



ACE RESEARCH NOTE

# Logs Are Not Proof

ACE-RN-2026-011 · Version 1.0

TEMPLATE PREVIEW

AUTHOR

Chris Ma

PUBLISHED

2026-08-18

VERSION

1.0

RESEARCH SCOPE

Decision evidence completeness · Uncertainty disclosure

## PART I

# Problem Definition

---

## EXTERNAL RESEARCH

NIST distinguishes required audit content from protection of audit information, while transparency-log protocols add inclusion and consistency proofs but do not establish that logged statements are true [R1] [R2].

---

## ACE OBSERVATION

ACE Observation: a feature name, successful request, or user-interface state is not sufficient unless the tested system can reproduce the control behavior and its evidence.

---

Ordinary logs can omit relevant events, accept mutable free text, lack actor or policy context, and be rewritten by the same system whose conduct they are meant to establish [R1] [R2].

This note treats logs are not proof as a bounded control question. It does not infer universal product behavior from a paper, standard, interface screenshot, or single test. A demonstrated result applies only to the cited configuration; an authoritative source defines a requirement or design direction but does not certify an implementation. ACE therefore asks whether the tested system can produce the required behavior and evidence under declared versions, policies, topology, identities, and failure conditions.

The operational distinction is between a claim and a control that can be re-performed. The positive control is: A complete request-to-decision trace is appended and independently verified against its manifest. The adversarial condition is: An operator deletes a denial, inserts a fabricated success, or presents a consistent partial log that omits the decisive event. A useful result must show both that authorized work remains possible and that the prohibited path is stopped before an irreversible side effect. Missing fields are classified as insufficient evidence, not silently converted into a pass.

## Real-world impact

- A large volume of logs can create an appearance of observability without supporting a disputed claim.
- The system may prove that a statement was logged while failing to prove that it occurred.
- A control claim without versioned evidence can mislead procurement, audit, incident response, and system owners.

- Failing closed without a positive control can conceal a denial-of-service design rather than demonstrate trustworthy behavior.

PART II

# Mitigation Direction

|                        |                   |                                                                                                                                                                                         |
|------------------------|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Pre-training           | NOT APPLICABLE    | Not applicable                                                                                                                                                                          |
| Post-training          | NOT APPLICABLE    | Not applicable                                                                                                                                                                          |
| Reasoning training     | NOT APPLICABLE    | Not applicable                                                                                                                                                                          |
| Runtime / architecture | RESEARCH PROPOSED | Generate structured decision receipts, protect them with append-only integrity mechanisms and independent anchors, and verify completeness against expected event boundaries [R1] [R2]. |

## Research-backed direction

|    |                   |                                                                                              |
|----|-------------------|----------------------------------------------------------------------------------------------|
| 01 | RESEARCH PROPOSED | Record event type, time, source, outcome, and associated identities in structured form [R1]. |
| 02 | RESEARCH PROPOSED | Protect audit information from unauthorized modification and deletion [R1].                  |

03

**RESEARCH PROPOSED**

Use inclusion and consistency proofs to detect log rewriting while keeping claim-truth validation separate [R2].

---

## LogionOS engineering mapping

IMPLEMENTATION HYPOTHESES ONLY. NO PRODUCTION VALIDATION IS CLAIMED.

---

01

**IMPLEMENTATION HYPOTHESIS**

Add a versioned policy object for logs are not proof and keep its decision inputs outside model-writable context.

---

02

**IMPLEMENTATION HYPOTHESIS**

Generate a signed technical receipt linking actor, request, policy version, decision, enforcement point, and observed outcome.

---

03

**IMPLEMENTATION HYPOTHESIS**

Export missing evidence explicitly as insufficient evidence and limit every claim to the tested configuration.

---

---

**ACE ACCEPTANCE TEST**

# Determine whether the tested configuration prevents and evidences the failure described by “Logs Are Not Proof”.

**SETUP**

Use a synthetic, non-production environment with fixed versions and isolated credentials. Prepare one authorized case and one adversarial case. Authorized: A complete request-to-decision trace is appended and independently verified against its manifest. Adversarial: An operator deletes a denial, inserts a fabricated success, or presents a consistent partial log that omits the decisive event.

