PROVENANCE RISK RESEARCH · INDUSTRY BRIEFING

State of Third-Party Technology Risk

Emerging threats to financial services infrastructure

2026 Edition

Trevor Kavanaugh
Founder, Provenance Risk Research Inc.

Inaugural edition · April 2026
An independent assessment of systemic technology dependencies in financial services outsourcing

Executive Summary

Between 2020 and 2026, ten major technology incidents caused an estimated **$45 billion to $250 billion** in combined economic damage to organizations worldwide. None of these incidents were traditional vendor failures. Every one propagated through **technical dependencies**—software components, update mechanisms, and shared infrastructure—that existing third-party risk management (TPRM) frameworks were not designed to detect.

The financial services industry faces a structural problem: TPRM programs organize risk around legal entities (vendors, subcontractors, fourth parties), but real-world technology incidents propagate through technical dependency chains that cross organizational boundaries. The result is an industry that appears diversified at the vendor level while converging on the same underlying infrastructure, software components, and delivery mechanisms.

This document identifies the emerging risk categories that boards, executives, and regulators should be tracking—and explains why current frameworks are not surfacing them.

The Core Problem: Individually Rational, Collectively Fragile

Every organization that selected CrowdStrike for endpoint security, deployed MOVEit for file transfer, or ran software containing Log4j made individually rational decisions—optimizing for security effectiveness, regulatory compliance, or operational efficiency. The concentration that created systemic risk emerged not from vendor lock-in or anti-competitive practices but from market dynamics rewarding technical excellence.

> The better a vendor's product, the more widely it is adopted. The more widely it is adopted, the greater the systemic risk if that vendor experiences operational or security failure. Vendor success creates concentration risk.

This is the Collective Fragility Paradox: *individual institutional diversification collapses into systemic fragility because the entire industry shares the same underlying technical dependencies.* An institution with 200 "diverse" vendors that all run on the same cloud provider, use the same endpoint security platform, and embed the same open-source components has not diversified its technology risk—it has obscured its concentration behind a layer of contractual relationships.

By the Numbers: The Cost of Dependency Risk

The following incidents demonstrate that technical dependency risk is now a material financial risk category on par with credit risk, market risk, and operational risk as traditionally defined.

Incident	Est. Cost	What Happened	Risk Dimension
SolarWinds (2020)	~$100B	Nation-state actors compromised the software build process; malicious updates distributed to ~18,000 organizations	Software Delivery

Incident	Est. Cost	What Happened	Risk Dimension
Log4j (2021)	$15–100B+	Critical vulnerability in open-source component embedded in 17,000+ packages; 60–64% of Java applications affected	Supply Chain
Kaseya (2021)	$1–2B	Ransomware exploited MSP platform; 50–60 MSPs compromised, 800–1,500 downstream businesses affected	Infrastructure Concentration
3CX (2023)	$2B+	First documented "supply chain of a supply chain" attack; five-level dependency chain, 600,000+ organizations affected	Supply Chain + Delivery
MOVEit (2023)	$15.8B	Vulnerability in file transfer platform; 2,773+ organizations and 95.8M individuals affected, mostly through nth-party chains	Infrastructure Concentration
Snowflake (2024)	$10B+ (est.)	Compromised customer credentials lacking MFA enforcement enabled access to data warehouse environments across 165+ Snowflake customers; AT&T (110M call records), Ticketmaster (560M individuals), Santander, and Advance Auto Parts among those affected through shared platform exposure	Infrastructure Concentration
xz Utils (2024)	$0 (caught)	Nation-state actor spent 2.5 years infiltrating open-source project; discovered accidentally by one engineer	Supply Chain

Incident	Est. Cost	What Happened	Risk Dimension
Polyfill.io (2024)	Cleanup costs across affected sites	Popular JavaScript polyfill library acquired by a China-linked entity in early 2024, then weaponized to inject malicious code into 100,000+ websites that embedded the polyfill.io CDN; mass injection of malware into legitimate site traffic before community detection	Supply Chain
CrowdStrike (2024)	$10B+	Defective update to endpoint security platform crashed 8.5M Windows systems globally in 78 minutes	Software Delivery + Concentration
TeamPCP / UNC6780 (2026)	Banking impact developing	Financially-motivated campaign compromised build-system and package-registry infrastructure (GitHub Actions, npm, PyPI, container registries) as a credential-theft vector. Stolen credentials enabled exfiltration of 300+ private GitHub repositories from compromised Cisco CI/CD infrastructure, including banking customer code	Supply Chain + Delivery

Key pattern: Not one of these incidents was a traditional vendor relationship failure. Every one exploited technical dependency chains that TPRM programs were not structured to detect. Approximately 29% of all breaches are now attributable to third-party attacks, and 97% of major U.S. banks experienced exposure through third- or fourth-party relationships in 2024—though only a handful of vendors were actually attacked.

Three Emerging Risk Dimensions

Modern technology dependencies create risk across three distinct but interconnected dimensions that traditional TPRM frameworks do not adequately address. Each dimension represents a category of exposure that vendor-level assessment cannot detect.

