Trusted Systems in an Autonomous, Post-Quantum World

Why Encryption Alone Will Not Protect Your Organization

SecureFi Institute Special Brief 002

Trusted Systems in an Autonomous, Post-Quantum World

Why Encryption Alone Will Not Protect Your Organization

Date: April 2026
SecureFi Institute

Table of Contents

PREFACE

Author's Perspective

I was trained in an era where trust was engineered into systems at the lowest levels.

During my time at Digital Equipment Corporation, Compaq, and Hewlett-Packard, security was not an afterthought, or a feature layered on top of infrastructure. It was built into the foundation. Hardware integrity, operating system controls, and data handling were tightly integrated, particularly in environments supporting federal and defense missions. Systems such as secure desktops, leveraging technologies like SE Linux and NetTop architectures, enforced separation and control in ways that were deliberate and measurable.

My work supporting defense programs, including early data environments tied to GMTI and Joint STARS, reinforced a critical lesson that has only grown more relevant over time. Data does not lose value with age. In many cases, it becomes more valuable. The assumption that data is only at risk in the moment it is created or transmitted has never aligned with how adversaries operate. Persistence, patience, and long-term collection have always been part of the threat model.

Later in my career, working with data platforms and visualization technologies, including my time supporting federal initiatives at Tableau, that perspective expanded. It became clear that the challenge was no longer just how data is protected, but how it is governed, accessed, replicated, and ultimately understood. Data governance, lineage, and visibility became just as critical as encryption itself. Organizations were generating and sharing data at a scale that made complete control increasingly difficult.

What has changed is the scale and the timeline.

Today, we operate in environments that are distributed, software-defined, and increasingly autonomous. Infrastructure is assembled across global supply chains. Data is replicated across systems, partners, and platforms. Artificial intelligence is accelerating both capability and exposure. At the same time, the emergence of post-quantum cryptography is reshaping assumptions about how long current protections will hold.

The industry has responded, but largely in isolation, focusing on encryption, or supply chain, or AI governance as separate challenges. In practice, these are not separate problems. They are interconnected layers of a single issue.

Trust is no longer guaranteed by any single control point.

In many cases, it is not guaranteed at all.

This paper is built on a different premise.

Trust must be engineered across the entire system, from silicon to software to data to algorithm, and continuously validated over time.

The goal is not to predict when a specific technological threshold will be crossed. It is to understand what is already happening, where current approaches break down, and how organizations can establish environments where trust is not assumed, but built, measured, and sustained.

We are not preparing for a future event.

We are responding to a system that is already in motion.

ACT I - THE THREAT REALITY

1. Executive Summary

The security assumptions that underpin modern computing are under sustained pressure from three converging forces: continuous adversary data collection, accelerating regulatory mandates, and the advancing trajectory of quantum computing.

These forces do not align in time.

Data is being collected now, often in encrypted form, with the expectation that it can be decrypted later. Regulatory bodies are requiring organizations to inventory cryptographic systems and prepare for transition. At the same time, quantum capability is progressing toward thresholds that will invalidate widely deployed public key cryptography.

This creates a structural shift in risk.

Exposure is no longer defined solely by unauthorized access. It is defined by whether sensitive data has already entered an external collection environment where future decryption may convert it into actionable intelligence. In this model, the breach and the impact are separated in time.

Systems may appear secure today while accumulating long-term exposure that is not visible through traditional controls.

This condition is commonly described as Harvest Now, Decrypt Later. It reflects a transition from immediate compromise to deferred disclosure.

Most organizations are not prepared for this shift.

Cryptographic dependencies are deeply embedded across applications, infrastructure, identity systems, and supply chains. Data is replicated across environments, often without complete visibility. Machine identities outnumber human identities and are not consistently managed. Supply chains introduce components that cannot be fully verified. Artificial intelligence systems accelerate decision-making while expanding data exposure and reducing transparency.

These factors create an environment in which trust is assumed across layers that are not validated as a system.

This paper presents the Trusted Systems Framework as a response.

The framework defines nine interdependent pillars required to establish and maintain trust across modern environments: origin, integrity, visibility, identity, control, resilience, adaptability, autonomy governance, and temporal awareness. Together, these pillars provide a structure for understanding where trust breaks down and how it can be re-established.

A maturity model is introduced to assess capability across these pillars, with particular emphasis on temporal alignment between data sensitivity and cryptographic durability. Scenario-based illustrations demonstrate how failures propagate across supply chain, identity, and data layers, and how a system-level approach would alter those outcomes.

The central conclusion is direct.

Post-quantum transition is not a cryptographic upgrade. It is a system-wide transformation that requires coordinated change across hardware, software, data, identity, and governance.

Organizations that act early can align this transition with system lifecycles, reduce long-term exposure, and establish measurable control. Organizations that delay will face compressed timelines, incomplete visibility, and increasing regulatory pressure, while exposure continues to accumulate.

The defining factor will not be how quickly new algorithms are deployed.

It will be whether trust is engineered as a system property, rather than assumed at individual layers.

2. Exposure Is Already Underway

The prevailing assumption in most organizations is that sensitive data is at risk primarily at the moment it is created, transmitted, or accessed. Once encrypted and stored, that data is generally considered protected, subject only to future breach or unauthorized access.

That assumption is no longer technically sufficient.

Adversaries do not need to break encryption at the time of collection to derive value from data. They only need to collect it. Advances in storage, distributed collection capabilities, and long-term intelligence strategies have made it feasible to capture and retain large volumes of encrypted data for extended periods. This data includes communications, transaction records, identity exchanges, telemetry, and intellectual property.

This model, commonly described as Harvest Now, Decrypt Later, fundamentally changes how exposure must be understood.

Under this model, compromise is no longer defined by immediate access to plaintext. It is defined by successful acquisition and retention of encrypted data that can be decrypted at a later time. The breach event and the impact event are separated, sometimes by years. In many cases, there is no observable signal at the time of collection that distinguishes a routine transaction from a long-term compromise.

From a system perspective, this creates a temporal exposure problem.

Data with long-term sensitivity, such as financial records, defense information, identity systems, and intellectual property, often persists within enterprise and government environments for periods ranging from five to twenty years or more. If that data is collected today and stored externally, its eventual exposure becomes a function of time and computational capability rather than ongoing access.

The key implication is that encryption protects confidentiality in the present, but does not prevent future disclosure if the underlying cryptographic assumptions are broken.

This exposure is compounded by the way modern systems handle data.

Data is no longer confined to a single controlled environment. It is routinely replicated across backup systems, analytics platforms, partner integrations, cloud services, and machine learning pipelines. Copies are created for resilience, performance, and operational convenience. In many environments, these copies are not fully tracked, and their cryptographic protections are not consistently managed over time.

As a result, organizations often cannot answer a set of basic but critical questions:

- Where does sensitive data reside across all environments
- How many copies of that data exist

- Which cryptographic mechanisms protect each instance
- How long that data is expected to remain sensitive
- Whether that data has already been collected by external actors

This lack of visibility does not indicate a failure of individual controls. It reflects the architectural reality of distributed systems operating at scale.

The challenge is not limited to data storage. It extends to data in transit and data in use.

Encrypted communications, including TLS-protected sessions, API calls, and inter-system exchanges, are observable at the network level. While their contents are protected in real time, they can still be captured and stored. Identity exchanges, including authentication tokens and certificate-based negotiations, are similarly exposed to collection. Over time, these interactions form a detailed record of system behavior, relationships, and access patterns.

In environments where artificial intelligence is used, the exposure surface expands further. Data provided to external models, embedded in prompts, or incorporated into training workflows may be retained, processed, or redistributed in ways that are not fully visible to the originating organization. This introduces additional pathways for long-term data persistence outside of controlled environments.

From the perspective of an adversary, this model is efficient and scalable.

Collection can occur passively, without triggering alerts associated with active exploitation. It does not require persistence within a target environment. It does not depend on bypassing endpoint controls or maintaining access over time. It leverages the fact that organizations continue to generate, transmit, and store valuable data using cryptographic systems that are expected to weaken under future computational advances.

From the perspective of the defending organization, this model is difficult to detect and even more difficult to quantify.

There is no single event that can be identified as the point of compromise. There is no alert that indicates data has been successfully harvested for future decryption. Standard security controls, which are designed to detect unauthorized access or anomalous behavior, are not structured to detect passive collection of encrypted data at scale.

The result is a condition in which exposure exists without visibility and compromise occurs without immediate consequence.

This is the context in which post-quantum risk must be evaluated.

The question is not whether quantum computing will eventually impact current cryptographic systems. The question is how much data of long-term value has already entered an exposure window in which future decryption will convert historical collection into actionable intelligence.

For many organizations, the honest answer is unknown.

That uncertainty is not a temporary gap that will be resolved through incremental improvement. It is a structural condition created by the interaction of data proliferation, distributed architectures, and adversary collection strategies.

The implication is direct.

Organizations are not starting from a clean state as they plan for post-quantum migration.

They are operating within an environment where elements of compromise may already be in place, waiting for the conditions under which they can be realized.

3. Why Action Is Now Mandatory

The transition to post-quantum security is no longer driven by technical curiosity or long-term research planning.

It is being driven by a convergence of operational reality and regulatory enforcement.

The complexity of modern environments means that cryptographic systems are deeply embedded across infrastructure, applications, identity, and data flows. These dependencies are rarely centralized and often not fully documented. As a result, the time required to identify, assess, and replace vulnerable cryptographic mechanisms is measured in years, not months.

Regulators have recognized this.

Mandates are now focused not only on future adoption, but on immediate visibility. Organizations are being required to identify where cryptography is used, how it is implemented, and what dependencies exist across their systems. This shift reflects an understanding that transition timelines must begin before the risk is fully realized.

The challenge is that the threat model does not wait for that process to complete.

Guidance from National Institute of Standards and Technology has resulted in the standardization of post-quantum cryptographic algorithms, providing a technical foundation for migration. These standards define what to implement, but they do not reduce the complexity of implementing change across existing systems.

In parallel, federal directives such as OMB M-23-02 require agencies to perform a comprehensive inventory of cryptographic systems and begin planning for migration. This requirement forces organizations to answer a question many have not previously addressed:

Where are all of our cryptographic dependencies, and how are they used?

For most organizations, the answer is incomplete.

National security guidance further accelerates the timeline. The Commercial National Security Algorithm Suite 2.0 defines a transition path for quantum-resistant cryptography across national security systems, with adoption expected in the near term and required implementation within this decade.

The implication is clear.

Organizations must begin the transition before the risk is fully realized.

The regulatory timeline assumes a structured progression from inventory to planning to implementation. The threat model does not follow that structure.

Adversary collection operates continuously. Data is accumulated without regard for compliance milestones or migration schedules. This creates a misalignment between how organizations plan and how exposure actually develops.

The result is a narrowing window.

Organizations must begin transition before they fully understand their environment, while the volume of potentially exposed data continues to grow.

From a technical standpoint, the challenge is not limited to adopting new algorithms.

Post-quantum cryptography introduces new requirements for key sizes, computational performance, and protocol design. Systems that were not designed for cryptographic agility may require significant modification or replacement. Hardware platforms, firmware signing mechanisms, identity systems, and communication protocols must all be evaluated for compatibility with new standards.

From an operational standpoint, the challenge is coordination.

Cryptographic systems are embedded across security, infrastructure, application development, and third-party services. Migration requires alignment across these domains, along with clear ownership, funding, and governance. Without this coordination, organizations risk partial implementations that introduce new vulnerabilities rather than eliminating existing ones.

From a compliance standpoint, the challenge is accountability.

Regulatory mandates are increasingly specific in their expectations. Organizations are required not only to plan for migration, but to demonstrate progress, document dependencies, and validate that controls are in place. Failure to meet these expectations will result in audit findings, remediation requirements, and potential operational restrictions.

Taken together, these factors create a condition in which inaction is no longer neutral.

Delaying inventory delays understanding. Delaying understanding delays planning. Delaying planning compresses implementation into a shorter timeframe, increasing both cost and risk. At the same time, the exposure window continues to expand.

The decision point is no longer whether to act.

It is whether the organization can establish sufficient visibility and coordination to act effectively before regulatory pressure and technical risk converge.

Those that move early will shape their transition under controlled conditions.

Those that delay will be required to execute under constraint, with incomplete information and reduced margin for error.

4. The Full Attack Surface

Post-quantum risk is often framed too narrowly. In many discussions, the attack surface is reduced to a limited set of cryptographic functions such as TLS key exchange, VPN tunnels, or digital signatures. Those functions are important, but they represent only a small portion of the environment in which trust is actually established and maintained.

In operational terms, the attack surface is the total set of technical pathways through which data, control, identity, and system state can be influenced, observed, or compromised. In a post-quantum context, that surface is expanded not only by vulnerable algorithms, but by the interdependence of hardware, firmware, software, data flows, identity systems, supply chain inputs, and increasingly autonomous systems.

The critical issue is that these layers do not fail independently. Weakness in one layer can invalidate assumptions in another. A system may have strong encryption in transit, but untrusted firmware. It may have signed software packages, but opaque third-party dependencies. It may have well-defined user authentication, but unmanaged machine identities. It may have a migration plan for cryptographic libraries, but no method for revalidating archived data or hardware-rooted trust anchors.

As a result, organizations that approach post-quantum transition as an isolated cryptographic upgrade are likely to underestimate both the scope of exposure and the complexity of remediation.

4.1 Hardware as the Physical Trust Boundary

The hardware layer remains the lowest practical trust boundary in most computing environments. It includes processors, memory subsystems, motherboards, embedded management controllers, peripheral buses, network interfaces, storage controllers, and the components that anchor secure boot and attestation.

This layer matters because higher-level controls assume its integrity.

If a hardware platform is altered during manufacturing, staging, transport, integration, or maintenance, then software-based security mechanisms may operate on a compromised foundation. A malicious or modified component can affect system behavior below the visibility of the operating system, endpoint security tools, and application monitoring platforms. Examples include modified firmware images, malicious controllers, unauthorized debug interfaces, altered boot chains, and hidden management pathways.

From a post-quantum perspective, the hardware layer introduces several distinct risks.

First, hardware lifecycles are much longer than software lifecycles. A server, appliance, embedded platform, industrial controller, or mission system may remain in service for five to ten years or longer. If cryptographic protections built into those systems are not crypto-agile, they may outlive the validity of the assumptions on which they were deployed.

Second, hardware trust often depends on device identity, firmware signing, secure enclaves, or root-of-trust mechanisms that themselves rely on asymmetric cryptography. If those trust anchors cannot evolve to post-quantum algorithms, then the system may retain a secure appearance while its underlying validation model degrades over time.

