NORMATIVE

`T = I  det = +1  (x, y) → (x, y)`

“This is the standard.”

RFC 2119 MUST / SHALL

DNS Tool

Confidence-Scored Analysis of Domain Security Infrastructure

Carey James Balboa

Independent DNS Security Researcher

ORCID 0009-0000-5237-9065
DOI 10.5281/zenodo.19468134

PROJECT dnstool.it-help.tech **SOURCE** github.com/IT-Help-San-Diego/dns-tool-intel

VERSION 26.46.14 · **LICENSE** BUSL-1.1

Independent research artifact accompanying the DNS Tool software platform.

Abstract

DNS Tool is an open-core OSINT platform designed to analyze domain security posture using RFC-compliant signals from DNS and email authentication infrastructure. The system collects DNS records, evaluates configuration compliance with relevant RFC standards, and applies a confidence-scored interpretation model to produce structured security intelligence outputs. Unlike traditional scanners that report raw DNS results, DNS Tool emphasizes confidence scoring and reproducibility, enabling analysts to distinguish between verified security signals, ambiguous observations, and unsupported conclusions.

The platform focuses on five major areas: email authentication (SPF, DKIM, DMARC), transport security (MTA-STS, DANE), DNS integrity (DNSSEC), brand protection indicators (BIMI), and domain configuration analysis (CAA, TLS-RPT). This document describes the methodology used by DNS Tool to transform raw DNS observations into structured intelligence outputs.

1. Problem Statement

Modern domain security analysis is fragmented across multiple DNS-based mechanisms defined by different RFC specifications: SPF (RFC 7208), DKIM (RFC 6376), DMARC (RFC 7489), DNSSEC (RFC 4033–4035), MTA-STS (RFC 8461), DANE for SMTP (RFC 7672), BIMI (RFC 9495), TLS-RPT (RFC 8460), and CAA (RFC 8659).

Most existing tools present raw DNS data without distinguishing between valid security signals, partial configurations, ambiguous results, and misconfigurations. This creates a common problem for analysts: interpretation uncertainty.

DNS Tool was developed to address this problem by introducing a structured evaluation process that: (1) collects DNS evidence, (2) verifies RFC compliance, and (3) applies a confidence model to interpretation.

2. Data Collection

2.1 DNS Query Process

DNS Tool collects records by querying multiple upstream resolvers (including Google Public DNS, Cloudflare, and Quad9) and comparing responses for consistency. The multi-resolver approach improves confidence by detecting resolver-specific anomalies such as caching artifacts, SERVFAIL responses, or DNSSEC validation failures.

All queries use standard DNS protocols over UDP/53 and TCP/53. DNS-over-HTTPS (DoH) is used as a secondary validation channel when available.

2.2 Record Types Collected

For each domain analyzed, DNS Tool collects and evaluates:

- **A / AAAA**: Host address records
- **MX**: Mail exchanger records
- **TXT**: SPF, DMARC, DKIM, BIMI, MTA-STS, TLS-RPT policies
- **TLSA**: DANE/TLSA certificate association records
- **CNAME**: Canonical name records (indirect hosting detection)

- **NS**: Nameserver delegation records
- **SOA**: Start of Authority records
- **CAA**: Certificate Authority Authorization records
- **DNSKEY / DS / RRSIG**: DNSSEC validation chain records

2.3 Evidence Preservation

All collected DNS responses are preserved with timestamps, resolver identification, TTL values, and response codes. This evidence chain enables reproducibility and audit trail verification.

3. Analysis Methodology

3.1 Protocol-Specific Evaluation

Each protocol is evaluated against its defining RFC specification. The evaluation produces three outputs:

1. **Finding**: A factual observation about what was found
2. **Compliance Status**: Pass, Fail, Warn, or Info relative to the RFC
3. **Confidence Score**: How certain the tool is about the interpretation

3.2 SPF Analysis (RFC 7208)

SPF evaluation checks: presence of a valid SPF TXT record, syntax validation against RFC 7208, mechanism count (10-lookup limit enforcement), include chain resolution and depth analysis, qualifier analysis (pass, fail, softfail, neutral), and duplicate record detection.

