

Security Data and Documentation: If It Isn't Recorded, It Didn't Happen

Mon, Aug 24, 6:00 PM EDT · <https://scottslab.io/posts/security-data-and-documentation>



TL;DR

A security programme runs on two kinds of data: technical logs and the process records proving your controls actually ran. Three things quietly decide whether either survives contact with an auditor or a court: synchronised clocks, because unsynchronised logs cannot be correlated and are trivially challenged; tamper-evident storage, because a log an administrator can edit proves nothing about that administrator; and contemporaneous records, because version history does not lie. Backfilled evidence isn't evidence. It's a confession with extra steps.

Running a security programme means proving it works, and proof is data. It comes in two flavours people tend to conflate.

Technical data is machine exhaust: logs from servers, firewalls, the IPS, access control, pulled together and made sense of in a SIEM. **Process data** is the paperwork: the records showing your processes happened at all, that backups ran, that account reviews were done, that scans went out.

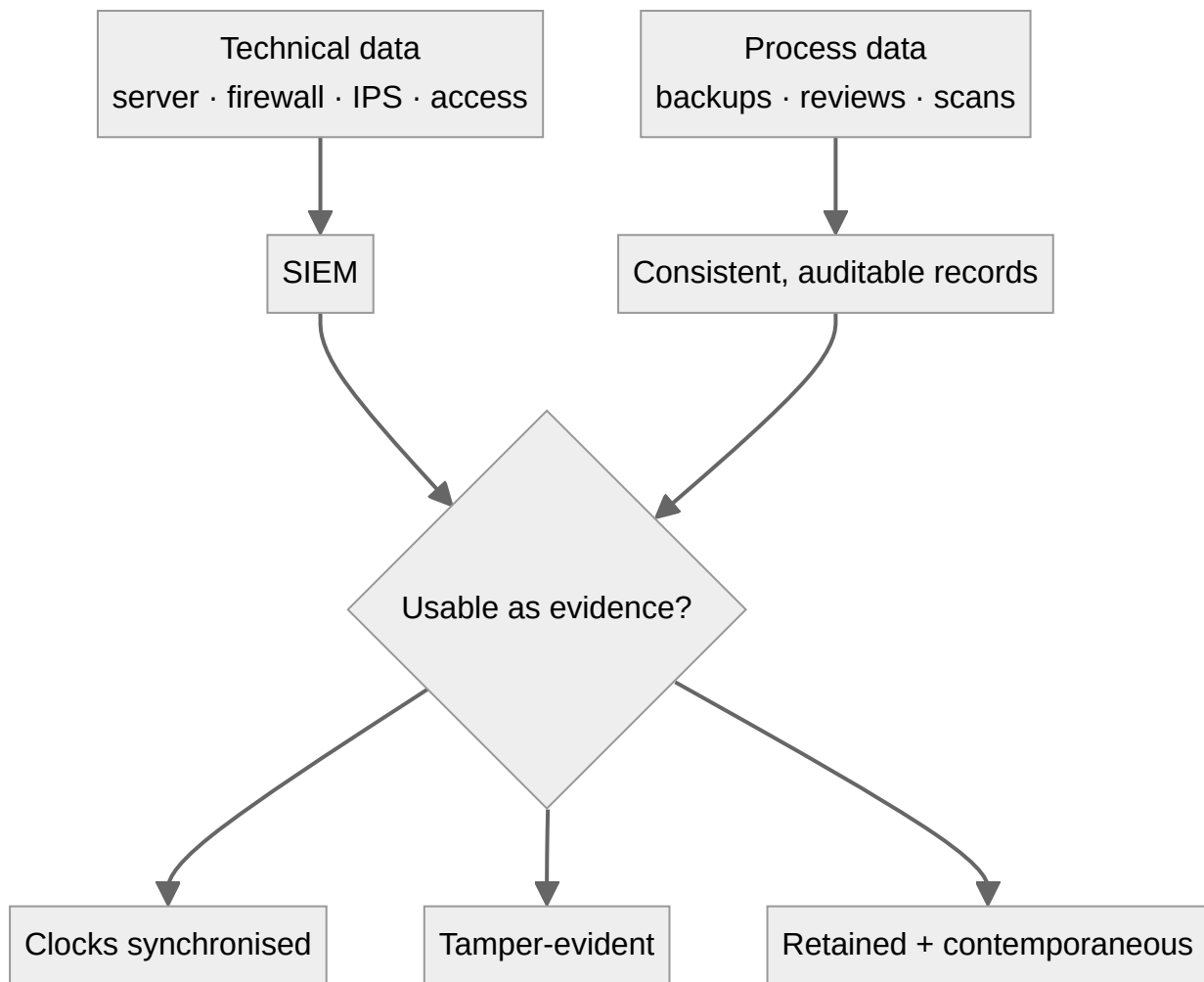
The SIEM tells you what the systems did. The process records tell you what the humans did. Auditors ask for both, and most programmes are far weaker on the second.

