LINTTRUST COMPARISON REPORT

LintTrust Versus zlint and pkilint: What the Comparison Showed

Published 28 September 2026. The comparison was run on 26 September 2026.

A read-only review compared LintTrust with the two linters the WebPKI uses every day, on the same certificates, and kept its scripts and raw outputs. This report gives the commercial argument with the proof beside each claim. Certificates are named by their public subject or not at all, and no natural person appears.

1. What Was Compared, and How

The two linters, **zlint 3.7.2** and **pkilint 0.13.3** (its CABF serverauth module, which declares Baseline Requirements 2.2.5), were run on the exact DER bytes LintTrust stores. They ran on two corpora LintTrust had judged against WTCA TLS BR 2.2.5 (current edition TLS BR 2.3.0, no audit period). The first is a reference sample of 116 files and 85 live certificates: public roots, Let's Encrypt and ISRG chains, badssl test certificates, a national CA's certificates, and public TLS certificates. The second is a set of 17 case files, 23 live after retrieval.

Every BR section named on a LintTrust evidence row (65 sections) was mapped to the lints that cite it, and each pair of a tool, a certificate, and a section was classified as an agreement, a disagreement, or out of scope. That made **14,040 pairs** in all. The **424 disagreements** were then read one by one against the BR text by the reviewer, each with the certificate, the BR row, LintTrust's Expected and Found, the linter's finding, and a decision.

The first thing the comparison shows is a difference of nature. **10,040 of the 14,040 pairs are outside a population or an applicability**, because a standalone linter judges every certificate as a TLS certificate, while LintTrust reads BR 1.1 first (which certificates the Baseline Requirements bind at all) and answers “not applicable, and why” on the others. A linter gives lines. LintTrust gives a WTCA verdict with its population, its edition, and its reason.

2. The Arguments, with the Proof

Argument 1: LintTrust Finds Non-Conformities the Linters Cannot See

There are **25 pairs** where LintTrust fails a certificate on a mandatory BR row and neither linter fires, because the check needs what a standalone linter does not have (the issuer's key, the workspace, the certificate's declared type) or is simply not implemented.

BR section	What LintTrust found	Why a standalone linter is silent
7.1.3.2 and 7.1.3.2.2 (signature algorithm bound to the signing key)	An ECDSA signature with SHA-256 made by a P-384 key, where the BR requires SHA-384 for P-384: 22 certificates under one issuing CA of a national CA on the full corpus, several in the sample.	The rule binds the hash to the **issuer's** curve. The linter sees one certificate and does not hold the issuer's key. LintTrust holds the chain.
7.1.4.2 (subject attributes: the table's order, string types, and lengths)	The attributes encoded in an order the table forbids (commonName before countryName) on five root and issuing CAs of a national CA.	The attribute order rule of 7.1.4.2 is not a lint in either tool.
7.1.2.7.2 (the Domain Validated subject profile)	An organizationalUnitName (“Domain Control Validated”, “PositiveSSL”) in a DV certificate, which the DV profile forbids.	The profile check needs the certificate's type (DV, OV, or EV), which LintTrust reads from the policy OID and the auditor's declaration.
7.1.2.11.2 (CRL distribution points)	A subscriber certificate with no cRLDistributionPoints and no OCSP pointer, and not Short-lived.	The exemption depends on the Short-lived definition and the OCSP pointer together. Neither linter fires.
4.2.2 (internal names)	`old-system.linttrust.test`: a name under a TLD that is not in IANA's Root Zone Database.	LintTrust reads the IANA registry. zlint's internal-name lint does not cover this case.
7.1.2.3.1 (the technically constrained TLS Subordinate CA's extension table)	Extensions present or absent against the table of the CA's profile.	The profile of a subordinate CA (7.1.2.3 to 7.1.2.6) depends on what the CA is, which LintTrust decides from the chain and the declaration.

Argument 2: The Linters Make Mistakes LintTrust Does Not Make

There are **14 pairs** where pkilint rejects, as an error, a subjectAltName that the BR explicitly permits: fourteen wildcard names of public sites. pkilint's generic PKIX domain-name syntax refuses the asterisk, while BR 7.1.2.7.12 permits a validated Wildcard Domain Name. LintTrust passes them, and the reviewer confirmed it against the text. A CA using pkilint alone would chase fourteen false non-conformities.

There are **6 pairs** where pkilint applies a subscriber-profile rule (the prohibition of duplicate attribute types) to historical roots (VeriSign, Microsoft), or classifies a malformed subscriber certificate as a root and judges it against the root validity rule instead of the subscriber's 200-day limit. LintTrust reads the certificate's type from the chain and applies the right table.

