Quintessence Labs

ACCELERATING POST-QUANTUM CRYPTOGRAPHY (PQC) TRANSITION WITH qSENTRY

Active Governance vs. Passive Discovery

Table of Contents

1. Executive Summary 2
2. The qSentry Architecture: Centralized Control Plane & Policy Oracle 2
3. How qSentry Solves the Problem 3
4. CBOM Relevancy Analysis: Defeating Scanner Fatigue 4
5. Remediation Approaches 4
6. Real-world qSentry Demo 5
   - Executive Summary 5
   - The Approach 5
   - Baseline CBOM 5
   - qSentry Initial Scan 9
   - qSentry – Policy Definition 11
   - qSentry – Policy API Access 13
   - Integration with Infrastructure Orchestration Rails – such as Ansible, Puppet and Chef 13
   - Our PQC Posture After qSentry 14
   - CBOM Analysis After qSentry Remediation 16
   - Conclusion 16
7. Conclusion & Roadmap for Enterprise Adoption 16

©2026 by QuintessenceLabs Pty Ltd. All rights reserved. No part of this document may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without prior written permission of QuintessenceLabs Pty Ltd.

1. EXECUTIVE SUMMARY

As organizations march toward the 2029 NIST Post-Quantum Cryptography (PQC) migration horizon, CISOs and security governance boards face an unprecedented operational crisis: decentralized, unmanaged, and hardcoded cryptographic debt. Enterprise infrastructure has evolved into a fragmented dependency layer where cryptographic logic is locked directly into application source code, configuration files are maintained inconsistently across servers, and security teams lack real-time visibility into actual cryptographic usage.

To satisfy escalating regulatory mandates (such as US Executive Order M-23-02, NIST guidance, and international standards such as ASD and others), enterprises are forced to perform comprehensive cryptographic discovery. However, traditional crypto-discovery tools suffer from a fatal operational flaw: Scanner Fatigue. Security teams are routinely flooded with massive, 500-page Cryptographic Bills of Materials (CBOMs)—often cataloging millions or more raw entities—with zero guidance other than an untenable directive: 'Good luck manually fixing all these configs, hardcoded private keys, and legacy curve parameters.'

> **The Operational Dilemma: Passive Discovery vs. Active Remediation**
> Passive scanners successfully diagnose cryptographic debt by finding vulnerabilities (e.g., exposed .key files on disk and classical RSA-2048 / ECDSA curves). However, diagnosis is not remediation. Manually editing configurations across hundreds of Apache, NGINX, MySQL, Postfix, Dovecot, and OpenSSH instances introduces severe outage risks, configuration drift, and months of engineering delays.

2. THE qSENTRY ARCHITECTURE: CENTRALIZED CONTROL PLANE & POLICY ORACLE

qSentry breaks this cycle by introducing a centralized cryptographic control plane and dynamic Policy Oracle. Rather than treating cryptography as a local application concern, qSentry decouples key management and algorithm selection from underlying enterprise services, governing infrastructure through two foundational abstraction principles:

- **Principle 1: Identity Abstraction (Keys):** Eliminates raw private key files sitting exposed on local filesystems. Instead, static keys are abstracted and bound to hardware-backed or software-simulated token URIs (e.g., PKCS#11 paths like `pkcs11:token=TSF_Token;object=office_identity;type=private?pin-value=1234`), completely removing key extraction risk. Other alternatives are secret management systems.

- **Principle 2: Algorithm Abstraction (Curves & Ciphers):** Centralizes cryptographic protocol bounds and cipher suites. It dynamically pushes modern hybrid post-quantum key exchange groups (such as NIST-compliant mlkem768:X25519) across enterprise servers without requiring application source code refactoring.

These two principles provide the abstraction of the cryptographic elements from the application itself. This is what crypto-agility is all about and ensures that you don't have to go through the same pain ever again if you need to change any of your cryptographic parameters.

> **Gartner predicts over 40 new crypto algorithms in the next decade**
> Cryptography is not a static field; it is a constantly moving target and it is very common for algorithms to be replaced. We have done this hundreds of times for all sorts of reasons. MD5 was designed in 1991, and it was cryptographically broken in 2008 and unsuitable for use. Yet in 2019, one quarter of widely used content-management systems still use MD5 for password hashing. That shows the risk organizations inherit if crypto-agility isn't taken seriously.

3.HOW qSENTRY SOLVES THE PROBLEM

qSentry is a platform that manages the whole lifecycle of your PQC migration and ongoing monitoring of your applications that perform cryptographic operations. It's not a do it once and then never again, it is a constant part of continuous risk management.

Cryptographic assets are managed in groups (e.g. network segments), and a continuous cryptographic discovery and visibility process will run periodically in the background.

qSentry supports network-based detection, file and host-based detection as well as code scanning. qSentry has the ability to ingest 3rd party CBOM structures to provide a holistic view of cryptographic exposure. However, a key concept in qSentry is to manage cryptographic applications as these are the ones that use any of the cryptographic assets. A legacy RSA key on a file server poses very little risk if nobody uses that anymore. However, if an application uses that RSA key to encrypt web traffic with a web server, then this web server needs to be remediated immediately.

qSentry enabled enterprises to hide all the nitty-gritty details of the crypto definitions behind policies and as an administrator you can define policies to configure any of the applications within your environment that need to be managed.

qSentry provides an extensive API to integrate these policies natively with existing infrastructure orchestration rails – such as Ansible, Puppet and Chef. When an administrator pushes a compliance policy (such as Hybrid_PQC_2029), the automation workflow executes seamlessly:
**Note: Ansible is used below as an example; the API works natively with all common automation frameworks.*