Three things that decide whether your logs are worth anything

Synchronised clocks. This is the boring one that ruins investigations. Correlating an event across a firewall, a server and an identity provider requires their timestamps to mean the same thing. If one host's clock drifts by four minutes, your timeline is wrong in a way that looks *plausible*. That's worse than obviously broken, because you'll believe it. Run NTP everywhere, log in UTC, monitor for drift, and record the timezone explicitly. In any proceeding, unsynchronised timestamps are the first thing a competent opponent attacks, and it works.

Tamper-evident storage. A log file an administrator can edit tells you nothing about what that administrator did. If logs matter as evidence, they need write-once storage, append-only retention, or hash-chaining so alteration is detectable, and they need to leave the originating host promptly, because the first thing a capable intruder does is clean up after themselves. Centralised logging isn't just for convenience; it means the attacker has to compromise two systems to erase their tracks instead of one.

Retention that matches your obligations. Know the number for your regime rather than guessing. PCI DSS, for instance, requires audit log history be retained for **at least 12 months, with the most recent three months immediately available for analysis**. "Immediately available" is doing real work in that sentence, because a year of logs in cold storage you can't query in an afternoon does not satisfy it.



Three things that decide whether your logs are worth anything

What an auditor is actually testing

This is the part that surprises people the first time. An auditor is generally not checking whether a control is in place today. They're checking whether it **operated throughout the period**. That difference decides how much of your year you spend on the audit.

So they ask for a **population** and then sample it. If the control is quarterly access reviews, the population is all four reviews for the year, and you must be able to produce the complete list before they pick which to examine. A programme that can produce three of four has not passed three-quarters of the test; it has demonstrated the control does not reliably run.

Which means the useful question when designing any control is not "can we do this?" but "**what artefact will this produce, automatically, every time it runs?**" A control that leaves no trace is unauditably no matter how diligently you perform it.

Whatever holds the process data, a GRC platform or, honestly, a well-kept spreadsheet, needs to be consistent and auditable. The tooling matters less than the discipline. What matters is that records are complete, follow a consistent format, and can be handed over without a translation layer.

Contemporaneous, or it isn't real

The part that trips people up is recording things **when they happen**, not reconstructing them the night before the audit.

Modern tools quietly enforce this. A shared spreadsheet keeps version history, so "we've been reviewing accounts quarterly" collapses instantly when the history shows no edits between March 2018 and December 2020. The same is true of a document created three days before fieldwork, or twelve months of entries in identical handwriting and one ink.

Backfilled records aren't evidence. They're a confession with extra steps. Unlike a missing record, which is a finding, a fabricated one is a different category of problem entirely.

The habit is boring and non-negotiable: record the control as you perform it, in something that timestamps you honestly. Because a programme you can't prove you ran is, to an auditor, a programme you didn't run. To a court, it's worse than that.

Written by Scott S. — Contact: scott@scottschlangen.com — scottslab.io — <https://scottslab.io/posts/security-data-and-documentation>