

How one observation was made

SAMPLE. Written for a real observation of our own domain machinewitness.eu, with its actual parameters, by the same tool as every statement. It is published as an example and therefore not signed. A statement you order is written for the observation you name, read, and signed.

1. Subject of this statement

This statement describes how one observation was made: the address `https://machinewitness.eu/robots.txt` on the day **2026-09-28** (UTC), retrieved independently by 2 witnesses. It explains the method with the actual parameters of that observation, sets out each step so that a third party can repeat it without us, and names the objections usually raised against the method. It says nothing about the content that was retrieved or about any matter in which it is used.

The parameters below are read from the witnesses' packages and re-derived where possible; nothing in them was typed by hand.

WITNESS-1

Witness	witness-1
Retrieved at (UTC)	2026-09-28T03:12:53+00:00
Request	GET, no cookies, redirects followed, HTTP/1.1
Final address	https://machinewitness.eu/robots.txt
HTTP status	200
TLS	TLSv1.3 TLS_AES_256_GCM_SHA384
Certificate subject	CN=machinewitness.eu
Certificate issuer	C=US, O=Google Trust Services, CN=WE1
Certificate valid	Sep 20 12:45:20 2026 GMT to Dec 19 13:45:17 2026 GMT
Certificate chain (SHA-256 of each DER)	aec098364a0137418a01ed8935ac9d0070ffb370429a16df8b70a8cd73cd5e53 · 1dfc1605fbad358d8bc844f76d15203fac9ca5c1a79fd4857ffaf2864fbeb96 · 76b27b80a58027dc3cf1da68dac17010ed93997d0b603e2fadbe85012493b5a7
Completeness	Stored in full; no byte cap applied to this observation.
SHA-256 served bytes	ae6ef8da7be4397f9d37a57be7b522f3f2b84d2b6af133ba26e8379acab36e32
SHA-256 of the provenance	5819ce2ac282b3651477abf7db916cf1f3afd222a8aa3a0abef9fa9c580dc9c6
Leaf	c6ae3444c00b163a044151ea59f634a6b0b39a4a3e29ca5b57f61abab78ec1ca
Position in the day's tree	index 44,025 of 642,761 leaves, 20 steps to the root
Day root	f1437269e4708b3dcc32a0c2947bdec58c1dbe084739ff912f14a03cacad48a6
Anchors of the day root	RFC 3161 time-stamp (FreeTSA) · OpenTimestamps (Bitcoin) · qualified eIDAS time-stamp (GLOBALTRUST)
Package	machinewitness-extract-witness-1-2026-09-28-d0199b52c7cc.zip, SHA-256 3188c88a114c739d32122f7a89c6f4fa06c96c70eb1105886a2c4710c5633fb7

WITNESS-2

Witness	witness-2
Retrieved at (UTC)	2026-09-28T03:01:20+00:00
Request	GET, no cookies, redirects followed, HTTP/1.1
Final address	https://machinewitness.eu/robots.txt

HTTP status	200
TLS	TLSv1.3 TLS_AES_256_GCM_SHA384
Certificate subject	CN=machinewitness.eu
Certificate issuer	C=US, O=Google Trust Services, CN=WE1
Certificate valid	Sep 20 12:45:20 2026 GMT to Dec 19 13:45:17 2026 GMT
Certificate chain (SHA-256 of each DER)	aec098364a0137418a01ed8935ac9d0070ffb370429a16df8b70a8cd73cd5e53 · 1dfc1605fbad358d8bc844f76d15203fac9ca5c1a79fd4857ffaf2864fbeb96 · 76b27b80a58027dc3cf1da68dac17010ed93997d0b603e2fadbe85012493b5a7
Completeness	Stored in full; no byte cap applied to this observation.
SHA-256 served bytes	ae6ef8da7be4397f9d37a57be7b522f3f2b84d2b6af133ba26e8379acab36e32
SHA-256 of the provenance	a20e01fd655b9927a077d667fbcf68afe2a2600552863b6ac8ae84297974a730
Leaf	10f22b28b520a9ea35e4e1be81f525f7aa04035157b9d1cb22ced67375f3fd52
Position in the day's tree	index 24 of 586,787 leaves, 20 steps to the root
Day root	d568805980fc63d4fbd54959ec16efda605b13fc1cae54b88fb94a7ebdc55d
Anchors of the day root	RFC 3161 time-stamp (FreeTSA) · OpenTimestamps (Bitcoin) · qualified eIDAS time-stamp (GLOBALTRUST)
Package	machinewitness-extract-witness-2-2026-09-28-d0199b52c7cc.zip, SHA-256 d69cce6fbdda19d2a951419677ace17a25d2e079ff7799e2fea49ef21e5ed5f6

Both witnesses received the same bytes: the SHA-256 of the served bytes is identical in both packages. Their leaves differ, as they must: each leaf also binds the witness's own time of retrieval and its own provenance record.

2. Retrieval

Each witness is a separate machine at a separate provider in a separate country, with its own keys. Neither can write to the other. Each retrieves the address with a plain HTTP GET request, identifying itself with the published user agent `MachineWitnessBot/0.1 (+https://machinewitness.eu/bot; evidence archive; contact: bot@machinewitness.eu)`, sending no cookies and following redirects; the address that finally answered is recorded as the final address. The witness records the status, all response headers, the TLS version and cipher, the server certificate and the SHA-256 of every certificate in the chain presented, and, where recorded, its own address and the address that answered. It does not log in, click, fill in anything or run the page's scripts. What is recorded is therefore what this address served to this request at this moment, not what it may have shown to another visitor at another moment.