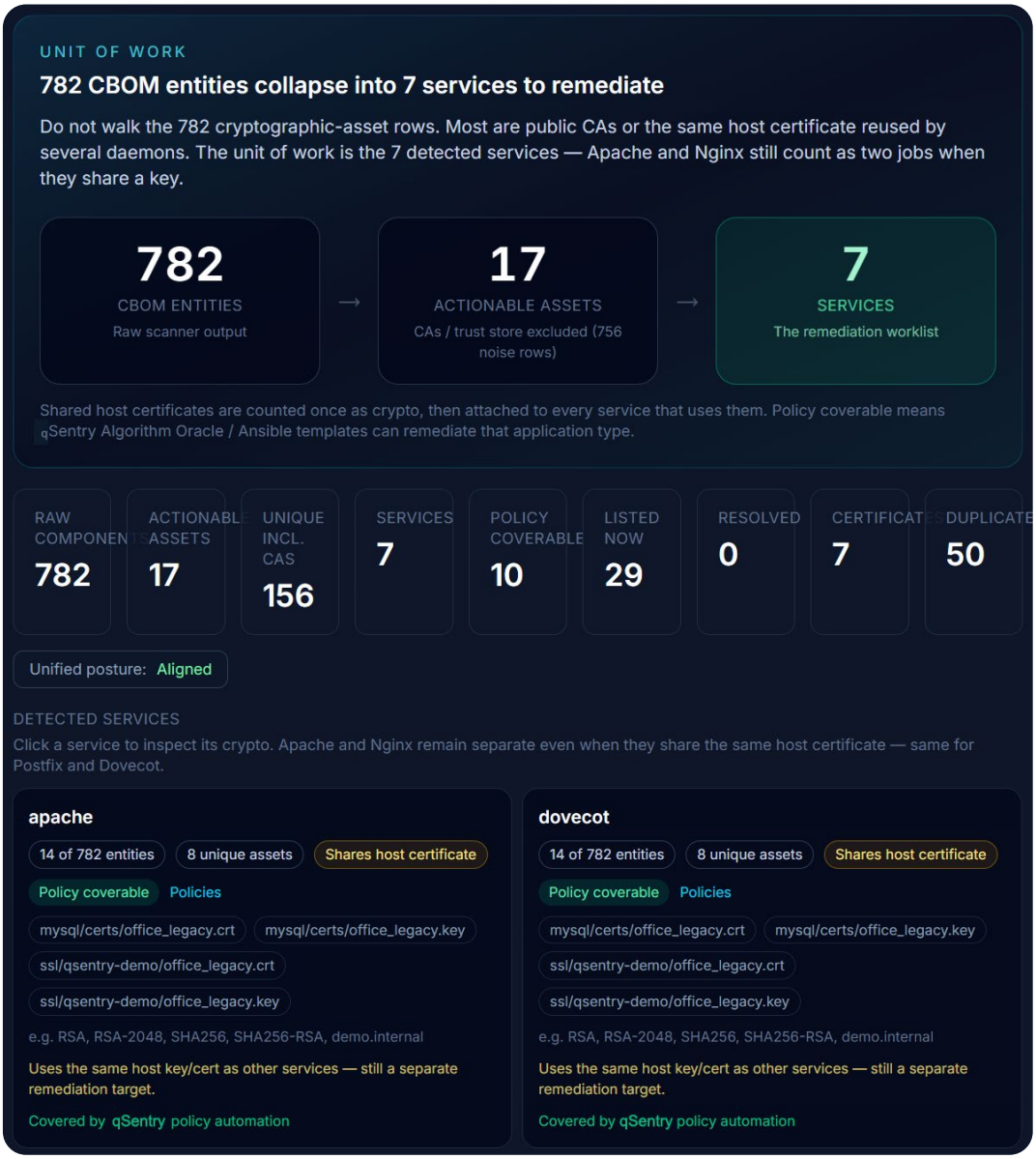
1. **Policy Trigger & API Query:** The orchestration master executes the Ansible playbook targeting infrastructure hosts, specifying the active compliance policy.
2. **Oracle Resolution:** Ansible queries qSentry's Endpoint Identity and Algorithm Oracles to fetch the exact cryptographic parameters required for each target service.
3. **Zero-Agent Configuration Push:** Ansible updates local configuration files (e.g., mysqld.cnf, default-ssl.conf, main.cf, 10-ssl.conf) with the standardized PKCS#11 key paths, or secrets configuration and hybrid curve lists.
4. **Automated Service Lifecycle Handlers:** Built-in handlers cleanly reload or restart services (Apache, NGINX, MySQL, Postfix, Dovecot) in under 60 seconds, ensuring zero operational disruption.

4.CBOM RELEVANCY ANALYSIS: CUTTING THROUGH CRYPTOGRAPHIC NOISE

A central contribution of the qSentry demonstration is debunking the fear of large Cryptographic Bills of Materials (CBOMs). When organizations run automated discovery on a standard server, the scanner typically returns thousands of entities per server. Sometimes even in the hundreds of thousands.

Without context, security boards panic over a many thousand-item finding list. However, a rigorous cryptographic relevancy analysis reveals that the vast majority of these items represent inert passive inventory—such as standard CA trust store root bundles (e.g., ca-certificates.crt), operating system documentation, and non-actionable framework references that pose zero active risk.

qSentry will run a full deduplication round and analyse the CBOM against the configured applications thus reducing repetitive and static components enormously. This powerful analysis results in two benefits. Firstly, the actual active footprint of the CBOM gets reduced to a handful of critical service endpoints and more importantly, executing qSentry's automated policy-push for these services neutralized active transport risk entirely without requiring manual inspection of those entities line-by-line.



5. REMEDIATION APPROACHES

The only viable remediation approach is to bring all applications that perform cryptography into a holistic management layer. The cryptographic operations need to be abstracted onto its own layer (crypto-agility).