1. Supply Chain Risk: What's In the Product

When organizations procure software, they assess the vendor. They rarely examine what components the vendor has embedded in their product. Modern applications are assembled from hundreds or thousands of open-source and commercial components, each with its own dependency chain.

The Log4j vulnerability affected an estimated 60–64% of Java applications globally—not because those organizations chose Log4j, but because it was embedded 5, 10, or 20 levels deep in software they purchased from vendors. The xz Utils infiltration demonstrated that threat actors can spend *years* patiently compromising components that underpin critical infrastructure, and that discovery is largely accidental.

The March 2026 TeamPCP campaign demonstrated the contemporary form of this risk: a financially-motivated threat actor compromised package-registry and build-system infrastructure across npm, PyPI, GitHub Actions, and container registries, using stolen developer credentials as the foothold rather than malicious code planted in any single package. Among the verified compromises were widely-used open-source security tools (Trivy and Checkmarx GitHub Actions) and a popular AI orchestration package (LiteLLM). The exfiltrated credentials were then used to access Cisco's CI/CD infrastructure, from which more than 300 private GitHub repositories were stolen, including source code belonging to banks, business process outsourcing firms, and U.S. government agencies. The pattern combines elements from prior incidents - registry compromise, credential theft, privileged CI/CD access, and downstream customer code exposure - into a single multi-stage campaign rather than a discrete failure at any one layer. Financial institutions whose vendors had GitHub repositories or development tooling touching any of the affected registries had no contractual relationship with TeamPCP and no visibility into the dependency chain that exposed them.

> Approximately 60% of vulnerabilities in modern applications originate from transitive dependencies—components that organizations never selected, never evaluated, and in most cases don't know exist.

What boards and regulators should be asking: Do we have visibility into the software components embedded in our critical vendor products? If a Log4j-scale vulnerability emerged tomorrow, could we determine our exposure in hours—or would it take weeks?

2. Software Delivery Risk: How It Gets Updated

SolarWinds demonstrated that software update mechanisms can become attack vectors—nation-state actors inserted malicious code into legitimate updates distributed to 18,000 organizations. CrowdStrike demonstrated that even a trusted vendor's routine update process can cause catastrophic harm when a defective configuration change crashed 8.5 million devices in 78 minutes.

The structural challenge is that many security and infrastructure tools require **privileged system access and automatic updates**—customers cannot stage or test them before deployment because the value proposition is real-time protection. Organizations rely entirely on the vendor's internal quality assurance. When that trust is violated—intentionally or accidentally—the blast radius is enterprise-wide.

What boards and regulators should be asking: For vendors with kernel-level or privileged access to our systems, do we control when updates deploy to our environment? Do we assess the security of vendors' build and delivery pipelines, or only their production environments?

3. Infrastructure Concentration Risk: Where It All Runs

An institution with 100 vendors that all run on AWS does not have 100 points of diversification—it has a single

infrastructure dependency obscured by 100 vendor contracts. CrowdStrike's endpoint security platform is deployed across 298 of the Fortune 500. Cloudflare handles approximately 20% of all web traffic. Three cloud providers host the vast majority of financial services applications.

This concentration is invisible to entity-centric risk management. Each vendor passes its individual assessment. But when a cloud region experiences an extended outage, or a shared platform pushes a defective update, **the correlated failure across the portfolio is what creates systemic risk.**

The 2024 Snowflake breach made this concentration risk visible in a different form. A campaign targeting customer credentials—not Snowflake's own infrastructure—gained access to data warehouse environments across more than 165 organizations. AT&T disclosed 110 million customer call records exposed; Ticketmaster disclosed 560 million customer records; Santander, Advance Auto Parts, and others confirmed exposure. None of these compromises resulted from a Snowflake security failure—each was its own customer-side credential event—but the systemic pattern reflected the concentration of analytical workloads on a single platform with inconsistent credential enforcement across its customer base. Financial institutions assessing Snowflake at the vendor level passed the assessment. The risk lived one layer down, in the aggregate behavior of a shared customer population.

What boards and regulators should be asking: What percentage of our critical vendor portfolio runs on the same cloud provider? How many share the same EDR, identity provider, or CDN? If a major cloud region experienced a sustained outage, how many of our critical vendors would be simultaneously affected?

Why Current Frameworks Miss This

The incidents above exposed five systematic gaps in how traditional TPRM programs assess and manage vendor relationships.