3. Storage and fingerprint

The bytes served are stored as received. Their SHA-256 fingerprint is computed over exactly those bytes. Whether the stored bytes are complete is itself recorded in the provenance (truncated), and the completeness line above repeats it. The provenance record (final address, headers, TLS data, certificate hashes, addresses) is serialised as compact JSON with sorted keys and fingerprinted in turn.

4. The leaf

The observation is reduced to one value, the leaf:

```
leaf = SHA-256( 0x00 || "mwleaf-v1|" url "|" observed_at "|" payload_sha256 "|" status "|"
provenance_sha256 )
```

The prefix byte `0x00` separates leaves from inner nodes of the tree, so that no inner node can be presented as a leaf. For this statement the leaf of each witness was recomputed from its recorded fields and matches the leaf in the package.

5. The day's tree and root

Every observation a witness makes on a UTC day becomes one leaf of that day's Merkle tree. Inner nodes are `SHA-256(0x01 || left || right)`. The single value at the top, the day root, changes if any byte of any observation of that day changes. The package contains the path from this leaf to the root: the sibling hashes and their side. Folding them in order reproduces the root. The root is published in the public log of this archive under a fixed address for the day.

6. Anchors

The same night the day root is handed to parties outside this archive: an RFC 3161 time-stamp authority, the OpenTimestamps calendars, which aggregate it into the Bitcoin blockchain, and, since 31 July 2026, GLOBALTRUST (e-commerce monitoring GmbH, Austria), a qualified trust service provider on the EU Trusted List, which issues a qualified electronic time-stamp. Each anchor states that the root existed at the time it names. After that, a root cannot be replaced without the anchors no longer matching. The qualified time-stamp is the one that carries the presumption laid down in Article 41(2) of Regulation (EU) No 910/2014 as to the accuracy of its date and time and the integrity of the data it is linked to.

7. Two witnesses

The second witness makes its own retrieval, builds its own tree and takes its own anchors. The two roots are not identical and are not meant to be: each witness observes on its own schedule and records its own time of retrieval. What they allow is a comparison: for the same address on the same day, each witness's package shows what that witness received, and the two can be held against each other. Both witnesses received the same bytes: the SHA-256 of the served bytes is identical in both packages. Their leaves differ, as they must: each leaf also binds the witness's own time of retrieval and its own provenance record.

8. Repeating the steps without us

1. **Bytes.** `sha256sum payload-<id>.bin` gives the value in `leaf-<id>.json` as `payload_sha256`.
2. **Provenance.** `sha256sum provenance-<id>.json`, the file exactly as it is, gives `provenance_sha256`.
3. **Leaf.** The formula in section 4, with the five fields from `leaf-<id>.json`, gives `leaf`. Any tool that computes SHA-256 will do.
4. **Root.** Fold the path in `proof-<id>.json` as described in section 5; the result equals the root in `<day>.root.txt` and in the public log.
5. **Anchors.** `openssl ts -verify` for the two RFC 3161 tokens against the issuers' certificate chains; `ots verify` for the OpenTimestamps proof.

Each package carries these steps in a file `VERIFY.txt`. None of them needs software from this archive.

9. Objections usually raised, and the technical answer

The operator could have changed the record afterwards.

A change to any byte changes the leaf, the root and the day's anchors. The anchors were issued the night the day was sealed and are held by third parties. A changed record no longer matches them, and this can be

checked without us (section 8, steps 4 and 5).

The record shows what the witness chose to store, not what the server sent.

The bytes are stored as received and fingerprinted before anything else happens to them. Completeness is recorded as a sealed field, not stated afterwards. Where a byte cap applied, the record says so for that observation.

The server may have shown other visitors something else.

That is correct and it is stated, not hidden: the record covers what this address served to this request, identified by the published user agent, at the moment given. It makes no statement about other requests.

The time of retrieval rests on the witness's own clock.

The time of retrieval is recorded by the witness. The anchors independently fix the latest point by which the whole day, this observation included, existed. The qualified time-stamp states its time under the supervision described in section 6.

One machine could be wrong.

There are two, at different providers, with separate keys, and neither can write to the other (section 7).

10. What this statement is not

It contains no statement about any matter, no assessment of anyone's conduct and no view on the prospects of a case. It describes a procedure and the parameters of one observation. What follows from it is for the parties and the court to decide.

Part B · Written technical answers, round 1

Questions received in writing about the observation above, answered in writing. The answers cite the published method and say nothing about the matter.

The extract gives two different day roots for 28 September 2026, one per witness. Which of the two is the correct one?

Both. Each witness builds its own tree over its own observations of that day and anchors its own root; the two trees contain different sets of leaves, recorded at different times, so their roots differ by construction. A root is checked against the log and the anchors of the witness that produced it. What the two witnesses have in common is not the root but the observation: for this address both packages carry the same SHA-256 of the served bytes.

How can it be checked that robots.txt was not altered after it was stored?

By recomputing, without us: the SHA-256 of payload-`<id>.bin` must equal payload_sha256 in the leaf file; the leaf must recompute from its five fields (section 4); the path in proof-`<id>.json` must fold to the day root; and the day root must match the anchors issued the same night (section 8). If any byte had been changed after 28 September 2026, one of these four checks would fail.

Place, date

Martin Schenk, for MachineWitness (Martin Schenk S.L., Madrid)

SAMPLE, not signed.