PITHY NOTES PUBLICATIONS

PithySignal

PITHY SIGNAL: HYPER TPRM

SIGNAL BRIEF | ISSUE 003

# A Software Bill of Materials Is Becoming a Machine-Readable Regulatory Control

*Why software transparency is moving from supplier documentation to continuous, usable third-party risk evidence.*

A software bill of materials is becoming structured, machine-readable evidence that third-party risk teams can connect to vulnerability monitoring, supplier remediation, and accountable decisions.

**Created by Pithy Notes Publications | Published under Pithy Signal**

KEY SECTIONS



**Category: Hyper TPRM**
**Publication Date: 2026-09-01**
**Issue: 003**
Tags: Hyper TPRM | Software Supply Chain | SBOM | Continuous Evidence | Regulatory Readiness

This publication was created by Pithy Notes Publications and published as part of the Pithy Signal governance intelligence series.

## Executive Summary

A software bill of materials (SBOM) is moving beyond its original role as a technical inventory. In several important regulatory and assurance contexts, it is becoming structured evidence: a machine-readable record that can help organizations identify software components, connect them to known vulnerabilities, and evaluate how suppliers respond as risk changes.

The shift matters for third-party risk management. Traditional programs often collect a document during onboarding or an annual review, confirm that it exists, and treat the request as complete. That approach is poorly matched to software that changes continuously. New releases introduce dependencies. Vulnerability information changes. Suppliers patch, replace, or accept components over time.

The more useful question is no longer simply whether a supplier has an SBOM. It is whether the organization can use the SBOM as continuing control evidence.

That requires more than possession. It requires completeness, version alignment, machine readability, freshness, ownership, exception handling, and a connection to vulnerability and remediation workflows. It also requires restraint: an SBOM can reveal what software contains, but it does not by itself prove exploitability, patch quality, or effective risk reduction.

For Hyper TPRM, the emerging signal is clear. Software transparency is becoming operational evidence, and governance programs need the processes and systems to use it.

## 1. Software Transparency Is Becoming Operational

An SBOM is a structured inventory of the components that make up a software product. It may include proprietary components, open-source packages, commercial dependencies, versions, identifiers, and dependency relationships.

Historically, many organizations treated this information as a specialist artifact for product security or software composition analysis. That boundary is changing.

Regulators and public-sector security authorities increasingly describe software transparency in terms that support automation and repeatability. The Cybersecurity and Infrastructure Security Agency's current minimum elements emphasize an SBOM associated with each software version or update and coverage that includes direct and transitive dependencies. The European Union's Cyber Resilience Act requires a commonly used, machine-readable SBOM for products with digital elements, covering at least top-level dependencies. In the United States, Food and Drug Administration requirements connect SBOMs to cybersecurity documentation for cyber devices.

These are not identical obligations, and they do not create one universal rule for every product or sector. Together, however, they indicate a broader direction: software component evidence is expected to be structured, current enough to match a product version, and usable by systems rather than readable only by people.

## 2. Machine Readability Changes the Control Model

A static document can satisfy a request without improving oversight. A machine-readable artifact can participate in an operating process.

When an SBOM is normalized and connected to security tooling, an organization can use it to:

- identify affected supplier products when a new vulnerability is disclosed;
- distinguish one software version from another;
- trace direct and transitive component exposure;
- prioritize outreach to suppliers with relevant dependencies;
- compare supplier claims with observed remediation activity; and
- maintain an evidence trail across releases and review cycles.

This is the difference between collecting evidence and activating evidence.

In a traditional assessment, the artifact is often evaluated once. In a continuous assurance model, the artifact becomes an input to monitoring, triage, exception management, and decision-making. The control is not the file itself. The control is the repeatable ability to receive, validate, interpret, and act on it.

## 3. Why This Matters for Hyper TPRM

Hyper TPRM extends third-party risk management beyond periodic questionnaires and static evidence snapshots. It focuses on operational signals, changing dependencies, and the ability to maintain trust as systems and suppliers evolve.

SBOMs fit this model because software risk is not stationary. A component that appears low risk at onboarding may become vulnerable months later. A supplier may release an update that removes one dependency and introduces another. A remediation commitment may be technically valid for one version and irrelevant for the version the organization actually uses.

An effective program therefore needs to link the SBOM to four operating questions:

1. Scope: Which supplier products and versions are in use?
2. Exposure: Which components and dependency paths are relevant?

3. Response: What has the supplier done about the identified risk?
4. Decision: What action should the organization take, and who owns it?