Third, hardware provenance is often poorly understood once systems move beyond the original procurement event. Organizations may know the vendor, but not the full origin of subcomponents, firmware development chains, or integration pathways. That creates a provenance gap between the purchased system and the trusted system.

In practical terms, post-quantum readiness at the hardware layer requires more than support for new algorithms. It requires confidence that the platform can maintain trusted boot, trusted update, trusted attestation, and trusted identity under new cryptographic conditions without requiring full replacement of the deployed base.

4.2 Firmware and the Management Plane

Firmware occupies one of the most consequential and least visible portions of the attack surface. It includes BIOS and UEFI components, embedded controller code, BMC and out-of-band management firmware, storage controller firmware, NIC firmware, and device microcode.

Firmware is strategically important because it persists below the operating system, executes early in the boot chain, and often retains privileged access to system state. A compromise at this layer can survive reimaging, bypass host-based controls, manipulate telemetry, and establish long-term persistence.

The management plane further amplifies this risk. Out-of-band controllers are designed specifically to provide administrative control, remote recovery, power management, provisioning, and lifecycle operations. They are supposed to be trusted by design. If they are

compromised, misconfigured, or cryptographically outdated, they can become the most powerful attack path in the environment.

From a post-quantum perspective, firmware and management systems create several technical challenges:

- Firmware signing mechanisms may rely on digital signature algorithms that require migration.
- Boot-time validation may depend on certificate chains that are not post-quantum ready.
- Secure update channels may require protocol and library changes to support new cryptographic primitives.
- Device identity and attestation models may need reissuance at scale across hardware fleets.
- Management systems may become a bottleneck if they cannot coordinate or validate quantum-safe updates across heterogeneous infrastructure.

This is one reason hardware-rooted security and integrated management architectures matter. If post-quantum protections are only introduced at the application or network layer, but not at the firmware and management layers, the result is a fragmented trust model. Sensitive workloads may appear protected while underlying control paths remain susceptible to persistence, spoofing, or downgrade risk.

In short, firmware is not a lower-priority layer to be addressed after software migration. It is one of the first layers that must be assessed because it governs how trust is established at system startup and maintained across the platform lifecycle.

4.3 Software and Dependency Chain Exposure

Software is often treated as the most visible portion of the attack surface because it is where organizations spend most of their engineering time and where many traditional vulnerabilities are discovered. In reality, software risk is not limited to application code. It includes libraries, operating systems, hypervisors, orchestration layers, middleware, APIs, CI/CD pipelines, container images, and all transitive dependencies that influence runtime behavior.

The post-quantum dimension of software risk is broader than replacing one algorithm implementation with another.

Applications may embed cryptographic logic directly in code, rely on older libraries with limited support for PQC primitives, or depend on middleware and frameworks that were never designed for cryptographic agility. Some systems will require only a library refresh. Others will require protocol redesign, certificate workflow changes, or full application modernization.

Dependency chains complicate this further. A system may be nominally upgraded to a post-quantum-capable library while continuing to depend on components that assume classical certificates, classical handshakes, or hardcoded key sizes. In distributed systems, these

assumptions are often hidden inside messaging frameworks, identity brokers, API gateways, and service meshes.

Software supply chain integrity also remains a primary concern. Signed packages, reproducible builds, SBOMs, artifact registries, and pipeline controls can improve visibility, but they do not eliminate the problem of trust transitivity. If a trusted build system consumes an untrusted dependency, or if a signing environment is compromised, the resulting artifact may preserve formal integrity while violating actual trust assumptions.

Post-quantum transition at the software layer therefore requires several parallel activities:

- discovery of algorithm use across applications and libraries
- mapping of transitive cryptographic dependencies
- validation of protocol compatibility and performance impact
- redesign of systems that cannot support cryptographic agility
- assurance that software provenance and build integrity remain trustworthy throughout migration

The software attack surface is not simply where encryption is used. It is where trust assumptions are instantiated in code.

4.4 Data as a Distributed and Persistent Exposure Layer

Data is not only an asset. It is also a propagation medium for risk.

Most enterprise and government environments treat data protection as a combination of encryption, access control, backup, retention policy, and segmentation. Those controls remain important, but they are not sufficient when data is duplicated across environments and retained over long time horizons.

From a systems perspective, data moves through multiple states:

- data at rest in databases, file stores, object stores, and archives
- data in transit across internal networks, partner connections, APIs, and external services
- data in use within analytics pipelines, business applications, memory, model workflows, and user sessions
- derived data in logs, reports, embeddings, indexes, metadata stores, and training artifacts
- replicated data in backups, snapshots, disaster recovery environments, and external partner ecosystems

Each state introduces a different exposure model. The issue is not only whether the primary data set is encrypted. It is whether all instances, derivatives, replicas, and metadata artifacts are protected consistently and remain governable over time.

This is where post-quantum risk becomes materially different from a conventional breach model.

In a conventional breach, the central question is whether an unauthorized actor gained access to the environment. In a delayed exploitation model, the central question is whether data of long-term value was collected, intercepted, or copied in a form that can be decrypted later. That means data sensitivity must be evaluated not only by business value, but by time horizon.

A short-lived session artifact may have limited future value. A settlement record, health history, defense communication, engineering design, or identity ledger may remain operationally or strategically valuable for years. That data enters a quantum exposure window the moment it is transmitted or stored under vulnerable cryptographic assumptions.

The data layer is therefore not passive. It is the layer in which the consequences of all other trust failures accumulate.

4.5 Identity as a High-Consequence Trust Layer

Identity is one of the most critical and underappreciated parts of the post-quantum attack surface.

Modern digital systems rely on identity not only for human access, but for service authentication, workload trust, API authorization, system-to-system communication, device enrollment, code signing, provisioning, orchestration, remote management, and administrative delegation. In most environments, these functions are implemented through PKI, certificate chains, key pairs, hardware-backed credentials, federation protocols, and token services.

This means identity is deeply dependent on asymmetric cryptography.

When organizations discuss post-quantum migration, they often focus first on encryption of data in transit. In practice, identity may prove even more difficult. Machine identities now outnumber human identities by orders of magnitude in many environments. Certificates are embedded in applications, appliances, services, containers, mobile devices, network infrastructure, and management systems. Many organizations do not have a complete inventory of these identities, much less an automated lifecycle process for replacing them at scale.

There are several technical reasons identity deserves special attention:

- certificate reissuance at scale is operationally difficult
- trust chains may span internal and external domains
- identity protocols may assume message sizes and latency profiles that change under PQC
- tokens and federation services may depend indirectly on vulnerable signing infrastructure
- machine-to-machine trust frequently lacks the governance applied to user access

If identity migration lags behind the rest of the environment, organizations can end up with an inconsistent trust architecture. Systems may support post-quantum encryption while still depending on classical certificates for authentication and integrity. That is not a secure end state. It is a partial migration with hidden failure points.

Identity is not just another layer in the stack. It is the mechanism by which the stack authorizes itself.

4.6 Networking and the Exposure of Encrypted Transit

The network layer remains central to both real-time operations and adversary collection. It includes core routing and switching infrastructure, firewalls, VPNs, load balancers, SD-WAN platforms, DNS, network access control, interconnect gateways, service meshes, east-west traffic paths, and edge connectivity.

In classical cyber defense, network security often focuses on segmentation, intrusion detection, access control, and traffic monitoring. In a post-quantum context, the network must also be understood as a collection surface.

Encrypted traffic is still traffic. It is observable, interceptable, storable, and classifiable. Session metadata, certificate negotiations, endpoint relationships, timing patterns, volume patterns, routing behavior, and protocol usage all provide intelligence value even before decryption. Once future decryption becomes feasible for captured payloads, the network becomes both a current reconnaissance layer and a future disclosure layer.

There are additional networking challenges tied to migration:

- network devices may have long refresh cycles
- embedded cryptographic functions in network appliances may not be easily upgraded
- management protocols, SSH sessions, VPN tunnels, and device certificates may all require separate migration paths
- operational teams may resist performance tradeoffs introduced by larger keys and new handshake models
- hybrid environments may require coexistence between classical and post-quantum protocols during transition

This coexistence period is especially important. Most organizations will not move from classical cryptography to post-quantum cryptography in a single event. They will operate hybrid trust models for years. That introduces downgrade risk, interoperability issues, and policy enforcement challenges at the network layer.

4.7 Supply Chain as a Multi-Layer Insertion Surface

Supply chain is often treated as an external issue, but technically it is an insertion surface that cuts across all internal layers.

A supply chain event can affect hardware provenance, firmware development, software dependencies, third-party APIs, managed services, cloud integrations, and update mechanisms. It can occur before deployment, during maintenance, during build and release, or through upstream

dependency compromise. In many cases, the resulting exposure is inherited silently by the downstream organization.

From a trusted systems perspective, supply chain risk is not limited to counterfeit hardware or banned vendors. It includes any condition in which the organization cannot establish confidence in origin, development control, update integrity, access pathways, or dependency inheritance.

The most difficult part of supply chain risk is asymmetry of knowledge. Suppliers know more about their internal development processes than customers do. Integrators know more about assembly steps than operators do. Service providers know more about internal access than clients do. The consuming organization must therefore establish trust under incomplete information.

This is why provenance, attestation, SBOMs, supplier intelligence, and contractual transparency requirements matter. They do not solve the problem completely, but they reduce the number of hidden trust assumptions built into the environment.

In a post-quantum world, supply chain risk becomes even more consequential because it can affect the mechanisms by which cryptographic migration itself is performed. If the systems, firmware, update channels, or signing environments used for migration are not trustworthy, the transition can introduce new compromise paths while attempting to solve old ones.

4.8 AI and Model-Mediated Exposure

Artificial intelligence expands the attack surface in two ways. First, it creates new locations where sensitive data is processed, stored, inferred from, or derived. Second, it introduces decision pathways that may influence access, prioritization, or operational action without full human inspection.

AI systems are not just applications. They are composed environments that may include prompts, context windows, embeddings, vector stores, model weights, orchestration logic, external tools, retrieval pipelines, logs, memory layers, and training or fine-tuning datasets. Each of these can become a location of persistence or leakage.

Several technical issues are especially relevant:

- prompts may contain sensitive operational data
- model outputs may reveal protected information indirectly
- retrieval systems may expose indexed content beyond intended boundaries
- model providers may retain data for telemetry, tuning, or abuse detection
- orchestration layers may invoke external tools or APIs with privileged context
- agentic systems may chain actions across systems faster than human review cycles

These issues intersect directly with trust. An AI workflow may appear to operate within policy while exposing data through logs, embeddings, intermediate reasoning artifacts, or downstream tool usage. It may also become dependent on identity, networking, and supply chain controls that were not originally designed for model-mediated actions.

From a post-quantum perspective, AI also increases the value of collected data. Historical datasets, model training corpora, and long-lived enterprise knowledge bases may become more strategically useful over time, not less. That makes them attractive targets for HNDL-style collection even if they are not immediately exploitable.

4.9 Autonomous and Edge Systems

The edge layer includes IoT devices, industrial systems, mobile endpoints, field sensors, remote platforms, smart infrastructure, robotics, medical devices, embedded defense systems, and distributed operational technology. Many of these systems are resource-constrained, difficult to patch, physically exposed, or deployed in environments with limited connectivity and extended lifecycles.

Autonomous behavior increases the risk profile further. Once systems begin sensing, deciding, and acting with reduced human latency, trust failures can propagate more quickly and with fewer opportunities for intervention.

Post-quantum transition in these environments is especially difficult because:

- devices may have limited compute and memory headroom
- firmware may be rarely updated
- hardware replacement cycles may be long
- device identity may be inconsistent or absent
- central visibility may be incomplete
- operational disruption from migration may be unacceptable

This means the edge and autonomous layer cannot be treated as an afterthought. In many sectors, it may become the slowest-moving and least-visible part of the migration, which makes it a likely concentration point for residual risk.

4.10 The Systemic Nature of the Attack Surface

The most important conclusion is that the attack surface is not additive. It is systemic.

Hardware trust affects firmware validation. Firmware trust affects software assurance. Software trust affects data handling. Identity trust affects access decisions across all layers. Network trust affects collection and disclosure risk. Supply chain trust affects every stage from origin to update. AI and autonomy amplify the speed, scale, and opacity of decisions made within the system.

This means an organization can be highly mature in one domain and still be systemically exposed.

A strong PKI program cannot compensate for compromised firmware. A cryptographic migration roadmap cannot compensate for unknown data copies. A secure hardware platform

cannot compensate for unmanaged machine identities or uncontrolled AI workflows. A clean compliance audit cannot compensate for long-term exposure that current controls were never designed to detect.

The full attack surface must therefore be understood as a trust surface.

The relevant question is not simply where an attacker might gain access.

The relevant question is where trust can be subverted, inherited incorrectly, or allowed to decay over time without detection.

That is the problem the remainder of this paper addresses.

5. State of Play: Converging Timelines and Expanding Risk

The dynamics described in the previous sections are difficult to fully understand when considered independently. Each element evolves on its own timeline, is governed by different constraints, and is often managed by different parts of the organization.

Post-quantum risk emerges not from any single factor, but from the interaction of three.

The first is adversary data collection, which is already active and operating at scale. The second is regulatory and policy enforcement, which is accelerating in defined stages and introducing mandatory timelines. The third is quantum capability, which continues to progress toward a threshold at which current cryptographic assumptions will no longer hold.

These timelines do not align by design.

They converge.

