PQCAT™: A Post-Quantum Cryptography Compliance Assessment Tool

Discovery, Risk Quantification, and Verifiable Evidence for the PQC Migration

J. Casey Wilson
Founder & CEO, Soqucoin Labs Inc.
labs@soqu.org

July 2026 (Revision 1)
Describes PQCAT v2.8

Abstract

Every federal agency and most regulated enterprises now carry a written obligation to inventory their cryptography and migrate it to post-quantum algorithms. Almost none of them can answer the first question that obligation implies: where, exactly, is the vulnerable cryptography? PQCAT is a cryptographic discovery and compliance assessment tool built to answer that question with evidence. It scans TLS endpoints, SSH services, software bills of materials, PKI stores, source code (564 detection patterns across more than 40 file types), service configurations, HSM and KMS interfaces, SCAP content, container images, and AWS cloud services; classifies every finding against the quantum threat; and converts the results into a defensible 0–100 compliance score under any of eleven regulatory frameworks, from CNSA 2.0 and OMB M-23-02 to PCI DSS 4.0 and HIPAA. Two patent-pending techniques distinguish it: a harvest-now-decrypt-later (HNDL) risk engine that scores findings by how long the data they protect must outlive the arrival of a cryptanalytically relevant quantum computer, and a Confidential Compliance Engine that lets an organization prove its compliance posture to an auditor or regulator without disclosing the location of a single vulnerable asset. PQCAT ships as a single static binary with zero C dependencies, in a free air-gapped edition whose absence of network code is enforced by the compiler, and in a licensed Pro edition adding a REST API, role-based dashboard, SIEM export, and ML-DSA-signed reports. It is built by the team that performed one of the first production migrations of a blockchain from ECDSA to NIST FIPS 204 ML-DSA signatures.

Foreword

I spent over a decade building automated vulnerability and compliance assessment systems for federal environments, including FedRAMP and DoD authorization work, before I ever touched a lattice. What that decade taught me is that compliance programs fail at inventory. Not at policy, not at intent: at knowing what you actually have. The post-quantum migration is the largest cryptographic inventory problem ever mandated, and most organizations are approaching it with spreadsheets.

I also learned the problem from the other side. In 2025 my team removed ECDSA from a production blockchain and replaced it with FIPS 204 ML-DSA signatures end to end, through an independent security audit. We had to find every place classical cryptography lived in a codebase of consequence, decide what mattered first, and prove the result to auditors. PQCAT is that experience, generalized into a tool.

Two principles from that work are load-bearing here. First, findings without prioritization are noise: an RSA certificate protecting data that expires next week and an RSA key wrapping records that must stay confidential for 25 years are not the same finding, and a tool that scores them the same is lying to you. That is what the HNDL engine is for. Second, the most security-conscious customers are precisely the ones who cannot show you their findings. An agency cannot hand a vendor, or sometimes even its own auditor, a map of its weakest cryptography. The Confidential Compliance Engine exists so the score can travel without the map.

If you evaluate tools for a living, the reproducibility notes in this paper are for you. Where a claim can be checked against the open-source scanner code, I say so. Where a capability is proprietary, I describe what it does and does not do. And where something is roadmap rather than product, it is labeled roadmap.

— **J. Casey Wilson**
Founder & CEO, Soqucoin Labs Inc.

1 The Problem: Mandated Migration, Unknown Inventory

The regulatory clock on classical cryptography is already running:

- **OMB M-23-02** requires federal agencies to inventory cryptographic systems and submit prioritized migration plans, with progress reporting to OMB on a fixed schedule.
- **CISA BOD 23-02** requires agencies to maintain an inventory of internet-exposed management interfaces, the same asset population where weak TLS and SSH cryptography concentrates.
- **NSA CNSA 2.0** sets algorithm-by-algorithm deadlines for National Security Systems, with key milestones in the early 2030s and a stated preference for exclusive post-quantum use as vendor support matures.
- **NIST** finalized the post-quantum standards in August 2024 (FIPS 203 ML-KEM, FIPS 204 ML-DSA, FIPS 205 SLH-DSA), and its draft transition guidance deprecates 112-bit classical cryptography by 2030 and disallows it by 2035.
- **PCI DSS 4.0, HIPAA, NYDFS 500, SWIFT CSP, SOX, CMMC**: sector frameworks that do not name quantum computing still require current, non-deprecated cryptography and documented cryptographic inventories, which makes classical-crypto findings audit findings today.

