Enable Linux File/Folder Auditing for exacqVision Recordings and Logs (Auditd)

Introduction

This article explains how to enable Linux auditing so you can see who created, modified, or deleted files inside exacqVision storage locations (recorded footage folders) and the exacqVision log directory on Linux. The recommended method uses the Linux Audit Framework (auditd), which records file activity at the system-call level.

Problem

Recorded video or log files appear to be missing or changed on an exacqVision Linux recorder, and you need evidence of which user account or process performed the action.

Cause

Linux does not keep a complete “who changed this file” history by default. You must enable auditing and define rules that watch the folders you care about. Without audit rules, the system will not generate events for file create/delete/write operations.

Solution

Summary steps:

A) Install and enable auditd (Linux Audit Daemon).

B) Add audit rules for the recording folder(s) and exacqVision log folder.

C) Generate a test event (create/delete a file) and confirm logs are recorded.

D) Use ausearch/aureport to identify WHO (auid/uid), WHAT (file path), and HOW (process/syscall).

A) Install and start auditd

On Ubuntu/Debian:

sudo apt update
sudo apt install -y auditd audispd-plugins

sudo systemctl enable --now auditd
sudo systemctl status auditd

On RHEL/CentOS/Fedora (examples):

sudo dnf install -y audit # Fedora/RHEL
sudo systemctl enable --now auditd

Audit logs are written to /var/log/audit/audit.log by default.

B) Add folder watches (recommended for most cases)

Use watch rules for directories that contain recorded footage and logs. Watches support permissions flags: r=read, w=write, x=execute, a=attribute change. For file change tracking, start with "wa".

Example: watch a recordings folder and exacqVision logs folder (replace paths for your server):

Runtime (until reboot):
sudo auditctl -w /mnt/exacq/recordings -p wa -k exacq_recordings
sudo auditctl -w /var/log/exacqvision -p wa -k exacq_logs

Verify loaded rules:
sudo auditctl -l | grep exacq

Make rules persistent (survive reboot): create a rules file under /etc/audit/rules.d/ and load it with augenrules or restart auditd.

sudo tee /etc/audit/rules.d/exacq-filewatch.rules >/dev/null <<'EOF'
-w /mnt/exacq/recordings -p wa -k exacq_recordings
-w /var/log/exacqvision -p wa -k exacq_logs
EOF

sudo augenrules --load # or: *sudo systemctl restart auditd*

Note: Directory watches are recursive until a mount point boundary; watch the correct filesystem roots if recordings are on separate mounts.

C) Test and confirm logging

Create and delete a test file inside the watched folder (example):

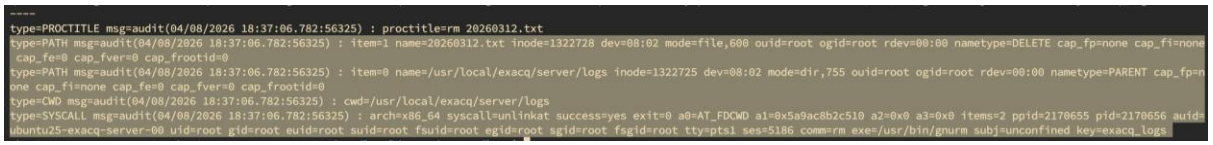
sudo touch /mnt/exacq/recordings/_audit_test.txt
sudo rm -f /mnt/exacq/recordings/_audit_test.txt

Search for events by key and interpret IDs into names:

sudo ausearch -k exacq_recordings -ts recent -i
sudo ausearch -k exacq_logs -ts recent -i

In results, focus on these fields: auid (original logged-in user), uid/euid (effective user), exe/comm (process), syscall (unlink/rename/openat), and PATH records (the file name).

```
/usr/local/exacq/server/logs$ sudo rm 20260312.txt
/usr/local/exacq/server/logs$ sudo ausearch -k exacq_logs -ts recent -i
```



Optional: produce a summary report of file events:

sudo aureport -f -i --summary
Drill into a specific event number from aureport:
sudo ausearch -a <EVENT_NUMBER> -i

D) Map actions to “Add / Edit / Delete”

Use the syscall and PATH “nametype” to understand what happened:

• Create/Write: open/openat/creat with write flags; PATH shows the file and type.

• Delete: unlink/unlinkat; PATH nametype may show DELETE.

• Rename/Move: rename/renameat; PATH shows old/new names.

• Permission changes: chmod/chown/setxattr; PATH shows the target file.

Tip: auid is often the best “human user” to report, because processes may run as root (uid=0) while still being attributable to the logged-in user (auid).

Common pitfalls

1) High log volume: Watching large recording trees can produce many events. Start with the smallest folder scope you can.

2) Missing events after reboot: rules added with auditctl are not persistent; use /etc/audit/rules.d/*.rules and augenrules.

3) Separate mounts: if recordings are stored on a separate mount, ensure the watch path is on the correct mount point.

4) Containers/services: if exacqVision runs as a service account, events may show the service user; use auid to identify the original login session when applicable.

> **Note**
>
> Focus on these fields: **auid** (original logged-in user**), uid/euid** (effective user), exe/comm (process), **syscall** (unlink/rename/openat), and **PATH** records (the file name).

References

auditd overview and files (audit.log, audit.rules, rules.d): https://manpages.ubuntu.com/manpages/xenial/man8/auditd.8.html

auditctl man page (configure rules, keys, permissions): https://www.man7.org/linux/man-pages/man8/auditctl.8.html

audit.rules man page (watch rules, recursive directory behavior): https://www.man7.org/linux/man-pages/man7/audit.rules.7.html

Example watch usage (-w, -p, -k): https://access.redhat.com/solutions/10107

aureport usage and interpreting audit logs: https://www.man7.org/linux/man-pages/man8/aureport.8.html