Argument 3: The Linters Report Recommendations as Findings, While LintTrust Keeps the Distinction and Lets the Auditor Choose

There are **195 pairs** where a linter emits a warning or an error on a sentence the BR writes as RECOMMENDED, NOT RECOMMENDED, SHOULD, or MAY: a commonName present in a subscriber certificate (35 zlint warnings, 33 pkilint), a subjectKeyIdentifier present in a subscriber certificate (69), both digitalSignature and keyEncipherment set on an RSA key (34), an authorityKeyIdentifier absent from a root (9), a policy qualifier present, the reserved policy not first, and a subordinate CA's AIA without a calssuers URL. None of these is a non-conformity under the Baseline Requirements.

LintTrust records each of them on the certificate's row, marked as the recommendation it is, and the criterion stays compliant. The working paper lists them apart as observations on recommendations. And because a firm may want to hold a CA to the recommendations, **Strict analysis** turns every breached recommendation into a failure, deliberately and workspace-wide, with the auditor's acceptance where the auditor decides otherwise. The auditor chooses the standard to apply. A linter makes that choice for the auditor.

Argument 4: LintTrust Reads BR Sections No Linter Reads at All

Of the 65 BR sections LintTrust's evidence names, **28 have no lint in zlint and 28 have none in pkilint** (not the same 28), and 970 of the 14,040 pairs had no candidate on the linter's side at all. Among them are 6.1.7 (what a Root CA's private key may sign, which needs the whole PKI), the attribute order and string types of 7.1.4.2, and most of the per-type CA profile tables (7.1.2.2 to 7.1.2.6: the cross-certificate, the technically constrained TLS and non-TLS subordinate CAs, and the Precertificate Signing CA). They also include the SHA-1 case's conditions of 7.1.3.2.1 (which compares the new certificate with an existing one of the same issuer) and the name constraints of 7.1.2.10.8 read together with 7.1.2.5.2.

A linter checks a certificate's syntax against the rules it can see in isolation. LintTrust reads the profile tables of the Baseline Requirements as the auditor reads them, on the certificate in its PKI. That is where the analysis goes further, and it is also where a comparison cannot vouch for it: those sections are verified by reading sheets against the text and by probes written from the text, not by another tool.

The Scope of This Comparison, Said Plainly

The comparison covers 108 certificate occurrences (the two corpora) and the 37 BR sections of the 65 for which at least one linter has a lint. For the 28 sections per tool that have none (6.1.7, the attribute order of 7.1.4.2, most of the per-type CA profile tables), the comparison proves nothing either way.

No certificate of the corpora exercises the SHA-1 closing date of 2026-09-15, the reverse-zone rule of 2026-03-15, the 100-day and 47-day validity steps, an EV profile, a Precertificate Signing CA, or an IDN name, so those rules were not compared. The composition rules that have no equivalent in a linter (the BR 1.1 population, the two editions, the observation, and the auditor's appreciation) and the chain evaluation were not compared either. A second comparison is planned on a purpose-built PKI (one certificate per rule, each dated rule exercised on both sides of its date) and on a larger corpus of a national CA.

3. What the Comparison Found Against LintTrust, and What Was Done

Honesty is part of the argument. Of the 424 disagreements, **3 were LintTrust's error**. One rule, the root certificate's validity (7.1.2.1.1), was counted in whole days, dropping the final second, so two roots whose validity is exactly 9,132 days and one second read “9132 days, pass” instead of failing the 25-year maximum. pkilint counted the second and was right. The defect was found by this comparison on 2026-09-26 and corrected the same day: one definition of the Validity Period for the whole engine, inclusive to the second, with boundary tests, and the affected verdicts recomputed and recorded. **No other false compliant was established on any of the 65 sections.**

The rest of the 424 are these: 100 pairs where a LintTrust composite (a whole profile table) fails on one of its rows and the linter's lint maps to the parent, not the child, which is not a disagreement. There are 55 where the section number matches but the sub-requirement differs, and 26 where the linter and LintTrust apply a different edition or effective date (mostly roots issued before the oldest edition LintTrust holds, which LintTrust judges against that edition and says so). The last 6 are the profile differences above.

Linters remain useful complements, faster and broader on PKIX syntax. LintTrust does not claim to cover every BR rule: it evaluates the 15 WTCA criteria a certificate can decide, and says so.

4. The Evidence

The mapping of the 65 sections to the lints, the 424 adjudications with their rationale, the counts, the per-certificate outputs of zlint and pkilint, LintTrust's evidence, and the scripts that rerun the whole comparison are available on request.