

Certificate of Authenticity

Permavault Archive

What this certifies

This certificate verifies the technical integrity and provenance of the capture. It does not assess the truth, accuracy, or legality of the captured content. Designed to support authentication under FRE 902(13) and 902(14). This document is not legal advice or a determination of admissibility.

Capture

Capture ID	4f868e92-f12f-4fb7-bb5a-cb096b82eac8
Page title	Permavault · Archive Anything. Permanently.
Source URL	https://permavault.xyz
Final URL (after redirects)	https://permavault.xyz/
Captured at (UTC)	2026-07-23T08:33:19.462Z
Time source	system
Stored at (UTC)	2026-07-23 08:33:23

Parties

Requested by	wassim.bendella@gmail.com
Captured by	Permavault neutral capture infrastructure, independent of the source website

Content hashes (SHA-256)

Web archive (WACZ)	ad093d193f31d4e6240fc06b6a12a752a9b7105b37d1a7178e0c9b76e268762b
Signed manifest	156c51a5b58a5afc8fc84cb870351f0df49b06440ad2835e169b79ebd5a227b7

Permanent record (Arweave)

Transaction ID	7iFluxfrjGu-BN6mdnfxUqQCCIB6z9rYNIjDMYokF3I
Permanent link	https://ar-io.dev/7iFluxfrjGu-BN6mdnfxUqQCCIB6z9rYNIjDMYokF3I

Prior versions of this URL

The capture certified by this document supersedes one earlier capture of the same URL made from the same account. Each prior version is a separate, independently stored and verifiable capture with its own content hash and permanent record. Listed oldest first.

Version 1

captured 2026-07-11T10:38:23.102Z (UTC), capture id
7cf7b4d8-4ff8-4cc1-9ffb-46fa88feeda8

sha256 ee7967d637be60792416f97a6f01a2566d038825387e4511979ee3b36559834e

arweave P6Qme_uh8u6Kk_w9D75q89AoHqqcoZ12AgjpbFraf8c
https://ar-io.dev/P6Qme_uh8u6Kk_w9D75q89AoHqqcoZ12AgjpbFraf8c

This section is included at the account holder's request. Its absence from another copy of a Permavault certificate does not indicate that prior versions do or do not exist.

Capture environment

Capture server IP	35.204.121.219
Browser user agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/149.0.7827.55 Safari/537.36
Capture software	Permavault 0.6.4, Scoop 0.6.59, Node 22.23.1
TLS certificate issuer	C=US, O=Google Trust Services, CN=WE1
TLS certificate subject	CN=permavault.xyz
HTTP status (final response)	200
Redirect hop 1 (200)	https://permavault.xyz/

Machine-readable proof

The manifest below lists every file in the capture package together with its SHA-256. One detached Ed25519 signature over the manifest seals the whole set. Anyone can verify it with the public key, which is published independently (see the verification steps).

Signature

Algorithm	ed25519
Signing key ID	98ae373420471711
Manifest SHA-256	156c51a5b58a5afc8fc84cb870351f0df49b06440ad2835e169b79ebd5a227b7
Signature (base64):	xjaKdgN80SJegQNC86MLvYv1yvG9VZBVl0fSk+1RF9b1QHXYNb3QRQMeIaIsjlncA0EQ8gy2eG5R/mJLfP0TDQ==

Signing public key

Ed25519 public key (SPKI, PEM):	-----BEGIN PUBLIC KEY----- MCowBQYDK2VwAyEAhHphwYb5zoexiY6Rl5tTmKBRwTR8SA7/kIBVh8BirtM= -----END PUBLIC KEY-----
Published at	https://app.permavault.xyz/.well-known/permavault-signing-key.json
On Arweave (tx)	y2rHSglMfqfkkVDT8OI3bv47QkJgUZ5Suw00SAumK_4
Public key link	https://ar-io.dev/y2rHSglMfqfkkVDT8OI3bv47QkJgUZ5Suw00SAumK_4

Bitcoin timestamp (OpenTimestamps)

Status	Pending Bitcoin confirmation (usually within about 24 hours)
The manifest hash has been submitted to the OpenTimestamps calendar servers. Once the Bitcoin attestation confirms, re-download this certificate and this section will show the block number and block time.	

