

Chain of Custody Appendix

Permavault Archive

Custody statement

The capture and its storage were performed by neutral, automated Permavault infrastructure operated independently of the content source (a source website, or a file supplied by the account holder). No human viewed, edited, or otherwise handled the artifact bytes in the course of capture, sealing, storage, or anchoring. Every lifecycle step was recorded as it happened in the append-only, hash-chained event log rendered below. This process is designed to align with the principles of ISO/IEC 27037 (identification, collection, acquisition and preservation of digital evidence) and to support authentication under FRE 902(13) and 902(14).

Capture

Capture ID 4f868e92-f12f-4fb7-bb5a-cb096b82eac8

Events recorded 4

Integrity of this log

Each entry below carries an entry hash: the SHA-256 of the entry's canonical content together with the previous entry's hash. Entry N therefore commits to the entire log before it; altering, removing, or reordering any entry changes every hash after it. The raw chain ships alongside this appendix as custody-log.json so the computation can be repeated independently, without Permavault.

Anchored in the signed manifest: the events up to sealing recompute to exactly the chain head this capture's signed manifest committed to (2 events, head shown with the sealing entry below). Because the manifest is signed and its hash is timestamped independently, the early history cannot have been rewritten, even wholesale with every hash recomputed to look consistent, without failing this comparison. Entries after sealing are outside this commitment by design and are corroborated by their own records (Arweave transactions and timestamp anchors).

Chain verification Verified: all 4 entries link and recompute correctly

Event chain

1. Capture requested 2026-07-23T08:33:18.484Z (UTC), recorded by permavault-capture-server

requestedBy: user
url: https://permavault.xyz
entry hash: fc36a1d85bbe7ffcd588b601aa59ac44680f9ec703d16fce0c91d29a69e63aec

2. Content captured and hashed 2026-07-23T08:33:19.462Z (UTC), recorded by permavault-capture-server

sha256: ad093d193f31d4e6240fc06b6a12a752a9b7105b37d1a7178e0c9b76e268762b
url: https://permavault.xyz
entry hash: 91c47b29f95be37b4c56cd71be32b546c990064f714452fe0e1f0895157a6fab

3. Signed manifest sealed 2026-07-23T08:33:19.462Z (UTC), recorded by permavault-capture-server

```
keyId: 98ae373420471711
manifestSha256: 156c51a5b58a5afc8fc84cb870351f0df49b06440ad2835e169b79ebd5a227b7
entry hash: 73b05c6d7caa1dec6ed095cdd9a0068eb18a226eacc6f4a64ad14bc3245ce727
```

4. Stored on Arweave

2026-07-23T08:33:25.015Z (UTC), recorded by permavault-capture-server

```
arweaveTx: 7iFluxfrjGu-BN6mdnfxUqQCClB6z9rYNIjDMYokF3I
entry hash: 6aa189a9ce39e9d0eb9805bb59ff416d6b8bb77710fca17a37766597c1336f5d
```

Deletion log

No user deletions are recorded for this capture. A deletion, had one occurred, would appear here with its time, actor, and the final entry hash of the chain it ended, because deletion records are kept separately and survive the deleted capture.

How to verify this log independently

Open custody-log.json from the evidence package. For each entry, serialize the fields seq, eventType, occurredAt, actor, and details as canonical JSON (fixed field order, details keys sorted), append a newline and the previous entry's hash (empty for the first entry), and compute the SHA-256. The result must equal the entry's recorded entry hash, and each entry's prevHash must equal the previous entry's entry hash.