The threat model that makes the deadlines binding is **harvest now, decrypt later (HNDL)**: an adversary records encrypted traffic and stolen ciphertext today and decrypts it when a cryptanalytically relevant quantum computer exists. HNDL means the migration deadline for any given system is not the arrival of the quantum computer. It is the arrival date minus the confidentiality lifetime of the data the system protects. A hospital record with a decades-long retention requirement, transported over classical TLS in 2026, may already be compromised in every sense that matters.

Answering "where is our vulnerable cryptography, and which of it matters first" requires looking in at least ten different kinds of places: live endpoints, key stores, source code, dependency manifests, configurations, hardware modules, containers, and cloud services. General-purpose vulnerability scanners see a fraction of this, and none of them reason about data lifetime. That is the gap PQCAT occupies.

2 What PQCAT Is

PQCAT (Post-Quantum Cryptography Compliance Assessment Tool) is a Go program organized in four layers:

1. **Discovery**: ten scanner modules plus auto-detection and CIDR range scanning produce a normalized inventory of cryptographic assets (§3).
2. **Intelligence**: a classifier assigns each asset a quantum-risk zone; the compliance engine converts the inventory into framework-specific scores, gap analyses, remediation plans, and HNDL risk rankings (§4).
3. **Security**: role-based access control, session authentication, a tamper-evident audit log chained with HMAC-SHA256, and post-quantum report seals (§6).
4. **Delivery**: terminal and TUI output, self-contained HTML and PDF reports, machine-readable exports (JSON, CSV for eMASS, XML for Archer/ServiceNow/Xacta, OSCAL, CycloneDX 1.6 cryptographic BOM, CISA BOD 23-02 inventory), SIEM streams, and a REST API with an embedded dashboard in the Pro edition.

Three engineering properties define the tool's deployment story:

Single static binary, zero CGO. Every build disables CGO; the SQLite store and the post-quantum cryptography are pure-Go implementations. The result is one file per platform with no runtime dependencies, which matters enormously in the environments PQCAT targets: SCIFs, sealed enclaves, and change-controlled federal networks where installing a dependency tree is a governance event.

Air-gap enforcement by the compiler. The free Enclave edition is not a connected build with networking switched off in configuration. Server, dashboard, and live threat-intelligence code is fenced behind Go build tags, so the air-gapped binary contains no listener and no outbound network code at all. The claim is checkable: the scanner layer is open source under Apache 2.0, and the binary can be audited for network syscalls. Threat intelligence reaches air-gapped deployments as a reviewable JSON sidecar file carried across the boundary by the operator.

Post-quantum cryptography inside the tool itself. Reports can be sealed with ML-DSA-44 (FIPS 204) signatures so their integrity is verifiable years later; Pro licenses are ML-DSA-65-signed documents validated fully offline. A tool whose job is flagging classical cryptography should not depend on it for its own trust chain.

PQCAT runs on Linux (amd64/arm64), macOS (signed universal builds), and Windows (installer and MSI for GPO/SCCM/Intune deployment), plus Docker and an AWS AMI. A typical deep TLS assessment of a single endpoint completes in 15 to 20 seconds; the fast profile takes about one second per target.

3 Discovery: Ten Scanners, One Inventory

Each module emits findings in a common asset model, so a certificate found on disk, the same certificate served by a load balancer, and the library that generated it all land in one inventory.