Approaches such as network overlays attempt to protect data in transit by wrapping an external encryption layer around packets traversing 'the pipe', suffer from a critical architectural flaw known as the Accidental Exposure Gap. The moment encrypted traffic reaches the destination server, the overlay strips the wrapper, and the local application processes the data. If the local database or web server is still authenticating via legacy RSA or unmanaged local keys, the system remains completely vulnerable to harvest-now-decrypt-later attacks and credential compromise. If a database still writes unencrypted data to the file system, this data is still at risk.

qSentry solves the root cause by immunizing the endpoints themselves. By governing native application configurations and enforcing hybrid PQC across edge servers, databases, and mail exchanges, qSentry establishes end-to-end sovereign trust.

There are cases where this approach is not suitable. Old legacy systems that nobody knows anymore or industrial applications that cannot easily be touched. There are many other ways to solve this problem, often called "Crypto-Shielding" where a modern security layer wraps around un-patchable legacy systems.

6. REAL-WORLD qSENTRY DEMO

To showcase qSentry and to provide some concrete datapoints to the "passive scanning (finding the debt)" and "active governance (fixing the debt)", we used the following approach.

We wanted to simulate a "normal" Office server that is commonly used in millions of enterprises around the world to see what the problems with a "standard" Crypto-Discovery (CBOM) solution is vs. qSentry.

Executive Summary

Just taking a very normal office server and using a CBOM scanner resulted in 782 entries – just for one little server! Good luck fixing a complex enterprise infrastructure in a medium organization.

However, there are only 8 applications that produce these 782 entries and by managing these applications with qSentry, we could immediately eliminate 98% of the risk fully automatically and provide a roadmap for the last 2%.

Cross-correlation revealed that many of the 782 entries relate to duplicate issues and going through one-by-one we could reduce this to 55 entries of interest. In the end there were just 5 highly relevant files that needed to be actioned upon immediately. qSentry managed these applications via an enterprise policy so there was only one item left that couldn't be remediated immediately and a roadmap was provided.

The Approach

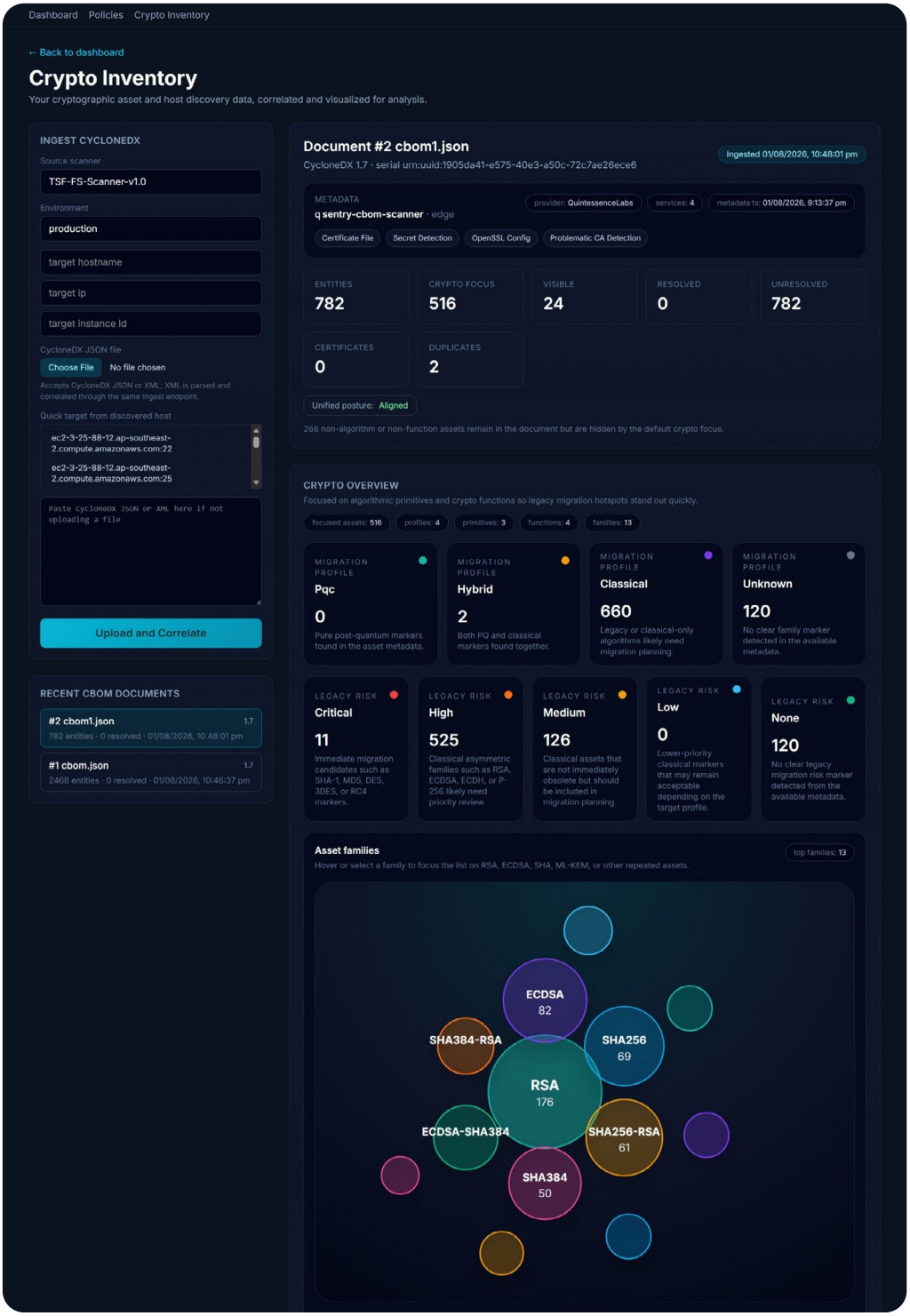
We provisioned an Amazon EC2 instance that resembled a "normal" office server consisting of:

- Apache webserver (for public internet)
- NGINX reverse proxy (for internal applications)
- Mysql database (for data storage)
- Dovecot/postfix for Email services (SMTP / IMAP)
- OpenVPN (for a secure VPN tunnel)
- OpenSSH (for administration)

Nothing special, that's it.