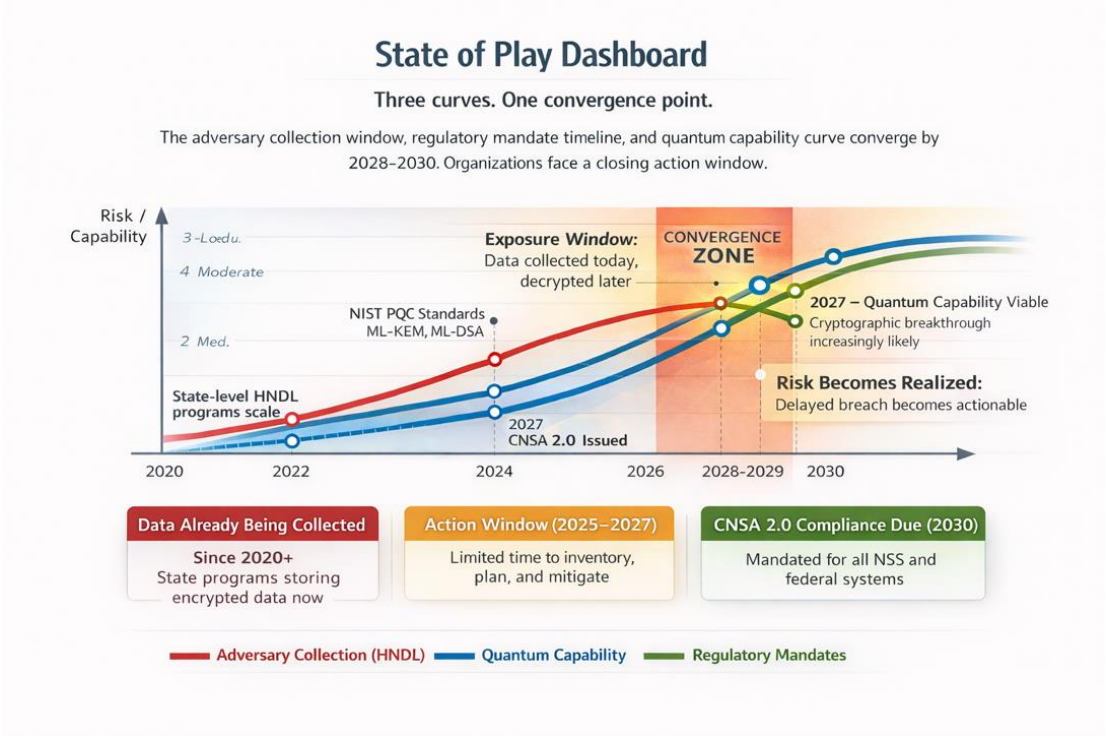


Figure 1. State of Play: Quantum Capability, Regulatory Mandates, and Adversary Collection

Description:
This figure illustrates the convergence of three independent timelines. The adversary collection curve reflects the ongoing capture and retention of encrypted data. The regulatory curve reflects increasing mandates for cryptographic inventory and post-quantum migration. The quantum capability curve reflects the progression toward cryptographically relevant quantum computing. The shaded region represents the exposure window in which data collected today may be decrypted in the future. The convergence zone highlights the period where capability, mandate, and accumulated exposure intersect.

The adversary collection curve is already elevated.

Data is being captured continuously across networks, endpoints, and external interfaces. This collection does not depend on immediate decryption capability. It is based on the assumption that stored data can be decrypted later, when computational methods allow it. From a technical standpoint, this is the most mature of the three curves. It is not speculative. It is operational.

The regulatory curve increases in discrete steps.

Mandates such as OMB M-23-02 require organizations to identify and inventory cryptographic systems. Guidance aligned to CNSA 2.0 establishes timelines for migration to quantum-resistant algorithms. Standards from National Institute of Standards and Technology define the technical basis for that transition. Each of these milestones introduces new requirements, compresses

timelines, and increases accountability. Unlike adversary activity, which is continuous, regulatory pressure arrives in phases, but each phase raises the baseline expectation.

The quantum capability curve progresses differently.

Its exact timing remains uncertain, but its direction is not. Advances in error correction, qubit stability, and algorithmic optimization continue to move the field toward cryptographically relevant thresholds. The critical point is not a specific year. It is the crossing of a capability boundary at which widely deployed public key systems can no longer be relied upon for confidentiality or integrity.

These curves intersect within a bounded period.

That period defines the exposure window.

Within this window, data of long-term value is collected under cryptographic protections that are expected to degrade. Organizations are simultaneously required to prepare for migration, often without complete visibility into their current state. As the window narrows, the margin for error decreases. Delays in inventory, planning, or execution directly increase the amount of data that enters a state of deferred exposure.

The convergence zone represents a structural shift in risk.

It is the point at which accumulated data collection, regulatory enforcement, and emerging technical capability align. At that point, exposure transitions from theoretical to actionable. Data that was previously protected by computational infeasibility becomes accessible under new conditions, without requiring additional compromise.

The most important implication is not tied to a future event.

It is tied to present conditions.

Data that enters the exposure window today cannot be retroactively protected once it has been collected. Even if future systems are fully migrated to post-quantum cryptography, historical data that was captured under previous assumptions may still be subject to decryption.

This changes the definition of security.

Security is no longer solely about preventing unauthorized access in the present. It is about limiting the amount of sensitive data that enters a state where future access becomes possible, regardless of current controls.

For organizations, this creates a requirement to think in terms of time, not just architecture.

Which data must remain confidential for ten years or more
Which systems cannot be easily migrated within that timeframe

Which identities, communications, and transactions create long-term exposure
Which layers of the environment introduce hidden persistence or duplication

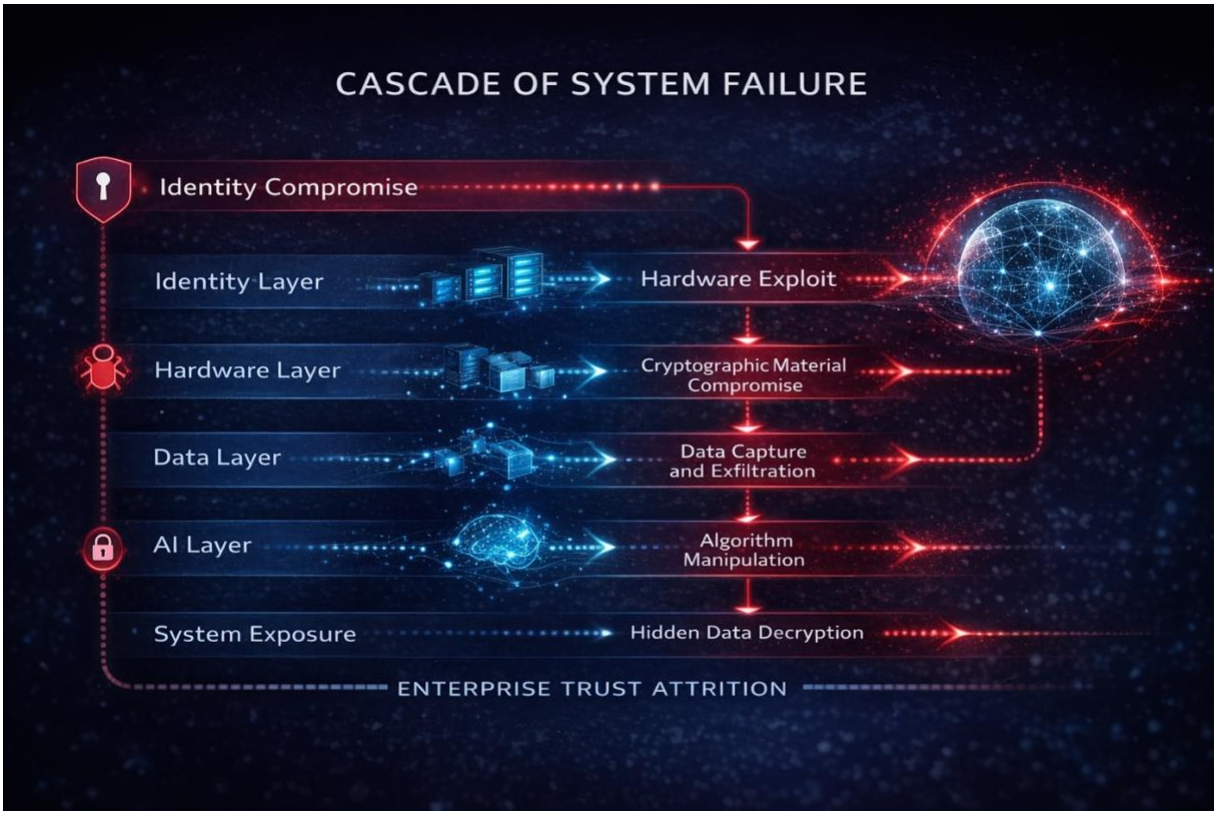
These questions are not theoretical. They are necessary inputs to any credible post-quantum strategy.

The State of Play is therefore not a forecast.

It is a representation of conditions that already exist and are continuing to evolve.

The remainder of this paper builds on this model by examining where trust breaks down across these layers and how it can be re-established as a system property rather than an assumed condition.

ACT II - WHY SYSTEMS FAIL



6. Identity as the First Point of Failure

Identity is the control plane of modern digital systems.

Every significant security function depends on it. Authentication, authorization, system-to-system communication, code signing, device enrollment, session establishment, API access, and administrative control all rely on the ability to establish and validate identity. In most environments, this validation is implemented through public key infrastructure, certificate chains, key pairs, token services, and hardware-backed credentials.

From a systems perspective, identity is not an isolated layer. It is the mechanism through which all other layers assert trust.

This makes it one of the highest consequence points of failure in a post-quantum context.

6.1 Cryptographic Dependence of Identity Systems

Modern identity systems are fundamentally dependent on asymmetric cryptography.

Protocols such as TLS, mutual TLS, S/MIME, SSH, and a wide range of federation and token-based systems rely on key exchange and digital signature mechanisms that are based on computational hardness assumptions. These include integer factorization and discrete logarithm problems, which are expected to be vulnerable to quantum algorithms once sufficient computational capability is available.

In practice, this means that identity validation today is anchored in cryptographic primitives that are not expected to remain secure over the full lifecycle of the data and systems they protect.

This creates a mismatch between identity validity and data sensitivity.

A certificate that is valid for one or two years may protect a transaction whose underlying data must remain confidential for a decade or more. If that transaction is captured and stored, the eventual compromise of the underlying cryptography affects not only the data, but the trust model used to authenticate the participants.

The identity layer does not degrade gracefully under these conditions. It fails categorically.

6.2 Scale and Fragmentation of Machine Identity

The challenge is amplified by the scale of machine identity.

In many enterprise and government environments, machine identities outnumber human identities by several orders of magnitude. These identities are embedded in servers, containers, microservices, APIs, network devices, storage systems, management platforms, and edge devices. Each identity may have its own certificate lifecycle, issuance process, renewal mechanism, and trust chain.

Unlike user identities, which are often centrally managed and governed, machine identities are frequently decentralized and application-specific. They are created by development teams, infrastructure automation, third-party products, and orchestration frameworks. As a result, organizations often lack a complete inventory of where machine identities exist, how they are used, and how they are validated.

This fragmentation introduces several technical risks:

- certificates that are not tracked or rotated
- hardcoded keys embedded in code or configuration
- inconsistent trust anchors across environments
- unmanaged certificate authorities or intermediate signers
- dependencies on legacy protocols that cannot support new cryptographic primitives

From a post-quantum perspective, this creates a migration problem at scale.

Transitioning identity systems to quantum-resistant algorithms requires coordinated reissuance of certificates, updates to trust chains, validation of protocol compatibility, and changes to

systems that consume or validate those identities. If the underlying inventory is incomplete, the migration will also be incomplete.

6.3 Identity Protocol and Performance Constraints

Post-quantum cryptographic algorithms introduce changes that directly affect identity protocols.

Many candidate algorithms involve larger key sizes, larger signatures, and different performance characteristics compared to classical elliptic curve or RSA-based systems. These changes impact handshake protocols, certificate sizes, latency, bandwidth utilization, and memory consumption.

In high-throughput environments, such as API gateways, service meshes, and distributed microservices architectures, even small increases in handshake cost can have measurable performance implications. Systems that were designed with tight assumptions about key sizes and message formats may require modification to support new algorithms.

There are also protocol-level considerations:

- TLS versions and extensions may need to support hybrid or post-quantum key exchange mechanisms
- certificate chains may grow in size, affecting transmission and validation
- legacy systems may not be able to parse or validate new certificate formats
- hardware accelerators may not support new primitives

These constraints mean that identity migration is not simply a cryptographic update. It is a protocol and system design challenge.

6.4 Trust Chains and Transitive Risk

Identity systems operate through chains of trust.

A root certificate authority signs intermediate authorities, which sign end-entity certificates, which are used by systems to authenticate and establish secure sessions. Trust is transitive. If a root or intermediate is compromised or invalidated, the entire chain is affected.

In a post-quantum scenario, the concern is not only compromise, but invalidation.

If the algorithms used to generate or validate signatures in the chain are no longer secure, then the assurance provided by the chain collapses. This affects not only active sessions, but historical records that relied on those signatures for integrity and non-repudiation.

This is particularly important for:

- code signing and software integrity verification
- firmware updates and secure boot processes

- audit logs and compliance records
- digitally signed documents and transactions

In each case, the trustworthiness of past actions may depend on cryptographic assurances that degrade over time.

6.5 Identity in Distributed and Federated Environments

Modern identity systems rarely exist within a single domain.

Organizations rely on federated identity models, third-party identity providers, cloud-based authentication services, and cross-domain trust relationships. Tokens issued in one environment are validated in another. Certificates issued by one authority are trusted by multiple systems across organizational boundaries.

This distribution introduces coordination challenges.

Migration to post-quantum identity requires alignment across:

- internal identity systems
- external identity providers
- cloud platforms
- partner organizations
- software vendors

If any part of the trust chain remains on vulnerable cryptography, it can become a point of failure. Hybrid environments, where some systems support post-quantum algorithms and others do not, introduce additional complexity. These environments must manage interoperability while preventing downgrade attacks and ensuring that weaker cryptographic paths are not used.

6.6 Identity as a System-Level Risk Amplifier

Identity does not simply fail in isolation. It amplifies failures across the system.

A compromised or invalid identity can:

- grant unauthorized access to sensitive data
- enable lateral movement across systems
- allow injection of malicious code or configuration
- bypass segmentation and network controls
- undermine audit and accountability mechanisms

In a delayed exploitation model, identity-related data such as certificate exchanges, authentication events, and key negotiations can also be captured and later analyzed or decrypted.

This provides adversaries with insight into system relationships, access patterns, and operational behavior, even before full decryption of protected payloads.

This dual role, both as a control mechanism and as a source of intelligence, makes identity a focal point for both immediate and future risk.

6.7 Implications for Post-Quantum Transition

The implications are direct.

Identity cannot be treated as a downstream dependency in post-quantum migration. It must be addressed as a primary domain, with its own inventory, roadmap, and governance model.

This includes:

- establishing a complete inventory of certificates, keys, and identity relationships
- implementing automated lifecycle management for machine identities
- evaluating protocol and system compatibility with post-quantum algorithms
- designing hybrid trust models that support transition without introducing downgrade risk
- coordinating identity changes across internal and external domains
- ensuring that hardware-backed identity mechanisms can evolve to support new cryptographic primitives

Most importantly, it requires a shift in perspective.