Module	What it inspects
`tls`	Live endpoints: dual certificate chains (RSA and ECDSA), protocol matrix from SSLv2/SSLv3 through TLS 1.3 including suites modern libraries refuse to negotiate, cipher enumeration with server-preference detection, OCSP stapling and SCT signature analysis, compression and renegotiation checks, HTTP security headers, ML-KEM hybrid group probing, DNSSEC/DANE
`ssh`	Key exchange, host key, cipher, and MAC classification with server version capture
`sbom`	CycloneDX and SPDX manifests against a 183-entry cryptographic library database with version-specific vulnerability mapping and transitive dependency analysis
`pki`	X.509 stores, Java keystores, PEM directories: chain walking, key-usage quantum risk, cross-signed certificates, per-certificate HNDL exposure windows
`code`	Source code: 564 detection patterns across more than 40 file types, covering algorithm selection, key sizes, deprecated primitives, and hardcoded secrets, with an algorithm resolver that infers effective strength from usage (a truncated key slice, a digest function name) rather than string matching alone
`config`	nginx, Apache, HAProxy, Envoy, Istio, RHEL crypto-policies, Java security properties, Hyperledger crypto-config
`hsm`	PKCS#11 slots, KMIP endpoints, and KMS interfaces, with vendor PQC-readiness assessed by firmware version
`scap`	XCCDF and ARF result content, mapped to cryptographic rules
`image`	Docker/OCI images: embedded certificates and keys, cryptographic libraries, keystore files
`cloud`	AWS: KMS keys, ACM certificates, ELB/ALB listeners, S3 encryption, Route 53 DNSSEC signing keys, IAM signing certificates, under a read-only IAM policy

Table 1: PQCAT discovery modules (v2.8). Auto-detection selects modules per target; CIDR mode sweeps address ranges for TLS/SSH endpoints. Azure and OCI cloud scanners are roadmap (§8).

Every asset is classified into one of three zones. **Red**: algorithms broken by Shor's algorithm on a cryptanalytically relevant quantum computer (RSA, ECDSA, EdDSA, DSA, and classical Diffie-Hellman in all curve and modulus sizes). **Green**: NIST post-quantum standards (FIPS 203/204/205) plus stateful hash-based signatures (LMS/XMSS/HSS) and adequately sized symmetric cryptography, which Grover's algorithm degrades but does not break. **Yellow**: hybrid constructions and parameter sets at the low security margin. The classifier is part of the open-source layer, so the categorization rules are inspectable rather than asserted.

4 Risk Quantification: Scoring That Reasons About Time

4.1 The compliance score

PQCAT reduces an inventory to a 0–100 score that is designed to be defended in front of an auditor, not just displayed on a dashboard:

$$\text{Score} = 100 - \frac{\sum_i r_i \cdot u_i \cdot c_i \cdot h_i}{\sum_i 10 \cdot u_i \cdot c_i \cdot h_i} \times 100$$

where, for each asset i: r_i is the zone risk (green 0, yellow 6, red 10); u_i is urgency, a 1–4 multiplier driven by the distance to the governing framework's deadline for that asset class (overdue items score 4); c_i is criticality (standard 1, High Value Asset 2, National Security System 3); and h_i is the HNDL temporal multiplier described below. The denominator is the worst case for the same inventory, so the score answers "how far is this estate from its own worst case," which keeps it stable as the inventory grows. A companion metric, quantum exposure, reports the plain percentage of red assets so no one has to reverse-engineer the weighting to get a raw number.

4.2 The HNDL Risk Engine (patent pending)

The HNDL engine scores each finding by four additive components on a 0–100 scale: cryptographic strength of the protecting algorithm (0–40), data sensitivity under FIPS 199 categorization (0–30), exposure window (0–20), and attack surface (0–10). The exposure window is the heart of it: retention lifetime of the protected data minus the years remaining until the assumed arrival of a cryptanalytically relevant quantum computer. Data that must remain confidential past that date is already exposed in the HNDL model, whatever the calendar says. Framework profiles set the assumed arrival year and default

retention (the CNSA 2.0 profile is the most aggressive; the default planning year is 2035), and both are operator-tunable. Findings emerge as CRITICAL, HIGH, MODERATE, or LOW, giving migration planners an ordering that a zone color alone cannot provide.

The same temporal reasoning feeds the compliance score as the h_i multiplier, which scales from 1.0 for data retained under two years up to 6.0 for data retained 25 years or longer. Two identical RSA certificates, one on a marketing site and one wrapping long-retention records, produce very different score impacts, which is exactly the behavior a migration plan needs.