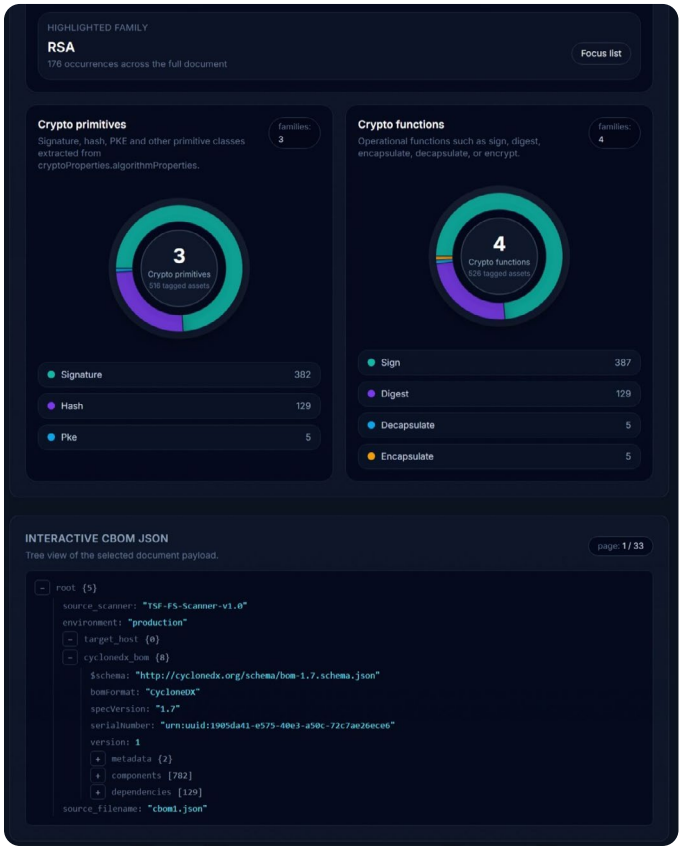
Baseline CBOM

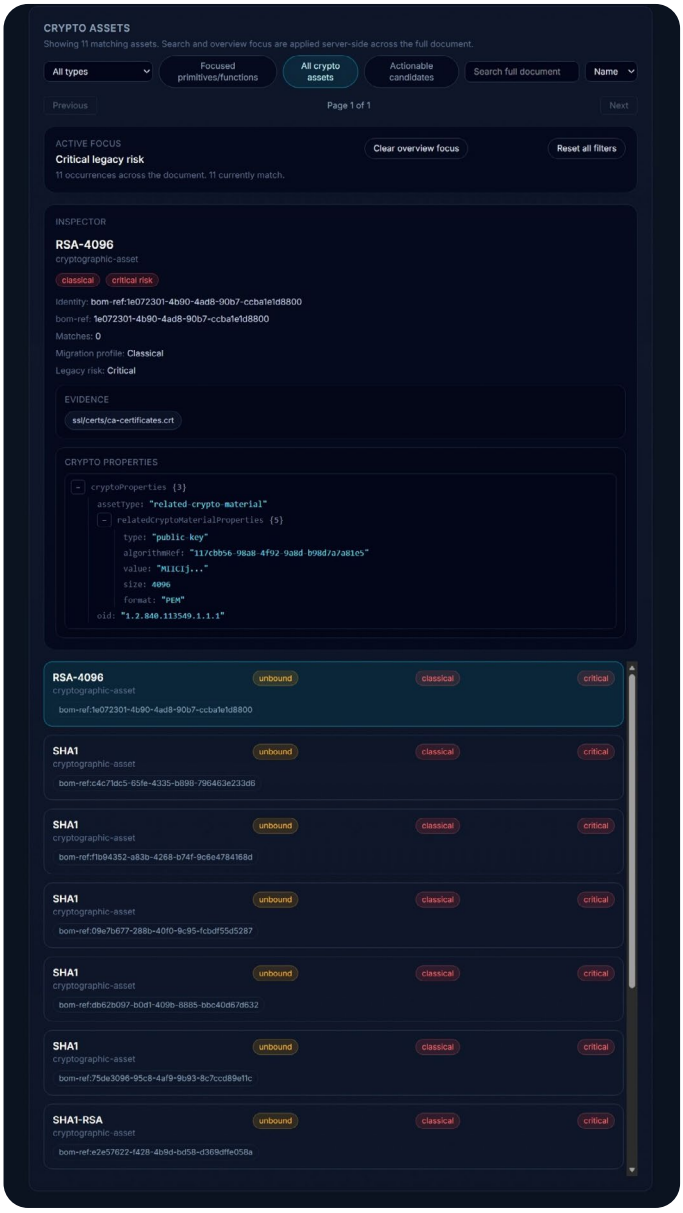
After installing this server, we simply ran a CBOM scanner to produce a baseline. qSentry can ingest 3rd party CBOM's so we can visualize it easily.



Interestingly there were some hybrid (classical & PQC) assets found already, but a standard server that produces 536 entries (68%) that are either critical or high was concerning.

The user interface allows complete filtering control and by filtering only the Critical and RSA assets, we can drill down a bit more.