3.3 DKIM Analysis (RFC 6376)

DKIM evaluation checks: public key record presence for known selectors, key type and size validation (RSA minimum 1024-bit, recommended 2048-bit), selector discovery using common selector patterns, and key rotation indicators.

3.4 DMARC Analysis (RFC 7489)

DMARC evaluation checks: policy record presence and syntax, policy strength (none, quarantine, reject), alignment modes (strict vs. relaxed) for SPF and DKIM, reporting URI validation (rua, ruf), subdomain policy (sp) analysis, and percentage (pct) field evaluation.

3.5 DNSSEC Validation (RFC 4033–4035)

DNSSEC evaluation checks: presence of DNSKEY, DS, and RRSIG records, signature validity and expiration, algorithm identification, chain of trust from root to domain, and NSEC/NSEC3 presence for authenticated denial of existence.

3.6 MTA-STS Analysis (RFC 8461)

MTA-STS evaluation checks: DNS TXT record presence (_mta-sts.domain), policy file retrieval via HTTPS, policy mode (enforce, testing, none), MX host matching against policy, and policy max_age validation.

3.7 DANE/TLSA Analysis (RFC 7672)

DANE evaluation checks: TLSA record presence for MX hosts, certificate usage field validation, selector and matching type verification, DNSSEC requirement verification (DANE requires DNSSEC), and cross-reference with actual TLS certificates.

3.8 BIMI Analysis (RFC 9495)

BIMI evaluation checks: BIMI TXT record presence, SVG logo URL validation, VMC (Verified Mark Certificate) presence, and DMARC policy requirement verification (BIMI requires DMARC enforcement).

3.9 CAA Analysis (RFC 8659)

CAA evaluation checks: CAA record presence, authorized certificate authority listing, wildcard policy analysis, and iodef (incident reporting) configuration.

3.10 TLS-RPT Analysis (RFC 8460)

TLS-RPT evaluation checks: TLS-RPT TXT record presence, reporting URI syntax validation, and version field verification.

4. Confidence Scoring Model

4.1 Intelligence Confidence Audit Engine (ICAE)

DNS Tool applies a confidence scoring model inspired by intelligence community analytic standards (ODNI ICD 203). Each protocol finding receives a confidence level:

- **Gold Master**: Sustained correctness across 5,000+ consecutive passes over 180+ days — the highest maturity tier
- **Gold**: High correctness with 500+ consecutive passes over 90+ days
- **Consistent**: Reliable correctness with 100+ consecutive passes over 30+ days
- **Verified**: Correctness demonstrated with 10+ consecutive passes over 7+ days
- **Development**: Insufficient test history to establish confidence — the initial tier for all new protocols

4.2 Confidence Calibration

Confidence calibration uses a reliability-weighted shrinkage estimator. For each protocol: (1) protocol-specific priors — empirically determined base rates encoding historical detection reliability per protocol; (2) resolver agreement ratio — the fraction of queried resolvers that return consistent results, used as measurement quality weight; (3) shrinkage toward prior — when resolver agreement is low, the calibrated score is pulled toward the prior mean; when agreement is high, the observation dominates. This produces a calibrated confidence score per protocol, distinct from the raw detection score. The calibration formula is a shrinkage estimator — not a true Bayesian posterior (see EDE-006 for the correction history on this distinction).

4.3 Overall Posture Score

The overall domain security posture is determined by the unified confidence engine, which combines ICAE accuracy scores, ICuAE currency scores, and ICAE maturity level. The maturity level imposes a ceiling on the maximum achievable confidence — a protocol in the “development” tier cannot reach full confidence regardless of its accuracy score. The weakest protocol dimension determines the overall confidence level.