This turns the SBOM from a procurement attachment into a shared control input for security, risk, procurement, engineering, legal, and resilience teams.

## 4. Evidence Quality Determines Usefulness

An SBOM can be present and still be operationally weak. Programs should evaluate at least five qualities.

### Completeness

Does the artifact include the dependencies needed to understand meaningful exposure, including transitive dependencies where appropriate? Missing components create false confidence.

### Version alignment

Does the SBOM match the exact product version or update in use? A valid inventory for the wrong release is not reliable evidence for the deployed environment.

### Machine readability

Can systems parse the format consistently? A spreadsheet or PDF may be understandable to a reviewer while remaining difficult to validate and monitor at scale.

### Freshness

Is the artifact updated when the product changes? The value of component evidence decays as releases, dependencies, and vulnerability intelligence change.

### Traceability

Can the organization connect findings to supplier responses, internal decisions, exceptions, and remediation records? Evidence without an accountable workflow does not complete the control.

These qualities should be assessed in proportion to risk. A safety-critical medical device, a widely deployed infrastructure product, and a low-impact internal utility do not necessarily require the same assurance depth.

## 5. An SBOM Is Not Proof That Risk Is Controlled

The growing importance of SBOMs creates a risk of overclaiming their value.

An SBOM may help identify that a component is present. It does not automatically establish that a vulnerability is exploitable in the product's configuration. It does not prove that a patch is effective, that compensating controls work, or that the supplier will respond within an acceptable time.

It may also contain errors, omissions, ambiguous identifiers, or outdated dependency information. Different tools can produce different results. Suppliers may have legitimate constraints on disclosure, and smaller organizations may need time and support to mature their practices.

The right governance model treats the SBOM as one evidence layer. It should be combined with vulnerability context, product configuration, supplier attestations, remediation data, monitoring signals, and business-impact analysis.

The counter-signal is important: structured transparency improves visibility, but visibility is not the same as risk reduction.

## 6. The Operating Model Must Mature

Organizations preparing for this shift should define a practical operating model rather than adding another questionnaire field.

### People

- Assign ownership for receiving and validating supplier SBOMs.
- Define when security, procurement, legal, engineering, and business owners become involved.
- Establish escalation paths for missing, incomplete, or high-risk evidence.

### Process

- Request version-specific SBOMs based on product and risk scope.
- Validate format, identifiers, dependency coverage, and freshness.
- Connect findings to vulnerability triage and supplier remediation.
- Document exceptions, acceptance decisions, and review dates.
- Preserve evidence across product updates and contract lifecycles.

### Technology

- Support commonly used machine-readable formats.
- Normalize component identifiers where possible.

- Integrate SBOM data with asset, vulnerability, vendor, and case-management systems.
- Automate matching and alerts without removing human judgment.
- Retain provenance so reviewers know what the evidence represents and when it was received.

Maturity will be uneven. The goal is not immediate perfection. It is a defensible progression from document collection to evidence operations.

## 7. What Organizations Should Do Next

Organizations can begin with a focused readiness sequence:

1. Inventory the current state. Determine where SBOMs are already requested, received, stored, or analyzed.
2. Define risk-based scope. Identify products, suppliers, and regulatory contexts where software component transparency matters most.
3. Set minimum evidence criteria. Establish acceptable formats, version alignment, dependency coverage, update frequency, and ownership.
4. Pilot an operational workflow. Connect a limited set of high-priority supplier SBOMs to vulnerability monitoring and remediation tracking.
5. Measure usability, not collection. Track whether the evidence changes triage, decisions, or supplier action.
6. Preserve human accountability. Use automation to surface and organize signals while keeping risk acceptance and escalation with named decision-makers.

This sequence allows a program to learn before scaling and avoids turning SBOM collection into a new form of evidence theater.

## Closing Signal

The software bill of materials is becoming more than a technical inventory.

It is becoming a machine-readable regulatory and assurance control—one that can support continuous third-party oversight when it is complete, current, interpretable, and connected to action.

The organizations that benefit most will not be those that collect the largest number of files. They will be those that can turn structured software transparency into timely, accountable decisions.

## Signal Sources

1. Cybersecurity and Infrastructure Security Agency, Minimum Elements for a Software Bill of Materials (SBOM).
2. European Parliament and Council of the European Union, Regulation (EU) 2024/2847 (Cyber Resilience Act).
3. U.S. Food and Drug Administration, Cybersecurity in Medical Devices: Quality Management System Considerations and Content of Premarket Submissions.