Identity is not just an access control mechanism.

It is the foundation on which trust is asserted across the entire system.

If that foundation is not made resilient to post-quantum conditions, then improvements in other areas will not prevent systemic failure.

7. Supply Chain and Provenance Failures

The integrity of a system is established long before it is deployed.

Every component within a modern environment originates from a chain of design, development, manufacturing, assembly, distribution, and integration processes that span multiple organizations and geographies. This includes hardware components, firmware images, operating systems, application code, libraries, container images, orchestration templates, and update mechanisms.

From a security perspective, this chain is not a linear process. It is a layered and branching system in which trust is inherited across multiple boundaries.

Each boundary introduces uncertainty.

7.1 Provenance as a Technical Requirement

Provenance is the ability to establish the origin, integrity, and custody of a component throughout its lifecycle.

In practice, most organizations do not have complete provenance for the systems they operate. They may know the vendor of record, but not the origin of subcomponents, the environment in which firmware was developed, the dependencies included in software builds, or the controls applied during assembly and distribution.

This creates a condition in which systems are trusted by assertion rather than verification.

From a post-quantum perspective, this matters for two reasons.

First, trust anchors such as firmware signing keys, code signing certificates, and device identities are established somewhere within the supply chain. If those anchors are compromised or generated within untrusted environments, then the resulting systems may carry valid signatures that do not represent valid trust.

Second, the process of migrating to post-quantum cryptography depends on the same supply chain mechanisms. If the systems responsible for generating, signing, distributing, and validating new cryptographic material are not trustworthy, then the transition itself becomes a vector for compromise.

Provenance is therefore not an abstract compliance concept. It is a technical prerequisite for establishing whether trust assumptions are valid.

7.2 Hardware Supply Chain and Embedded Risk

Hardware supply chains are complex and multi-tiered.

A single system may include components sourced from multiple manufacturers, assembled in different facilities, integrated with firmware developed by separate teams, and distributed through various logistics channels. At each stage, there is potential for modification, substitution, or insertion.

Unlike software, hardware modifications are often difficult to detect after the fact. A compromised component can introduce behavior that is not visible at the operating system level, including:

- hidden communication channels
- altered instruction paths
- modified firmware interfaces
- unauthorized debug or management access
- persistent backdoors that survive reinstallation or reset

These risks are not hypothetical. They are inherent to the complexity of modern manufacturing and integration processes.

From a trusted systems perspective, the issue is not whether a specific vendor is trustworthy. It is whether the organization can establish a verifiable chain of custody and integrity for the components it depends on.

This is particularly important for systems with long lifecycles, where hardware deployed today may remain in operation well into the period where post-quantum threats become operationally relevant.

7.3 Firmware Development and Update Pathways

Firmware represents the intersection of hardware and software within the supply chain.

It is developed in specialized environments, often by vendors or partners, and distributed through update mechanisms that may include signed images, staged rollouts, and remote management systems. Firmware controls device initialization, hardware interaction, and in many cases security enforcement at the lowest levels.

The supply chain risks associated with firmware include:

- compromise of development environments
- unauthorized modification of firmware images prior to signing
- compromise of signing keys or certificate authorities
- insertion of malicious logic during build or integration
- manipulation of update distribution channels

These risks are particularly concerning because firmware is typically trusted implicitly by higher layers of the system. If the firmware is compromised, it can present a trusted interface while performing untrusted actions.

From a post-quantum perspective, firmware introduces additional complexity.

Firmware signing and validation mechanisms often rely on cryptographic algorithms that must be updated. The ability to securely distribute and verify firmware updates depends on maintaining trust in the signing infrastructure. If that infrastructure is compromised or not upgraded to support new algorithms, the organization may be unable to establish trust in future updates.

This creates a dependency loop.

The system relies on firmware to establish trust. Firmware relies on cryptographic mechanisms that must be updated. The update process relies on trusted infrastructure that itself may depend on the system being trusted.

Breaking this loop requires explicit design for cryptographic agility and verifiable update pathways.

7.4 Software Supply Chain and Transitive Trust

Software supply chains introduce a different but equally complex set of risks.

Modern software is rarely developed from first principles. It is assembled from a combination of proprietary code, open source components, third-party libraries, container images, and external services. Each of these components may have its own dependencies, creating a transitive dependency graph that is difficult to fully enumerate.

This introduces the concept of transitive trust.

An organization may trust its internal development process, but that process may depend on external components that are not fully understood or controlled. A vulnerability or compromise in any upstream component can propagate downstream, affecting systems that appear to be secure.

Key risk areas include:

- compromised open source packages or repositories
- malicious or vulnerable dependencies introduced through automated builds
- tampering with artifacts in registries or distribution channels
- compromise of CI/CD pipelines used to build and deploy software
- inadequate validation of third-party code or updates

Efforts such as software bills of materials and signed artifacts improve visibility, but they do not eliminate the problem. They provide a map of dependencies, but not necessarily assurance of their integrity.

From a post-quantum perspective, the software supply chain must also support migration.

Libraries must implement new algorithms correctly. Protocols must be updated without introducing incompatibilities. Applications must be tested under new cryptographic conditions. All of this must occur while maintaining confidence that the components used to perform these updates are themselves trustworthy.

7.5 Supply Chain as a Persistence Mechanism

One of the most significant aspects of supply chain compromise is persistence.

Unlike runtime attacks, which may be detected and remediated, supply chain compromises can be embedded into systems before they are deployed. Once deployed, they may appear as legitimate functionality, signed code, or expected behavior.

This persistence can take several forms:

- firmware that reintroduces compromised behavior after system reset
- software updates that re-establish unauthorized access
- libraries that contain hidden functionality triggered under specific conditions
- management tools that provide covert access through legitimate interfaces

In a delayed exploitation model, this persistence is particularly valuable.

A compromised component can enable ongoing data collection without triggering immediate alarms. It can also provide a pathway for future access or manipulation once external conditions, such as the availability of quantum decryption capabilities, make that access more valuable.

7.6 Supply Chain Visibility and Intelligence

Addressing supply chain risk requires more than static assessment.

Traditional approaches rely on vendor questionnaires, certifications, and periodic audits. These provide a snapshot of conditions at a point in time, but they do not reflect the dynamic nature of supply chains.

Supply chains evolve continuously. New dependencies are introduced. Development teams change. infrastructure is updated. external threats shift. A static assessment cannot capture these changes.

This is why supply chain intelligence platforms, such as Exiger, are increasingly relevant. These platforms aggregate data on suppliers, dependencies, geopolitical risk, compliance status, and operational behavior to provide a more continuous view of supply chain exposure.

However, visibility alone is not sufficient.

Organizations must be able to act on this information by:

- prioritizing high-risk components and suppliers
- enforcing security requirements in contracts and procurement
- validating integrity through technical controls such as attestation and signed artifacts
- integrating supply chain intelligence into operational decision-making

7.7 Implications for Trusted Systems

The implications are systemic.

Supply chain is not a separate risk domain. It is the mechanism through which all other domains inherit risk.

A hardware platform inherits the integrity of its components and firmware. A software system inherits the integrity of its dependencies and build process. Identity systems inherit the integrity of their certificate authorities and key generation processes. Data systems inherit the integrity of the platforms on which they operate.

If provenance cannot be established, then trust cannot be validated.

From a post-quantum perspective, this has a direct consequence.

Organizations are preparing to replace cryptographic primitives that underpin trust across the system. If the supply chain mechanisms used to implement that replacement are not themselves trustworthy, the transition will not reduce risk. It will redistribute it.

The question is not simply whether an organization can deploy post-quantum cryptography.

The question is whether it can do so using components, processes, and pathways that it can verify and trust.

8. Human and Organizational Failure

The technical challenges described in the previous sections are significant, but they are not the primary reason most organizations will struggle with post-quantum transition.

The primary reason is organizational.

Modern enterprises and government agencies are not structured to manage trust as a system property. Responsibility for the components that establish and maintain trust is distributed across multiple functions, each with its own priorities, tooling, and operational model. As a result, critical dependencies are fragmented, and no single authority has a complete view of how trust is established across the environment.

This fragmentation creates conditions in which technically sound controls fail to produce a secure system.

8.1 Fragmentation of Responsibility

Cryptographic systems, identity infrastructure, hardware platforms, software development, network operations, and data governance are typically owned by different teams.

- Security organizations define policy, manage risk, and oversee compliance.
- Infrastructure teams manage hardware, operating systems, and core services.
- Application teams develop and maintain software and APIs.
- DevOps and platform engineering teams control build pipelines and deployment.
- Data teams manage storage, analytics, and governance.
- Procurement and vendor management oversee supply chain relationships.

Each of these domains interacts with cryptographic mechanisms and trust assumptions, but none of them owns the full lifecycle.

This leads to a set of predictable failure patterns:

- cryptographic dependencies embedded in applications without central visibility
- certificates issued and managed outside of enterprise identity systems
- hardware platforms deployed without alignment to long-term cryptographic requirements
- data retention policies that do not account for future decryption risk
- supply chain decisions made without technical validation of trust assumptions

From a system perspective, these are not isolated issues. They are manifestations of a single problem.

No function owns trust end to end.

8.2 Lack of Cryptographic Inventory and Visibility

Most organizations do not have a complete inventory of their cryptographic assets.

This includes:

- where encryption is used in applications and protocols
- which algorithms are in use
- where keys are generated, stored, and rotated
- how certificates are issued and validated
- how identity is established for users, devices, and services

Without this inventory, it is not possible to plan or execute a coherent migration.

Regulatory directives such as OMB M-23-02 are forcing organizations to begin this process, but the effort is often more complex than anticipated. Cryptographic usage is frequently embedded deep within systems, inherited through dependencies, or implemented in ways that are not centrally documented.

The result is a visibility gap.

Organizations cannot fully assess their exposure, cannot prioritize migration effectively, and cannot validate that changes have been applied consistently.

8.3 Key Management as a Persistent Weak Point

Key management has historically been one of the most difficult areas to standardize and enforce.

Keys are generated in different contexts, stored in different systems, and used for different purposes. Some are protected by hardware security modules. Others are stored in application configuration files, environment variables, or unmanaged keystores. Rotation policies vary, and enforcement is often inconsistent.

In a post-quantum context, key management becomes even more complex.

- new algorithms introduce new key formats and sizes
- migration requires reissuance and redistribution of keys at scale
- hybrid environments require coexistence of classical and post-quantum keys
- systems must be able to distinguish and enforce correct usage

If key management practices are not already mature, introducing new cryptographic mechanisms will amplify existing weaknesses.

8.4 Siloed Migration Efforts

A common failure mode is the initiation of migration efforts within a single domain.

For example:

- a security team may begin evaluating post-quantum algorithms
- an infrastructure team may plan hardware refresh cycles
- an application team may update libraries or frameworks

These efforts may be technically sound within their scope, but they are not coordinated.

Without coordination, several risks emerge:

- incompatible implementations across systems
- inconsistent trust models between services
- partial migrations that leave critical dependencies unchanged
- introduction of downgrade paths or interoperability gaps

Post-quantum transition requires synchronized change across multiple layers. A fragmented approach increases complexity without reducing risk.

8.5 Absence of Ownership for Post-Quantum Transition

Perhaps the most significant organizational gap is the absence of clear ownership.

Post-quantum transition sits at the intersection of security, architecture, infrastructure, and application development. It does not map cleanly to a single existing role.

As a result, organizations often delay assigning responsibility or distribute it informally. This leads to slow decision-making, unclear priorities, and lack of accountability.

A system-level problem requires system-level ownership.

Without it, even well-resourced organizations can fail to make meaningful progress.

8.6 Governance as a Technical Control

Governance is often viewed as a policy or compliance function. In this context, it must be treated as a technical control.

Effective governance defines:

- who is responsible for inventory, migration, and validation
- how decisions are made and prioritized
- how dependencies are identified and managed
- how progress is measured and enforced

Without this structure, technical controls cannot be applied consistently across the system.

The following model illustrates a practical distribution of responsibility.

PQC Governance Model

Function	Primary Responsibility
Security (CISO)	Define cryptographic policy, risk thresholds, and inventory requirements
Technology (CTO)	Align infrastructure and hardware roadmap with post-quantum requirements
Enterprise Architecture	Define transition strategy, system dependencies, and target state
Application and Platform Teams	Implement cryptographic changes in software and services
Data and Governance Teams	Define data sensitivity, retention, and exposure priorities
Procurement and Vendor Management	Enforce supply chain requirements and validate vendor capabilities
Operations (IT / DevOps)	Execute deployment, monitor systems, and enforce lifecycle management

This model does not eliminate complexity, but it makes it manageable.

It creates a structure in which responsibilities are explicit, dependencies are visible, and progress can be measured.

8.7 Implications for Trusted Systems

The technical and organizational dimensions of the problem are inseparable.

A system may have strong hardware controls, well-designed software, and modern cryptographic implementations. If the organization managing that system cannot coordinate changes, maintain visibility, or enforce consistent practices, the result will still be failure.

Conversely, organizations that establish clear ownership, maintain accurate inventories, and coordinate across domains can significantly reduce risk, even in complex environments.

The transition to post-quantum security is not simply a technical upgrade.

It is an organizational transformation that must align people, processes, and systems around a common objective.

Without that alignment, the technical solutions described in this paper cannot be applied effectively.

9. Hardware and System Integrity Failures

All higher-level security controls assume the integrity of the underlying system.

Operating systems assume that the hardware they execute on is functioning as designed. Applications assume that the operating system is enforcing isolation correctly. Identity systems assume that credentials are being processed and protected within a trusted execution environment. Cryptographic operations assume that keys are generated, stored, and used in a secure context.

If these assumptions are invalid, the controls built on top of them are weakened or rendered ineffective.

This makes hardware and system integrity a foundational requirement for trusted systems.

9.1 Hardware as the Root of Trust

In most architectures, the hardware platform serves as the initial root of trust.

This includes mechanisms such as:

- secure boot chains anchored in hardware
- trusted platform modules or equivalent secure elements
- hardware-based key storage
- measured boot and attestation capabilities
- embedded management controllers with independent control paths

These mechanisms are designed to establish an initial state of trust that can be extended upward through firmware, operating system, and application layers.

The effectiveness of this model depends on two conditions.

First, the hardware must be trustworthy at the time of deployment. Second, it must remain trustworthy throughout its operational lifecycle.

Both conditions are increasingly difficult to guarantee.