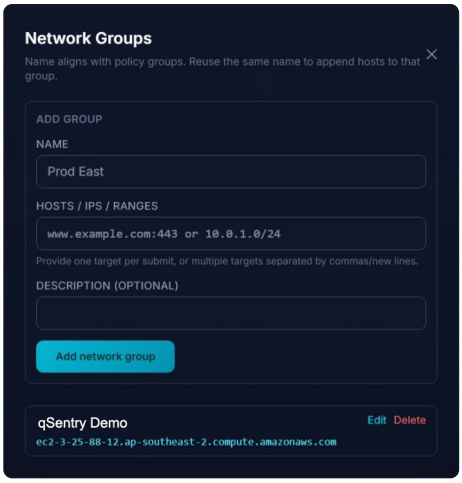




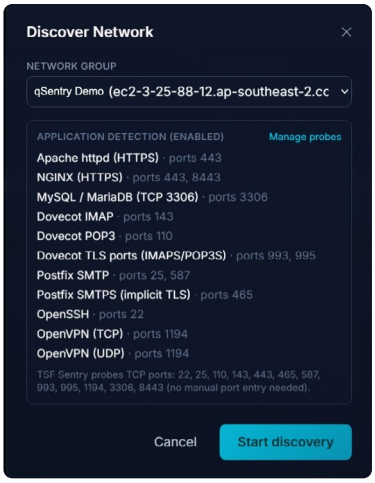
The idea of going through 782 entries for a small office server didn't seem like a good idea, so let's see what qSentry makes out of this.

qSentry Initial Scan

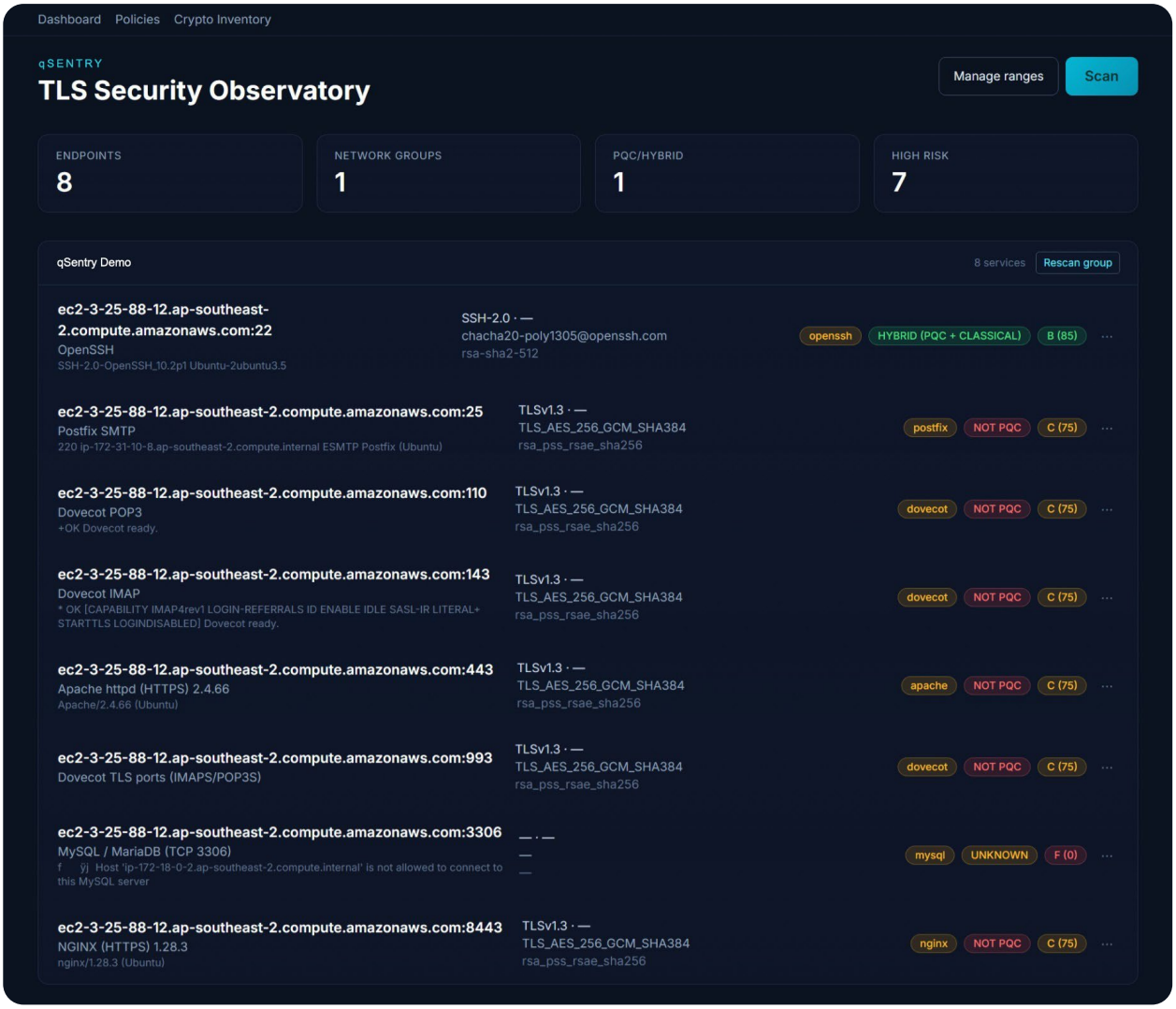
In qSentry, we simply defined a “Network Group” for our office server:



And kicked off the detection:

