

Capture System and Process Description

Permavault Archive

Scope of this document

This document describes the technical process by which Permavault captures and preserves content. It is provided to support a qualified person's certification or testimony about that process, and is designed to support authentication under FRE 902(13) and 902(14). It does not assess the truth, accuracy, or legality of any captured content, and it is not legal advice or a determination of admissibility.

Document identity

Process description version	1.2
Describes Permavault version	0.6.4
Capture ID	4f868e92-f12f-4fb7-bb5a-cb096b82eac8
Capture kind	web page
Capture made with	Permavault 0.6.4 (the version this document describes)

The capture process, step by step

Every Permavault capture follows the same automated pipeline. No step involves a human viewing, editing, or otherwise handling the captured bytes.

Step 1. Request intake

A capture begins when the account holder submits a URL. The request is received by Permavault's capture servers (neutral infrastructure operated independently of both the account holder's device and the source website) and a "requested" event is written to the capture's append-only event log.

Step 2. Safety and structure validation

The submitted URL is validated before any fetch: non-HTTP schemes, private and internal network addresses, and DNS rebinding attempts are rejected. This confines the capture to the public web resource the account holder named.

Step 3. Content acquisition and rendering

The page is fetched and rendered by the Scoop web-capture engine (developed by the Harvard Library Innovation Lab), which drives a Chromium browser via Playwright on Permavault's servers. The page is loaded the way an ordinary browser would load it, and every network exchange involved in rendering it is recorded. The capture environment (server IP, browser user agent, TLS certificate details, redirect chain, and software versions) is recorded in a forensic record alongside the capture.

Step 4. Hashing at capture

The captured exchanges are assembled into a WACZ web archive (an open, standardized format), and a SHA-256 hash is computed over each artifact before any further handling. Hashing happens first, so every later step can be checked against the hashes taken at capture time.

Step 5. Sealing with a signed manifest

A manifest listing every artifact and its SHA-256 is generated and signed with a detached Ed25519 signature. The signing public key is published at a stable URL and on Arweave, so any third party can verify the seal with standard tools (OpenSSL) and without Permavault.

Step 6. Permanent storage

The artifacts, the manifest, and the signature are stored on the Arweave network, which is designed for permanence (a storage endowment funds ongoing replication across independent nodes). Each stored file receives a transaction ID under which anyone can retrieve it directly from the network, independently of Permavault.

Step 7. Independent time anchors

The manifest's SHA-256 is anchored in time by services independent of Permavault: an OpenTimestamps commitment to the Bitcoin blockchain, and, for Legal-tier packages, an RFC 3161 timestamp token issued by a time-stamping authority. Each anchor is verifiable offline with open tools, and the package states each anchor's status honestly (pending or confirmed).

Step 8. Event logging throughout

Each lifecycle step above appends an entry to the capture's hash-chained event log: every entry's hash covers both its own content and the previous entry's hash, so an edited or reordered entry is detectable by recomputation. The chain up to (but excluding) the sealing event is additionally anchored in the signed manifest itself, which records that chain's head hash and event count; even a wholesale rewrite of the early history, recomputed to look internally consistent, produces a different head and fails against the anchor. Because the manifest is signed and its hash is timestamped independently (step 7), the pre-seal history inherits those anchors. Events recorded after sealing (storage and anchor confirmations) are outside this commitment by design and are corroborated by their own independent records, such as Arweave transaction ids and timestamp anchor rows. The chain is exported with the Legal package as custody-log.json and rendered in the chain-of-custody appendix.

Human involvement

The pipeline is fully automated. The account holder chooses WHAT to capture and WHEN; Permavault's infrastructure performs the capture, hashing, sealing, storage, and anchoring without human intervention, and no Permavault personnel view or handle the captured bytes in the ordinary course of operation.

Alignment with ISO/IEC 27037

The process is designed to align with the principles of ISO/IEC 27037 (guidelines for the identification, collection, acquisition and preservation of digital evidence): identification (the requested source and capture environment are recorded), collection and acquisition (content is acquired by a neutral automated system and hashed at acquisition time, before any transformation), and preservation (artifacts are sealed under a signed manifest, stored on a network designed for permanence, and covered by an append-only custody log). Alignment with these principles is a design statement, not a certification by any accreditation body.