4.4 Epistemic Correction Disclosure

When structural corrections to the confidence model are identified — such as recalibrated scoring weights, reinterpreted evidence thresholds, or corrected RFC compliance mappings — the system records these as Epistemic Disclosure Events (EDEs). Each EDE documents the original assessment, the correction applied, the confidence impact, and the verifiable commit reference. This practice is modeled on scientific corrigenda culture: corrections strengthen rather than undermine analytical credibility, provided they are transparent, traceable, and independently verifiable.

4.5 Calibration Validation

The ICAE confidence scoring model is empirically validated through a calibration framework that measures the statistical reliability of predicted confidence levels against observed outcomes.

Test Corpus: 129 golden test cases are evaluated across 5 resolver scenarios (Google, Cloudflare, Quad9, authoritative, and mixed-resolver), producing 645 individual predictions per calibration run.

Calibration Metrics:

- **Brier Score**: 0.0018 (excellent). The Brier Score measures the mean squared error between predicted confidence probabilities and actual outcomes. Values closer to 0 indicate better calibration; the ICAE score of 0.0018 demonstrates near-perfect probability estimation.
- **Expected Calibration Error (ECE)**: 0.031 (good). ECE measures the weighted average gap between predicted confidence and observed accuracy across probability bins. An ECE of 0.031 indicates that predicted confidence levels closely match empirical correctness rates.

Methodology: The calibration framework employs a shrinkage estimator that blends observed per-bin accuracy with the global base rate, regularized toward conservatism. This approach prevents overconfident predictions in low-sample bins while preserving sensitivity in well-populated confidence ranges.

Conclusion: The ICAE confidence model is conservatively calibrated — when the system reports high confidence, findings are correct at or above the stated rate. This conservative bias is an intentional design choice aligned with intelligence community analytic standards (ODNI ICD 203), where understating confidence is preferable to overstating it.

5. Output Products

5.1 Engineer's DNS Intelligence Report

A detailed technical report containing: per-protocol findings with evidence, RFC compliance status for each configuration, confidence scores with supporting rationale, remediation recommendations, and Big Picture Questions for strategic consideration.

5.2 Executive's DNS Intelligence Brief

A summarized report designed for non-technical stakeholders: overall security posture assessment, risk-prioritized findings, business impact analysis, and strategic recommendations.

6. Implementation Architecture

The DNS Tool system consists of three major components:

Web interface: Provides interactive domain analysis and visualization.

Analysis engine: Processes DNS records and performs RFC validation.

Supporting intelligence modules: Generate structured intelligence outputs from analysis results.

The implementation is written primarily in Go for the analysis engine with a web-based interface for user interaction. The system is designed to allow independent verification of DNS observations.

Core research logic and internal analysis pipelines are maintained in private repositories for security and intellectual property protection.

7. Reproducibility and Limitations

DNS Tool is designed for reproducible analysis. All DNS queries are logged with timestamps and resolver identification. Analysis logic is deterministic for a given set of DNS responses. The software is version-controlled with semantic versioning. This methodology document is versioned alongside the software. The software artifact is archived with a persistent DOI.

- The confidence scoring model is empirically calibrated against 129 golden test cases across 5 resolver scenarios (645 predictions), with calibration quality measured via Brier Score and Expected Calibration Error (see Section 4.5)

7.1 Epistemic Correction and Integrity Verification

DNS Tool maintains a public Epistemic Disclosure Event (EDE) register that documents all structural corrections to the confidence scoring model. Each EDE entry records the category of correction (e.g., scoring calibration, evidence reinterpretation, standards misattribution), the severity, the specific confidence impact, and a verifiable git commit hash linking to the exact code change.

To ensure the integrity of this correction record, DNS Tool computes SHA-3-512 cryptographic hashes at two levels:

1. **File-level hash**: A SHA-3-512 hash of the complete EDE register file (`integrity_stats.json`), independently verifiable via: `openssl dgst -sha3-512 static/data/integrity_stats.json`
2. **Per-event hash**: Each individual EDE entry receives its own SHA-3-512 hash computed from its JSON representation, enabling detection of single-entry tampering independently of other entries.

