perm	numeric, the file permission being used
perm_mask	numeric, file permission audit mask that triggered a watch event
pid	numeric, process id
prom	numeric, network promiscuity flag
qbytes	numeric, ipc objects quantity of bytes
range	alphanumeric, user's SE Linux range
rdev	numeric, the device identifier (special files only)
res	alphanumeric, result of the audited operation (success/fail)
result	alphanumeric, result of the audited operation (success/fail)
role	alphanumeric, user's SE Linux role
saddr	alphanumeric, socket address
sauid	numeric, sending login user id
scontext	alphanumeric, the subject's context string
seuser	alphanumeric, user's SE Linux user acct
sgid	numeric, set group id
spid	numeric, sending process id
subj	alphanumeric, lspp subject's context string
success	alphanumeric, whether the syscall was successful or not
suid	numeric, sending user id
syscall	numeric, the syscall number in effect when the event occurred

tclass	alphanumeric, target's object classification
tcontext	alphanumeric, the target's or object's context string
terminal	alphanumeric, terminal name the user is running programs on
tty	alphanumeric, tty interface that the user is running programs on
type	alphanumeric, the audit record's type
uid	numeric, user id
user	alphanumeric, account the user claims to be prior to authentication
ver	numeric, audit daemon's version number
watch	the file name in a watch record

Appendix C: /etc/audit/audit.rules for the VNX Control Station

This is the initial version of the file `/etc/audit/audit.rules` delivered on the VNX Control Station. The contents of this file determine what is audited when an administrator starts auditing (along with audit events from trusted programs regarding authentication events). These audit rules are intended to be mostly self-explanatory, at least after reading the auditctl(8) man page.

```
####################################
# Copyright 2008 EMC Corporation
# Unpublished - All Rights Reserved
#####################################

#
# This file contains the auditctl rules that are loaded
# when the audit daemon is started.  If you feel that you need to
# audit additional events, please expand the set of audit events
# carefully to avoid undue impact on the performance of your VNX
# Control Station.
#
# Details on what these entries mean can be found in the auditctl(8)
# man page.

## These rules are for those interested in roughly a "medium low" level of
## scrutiny. They are also designed to minimize the impact on Control
## Station functionality.

# First rule - delete all current rules so that this file completely
# defines the initial set of audit rules.
-D

###----------------Configure Audit System Parameters-----------------###

# Increase the buffer size to survive stress events
-b 256

# Enable Auditing
-e 1

# Print to console and syslog if audit failures occur
-f 1

###----------System Call Rules----------###

##
## These rules apply to all system call invocations, regardless of
## which process is involved.  Note that they are order-dependent.

# Audit exec operations performed by users.  This captures commands
# that are run by all users with an auid between 200 and 10,000.
# This is intended to exclude system processes, daemons, and unset
# auid values.

## NOTE:  If these rules are extended to capture additional audit
##        events, carefully monitor the impact to your Control Station
##        and to how quickly log files fill up and are rotated off the
##        Control Station.

-a exit,always -S execve -F success=1 -F auid>=200 -F auid<=10000

## Also note that user authentication events via SSH are also captured
## in the audit trail.

##----------Set File System Watches----------###

## These are audit events associated with particular files or
## directories on the system.  According to the auditctl(8) man page,
## the number of watches set does not impact performance.

-w /nas/quota -p rwa -k CFG_Backend
-w /nbsnas/quota -p rwa -k CFG_Backend
-w /nasmcd/quota -p rwa -k CFG_Backend

-w /nas/rootfs -p rwa -k CFG_Backend
-w /nbsnas/rootfs -p rwa -k CFG_Backend
-w /nasmcd/rootfs -p rwa -k CFG_Backend

# Audit changes (or attempted changes) to the audit configuration
# files.
#
-w /etc/audit/auditd.conf -p wa -k CFG_Audit
-w /etc/audit/audit.rules -p wa -k CFG_Audit

# Audit all attempts to access the audit logs
#
-w /celerra/audit -k Audit_Log_Access

## Changes to the PAM Configuration Files
#
-w /etc/pam.d/atd -p wa -k CFG_Authentication
-w /etc/pam.d/authconfig -p wa -k CFG_Authentication
-w /etc/pam.d/check_user -p wa -k CFG_Authentication
-w /etc/pam.d/chfn -p wa -k CFG_Authentication
-w /etc/pam.d/chsh -p wa -k CFG_Authentication
```

```
-w /etc/pam.d/crond -p wa -k CFG_Authentication
-w /etc/pam.d/cups -p wa -k CFG_Authentication
-w /etc/pam.d/halt -p wa -k CFG_Authentication
-w /etc/pam.d/internet -p wa -k CFG_Authentication-druid
-w /etc/pam.d/kbdrate -p wa -k CFG_Authentication
-w /etc/pam.d/login -p wa -k CFG_Authentication
-w /etc/pam.d/neat -p wa -k CFG_Authentication
-w /etc/pam.d/newrole -p wa -k CFG_Authentication
-w /etc/pam.d/other -p wa -k CFG_Authentication
-w /etc/pam.d/passwd -p wa -k CFG_Authentication
-w /etc/pam.d/poweroff -p wa -k CFG_Authentication
-w /etc/pam.d/ppp -p wa -k CFG_Authentication
-w /etc/pam.d/reboot -p wa -k CFG_Authentication
-w /etc/pam.d/remote -p wa -k CFG_Authentication
-w /etc/pam.d/rexec -p wa -k CFG_Authentication
-w /etc/pam.d/rlogin -p wa -k CFG_Authentication
-w /etc/pam.d/rsh -p wa -k CFG_Authentication
-w /etc/pam.d/run_init -p wa -k CFG_Authentication
-w /etc/pam.d/screen -p wa -k CFG_Authentication
-w /etc/pam.d/smtp -p wa -k CFG_Authentication
-w /etc/pam.d/smtp.sendmail -p wa -k CFG_Authentication
-w /etc/pam.d/sshd -p wa -k CFG_Authentication
-w /etc/pam.d/su -p wa -k CFG_Authentication
-w /etc/pam.d/sudo -p wa -k CFG_Authentication
-w /etc/pam.d/system-auth -p wa -k CFG_Authentication
-w /etc/pam.d/system-config-mouse -p wa -k CFG_Authentication
-w /etc/pam.d/system-config-network -p wa -k CFG_Authentication
-w /etc/pam.d/system-config-network-cmd -p wa -k CFG_Authentication
-w /etc/pam.d/system-config-network-druid -p wa -k CFG_Authentication


## Audit changes to user management and roles databases
#
-w /nas/site/role -p wa  -k CFG_User_Roles
-w /nas/site/user_db -p wa -k CFG_User_Roles
-w /nas/site/group_db -p wa -k CFG_User_Roles
```

Published August 2013

EMC believes the information in this publication is accurate of its publication date. The information is subject to change without notice.

The information in this publication is provided as is. EMC Corporation makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose. Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

For the most up-to-date regulatory document for your product line, go to the technical documentation and advisories section on the EMC online support website.