Hardware is produced through complex global supply chains, integrated across multiple vendors, and deployed in environments where physical and logical access cannot always be fully controlled. Once deployed, hardware may receive firmware updates, configuration changes, and maintenance interventions that alter its behavior over time.

From a post-quantum perspective, the root of trust must also be cryptographically durable.

If the mechanisms used to establish hardware identity, validate firmware, or attest to system state rely on cryptographic algorithms that become vulnerable, then the trust they provide degrades. This does not necessarily result in immediate compromise, but it introduces uncertainty into a layer that is assumed to be deterministic.

9.2 Firmware as the Persistence Layer

Firmware is the layer at which hardware and software converge.

It initializes system components, enforces low-level security controls, and often retains privileged access to system resources. Firmware executes before the operating system and, in many cases, operates independently of it through embedded controllers.

Because of its position in the stack, firmware is a natural persistence layer.

A compromised firmware component can:

- survive operating system reinstallation
- intercept or modify boot processes
- manipulate hardware behavior
- bypass host-based security controls
- alter telemetry and monitoring outputs

Firmware is also less visible than higher layers. Many security tools operate at the operating system or application level and have limited visibility into firmware behavior. Detection of firmware compromise often requires specialized tooling, hardware-level inspection, or comparison against known-good baselines.

From a lifecycle perspective, firmware updates are less frequent and more tightly controlled than software updates. This creates a tension between stability and security.

- infrequent updates increase the risk of unpatched vulnerabilities
- frequent updates increase the risk of introducing compromised or unverified code

From a post-quantum standpoint, firmware introduces additional requirements.

- firmware signing mechanisms must support new cryptographic algorithms
- update processes must ensure integrity under new trust models
- devices must be able to validate signatures and identities using updated primitives
- legacy devices that cannot be updated may become permanently non-compliant

This creates a segmentation challenge. Organizations may need to differentiate between systems that can be made post-quantum ready and those that must be isolated, replaced, or retired.

9.3 The Management Plane as a Control Surface

Modern systems include a management plane that operates independently of the primary compute environment.

This includes:

- baseboard management controllers
- out-of-band management interfaces
- remote provisioning systems
- lifecycle management platforms

These components are designed to provide administrative control regardless of the state of the operating system. They are used for provisioning, monitoring, recovery, and maintenance.

Because they operate with high privilege and independent access, they represent a powerful control surface.

If the management plane is compromised, an adversary can:

- access system state without interacting with the operating system
- modify firmware or configuration remotely
- capture or inject data
- disrupt or disable security controls
- maintain persistent access across reboots and updates

The security of the management plane depends on:

- strong identity and authentication mechanisms
- secure communication channels

- integrity of firmware and update processes
- isolation from untrusted networks

From a post-quantum perspective, the management plane must also support cryptographic migration.

- authentication mechanisms must transition to quantum-resistant algorithms
- device identities must be reissued and validated
- secure channels must support new key exchange methods

If the management plane cannot be trusted, it becomes a systemic vulnerability.

9.4 Cryptographic Agility at the Hardware Layer

A critical requirement for post-quantum transition is cryptographic agility.

Systems must be able to:

- support multiple cryptographic algorithms
- transition between algorithms without full system replacement
- validate and enforce algorithm selection across components
- maintain compatibility during hybrid operation

At the hardware layer, this introduces several constraints.

- hardware accelerators may be optimized for specific algorithms
- secure elements may have limited flexibility for new primitives
- firmware interfaces may not support dynamic algorithm selection
- legacy systems may lack the ability to update cryptographic functionality

If hardware platforms cannot support cryptographic agility, organizations are forced into binary decisions.

- continue operating with vulnerable algorithms
- or replace hardware at scale

Neither option is desirable at enterprise or government scale.

This is why newer platforms that incorporate quantum-resistant capabilities at the hardware and firmware level are significant. They provide a path for transition that does not rely solely on software-layer updates.

9.5 Integrity Over Time

Integrity is not a point-in-time property. It is a lifecycle property.

A system may be deployed in a trusted state and then diverge over time due to:

- firmware updates
- configuration changes
- patching cycles
- hardware replacements
- operational interventions

Each change introduces the possibility of drift from the original trusted state.

In a traditional model, integrity is verified at boot or during periodic checks. In a trusted systems model, integrity must be continuously validated.

This requires:

- measurement of system state at multiple points in time
- comparison against known-good baselines
- detection of unauthorized or unexpected changes
- ability to re-establish a trusted state when drift is detected

From a post-quantum perspective, integrity validation must also remain trustworthy as cryptographic assumptions evolve.

9.6 Implications for Trusted Systems

Hardware and system integrity are not background concerns. They are foundational.

If the root of trust is weak, higher-level controls cannot compensate. If firmware can be manipulated, software assurances are limited. If the management plane is not secure, administrative control cannot be trusted. If cryptographic mechanisms at the hardware layer cannot evolve, the system cannot adapt to new threat conditions.

This leads to a clear implication.

Post-quantum readiness must include the ability to establish and maintain trust at the lowest levels of the system.

This includes:

- verifiable hardware provenance
- secure and updateable firmware
- protected and validated management interfaces
- hardware-supported cryptographic agility
- continuous integrity measurement and attestation

Organizations that treat hardware as a static foundation will struggle to adapt.

Organizations that treat hardware as an active component of the trust model will have a viable path forward.

10. AI, Autonomy, and Algorithmic Risk

Artificial intelligence introduces a new class of risk that is not confined to data exposure or system compromise. It changes how decisions are made, how trust is propagated, and how actions are executed within the environment.

In traditional systems, security controls are designed around human-mediated workflows. Access is requested, validated, and granted. Actions are logged and reviewed. Even automated processes are typically bounded by predefined logic and deterministic behavior.

AI systems operate differently.

They ingest large volumes of data, generate outputs based on probabilistic models, and increasingly trigger downstream actions without requiring direct human approval. As organizations integrate AI into operational workflows, these systems become part of the control plane.

This introduces a shift in the attack surface.

The question is no longer only whether a system can be accessed or data can be decrypted. The question is whether the systems making decisions can be influenced, misled, or exploited in ways that alter outcomes at scale.

10.1 Model-Centric Data Exposure

AI systems are built on data.

Training datasets, fine-tuning inputs, retrieval sources, and prompt interactions all contribute to the behavior of the model. These data sources often include sensitive operational information, proprietary knowledge, user interactions, and derived insights.

From a technical standpoint, AI introduces multiple new persistence layers:

- training data stored in model development environments
- fine-tuning datasets specific to organizational use cases
- embeddings and vector representations used for retrieval
- logs of prompts and responses
- cached context and memory structures in agentic systems

Each of these layers can retain information beyond its original context.

In a post-quantum scenario, this becomes significant.

Data embedded in training corpora or derived representations may remain valuable over long periods. If captured or accessed, it may reveal patterns, relationships, or sensitive content that can be exploited once decryption capabilities evolve. Unlike structured data stores, these representations may not be easily discoverable or governed.

The result is an expansion of the data exposure surface into areas that are not traditionally treated as persistent storage.

10.2 Prompt and Inference Pathways

Prompts and inference interactions are a primary interface to AI systems.

These interactions may include:

- user-submitted queries containing sensitive data
- system-generated prompts that include operational context
- chained prompts in multi-step workflows
- outputs that may contain synthesized or inferred information

From a security perspective, prompts are both inputs and potential exfiltration vectors.

Sensitive information included in prompts may be:

- logged for monitoring or debugging
- retained for model improvement
- processed by external systems or APIs
- exposed through subsequent outputs

Inference pathways also introduce indirect leakage risks.

A model may not explicitly return sensitive data, but it may reveal information through inference, correlation, or pattern recognition. This is particularly relevant in systems that integrate internal knowledge bases, retrieval-augmented generation, or domain-specific training.

From a post-quantum perspective, these interactions create additional artifacts that may be collected and stored.

- encrypted prompt traffic can be intercepted and retained
- inference logs may persist in external systems
- derived outputs may expose structured or unstructured knowledge

These artifacts become part of the long-term exposure surface.

10.3 Autonomous Decision Loops

As AI systems become more integrated into operations, they are increasingly used to automate decision-making.

Examples include:

- automated threat detection and response
- resource allocation and scheduling
- financial transaction processing
- supply chain optimization
- infrastructure scaling and orchestration

In these systems, decisions are not always deterministic. They are influenced by model outputs, contextual data, and dynamic inputs.

This introduces a new risk model.

An adversary does not need to directly access a system to influence it. They may be able to:

- manipulate input data
- influence training datasets
- exploit model biases or weaknesses
- inject adversarial examples
- alter environmental signals

These actions can shift the behavior of the system without triggering traditional access controls.

The result is an indirect attack surface.

Instead of breaking into a system, an adversary shapes its decisions.

10.4 Algorithm vs Algorithm Dynamics

As systems become increasingly automated, the nature of conflict within digital environments shifts from human-directed interaction to machine-executed behavior.

In traditional security models, adversaries exploit vulnerabilities, and defenders respond through analysis, detection, and remediation. These processes are mediated by human decision-making, even when supported by automation.

In modern environments, that mediation is collapsing.

Systems now operate as interacting layers of algorithms:

- detection systems analyze telemetry and classify behavior
- response systems trigger actions based on predefined or learned conditions
- optimization systems adjust resource allocation and system configuration
- adversarial systems probe, adapt, and refine their methods continuously

These systems do not operate sequentially.

They operate concurrently.

This creates a condition in which algorithms are interacting with other algorithms in real time, often without human intervention.

From Event-Based Security to Continuous Interaction

In an algorithmic environment, the concept of a discrete attack event becomes less meaningful.

Instead of a single intrusion or exploit, adversarial influence can be applied incrementally:

- subtle manipulation of input data
- gradual shaping of system behavior
- exploitation of model assumptions
- iterative probing to map response boundaries

Each individual interaction may appear benign.

Collectively, they can alter system behavior in meaningful ways.

This is not an attack in the traditional sense.

It is a continuous interaction designed to influence outcomes.

Feedback Loops and System Amplification

A defining characteristic of algorithmic systems is the presence of feedback loops.

Outputs from one system become inputs to another:

- detection outputs feed response systems
- response actions generate new telemetry
- telemetry influences subsequent detection models
- optimization systems adjust behavior based on observed outcomes

In stable conditions, these loops improve efficiency and responsiveness.

Under adversarial influence, they can amplify small perturbations into large system effects.

For example:

- a slight shift in input data may alter classification thresholds
- altered thresholds may trigger different response actions
- those actions may change system state or data availability
- the resulting state reinforces the initial perturbation

This amplification can occur without any single control being violated.

Adversarial Adaptation

Adversarial systems are not static.

They observe system behavior and adapt.

This includes:

- identifying patterns in detection thresholds
- testing system responses to controlled inputs
- refining techniques to avoid triggering alerts
- exploiting timing, sequencing, and data dependencies

Over time, adversaries can develop an operational understanding of how a system behaves, even without direct access.

This creates an asymmetry.

The defender must secure the entire system.

The adversary needs only to identify exploitable patterns in behavior.

Interaction with Post-Quantum Risk

Algorithmic dynamics intersect with post-quantum risk in two critical ways.

1. Data Accumulation and Retrospective Analysis

The interactions between systems generate large volumes of data:

- telemetry
- logs
- model outputs
- decision traces

This data captures how systems behave under different conditions.

If collected and retained externally, it provides a detailed record of system dynamics.

In a post-quantum context, this data becomes more valuable over time.

Once decryption or advanced analysis becomes possible, adversaries can:

- reconstruct system behavior
- identify patterns and dependencies
- refine future attack strategies

This extends the concept of HNDL beyond static data to **behavioral intelligence**.

2. Acceleration of Decision Cycles

AI systems reduce the time between input and action.

This creates:

- faster response to legitimate conditions
- faster propagation of incorrect or manipulated inputs
- reduced opportunity for human validation

In such environments, errors or manipulations can propagate before they are detected.

When combined with future decryption capabilities, this creates a layered risk:

- immediate influence on system behavior
- delayed insight into how that behavior was shaped

Control Plane Implications

In an algorithm-driven environment, control shifts from direct intervention to constraint definition.

Organizations must define:

- acceptable input conditions
- boundaries for automated decision-making
- validation requirements for outputs
- escalation paths for uncertain conditions

This requires:

- monitoring not just outcomes, but decision pathways
- validating model behavior under adversarial conditions
- introducing checkpoints where human oversight is required

- limiting the scope of automated actions in high-risk contexts

Traditional perimeter and access controls do not address these requirements.

They operate at the wrong layer.

Detection Challenges

Detecting adversarial influence in algorithmic systems is fundamentally different from detecting traditional attacks.

Challenges include:

- distinguishing between normal variation and manipulated input
- identifying slow, incremental changes in system behavior
- correlating actions across multiple interacting systems
- attributing outcomes to specific inputs or decisions

These challenges are compounded by:

- limited visibility into model internals
- distributed system architectures
- high volumes of telemetry and interaction data

As a result, detection must shift from event-based alerts to pattern-based analysis over time.

Detection must therefore shift from event-based alerting to pattern-based analysis over time.

In practice, this means evaluating how systems behave across rolling time windows rather than reacting to isolated anomalies. For example, instead of triggering alerts on a single unexpected decision, organizations must monitor for gradual drift in model outputs, changes in decision thresholds, or evolving response patterns across similar inputs. These signals, while individually insignificant, can indicate sustained adversarial influence when observed collectively.

Implications for Trusted Systems

Algorithmic systems do not replace existing layers of risk.

They connect and accelerate them.

- identity decisions may be influenced by model outputs
- data exposure may occur through automated workflows
- system configurations may change dynamically
- supply chain dependencies may be integrated into automated processes

This creates a unified trust surface in which decisions, not just data, must be trusted.

Trusted systems must therefore include:

- governance over automated decision-making
- validation of inputs and outputs
- constraints on system behavior
- visibility into interaction patterns across systems

Trust must extend to how decisions are made, not just who or what is making them.

Closing Insight

In an algorithm-driven environment, control is no longer defined by access.

It is defined by influence.

The systems that will remain trustworthy are not those that react the fastest.

They are the ones that can constrain how decisions are made, even under conditions they do not fully control.

10.5 AI as a Trust Propagation Layer

AI systems do not operate in isolation.

They are integrated into workflows that span identity systems, data platforms, infrastructure, and external services. When an AI system makes a decision or generates an output, that output may be used to:

- grant or deny access
- trigger operational changes
- initiate transactions
- update configurations
- inform human decision-making

In effect, AI becomes a mechanism for propagating trust decisions.