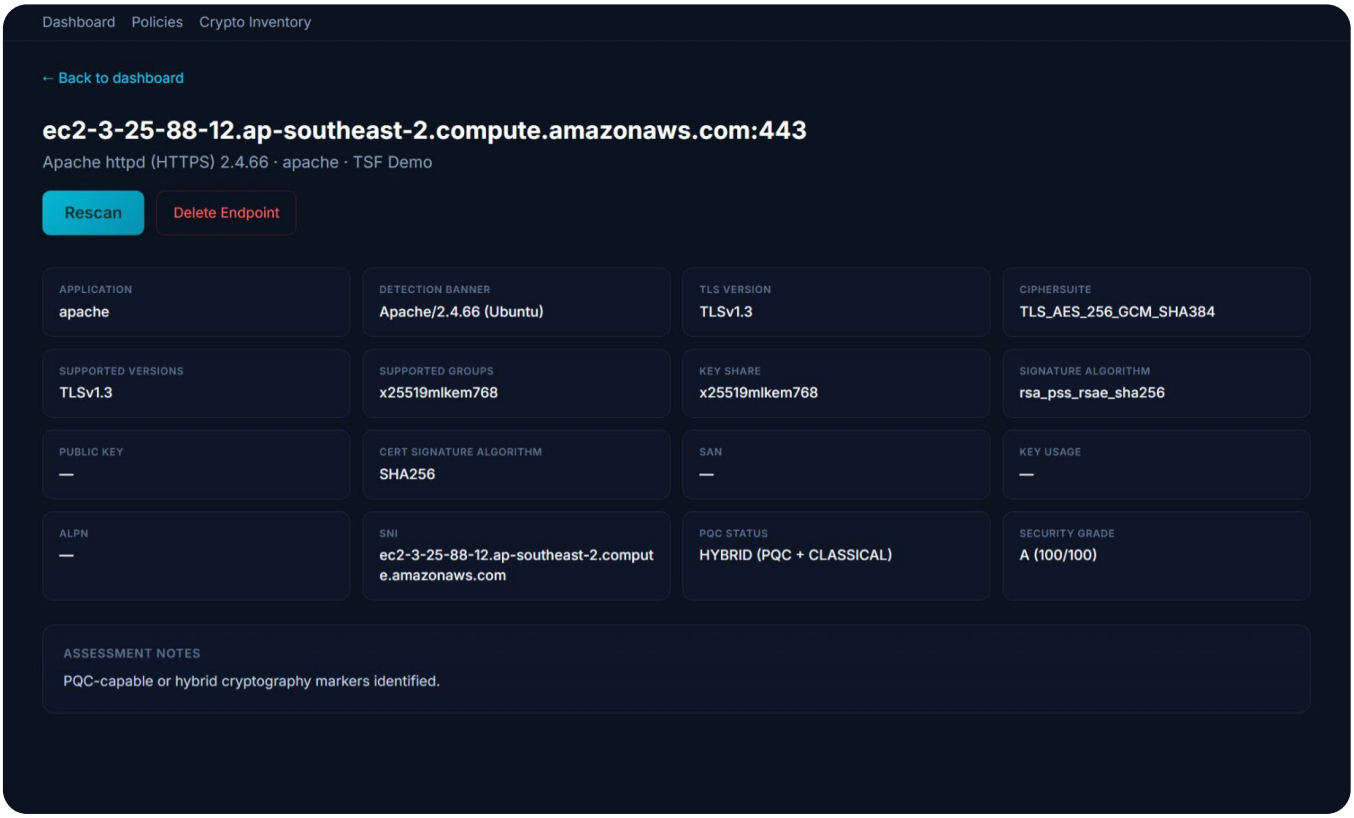


Which produced the following overview:



qSentry automatically detected 8 services where incidentally one was already PQC compliant by offering a hybrid of classical and PQC encryption.

We can easily drill down and look at the details of these entries if we want to review the current state:

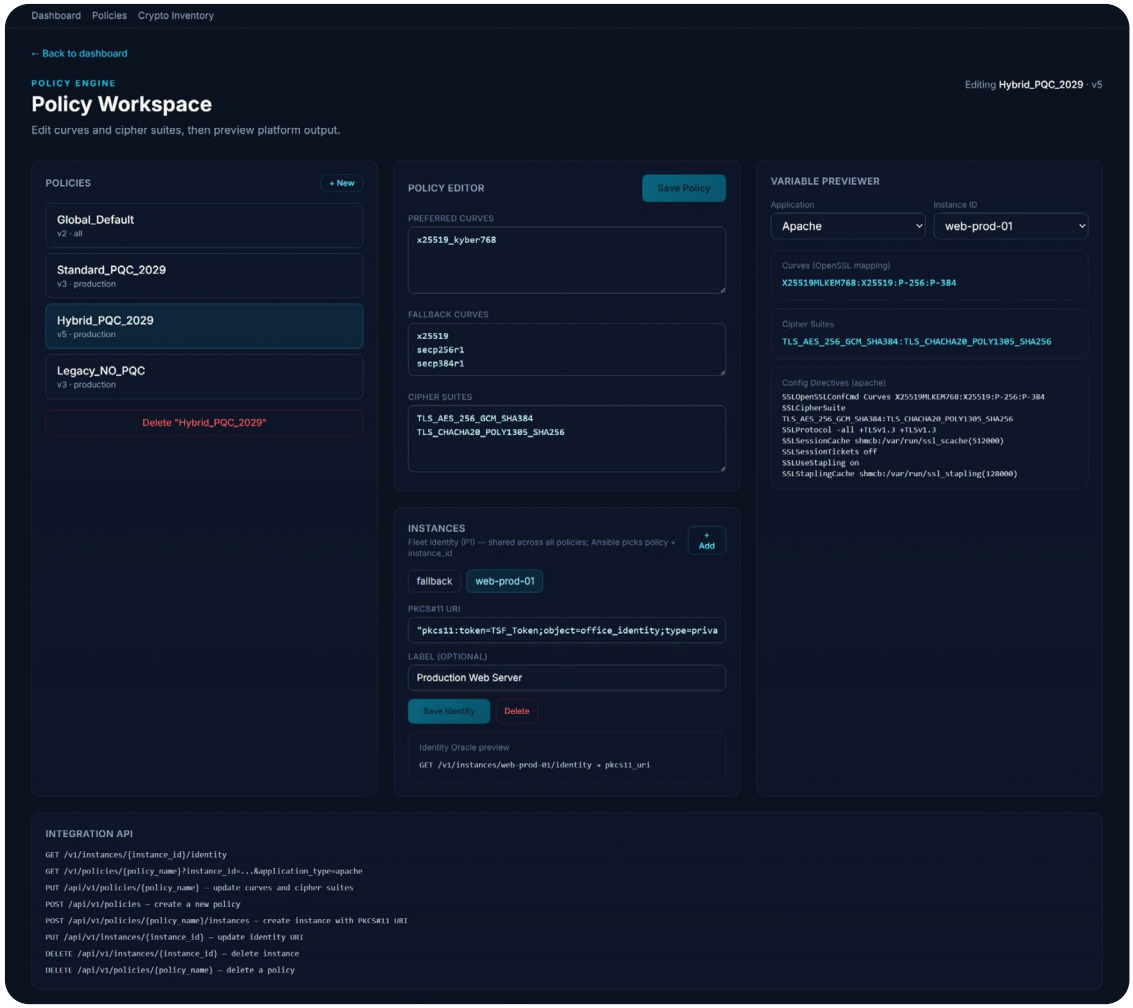


Let's remediate this by defining the right policies.

qSentry – Policy Definition

The main policy we wanted to demonstrate is a policy that pushed hybrid cryptography where we do both classical and post-quantum cryptography.