Published EDE entries are governed by a tamper resistance policy that permits amendments only on two explicitly declared grounds: factual error (with verifiable evidence) or dignity of expression (phrasing-only, with all factual fields locked). This framework is tamper-evident rather than tamper-proof — it is designed to make unauthorized modification detectable, not physically impossible. Full policy details, amendment records, and attack vector analysis are published as supplementary documentation on the project's EDE page.

7.2 Limitations

DNS Tool operates exclusively on publicly available DNS information. As a result, it cannot evaluate internal email infrastructure, private key security, or server-side enforcement mechanisms. The tool focuses on observable infrastructure posture rather than complete operational security evaluation. DNS observations may change over time, meaning results represent the configuration state at the time of analysis.

- DKIM analysis is limited to known selectors unless additional selectors are provided
- DNSSEC validation depends on resolver support and may vary across network environments
- Results represent a point-in-time snapshot; DNS configurations change frequently
- The confidence model is heuristic-based and may not capture all edge cases

Companion Artifact

The communication architecture and philosophical foundations underlying the Five Perspectives model, Socratic verification workflow, and narrative architecture are documented in a separate companion paper:

- Balboa, C. J. (2026). Philosophical Foundations for Security Analysis Communication. Available at: dnstool.it-help.tech/foundations

8. References

- RFC 7208 — Sender Policy Framework (SPF)
- RFC 6376 — DomainKeys Identified Mail (DKIM) Signatures
- RFC 7489 — Domain-based Message Authentication, Reporting, and Conformance (DMARC)
- RFC 4033, 4034, 4035 — DNS Security Extensions (DNSSEC)
- RFC 8461 — SMTP MTA Strict Transport Security (MTA-STS)
- RFC 7672 — SMTP Security via Opportunistic DNS-Based Authentication of Named Entities (DANE)
- RFC 9495 — Brand Indicators for Message Identification (BIMI)
- RFC 8659 — DNS Certification Authority Authorization (CAA)
- RFC 8460 — SMTP TLS Reporting (TLS-RPT)
- ODNI ICD 203 — Analytic Standards (Intelligence Community Directive)

Citation

If DNS Tool contributes to research or analysis, please cite:

```
@software{balboa2026dnstool,
  author       = {Balboa, Carey James},
  title        = {{DNS Tool}: Domain Security Audit Platform},
  year         = {2026},
  version      = {26.46.14},
  doi          = {10.5281/zenodo.19468134},
  url          = {https://dnstool.it-help.tech},
  license      = {BUSL-1.1}
}
```

OWL SEMAPHORE SYSTEM — CLASSIFICATION LEDGER

NORMATIVE

$T = I$ det = +1

$(x, y) \to (x, y)$

"This is the standard."

RFC 2119 MUST / SHALL

NON-NORMATIVE

$T = \sigma_v$ det = −1

$(x, y) \to (-x, y)$

"This reflects the standard."

Informative / Advisory (NOTE)

CRITICAL

$T = C_2$ det = +1

$(x, y) \to (-x, -y)$

"This inverts the standard."

RFC 2119 MUST NOT / SHALL NOT

METACOGNITIVE

$T = \sigma_h$ det = −1

$(x, y) \to (x, -y)$

"This audits the standard."

Observer audit / Frame inversion

KLEIN FOUR-GROUP V_4 — CLOSURE IN O(2)

$I \cdot \sigma_v \cdot C_2 \cdot \sigma_h$ | $\sigma \circ \sigma = I$ · $C_2 = \sigma_v \circ \sigma_h$ · Every element is its own inverse

DNS Tool v26.46.14 · dnstool.it-help.tech · DOI: 10.5281/zenodo.19468134

Related documents: Philosophical Foundations · Founder's Manifesto · Communication Standards · Reference Library · Sources & Citations