If the inputs to the system are untrusted, or if the model behaves unpredictably under certain conditions, the resulting actions may reflect that uncertainty.

This introduces a challenge for trusted systems.

Trust must now account for:

- the integrity of input data

- the behavior of the model under different conditions
- the transparency of decision-making processes
- the validation of outputs before action is taken

Traditional controls do not fully address these requirements.

10.6 Constraints on Governance and Visibility

AI systems are difficult to fully govern using existing frameworks.

- models may be hosted externally or accessed through APIs
- internal models may be trained on evolving datasets
- decision logic may not be easily interpretable
- logging may capture only partial context
- interactions may span multiple systems and services

This creates gaps in visibility.

Organizations may not be able to:

- fully trace how a decision was made
- determine what data influenced that decision
- validate that outputs meet security or compliance requirements
- enforce consistent policies across all AI interactions

These gaps are not unique to AI, but they are amplified by its scale and speed.

10.7 Implications for Post-Quantum Risk

AI does not directly break cryptography, but it amplifies the consequences of its failure.

- it increases the volume and value of data being generated and processed
- it introduces new locations where data is stored and replicated
- it accelerates decision-making and system response
- it creates new pathways for influence and manipulation

In a delayed exploitation model, this means that more data enters the exposure window, and that data may be more valuable when decrypted.

It also means that systems relying on AI may be influenced in ways that are difficult to detect, even before quantum capabilities are realized.

10.8 Implications for Trusted Systems

Trusted systems must extend beyond traditional controls.

They must account for:

- how data is used and transformed by models
- how decisions are generated and validated
- how actions are triggered and controlled
- how trust is propagated through automated workflows

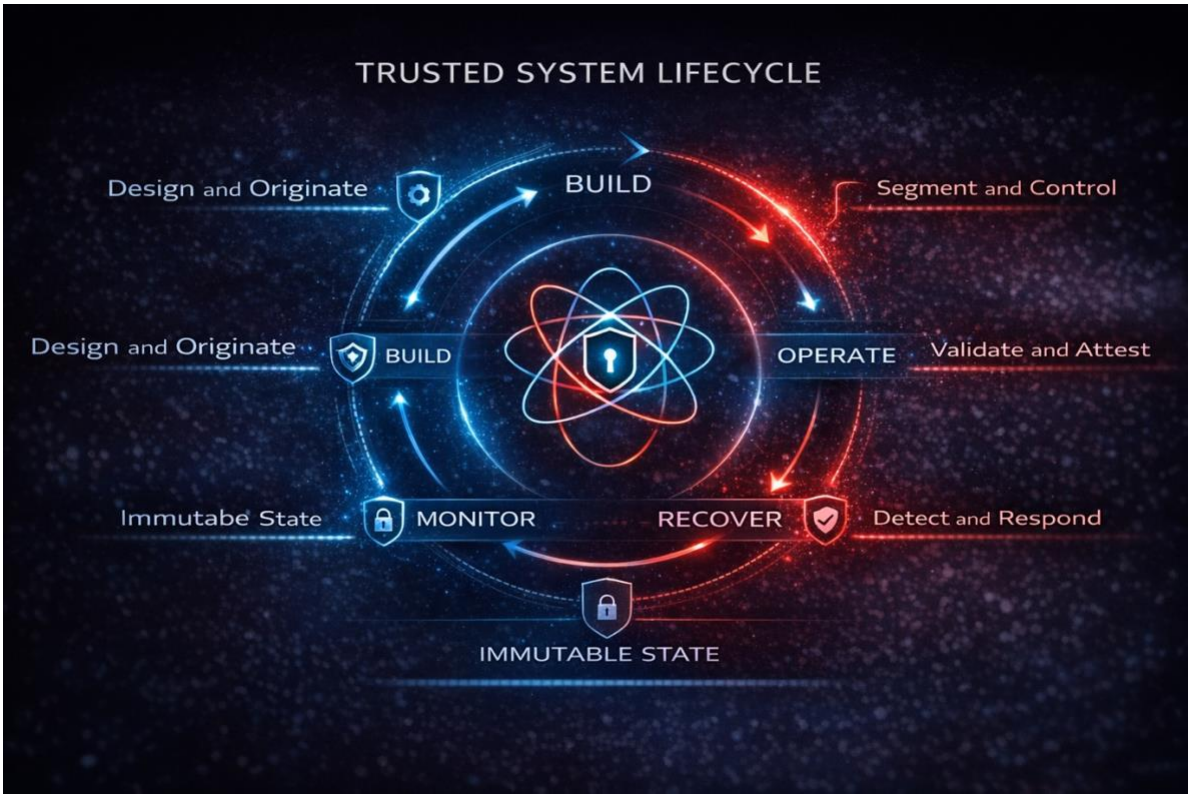
This requires:

- governance frameworks that include AI systems as first-class components
- validation mechanisms for model behavior and outputs
- controls on data input, retention, and reuse
- integration of AI systems into identity, data, and infrastructure trust models

AI does not replace existing layers of the attack surface.

It connects and accelerates them.

ACT III - THE TRUSTED SYSTEMS RESPONSE



11. The Trusted Systems Framework

The preceding sections demonstrate a consistent pattern.

Failures do not occur because a single control is weak. They occur because trust is assumed across layers that are not validated as a system.

Identity depends on cryptography that may not endure.
Supply chains introduce components that cannot be fully verified.
Hardware and firmware operate below the visibility of most controls.
Data is replicated beyond governance boundaries.
AI systems make decisions based on inputs that may not be trustworthy.
Organizational structures fragment responsibility across domains.

Each of these issues is typically addressed independently.

In practice, they are interdependent.

This creates a condition in which improving one domain does not eliminate risk if adjacent domains remain untrusted. A system can be compliant, patched, and monitored, and still be fundamentally exposed because trust is not established across the full environment.

The problem is not the absence of controls.

The problem is the absence of a system-level model for trust.

11.1 Defining Trust as a System Property

In traditional architectures, trust is often treated as an attribute of individual components.

- a device is trusted if it passes validation
- a user is trusted if authentication succeeds
- data is trusted if it is encrypted
- software is trusted if it is signed

These assertions are necessary, but not sufficient.

A trusted system requires that these conditions hold **simultaneously and continuously across all layers**. It also requires that the relationships between these layers are understood and validated.

Trust, in this context, must be treated as a system property.

A system property has three characteristics:

- it is established across multiple components
- it depends on the interaction of those components
- it must be maintained over time, not verified once

This leads to a different design requirement.

Trust must be engineered.

11.2 Framework Overview

The Trusted Systems Framework defines nine interdependent pillars that collectively establish and maintain trust across modern environments.

Figure 2. Trusted Systems Framework

Description:
This figure illustrates the nine pillars of the Trusted Systems Framework as an interconnected model. Each pillar represents a critical dimension of trust, and the relationships between them demonstrate how trust must be established and maintained across hardware, software, data, identity, and autonomous systems. The model emphasizes that trust is not isolated within any single layer, but emerges from coordinated assurance across the entire system.

These pillars are not independent controls or isolated capabilities. They are system-level functions that must operate together. Each pillar addresses a specific class of failure identified in the preceding sections, but their effectiveness depends on how they reinforce one another across the full architecture.

A system cannot be considered trusted if any of these pillars is absent or materially weak. Trust is not established through strength in a single domain. It emerges from coordinated assurance across all domains.

The Nine Pillars of Trusted Systems

1. Origin

Origin defines whether a system can establish trust at the point of creation. In a post-quantum environment, this extends beyond vendor identity to include verifiable provenance across

hardware, software, and supply chain pathways. Systems that cannot establish origin are forced to inherit trust assumptions that cannot be validated.

This includes hardware provenance, software source integrity, supplier validation, and chain of custody across all components.

Addresses: supply chain insertion and unknown dependencies

2. Integrity

Integrity defines whether a system remains in a known and trusted state throughout its lifecycle. It is not sufficient to verify integrity at deployment. Systems must be continuously validated to ensure that firmware, software, and configuration have not been altered in ways that undermine trust.

This includes secure and measured boot processes, firmware validation, code signing, and configuration integrity controls.

Addresses: firmware persistence and system drift

3. Visibility

Visibility defines whether an organization can observe and understand the state of its systems and the movement of data across them. Without visibility, trust cannot be measured, and exposure cannot be quantified. Most organizations operate with partial visibility, particularly across cryptographic usage, data flows, and system dependencies.

This includes cryptographic inventory, telemetry, dependency mapping, and data location and movement awareness.

Addresses: unknown exposure and lack of inventory

4. Identity

Identity defines how trust is asserted across users, devices, and services. It is the mechanism through which systems authenticate, authorize, and establish relationships. In a post-quantum context, identity systems must evolve to support new cryptographic models while maintaining continuity across distributed environments.

This includes PKI systems, certificate lifecycle management, machine identity governance, authentication protocols, and trust chain validation.

Addresses: identity failure and authentication breakdown

5. Control

Control defines whether policies can be enforced consistently across the system based on trust conditions. It is not enough to define policy. Systems must be able to restrict actions, segment environments, and enforce decisions in real time, even as conditions change.

This includes access control mechanisms, segmentation strategies, policy enforcement frameworks, and operational guardrails.

Addresses: unauthorized access and lateral movement

6. Resilience

Resilience defines whether a system can maintain or restore trusted operation under adverse conditions. In a delayed exploitation model, organizations must assume that some level of compromise may already exist. The ability to recover to a known and trusted state becomes a core requirement.

This includes backup and recovery architectures, immutable storage, system isolation, and validated restore processes.

Addresses: persistence and recovery after compromise

7. Adaptability

Adaptability defines whether a system can evolve in response to changing technical conditions without requiring complete replacement. Post-quantum transition depends on the ability to introduce new cryptographic mechanisms while maintaining operational continuity.

This includes cryptographic agility, modular system design, upgrade pathways, and support for hybrid environments during transition.

Addresses: inability to transition to post-quantum systems

8. Autonomy Governance

Autonomy governance defines whether automated and AI-driven systems operate within controlled and validated boundaries. As AI becomes part of the decision-making process, trust must extend beyond static controls to include how decisions are generated and executed.

This includes model validation, monitoring of inputs and outputs, human oversight mechanisms, and policy enforcement for automated actions.

Addresses: algorithmic manipulation and uncontrolled automation

9. Temporal Awareness

Temporal awareness defines whether an organization understands how risk evolves over time. This is the pillar that connects present-day security decisions to future exposure. Systems that are secure today may not remain secure over the lifespan of the data they protect.

This includes data sensitivity over time, exposure window analysis, retention strategy, and forward-looking risk modeling.

Addresses: delayed exploitation and HNDL exposure

Closing Bridge

These pillars do not operate independently.

A failure in origin can invalidate integrity. A lack of visibility can undermine control. Weak identity can compromise every other layer. Absence of temporal awareness can render otherwise strong systems ineffective over time.

Trust, in this model, is not asserted.

It is constructed, validated, and sustained across all nine dimensions.

11.3 Interdependence of the Pillars

The effectiveness of the framework depends on how these pillars interact.

A failure in one pillar can invalidate others.

- Strong identity controls cannot compensate for untrusted hardware origin
- Verified software integrity cannot compensate for compromised firmware
- Visibility cannot compensate for lack of control
- Control cannot compensate for lack of resilience
- Adaptability cannot compensate for absence of temporal awareness

This interdependence is why isolated improvements do not produce a trusted system.

Trust must be established across all pillars at a level appropriate to the sensitivity and lifecycle of the system.

11.4 Mapping the Framework to Failure Modes

Each pillar directly addresses failure modes identified earlier.

- Identity failures are addressed through identity lifecycle and cryptographic transition
- Supply chain failures are addressed through origin and integrity
- Organizational failures are addressed through visibility and control

- Hardware failures are addressed through integrity and adaptability
- AI-related risks are addressed through autonomy governance
- Delayed exposure is addressed through temporal awareness

This mapping is not conceptual. It provides a practical way to evaluate whether specific risks are being addressed or deferred.

11.5 From Controls to Systems

Most organizations measure security in terms of controls.

- number of vulnerabilities patched
- number of alerts detected
- compliance with standards
- coverage of security tools

These metrics do not measure whether the system is trusted.

The Trusted Systems Framework shifts the focus from control coverage to system assurance.

It asks:

- Can the origin of critical components be verified
- Can system integrity be measured and maintained over time
- Is there visibility into where data exists and how it moves
- Are identities managed consistently and at scale
- Can policies be enforced across all layers
- Can the system recover to a known trusted state
- Can the system adapt to new cryptographic requirements
- Are automated decisions governed and validated
- Is long-term exposure understood and managed

If these questions cannot be answered, trust is assumed.

11.6 Implications for Implementation

The framework is not a product or a single solution.

It is a structure for organizing how trust is established across the environment.

Implementation requires:

- aligning existing controls to the appropriate pillars
- identifying gaps where pillars are weak or absent
- prioritizing improvements based on data sensitivity and system criticality

- coordinating changes across technical and organizational domains

This structure also provides a foundation for measurement, which is addressed in the next section.

11.7 Transition to Maturity

The framework defines what must exist.

It does not, by itself, define how well it exists.

Organizations require a way to assess their current state and determine where to focus effort.

The next section introduces a maturity model that translates these pillars into measurable levels of capability and readiness.

12. Maturity Model: Measuring Trust Across the System

The Trusted Systems Framework defines the components required to establish trust across modern environments. It does not, by itself, indicate how effectively those components are implemented.

Organizations require a method to assess their current state, identify systemic weaknesses, and prioritize remediation efforts based on risk.

The Trusted Systems Maturity Model provides that method.

It translates the nine pillars of the framework into measurable levels of capability, allowing organizations to evaluate not only whether controls exist, but whether they operate consistently, at scale, and under changing conditions.

12.1 Purpose of the Maturity Model

The objective of the maturity model is not to produce a compliance score.

It is to expose gaps in system-level trust.

Traditional assessments focus on control presence. A system may pass an audit because encryption is enabled, access controls are configured, and policies are documented. These assessments do not determine whether trust can be sustained under conditions such as cryptographic transition, supply chain disruption, or delayed data exposure.

The maturity model addresses this by evaluating:

- completeness of implementation across each pillar

- consistency across systems and environments
- ability to operate at scale
- readiness for post-quantum transition
- resilience under failure conditions

This shifts the evaluation from static compliance to dynamic capability.

12.2 Maturity Levels

Each pillar is assessed across five levels of maturity.

Level 1 – Ad Hoc

Controls are minimal, inconsistent, or undocumented.
Trust is largely assumed.