The details of these next screenshots don't matter actually too much. This is exactly where we can "hide" all the technical details behind the policy. If we can get applications to be compliant with this policy and we have to change anything in the future ever again, you only need to change the policy and everything else is automated. That is what crypto-agility is all about.



You will see configuration for the preferred crypto configuration (which is X25519_KYBER768, which means that it is a hybrid configuration for classical (X25519) and post-quantum (KYBER768 – a PQC NIST standard algorithm).

You will also see that we endeavor to remove any private keys from being physically present on the file system. So, in this case we use a key management solution (our qCrypt solution) to store the private key in a central location with will access control, logging and auditing.

Not all applications support this (e.g. mysql transport encryption), so a backup alternative using secrets management solution is also supported.

As you can see, there are now two policies defined here. One legacy policy and one hybrid policy. We could easily add a third policy for full post-Quantum algorithms to start testing these in development environments.

qSentry – Policy API Access

We provide API access to these policies, which enable us now to fully automate the remediation for these applications. The remediation depends now on the application itself. Basically, there are three high-level outcomes:

- The application supports PQC already and can be configured to do so
- The application does not yet support PQC but will in the future, or
- The application is so legacy that it will never support PQC

The first case is by far the most common case and if you use mostly common applications or subscribe to cloud offerings from the big companies, you'll be fine.

The last case is the least common, but the most complex to deal with and the remediation is highly dependent on the particular circumstance. In some cases, it is possible to patch the application to enable PQC, but in many cases the best approach is to lock this down and do crypto-shielding – meaning put an application in front that can perform post-quantum-cryptography.

Integration with Infrastructure Orchestration Rails – Such as Ansible, Puppet and Chef

We can help you define Ansible, Puppet or Chef playbooks to ingest our Policy API to manage the configuration of the supported application. In our example we provided a simple Ansible playbook that connects to the qSentry API to get the policy and then pushes the changes out to the office server:

```
ubuntu@ip-172-31-10-8: ~/ansib
ubuntu@ip-172-31-10-8:~/ansible-playbook$ ansible-playbook apply_policy_api.yml -e policy_name=Hybrid_PQC_2029 -
e tsf_sentry_base_url=http://127.0.0.1:8080/tsf-sentry
```

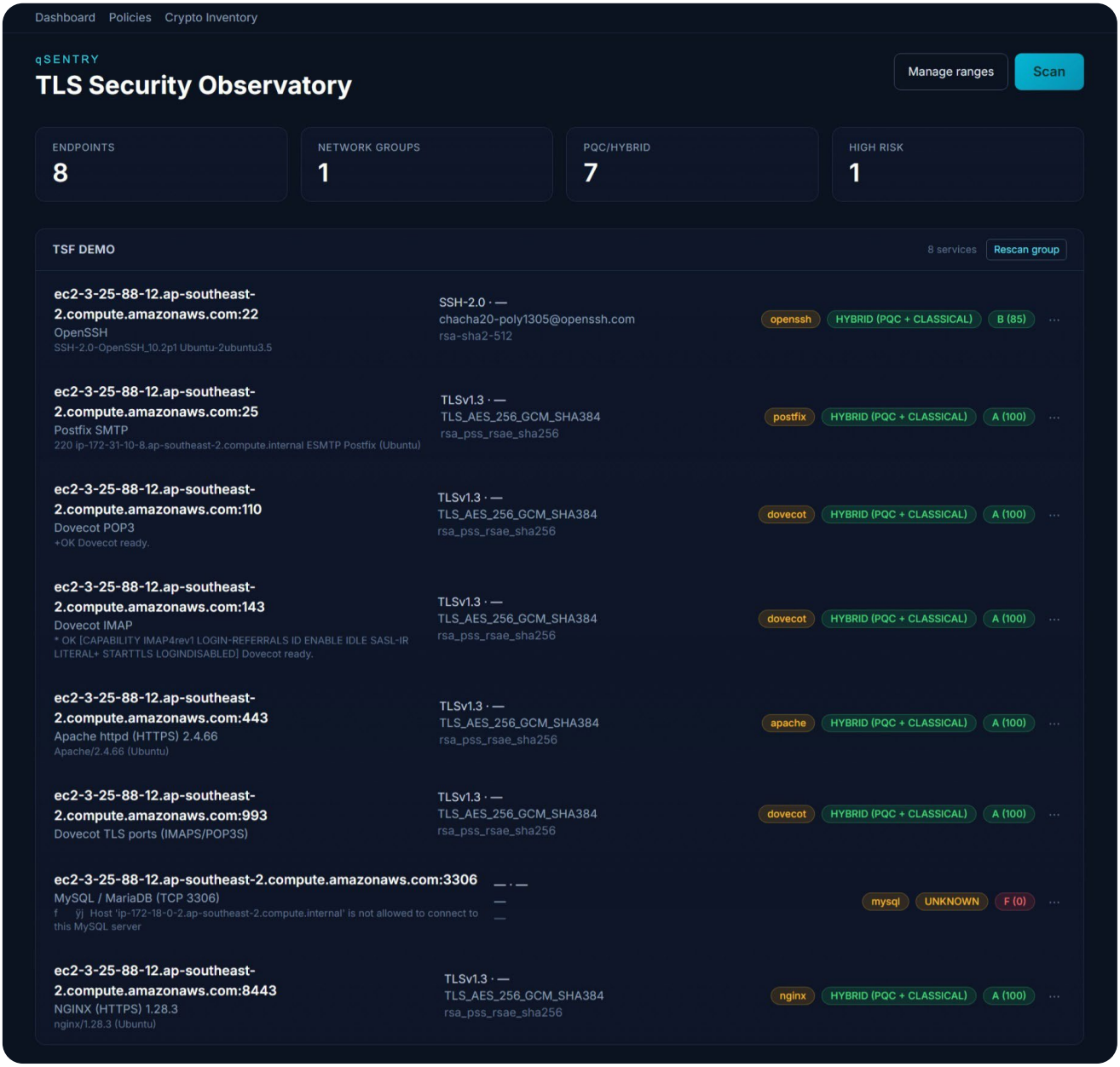
```
ubuntu@ip-172-31-10-8: ~/ansib
    "msg": [
        "policy=Hybrid_PQC_2029 v5",
        "P1 apache  (web-prod-01)=\"pkcs11:token=TSF_Token;object=office_identity;type=private?pin-value=1234\""
,
        "P1 fallback (fallback)=/etc/ssl/qsentry-demo/office_legacy.key",
        "P2 curves=X25519MLKEM768:X25519:P-256:P-384",
        "P2 ciphers=TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256",
        "P2 tls_min=TLS1.3"
    ]
}

RUNNING HANDLER [Reload Apache] ************************************************************************
changed: [localhost]

RUNNING HANDLER [Reload NGINX] *************************************************************************
changed: [localhost]

RUNNING HANDLER [Restart Postfix] **********************************************************************
changed: [localhost]

RUNNING HANDLER [Reload Dovecot] ***********************************************************************
changed: [localhost]

PLAY RECAP *********************************************************************************************
localhost                  : ok=37   changed=10   unreachable=0    failed=0    skipped=4    rescued=0    ignored
=0

ubuntu@ip-172-31-10-8:~/ansible-playbook$
```