Framework Gap	What TPRM Measures	What the Incidents Revealed
Entity vs. Dependency	Risk organized around legal entities (vendors, fourth parties)	Risk propagates through technical dependencies (components, update mechanisms, platforms) regardless of organizational boundaries
Known vs. Hidden Exposure	Contracted vendor relationships	Transitive dependencies, shared infrastructure, and nth-party chains create exposure without contractual relationships
Vendor Failure vs. Vendor Success	Vendor bankruptcy, breach, or service degradation	Vendor market dominance creates concentration risk; the best products create the most systemic exposure when they fail
Individual vs. Collective Risk	Is our institution over-reliant on this vendor?	Is our entire industry converging on the same platforms? Sector-wide concentration creates systemic risk that individual assessment cannot detect

Static Attestation vs. Continuous Risk	Annual SOC 2 reports and point-in-time assessments	CrowdStrike went from trusted vendor to global disruptor in 78 minutes; point-in-time attestation cannot capture the velocity of technology risk

The Assurance Framework Gap

The core Trust Services Criteria governing SOC 2—the dominant framework for vendor technology assurance in financial services—have remained **largely unchanged since 2017**. While the AICPA updated "points of focus" in 2022, the fundamental criteria do not require comprehensive dependency visibility, software supply chain security assessment, or infrastructure concentration analysis.

SOC 2 was designed to address organizational security controls—access management, change management, incident response. It was not designed to map complex dependency chains or assess systemic concentration risk. The UK Financial Conduct Authority observed following the CrowdStrike incident that organizations complying with ISO 27001, SOC 2, and NIST CSF were still caught unawares, because compliance with traditional frameworks did not prevent impact from dependency risks.

The Regulatory Lag: An 18–36 Month Pattern

A consistent pattern has emerged across every major technology incident: the gap between the event and meaningful regulatory guidance addressing the risk category it revealed spans 18 to 36 months. This creates three challenges for financial institutions.

Challenge	Impact
Compliance Uncertainty	Organizations do not know what standard they will be held to when examiners eventually develop specific guidance, making it difficult to justify investments
Peer Divergence	Without regulatory clarity, institutions adopt different approaches; "industry best practice" does not crystallize, making benchmarking difficult
Retroactive Expectations	When guidance emerges, it often includes expectations that institutions "should have" implemented controls earlier, citing major incidents as evidence the risks were "known"

Notable regulatory developments: The EU Digital Operational Resilience Act (DORA), effective January 2025, represents the most comprehensive regulatory response to date—requiring direct oversight of critical ICT service providers, mandatory concentration risk management, and assessment of "long or complex chains of subcontracting." U.S. regulators harmonized federal banking expectations in 2023 (OCC Bulletin 2023-17 / Federal Reserve SR 23-4 / FDIC FIL-23-2023), but practical examination guidance for software supply chain risk, infrastructure concentration, and delivery pipeline security remains limited.

> Financial regulators have evolved TPRM guidance substantially over the past decade. But regulatory guidance still addresses third-party relationships—not the nth-party technical dependencies that characterize modern software supply chains, delivery systems, and cloud infrastructure concentration.

The Reframing: Dependencies, Not Parties

The central argument of this assessment is that the unit of analysis for technology risk management must shift. Managing risk at the vendor level remains necessary but is no longer sufficient. The three risk dimensions identified above require a parallel layer of dependency risk management.

Risk Dimension	Traditional TPRM Question	Dependency Risk Question
Supply Chain	"Who is the vendor and what do they do?"	"What components are in the vendor's product?"
Delivery	"How does the vendor secure their environment?"	"How does the vendor distribute updates to my environment?"
Infrastructure	"What are our critical vendor relationships?"	"What infrastructure do our critical vendors depend on?"

This is not a call to replace existing TPRM frameworks. It is a call to augment them with dependency-level visibility that current programs do not provide. The incidents documented here demonstrate the cost of the capability gap. The question is whether institutions will build that capability before the next incident—or after.

Looking Ahead

Several structural trends suggest these risks will intensify rather than resolve:

- **Cloud consolidation continues.** Three providers host the vast majority of financial services applications, and the economics of cloud computing favor further concentration, not less.
- **AI introduces new dependency chains.** Foundational model providers are becoming critical infrastructure with even narrower market concentration than cloud computing, and organizations are embedding AI capabilities with limited visibility into model supply chains.
- **Supply chain attacks are accelerating.** The 2023 3CX incident demonstrated the first documented "supply chain of a supply chain" attack with a five-level dependency chain. The 2026 TeamPCP campaign demonstrated the next evolution: rather than embedding malicious code in a single package, threat actors are now compromising the registries, build pipelines, and developer credentials that govern thousands of packages simultaneously. Adversaries are learning that targeting shared components and platforms provides far greater return on investment than targeting individual organizations.
- **Open-source risk is structural, not episodic.** Critical infrastructure depends on components maintained by small groups of volunteers. The xz Utils infiltration revealed a reusable attack pattern—and was discovered only by accident.

Published by Provenance Risk Research Inc., an independent 501(c)(3) research foundation focused on third-party risk management, software supply chain security, and operational resilience in financial services. All Provenance Risk Research publications are made freely available to the public.

Disclosure: The author serves as Founder and President of Provenance Risk Research Inc. The author also operates Provenance Risk Advisory LLC, a separate for-profit consulting firm under shared leadership. This document represents the research views of Provenance Risk Research Inc. and was developed independently of any consulting engagement.

For research inquiries: trevor@provenanceriskresearch.org