4.3 The Confidential Compliance Engine (patent pending)

Regulated organizations face a disclosure paradox: proving compliance to a third party seems to require handing over a map of every weak point. The Confidential Compliance Engine (CCE) resolves this with three cooperating mechanisms, selectable per report as full, anonymized, or aggregate-only modes:

- **Selective anonymization.** Location-identifying fields (hostnames, IPs, certificate subjects and serials, file paths) are replaced by salted BLAKE2b-256 digests truncated to 96 bits, while algorithm and risk data pass through untouched. The report stays fully analyzable (an auditor can see *what* is wrong and how much of it) but no longer says *where*. The same asset hashes consistently within a report, so counts, groupings, and trend comparisons survive anonymization.
- **Aggregate mode.** Only distributions and scores leave the building; individual findings never appear.
- **Verifiable score binding.** Each confidential report carries a transparent, hash-based proof (a Merkle commitment over the findings with a Fiat-Shamir transcript, SHA-384 throughout, no trusted setup) that binds the published score and finding set to a single commitment. We are precise about what this proves and to whom. Any recipient can verify, from the proof alone, that the report is internally consistent and that the committed finding set has not been altered. A recipient who also holds the findings and the commitment salt (the scanned organization itself, or an auditor to whom it discloses under agreement) can additionally open the commitment and confirm that the published score, its range, and the asset count are genuinely derived from those exact findings; substituting a finding or altering the score fails this check. What the construction does not do, and what we do not claim, is prove score correctness to a party that never sees the findings: that is a zero-knowledge property, and a succinct zero-knowledge argument over the compliance statement is a stated research direction (§8), not a shipped capability. The privacy of a confidential report today comes from the anonymization and aggregation layers above, not from the proof.
- **Post-quantum seals.** The report and its proof sidecar are signed with ML-DSA-44, so the evidence chain itself is quantum-safe.

The combination changes who can consume a cryptographic posture assessment. A contractor can hand a prime or an agency a sealed, score-verified report; a CISO can give a board or a cyber-insurer real numbers; a vendor can answer a due-diligence questionnaire with evidence rather than assertions. None of them disclose a target list.

5 Compliance Frameworks and Evidence Formats

Findings are evaluated against the operator's choice of eleven framework profiles: CNSA 2.0, NSM-10, FISMA, FedRAMP, NIST SP 800-131A, CMMC, PCI DSS 4.0, SOX, HIPAA, NYDFS 500, and SWIFT CSP. Framework choice drives deadlines (and therefore urgency multipliers), report language, and gap analysis. The CNSA 2.0 profile also computes migration velocity: at the current rate of change between scans, will the estate meet each algorithm deadline, reported as on-track, at-risk, or critical.

Evidence leaves the tool in the formats compliance programs already consume: POA&M-ready CSV for eMASS, XML for Archer, ServiceNow, and Xacta, OSCAL, executive briefing PDFs written in NSM-10 language, ATO support packages with NIST 800-53 control mapping, a CISA BOD 23-02 inventory export extended with quantum-risk columns, and a CycloneDX 1.6 cryptographic bill of materials (CBOM) ranked by HNDL exposure. OMB M-23-02 milestone dates are built into the scoring calendar, so federal reports frame urgency in terms of the actual reporting obligations.

6 Enterprise Operations: The Pro Edition

The Pro edition adds the connected layer for teams:

- **REST API and dashboard.** A local-first API server (TLS by default, self-signed on first run) with an embedded web dashboard offering four persona views: auditor, CISO, CIO, and executive. Scan orchestration, history, trends, baselines and drift, POA&M, scheduling, and notifications are all API-addressable; Prometheus metrics are exposed for fleet monitoring.
- **Access control and audit.** Three roles (viewer, analyst, admin) enforced per route; session tokens with an 8-hour lifetime; every security-relevant action appended to an HMAC-SHA256 hash-chained audit log whose integrity is verifiable through the API. Denied actions are logged as first-class events.
- **SIEM integration.** Findings export natively as Splunk HEC events, Elasticsearch/OpenSearch bulk documents, and CEF; the real-time alerter also speaks HMAC-signed webhooks, Slack, syslog in CEF, LEEF 2.0, and RFC 5424 framing (UDP, TCP, or TLS), and SMTP.
- **Offline licensing.** Licenses are ML-DSA-65-signed documents validated entirely offline against an embedded public key and revocation list: no activation server, no phone-home, usable in a SCIF. A 14-day evaluation license self-activates on first run (introduced in v2.8), also offline, capped at 250 assets.
- **Accessibility.** The dashboard and reports target Section 508 / WCAG 2.1 AA.

A deliberate non-feature: PQCAT has no SaaS backend and no telemetry. Scan data never leaves the operator's infrastructure unless the operator exports it.

7 Editions and Verifiability

Edition	Contents	Trust story
Enclave (free)	All ten scanners, classifier, TUI, HTML/PDF/JSON/CBOM/OSCAL exports, SQLite history, offline threat-intel sidecar	Air gap enforced at compile time; scanner and classifier source is Apache 2.0 and independently buildable
Pro (licensed)	Enclave plus REST API, dashboard with persona views, RBAC, hash-chained audit log, SIEM export and alerting, scheduling, ML-DSA report seals, Prometheus	Offline ML-DSA-65 license validation; no telemetry; local-first server
Cloud	Pro plus the AWS scanner (KMS, ACM, ELB, S3, Route 53, IAM) under read-only credentials	Same binary discipline; AMI available

Table 2: PQCAT editions. The open-source scanner layer lives at `https://github.com/soqucoin-labs/pqcat`; install via `curl -sSL https://install.pqcat.io | sh`.

The open-core boundary is drawn on purpose: everything that determines *what gets flagged* (scanners, patterns, classifier) is public and auditable, because a compliance tool whose detection logic is secret is asking for faith. What is proprietary is the intelligence layer: scoring, HNDL quantification, the CCE, reporting, and the connected operations surface.

8 Roadmap

Stated as direction, not shipped capability. Azure and OCI cloud scanners extend the `cloud` module beyond AWS. The most significant cryptographic research item is a succinct zero-knowledge argument over the compliance statement: a proof that a published score is correctly computed over a real finding set, verifiable by a party that never sees the findings. Today's proof provides binding and tamper-evidence (§4.3); a true zero-knowledge argument would let an organization prove its posture to a regulator

or prime with no disclosure and no auditor-under-agreement step. The design constraint is that any such construction must be transparent (no trusted setup) and post-quantum sound to fit the tool's threat model, which points to hash-based or lattice-based arguments rather than pairing-based ones. Published scale benchmarks: the built-in benchmark harness measures scoring, HNDL evaluation, proof generation, and anonymization at inventory sizes from one hundred to one hundred thousand assets, and publishing reproducible results at the millions-of-assets scale is planned work. Windows EV code signing is in progress. Finally, we are designing assisted remediation: PQCAT already produces prioritized, framework-aware migration plans, and the next step is letting an operator's AI tooling consume PQCAT's structured outputs (CBOM, POA&M, JSON findings) to draft remediation changes for human review. The evidence chain stays the same: signed findings in, human-approved changes out.

9 About Soqucoin Labs

Soqucoin Labs Inc. (228 Park Ave S, PMB 85451, New York, NY 10003) is a Service-Disabled Veteran-Owned Small Business. The company builds post-quantum infrastructure: the Soqucoin L1 blockchain (the ECDSA-to-ML-DSA migration described in the foreword, independently audited by Halborn Security), the SoquShield mobile wallet, and PQCAT. Two provisional patent applications cover PQCAT's HNDL risk quantification and Confidential Compliance Engine, including U.S. Application No. 63/999,796 (filed March 2026). The scanner layer is open source under Apache 2.0. Enterprise and federal inquiries: `labs@soqu.org`.

This document describes PQCAT v2.8 (April 2026). Capability claims were verified against the shipping codebase at the time of writing; the open-source layer can be independently inspected at the repository above.