Level 2 – Defined

Basic controls are implemented, but not uniformly enforced.
Visibility is partial and often manual.

Level 3 – Managed

Controls are standardized and implemented across key systems.
Processes are documented and repeatable.

Level 4 – Integrated

Controls are integrated across domains.
Dependencies between systems are understood and managed.

Level 5 – Optimized

Trust is continuously validated across all layers.
Systems are designed for adaptability, resilience, and long-term exposure management.

12.3 Pillar-Based Assessment

Each of the nine pillars is evaluated independently, but interpreted collectively.

For example:

- An organization may have **Level 4 Identity** but **Level 2 Origin**
- Or **Level 3 Integrity** but **Level 1 Temporal Awareness**

These imbalances are critical.

The lowest maturity pillar often defines the effective trust level of the system.

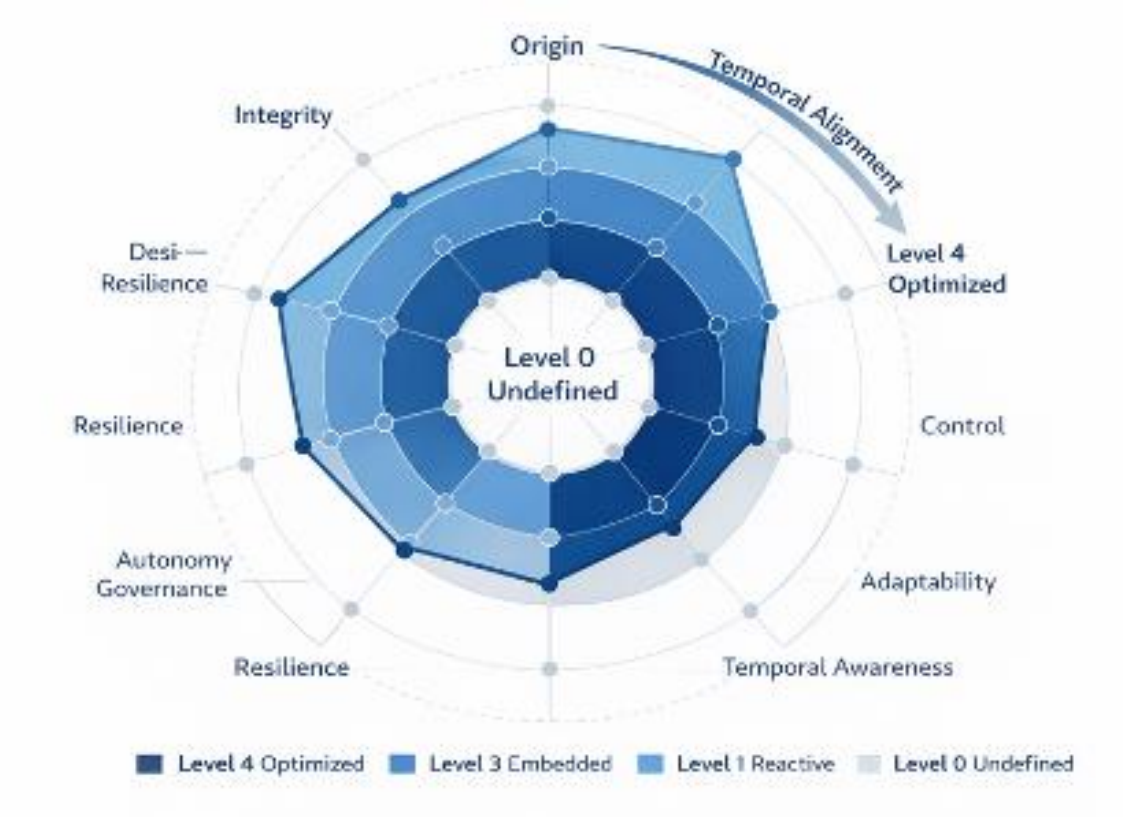


Figure 3. Trusted Systems Maturity Model

Description:
This figure illustrates the maturity of an organization across the nine pillars of the Trusted Systems Framework. The inner shape represents a typical current-state profile, reflecting uneven capability across domains. The outer shape represents a target-state profile in which trust is established consistently across hardware, software, data, identity, and autonomous systems.

The gaps between the two profiles highlight areas of systemic weakness and provide a visual prioritization of where capability must be strengthened to achieve system-level trust.

A full assessment tool is available as part of SecureFi Institute workshops and executive briefings.

12.4 Interpreting the Model

The maturity model should not be interpreted as a linear progression.

Organizations do not move uniformly from Level 1 to Level 5 across all pillars. Instead, maturity develops unevenly based on historical investments, operational priorities, and regulatory requirements.

This creates three common patterns:

Localized Strength

Strong capability in one or two domains, often driven by compliance or specific risk focus.

Systemic Gaps

Weakness in foundational areas such as origin, visibility, or temporal awareness that undermine other controls.

Fragmented Integration

Moderate capability across multiple domains without full coordination, resulting in inconsistent trust across systems.

From a post-quantum perspective, the third pattern is particularly risky. Partial migration or uneven capability can introduce new vulnerabilities, especially in hybrid environments where classical and post-quantum systems coexist.

12.5 Temporal Dimension of Maturity

Traditional maturity models evaluate systems based on their current state.

They assess whether controls are implemented, whether policies are enforced, and whether systems meet defined standards at a given point in time. This approach assumes that if a system is secure today, it can be considered secure for operational purposes.

That assumption no longer holds.

Post-quantum risk introduces a temporal dimension in which security must be evaluated over the full lifecycle of data, not just the current state of the system.

A system that is secure today but cannot maintain that security over time is not mature.

Security Over Time vs Security at a Point in Time

The distinction is fundamental.

Most organizations operate under a **point-in-time security model**:

- encryption is evaluated based on current strength
- identity is validated based on current certificates
- systems are assessed based on current configuration
- compliance is measured against current standards

In contrast, a **temporal security model** evaluates whether those same controls will remain valid for the duration in which the protected data retains value.

This introduces a new requirement:

Security controls must be aligned to the **lifespan of the data they protect**, not the lifespan of the system implementing them.

The Exposure Window

The concept of an exposure window becomes central to maturity.

The exposure window is defined as the period between:

- the moment data is created, transmitted, or stored
- and the point at which that data is no longer sensitive or valuable

In a post-quantum context, this window must also account for:

- the likelihood that encrypted data has already been collected
- the expected timeline for cryptographic degradation
- the ability of the organization to transition to new cryptographic models

If the exposure window exceeds the durability of the cryptographic controls protecting the data, then the system is not secure, regardless of its current state.

This is the condition created by Harvest Now, Decrypt Later.

Maturity Requires Temporal Alignment

Organizations must evaluate maturity based on whether their systems are aligned to this exposure window.

This requires answering a set of technical questions:

- How long must each class of data remain confidential
- Which cryptographic mechanisms protect that data today
- How long those mechanisms are expected to remain secure
- Whether the system can transition before those mechanisms degrade
- Whether historical data has already entered an exposure condition

These are not theoretical questions.

They are measurable, and they directly influence risk.

Temporal Gaps as a Source of Risk

A temporal gap exists when there is a mismatch between:

- the required confidentiality period of the data
- and the expected durability of the cryptographic controls

These gaps are often invisible in traditional assessments.

Examples include:

- financial records retained for regulatory purposes beyond cryptographic lifespan
- identity transactions that can be reconstructed from stored exchanges
- intellectual property that remains valuable for decades
- operational data that reveals long-term system behavior

In each case, the system may meet current security requirements while failing to meet future confidentiality requirements.

This is not a compliance failure.

It is a maturity failure.

Integration with the Maturity Model

Temporal awareness must be integrated across all nine pillars.

- Origin must consider whether components will remain trustworthy over time
- Integrity must account for drift and long-term validation
- Visibility must include data persistence and replication
- Identity must evolve with cryptographic transition
- Control must limit long-term exposure pathways
- Resilience must support recovery under new trust conditions
- Adaptability must enable migration before degradation occurs
- Autonomy governance must account for long-term data use and model behavior

Temporal awareness is not a separate control.

It is a condition that affects the effectiveness of every other pillar.

From Static Assessment to Forward-Looking Capability

The introduction of temporal awareness transforms the maturity model from a static assessment into a forward-looking capability.

Organizations must move from asking:

- "Are we secure today?"

to asking:

- "Will this system remain secure for as long as the data matters?"

This requires:

- modeling exposure over time
- prioritizing systems based on data lifespan
- aligning migration timelines with risk windows
- continuously reassessing assumptions as technology evolves

Organizations that incorporate this approach will naturally prioritize long-lived data, begin transition earlier for high-risk systems, and reduce the volume of data entering long-term exposure conditions.

Organizations that do not will continue to operate under a point-in-time model, accumulating exposure that is not visible in current assessments.

Closing Insight

The most important distinction is this:

A system that is secure today may already be compromised in the future.

Temporal maturity is the ability to recognize that condition and act before it becomes irreversible.

12.6 From Assessment to Action

The value of the maturity model lies in how it is used.

It enables organizations to:

- identify the lowest-maturity pillars that create systemic risk
- prioritize remediation based on data sensitivity and exposure duration
- align technical and organizational efforts across domains
- measure progress over time
- support decision-making for investment and architecture

This assessment becomes the foundation for structured action.

It connects directly to scenario analysis and informs the decisions outlined in the following sections.

12.7 Transition to Scenario Validation

The maturity model provides a static view of capability.

Real-world systems operate dynamically.

The next section applies the framework and maturity model to realistic scenarios, demonstrating how failures occur across multiple layers and how a trusted systems approach would alter those outcomes.

13. Scenario-Based Illustrations

Scenario 1: Federal Contractor Supply Chain Compromise

13.1 Situation Setup

A federal defense contractor operates a hybrid environment supporting classified-adjacent and sensitive unclassified workloads. The environment includes:

- on-premise data centers with modern server infrastructure
- cloud-based analytics platforms
- supplier-integrated software systems
- machine-to-machine communications across internal and partner networks

The organization has implemented standard security controls:

- encrypted communications using TLS
- certificate-based authentication for services
- code signing for software deployment
- endpoint detection and response tools
- compliance alignment with federal cybersecurity frameworks

Periodic audits show no critical findings. Systems are patched, monitored, and compliant.

From an operational perspective, the environment is considered secure.

13.2 Failure Chain

The compromise does not begin with a direct breach.

It begins in the supply chain.

A third-party software component used within the contractor's environment includes a dependency that was modified upstream. The modification is subtle and does not trigger signature validation failures. The software is built, signed, and deployed through the contractor's standard CI/CD pipeline.

Because the component is trusted and signed, it is incorporated into production systems.

Once deployed, the modified component introduces a passive data collection function. It does not alter system behavior in a way that triggers alerts. Instead, it observes and records:

- encrypted data exchanges between internal services
- authentication and certificate negotiation events
- metadata related to system interactions

This data is staged and periodically transmitted through legitimate outbound channels, blended with normal system traffic.

At this stage:

- no unauthorized access is detected
- no malware signatures are triggered
- no policy violations are recorded

The system remains operational and compliant.

Over time, the collected data accumulates outside the organization's control.

This includes:

- encrypted communications containing sensitive program data
- certificate exchanges and identity relationships
- system interaction patterns and dependencies

The organization continues to operate normally.

13.3 Point of Irreversibility

The point of irreversibility occurs when sufficient data has been collected and retained externally.

At this point:

- the organization no longer controls all copies of its sensitive data
- the exposure exists independently of current system security

- remediation within the environment does not eliminate external risk

Even if the compromised component is later identified and removed, the data already collected remains accessible to the adversary.

This is the transition from potential compromise to **deferred certainty of exposure**.

13.4 Invisible Factors

Several factors contribute to the invisibility of the compromise:

- **Valid signatures:** The software component is properly signed, preserving the appearance of integrity
- **Trusted pathways:** Data exfiltration occurs through allowed network channels
- **Encrypted content:** Captured data is encrypted, reducing immediate detection value
- **Distributed artifacts:** Data is collected in fragments across multiple systems and sessions
- **Lack of inventory:** The organization does not have a complete map of data flows or dependencies

Traditional controls are not designed to detect passive collection of encrypted data within trusted workflows.

The compromise exists below the level of visibility those controls provide.

13.5 Time to Detection vs Time to Impact

- **Time to Detection:** Not detected during initial operation
- **Time to Impact Realization:** Several years later, aligned to advances in decryption capability

The delay between these two points is the defining characteristic of the risk.

By the time impact is realized, the originating system may have been upgraded, replaced, or retired.

The exposure persists independently of those changes.

13.6 Trusted Systems Framework Intervention

Applying the Trusted Systems Framework would have altered the outcome at multiple points in the failure chain.

Pillar	Intervention
Origin	Verification of software provenance and dependency integrity before deployment
Integrity	Continuous validation of runtime behavior against expected baselines
Visibility	Full mapping of data flows, dependencies, and external communication patterns
Identity	Strong validation of service-to-service authentication and anomaly detection in certificate usage
Control	Restriction and monitoring of outbound data channels at a granular level
Resilience	Isolation of affected systems and ability to revert to known-good configurations
Adaptability	Ability to update cryptographic mechanisms and dependencies without full system disruption
Autonomy Governance	Monitoring of automated processes that handle data movement and system interaction
Temporal Awareness	Recognition that encrypted data exposure represents long-term risk, not immediate containment

The key difference is not a single control.

It is the presence of a system-level model that identifies and constrains the conditions under which such a compromise can occur and persist.

13.7 Outcome Comparison

Without Trusted Systems Approach:

- compromise remains undetected
- data is collected over time
- exposure is realized in the future
- remediation does not eliminate external risk

With Trusted Systems Approach:

- anomalous behavior is detected earlier
- data movement is constrained and monitored
- dependency risk is identified before deployment
- exposure is reduced, even if compromise occurs

13.8 Key Takeaways

- supply chain compromise does not require immediate exploitation
- encrypted data can be collected without triggering alerts

- compliance does not equate to systemic trust
- delayed exposure transforms minor events into major risk
- trust must be validated across origin, behavior, and time

Scenario 2: Financial Institution and Delayed Data Exposure

13.9 Situation Setup

A global financial institution operates a highly distributed environment supporting retail banking, institutional trading, and payment processing.