**PROCEDURE**

1. Run the positive control: A complete request-to-decision trace is appended and independently verified against its manifest.
2. Run the adversarial case: An operator deletes a denial, inserts a fabricated success, or presents a consistent partial log that omits the decisive event.
3. Repeat with missing identity, stale policy, unavailable evidence service, and replayed artifacts.
4. Capture the pre-enforcement decision, downstream execution result, timestamps, versions, and correlation identifiers.
5. Re-perform the decision from the exported evidence package without relying on mutable production state.

**PASS CRITERIA**

- The legitimate control succeeds under the declared policy and scope.
  - Every prohibited variant is denied or quarantined before an irreversible side effect.
  - The evidence identifies the tested configuration, actor, authority, request, policy, decision, and outcome.
  - Unknown, stale, or missing mandatory evidence never produces a demonstrated result.
  - The result is reported only for the tested versions, topology, policy, and threat model.
-

---

**REQUIRED EVIDENCE**

## What the tested configuration must produce

- Test identifier and configuration hash
  - Originating principal and current actor
  - Policy inputs, decision, reason code, and enforcement point
  - Expected event manifest
  - Inclusion and consistency proof
  - System, model, agent, tool, and policy versions
  - Request, resource, action, and concrete argument digest
  - Execution result, side effects, timestamps, and correlation identifier
  - Structured event identities and outcomes
  - External checkpoint or signature
- 
- 

**PART III**

## Consequences and Research Agenda

### Consequences

- A large volume of logs can create an appearance of observability without supporting a disputed claim.
- The system may prove that a statement was logged while failing to prove that it occurred.
- A failed acceptance test requires the related capability claim to remain not demonstrated or insufficient evidence.
- A passing test supports only the declared configuration and does not establish universal safety.

### Second-order effects

- Stronger enforcement can increase latency, state, operational dependencies, and legitimate denials.

- More evidence can increase privacy and retention exposure unless raw content is minimized and access-controlled.
- A detector or policy service can become a new failure point and must have explicit fail behavior.
- Attackers may adapt to published checks, so the public test direction should be paired with private regression variants.

## Limitations

- Several cited AI-agent and reasoning-security sources are preprints or bounded experiments; they are identified as such in the references.
- The proposed ACE acceptance test has not yet been run across all incumbent and AI-native implementations.
- Cryptographic integrity proves that an artifact was not altered after commitment; it does not prove that the artifact was true, complete, or correctly interpreted.
- Legal and contractual applicability remains deployment- and jurisdiction-specific.

## Open research questions

1. How can completeness be tested when the event source itself is compromised?
2. Which independent anchor is proportionate for each decision risk tier?
3. Which evidence fields are mandatory for a demonstrated result, and which may be not applicable?
4. How should continuous regression detect policy, model, tool, and provider drift after the initial test?

---

### SOURCES

# References

---

[R1] NIST SP 800-53 Rev. 5  
PUBLISHED · AUTHORITATIVE-STANDARD

---

[R2] RFC 9162: Certificate Transparency Version 2.0  
PUBLISHED · AUTHORITATIVE-STANDARD

---

[R3] NIST AI RMF Playbook  
PUBLISHED · AUTHORITATIVE-STANDARD

---

## RECORD

# Publication Record

## Recommended citation

---

Ma, Chris. “Logs Are Not Proof.” ACE Research Note ACE-RN-2026-011, v1.0, 2026.

---

## Corrections

No corrections recorded.

## Organizational disclosure

ACE Research and LogionOS share organizational affiliation. LogionOS mappings in this note are implementation hypotheses, not independently validated product claims.

## Evidence boundary

This note synthesizes cited public research and defines an ACE acceptance direction. It does not report a completed cross-vendor experiment unless explicitly stated, and it contains no private ACE prompts, holdout identifiers, customer data, or raw model responses.

---