That's it. All hard work is taken care of by Ansible and qSentry.

A subsequent scan reveals the updated cryptographic posture.

Our PQC Posture After qSentry

Once the policies are set up and the Ansible playbook runs, we can simply re-scan our environment to see what our new posture looks like:



qSentry now automatically manages all but one application, continuously monitoring them for compliance. Cryptographic parameters can be modified directly within the qSentry Policy, eliminating the need to edit complex configuration files. Updates are deployed immediately via an Ansible playbook.

The single application that couldn't be managed was the transport layer encryption for our database (mysql). The reason was that we installed MySQL 8.4 which doesn't support PQC yet. However newer versions of MySQL will support PQC, and we just need to update MySQL to a newer version, and it can be managed too.

This will certainly happen within any environment, and qSentry is the perfect tool to track the transition from the current status-quo to a full PQC compliant state. For small organizations that process may be weeks, but for large enterprises this is certainly a multi-year process.

CBOM Analysis After qSentry Remediation

As we mentioned before, we started with 782 items in our CBOM list. However just listing things doesn't make them relevant and we need to assess a neutralized or irrelevant risk and the shift to a cryptographic relevancy and risk state.

A standard CBOM catalogs everything it finds (algorithms, certificates, internal parameters, legacy references). However, under a modern hybrid compliance policy, the existence of an item in the inventory does not mean it poses an active threat.

So we took the CBOM and analyzed it manually. We could simply remove many Certificate Authority files (CA) that were irrelevant and not used as well as other irrelevant things and the list boiled down to 55.

A further review removed all but 5 critical components:

RSA-2048	mysql/certs/office_legacy.key
ECDSA	ssh/ssh_host_ecdsa_key
RSA-3072	ssh/ssh_host_rsa_key
RSA-2048	ssl/private/ssl-cert-snakeoil.key
RSA-2048	ssl/qsentry-demo/office_legacy.key

And these are exactly the components that qSentry manages for the applications. Most keys are gone by using PKCS#11 or secrets.

Conclusion

Simply installing "normal" office components on a server still poses a significant risk from a PQC posture point of view, but by using qSentry we've been able to mitigate all the risks (bar one) within 1h of analyzing this server and we successfully completed a full PQC readiness transition in that timeframe that is aligned with all regulatory institutions worldwide.

We have demonstrated the difference between Passive Inventory to Active Transport risk and hopefully we have demonstrated that just running a cryptographic inventory scanner (CBOM) will not solve any of your issues. It will just produce lots of work that is costly and inefficient.

7. CONCLUSION & ROADMAP FOR ENTERPRISE ADOPTION

The 2029 NIST PQC migration deadline leaves no room for manual configuration hunting or paralysis by scanner fatigue. Organizations cannot afford multi-year engineering projects to refactor legacy application stacks.

As demonstrated through qSentry, cryptographic complexity can be transformed into agile, automated control. By centralizing policy intelligence, abstracting vulnerable static keys behind hardware tokens, and orchestrating hybrid post-quantum standards through existing automation rails, qSentry empowers enterprises to achieve quantum-safe compliance in minutes rather than years.

GET IN TOUCH

QuintessenceLabs has formed proven partnerships with a variety of enterprises, where our quantum-based encryption and key management solutions have effectively strengthened data protection across entire organizations.

FOR MORE INFORMATION VISIT

www.quintessencelabs.com or contact info@quintessencelabs.com.

Quintessence Labs

AUSTRALIA
Unit 11, 18 Brindabella Circuit
Brindabella Business Park
Canberra Airport ACT 2609
+61 2 6260 4922

UNITED STATES
175 Bernal Road
Suite 220
San Jose CA 95119
+1 650 870 9920

www.quintessencelabs.com

Document ID: 7464-01

©2026 QuintessenceLabs. All rights reserved.