Sealed artifacts

permavault.xyz.wacz	
role: primary	
sha256	ad093d193f31d4e6240fc06b6a12a752a9b7105b37d1a7178e0c9b76e268762b
arweave	7iFluxfrjGu-BN6mdnfxUqQCClB6z9rYNIjDMYokF3I
screenshot.png	
role: screenshot	
sha256	b08ad419bbeb4ba3a378685546f9dddd558960a0361deffec93b1a721a71b8f1

arweave xtS-Q87_q8RqvQ-ZAo-9MdDYqsNTLSz0Jorxhd5AvTw

pdf-snapshot.pdf

role: pdf-snapshot

sha256 9c3a7ce16206b3ecc4a954c2427a9585df95f4a5d1ce992ba3b59ccca0970d91

arweave gHonz5j2PGjPwa6f6ceQ7FPYdqDL4nXTeKssqS3Fa0A

exhibit.pdf

role: exhibit-pdf

sha256 f06a34a4f654d23e5cf3c469c5af58bb9f77552eafd518fa5cb4bf692968981a

arweave sgmQXG-5l33jHlXotv8adKEZKT0GfVY_26qTgAr0T2c

capture-metadata.json

role: metadata-sidecar

sha256 8acd3e9ca3f2f72ef2b1fc6c5712cf95da5d34334b50da6fb9924c22e56ced15

arweave m03sNQtsQixRF-j5GGn627vbNhd799Ck73ta8hnIvdQ

manifest.json

role: manifest

arweave DkLbCjkmMs4aD_qey5jfPf5vhvg2th13k7e58ifItec

manifest.sig

role: manifest-signature

arweave 4hWLdxFEZxp9dbduWFHdep5PuW9Wm3mf0wjNoUs5TuM

manifest.sig.json

role: manifest-signature-meta

arweave uEn2DtpmCaV0h9gqa4IN-8URk2z0UiftMTxuC8t_bEA

How to verify this capture without Permavault

Every claim in this certificate can be checked with standard, open-source tools. You do not need Permavault, and you do not need to trust it. First gather the capture files: manifest.json, manifest.sig, and the artifacts listed above, each downloadable from your Permavault history or directly from Arweave using the transaction ids on the proof page. Then run the steps below.

Step 1. Confirm the files have not been modified

Compute each file's SHA-256 and compare it to the value recorded in manifest.json:

```
shasum -a 256 <file>          # macOS / Linux
certutil -hashfile <file> SHA256 # Windows
```

Expected hashes for this capture:

ad093d193f31d4e6240fc06b6a12a752a9b7105b37d1a7178e0c9b76e268762b	permavault.xyz.wacz
b08ad419bbeb4ba3a378685546f9ddd558960a0361deffec93b1a721a71b8f1	screenshot.png
9c3a7ce16206b3ecc4a954c2427a9585df95f4a5d1ce992ba3b59ccca0970d91	pdf-snapshot.pdf
f06a34a4f654d23e5cf3c469c5af58bb9f77552eafd518fa5cb4bf692968981a	exhibit.pdf
8acd3e9ca3f2f72ef2b1fc6c5712cf95da5d34334b50da6fb9924c22e56ced15	capture-metadata.json

Step 2. Confirm the signature over the manifest

Save the Permavault public key to permavault-public-key.pem:

```
-----BEGIN PUBLIC KEY-----
MCowBQYDK2VwAyEAhHphwYb5zoeXiY6Rl5tTmKBRwTR8SA7/kIBVh8BirtM=
-----END PUBLIC KEY-----
```

Then verify the detached signature over manifest.json (OpenSSL 3.0 or newer):

```
openssl pkeyutl -verify -pubin -inkey permavault-public-key.pem \
  -rawin -in manifest.json -sigfile manifest.sig
```

A "Signature Verified Successfully" result confirms manifest.json was signed by Permavault key 98ae373420471711 and has not changed since. Because the manifest lists every file's hash, verifying it plus the hashes in Step 1 proves the whole package is intact.

Step 3. Confirm the permanent record exists

Check that the archive is retrievable from the Arweave network, independently of Permavault:

```
curl -I https://ar-io.dev/7iFluxfrjGu-BN6mdnfxUqQCClB6z9rYNIjDMYokF3I
```

The signing public key is also published at the URL and Arweave transaction listed on the proof page, so this certificate can be verified even if Permavault is offline.