The architecture includes:

- encrypted transaction systems across internal and external networks
- cloud-based analytics and fraud detection platforms
- third-party integrations for payment processing and clearing
- identity systems supporting customers, employees, and automated services

The institution maintains strong security controls:

- TLS encryption for all communications
- certificate-based authentication for APIs and services
- hardware security modules for key protection
- continuous monitoring and fraud detection systems
- alignment with financial regulatory requirements

Sensitive data includes:

- transaction histories
- account identifiers
- authentication exchanges
- payment instructions
- customer behavioral data

Data retention requirements extend for multiple years, driven by regulatory and operational needs.

From a present-day perspective, the environment is considered secure.

13.10 Failure Chain

The exposure begins with normal operation.

Transactions are encrypted and transmitted between systems, partners, and clearing networks. Authentication is performed using certificate-based mechanisms. Data is logged for auditing, fraud detection, and analytics.

At each step:

- encryption protects content in real time
- identity systems validate participants
- logs and telemetry capture system activity

However, these interactions are observable at the network level.

Encrypted traffic, authentication exchanges, and transaction metadata are captured by an external adversary through passive collection methods. This includes:

- TLS session data
- certificate negotiation exchanges
- transaction timing and routing patterns
- API interaction sequences

The collected data is stored without immediate decryption.

Internally, the institution continues to operate without indication of compromise.

Over time, the collected dataset grows to include:

- large volumes of encrypted financial transactions
- identity relationships between systems and users
- behavioral patterns of accounts and services

This dataset is externally retained.

13.11 Point of Irreversibility

The point of irreversibility occurs when the volume and diversity of collected data reach a level sufficient to reconstruct meaningful financial and behavioral patterns once decryption becomes possible.

At this point:

- the institution no longer controls the full lifecycle of its sensitive data
- exposure is tied to future computational capability rather than current access
- historical data becomes a latent vulnerability

Even if the institution upgrades its encryption systems in the future, previously captured data remains at risk.

This is not a breach in the traditional sense.

It is a deferred disclosure condition.

13.12 Invisible Factors

The exposure remains invisible due to several factors:

- **Encrypted transport:** Data is protected during transmission, masking its content
- **Legitimate traffic patterns:** No anomalous behavior is required for collection
- **Compliance alignment:** Systems meet current regulatory and audit requirements
- **Distributed storage:** Data is replicated across multiple internal systems, increasing collection surface
- **Lack of temporal modeling:** Risk is evaluated based on current security, not future decryption potential

The institution has strong controls for preventing unauthorized access.

It does not have controls for preventing long-term external collection.

13.13 Time to Detection vs Time to Impact

- **Time to Detection:** No detection during collection phase
- **Time to Impact Realization:** Years later, when decryption becomes feasible

At the time of impact:

- systems may have been upgraded
- cryptographic algorithms may have been replaced
- operational teams may have changed

The exposure is disconnected from the original environment.

13.14 Trusted Systems Framework Intervention

The Trusted Systems Framework would reduce exposure at multiple points.

Pillar	Intervention
Origin	Validation of external integrations and data exchange pathways
Integrity	Assurance that transaction processing systems operate without hidden manipulation
Visibility	Mapping of data flows across internal and external systems
Identity	Strengthening of identity protocols and monitoring of authentication patterns

Pillar	Intervention
Control	Limiting unnecessary data replication and external transmission
Resilience	Protection of critical data through segmentation and controlled access
Adaptability	Early adoption of cryptographic agility and hybrid post-quantum mechanisms
Autonomy Governance	Oversight of automated fraud detection and analytics workflows
Temporal Awareness	Identification of data with long-term sensitivity and prioritization for protection

The most significant difference is the introduction of temporal awareness.

The institution evaluates data not only by current risk, but by how long it must remain confidential.

13.15 Outcome Comparison

Without Trusted Systems Approach:

- encrypted data is collected continuously
- exposure remains undetected
- long-term data becomes vulnerable to future decryption
- remediation cannot recover already collected data

With Trusted Systems Approach:

- data flows are reduced and monitored
- long-term sensitive data is prioritized for enhanced protection
- hybrid cryptographic strategies reduce future exposure
- risk is managed based on time horizon, not just current state

13.16 Key Takeaways

- encryption protects content in the present, not necessarily in the future
- financial data has long-term sensitivity that extends beyond current controls
- passive collection creates exposure without triggering alerts
- compliance does not account for delayed exploitation
- temporal awareness is critical for managing post-quantum risk

14. Recovery and Resilience Engineering

The preceding sections establish a consistent conclusion.

In modern environments, and particularly under post-quantum risk conditions, it is not possible to guarantee that all exposure can be prevented. Data may already have been collected. Components may already contain unknown dependencies. Identity systems may already rely on cryptographic assumptions that will not hold over time.

The objective, therefore, cannot be limited to prevention.

It must include the ability to recover, re-establish trust, and continue operation under degraded or uncertain conditions.

This is the role of resilience engineering.

14.1 From Prevention to Survivability

Traditional security models prioritize prevention and detection.

- prevent unauthorized access
- detect anomalies
- respond to incidents

These models assume that compromise is an event that can be identified and contained.

In a delayed exploitation model, compromise may occur without detection and without immediate impact. The effects may not be realized until years later, when conditions change.

This requires a different approach.

Systems must be designed to:

- assume that some level of compromise may already exist
- limit the impact of that compromise
- restore trusted operation when conditions change
- operate safely even when full trust cannot be established

This is not a failure of security.

It is an adaptation to a different threat model.

14.2 Defining a Known-Good State

Recovery depends on the ability to define and return to a known-good state.

A known-good state is not simply a recent backup. It is a system configuration that can be validated as trustworthy across all relevant layers.

This includes:

- verified hardware and firmware integrity
- trusted boot and system initialization
- validated software and configuration
- known cryptographic keys and certificates
- controlled and verified data sets

Without this definition, recovery becomes a restoration of unknown conditions.

In a post-quantum context, the concept of known-good must also include cryptographic validity over time. A system restored from backup may still rely on cryptographic mechanisms that are no longer considered secure.

14.3 Immutable and Segmented Recovery Architectures

Recovery systems must be designed to resist both active compromise and latent exposure.

This requires:

- **immutability:** backup data that cannot be altered once written
- **segmentation:** separation of recovery environments from operational systems
- **isolation:** controlled access paths to recovery assets
- **verification:** ability to validate integrity before restoration

Immutable storage ensures that recovery points are not modified by compromised systems. Segmentation prevents adversaries from accessing or manipulating backup environments through normal operational pathways.

These controls are critical in environments where compromise may persist undetected for extended periods.

14.4 Data-Centric Recovery

In many scenarios, the most critical asset is not the system, but the data.

Recovery strategies must therefore prioritize:

- identification of high-value data sets
- protection of those data sets across their lifecycle
- ability to restore data to a trusted state
- validation of data integrity after restoration

This is particularly important for data with long-term sensitivity.

If such data has been collected externally, recovery within the organization does not eliminate exposure. However, it can:

- ensure that internal operations are based on trusted data
- limit further propagation of compromised or unverified data
- support continuity of operations under new security conditions

Data-centric recovery aligns directly with the concept of temporal awareness.

14.5 Cryptographic Reconstitution

Post-quantum transition introduces a unique recovery requirement.

Organizations must be able to re-establish cryptographic trust.

This includes:

- replacing vulnerable algorithms with quantum-resistant alternatives
- reissuing certificates and keys
- updating trust chains and validation mechanisms
- ensuring compatibility across systems

This process can be viewed as cryptographic reconstitution.

It is not simply an upgrade. It is the re-establishment of trust relationships across the system.

In many environments, this will need to occur while systems remain operational. This requires:

- support for hybrid cryptographic models
- careful coordination to avoid breaking interoperability
- validation that new mechanisms are correctly implemented

Recovery, in this context, includes the ability to transition from one trust model to another.

14.6 Continuous Validation and Attestation

Recovery is not a one-time event.

Systems must be continuously validated to ensure that they remain in a trusted state.

This requires:

- measurement of system state at runtime
- comparison against known-good baselines
- detection of drift or unauthorized changes

- automated or guided remediation

Attestation mechanisms can provide cryptographic proof of system state, but they must themselves be trustworthy and adaptable to new cryptographic conditions.

Continuous validation closes the gap between recovery and operation.

14.7 Operational Resilience Under Uncertainty

In some cases, full trust cannot be immediately restored.

Systems may need to operate under conditions of partial confidence.

This requires:

- prioritization of critical functions
- restriction of non-essential operations
- enhanced monitoring and validation
- manual oversight for high-risk actions

This mode of operation is not ideal, but it may be necessary during transition periods or after significant compromise.

Designing for this condition in advance is a key aspect of resilience engineering.

14.8 Integration with the Trusted Systems Framework

Resilience is one of the nine pillars, but it depends on the others.

- Origin and integrity define what can be trusted
- Visibility identifies what must be restored
- Identity and control govern access during recovery
- Adaptability enables transition to new cryptographic models
- Temporal awareness defines which assets must be prioritized

Resilience is therefore not an isolated capability.

It is the ability of the system to re-establish trust across all pillars under adverse conditions.

14.9 Implications for Implementation

Organizations must incorporate resilience into their architecture, not treat it as an afterthought.

This includes:

- designing systems with recovery pathways from the outset
- aligning backup and recovery with trust validation requirements
- integrating cryptographic transition into recovery planning
- testing recovery processes under realistic conditions
- ensuring that recovery environments are themselves trustworthy

The objective is not simply to restore operation.

It is to restore trusted operation.

15. What Leaders Must Do Now

The transition to post-quantum security is not a single initiative. It is a coordinated transformation across architecture, cryptography, identity, and data management.

Organizations that approach this as a discrete upgrade will fall behind.

Organizations that treat it as a system-level priority can establish a controlled and measurable path forward.

The following actions define that path.

15.1 Establish Cryptographic Visibility Immediately

You cannot secure what you cannot identify.

The first requirement is a comprehensive inventory of cryptographic usage across the environment.

This includes:

- encryption mechanisms used in applications and protocols
- algorithms implemented across systems
- key generation, storage, and rotation practices
- certificate issuance, validation, and expiration
- dependencies on third-party cryptographic services

This inventory must extend beyond documented systems.

It must include:

- embedded cryptography in applications and libraries
- machine identities across services and infrastructure
- cryptographic functions within network devices and management systems

Without this visibility, all subsequent actions are based on incomplete information.

15.2 Identify Data with Long-Term Sensitivity

Not all data requires the same level of protection over time.

Organizations must classify data based on how long it must remain confidential.

Examples include:

- financial transaction records
- identity and authentication data
- defense and government information
- intellectual property and proprietary models
- long-lived operational datasets

This classification should define:

- which data is already within an exposure window
- which systems process or store that data
- which cryptographic mechanisms protect it

This step introduces temporal awareness into decision-making.

15.3 Map Identity and Trust Relationships

Identity systems must be understood as a complete graph, not a set of isolated components.

This requires:

- inventory of all certificates, keys, and identity providers
- mapping of trust relationships between systems
- identification of machine identities and their lifecycle
- analysis of authentication protocols and dependencies

Particular attention should be paid to:

- certificate authorities and trust anchors
- service-to-service authentication
- external identity providers and federation points

Identity is the mechanism by which trust is asserted.

If it is not fully mapped, it cannot be transitioned.

15.4 Validate Supply Chain and Provenance

Trust cannot be established without confidence in origin.

Organizations must:

- identify critical suppliers and dependencies
- validate software provenance and build integrity
- assess firmware and hardware trust mechanisms
- enforce security requirements in procurement processes

This includes both internal and external systems.

Where visibility is limited, organizations should augment internal analysis with supply chain intelligence capabilities such as Exiger to continuously evaluate risk across vendors and dependencies.

Supply chain validation is not a one-time activity.

It must be continuous and integrated into operational decision-making.

15.5 Introduce Cryptographic Agility

Systems must be able to evolve.

This requires:

- support for multiple cryptographic algorithms
- ability to transition between algorithms without disruption
- separation of cryptographic logic from application logic
- validation of protocol compatibility under new conditions

Organizations should:

- evaluate current systems for cryptographic agility
- prioritize upgrades where agility is limited
- design new systems with migration in mind

This is a prerequisite for post-quantum transition.

15.6 Plan and Execute Identity Migration

Identity migration is one of the most complex elements of the transition.

Organizations must:

- establish automated certificate lifecycle management
- prepare for large-scale reissuance of machine identities
- update authentication protocols to support new algorithms
- ensure interoperability during hybrid operation

This will require coordination across:

- internal systems
- cloud platforms
- partners and third-party services

Delays in identity migration will create systemic risk.

15.7 Strengthen Hardware and Firmware Trust

Post-quantum readiness must include the lowest layers of the system.

Organizations should:

- validate hardware root of trust mechanisms
- ensure firmware update processes are secure and verifiable
- assess management plane security and isolation
- align hardware lifecycle with cryptographic requirements

Platforms that support integrated security capabilities, such as those provided through hardware-level controllers like HPE Integrated Lights-Out, can enable more consistent enforcement of trust across system lifecycles.

This layer cannot be addressed after software migration.

It must be addressed in parallel.

15.8 Govern AI and Automated Decision Systems

AI systems must be treated as part of the trust model.

Organizations must:

- identify where AI is used in decision-making
- control data inputs and outputs
- monitor model behavior and drift
- enforce policy on automated actions

This includes:

- prompt handling and logging
- model training and fine-tuning data
- integration with external services

AI increases both the scale and speed of risk.

It must be governed accordingly.

15.9 Implement Data-Centric Controls

Reducing exposure requires controlling how data moves and persists.

Organizations should:

- limit unnecessary data replication
- enforce segmentation of high-value data
- monitor data flows across systems and boundaries
- apply encryption consistently across all states

This includes:

- data at rest
- data in transit
- data in use
- derived data and metadata

The objective is to reduce the amount of data entering long-term exposure conditions.

15.10 Build Recovery and Reconstitution Capabilities

Organizations must assume that some level of compromise already exists.

They must be able to:

- restore systems to known-good states
- validate integrity across all layers
- re-establish cryptographic trust
- operate under conditions of partial confidence

This requires:

- immutable and segmented recovery architectures
- defined recovery procedures aligned to trust validation
- testing under realistic conditions

Recovery is not an endpoint.

It is a continuous capability.

15.11 Establish System-Level Governance

All of these actions require coordination.

Organizations must establish clear ownership for post-quantum transition.

This includes:

- assigning responsibility across security, technology, architecture, and operations
- defining decision-making authority
- aligning funding and resources
- tracking progress and accountability

Without governance, technical efforts will remain fragmented.

15.12 Where to Focus Capability Investment

The capabilities required to support these actions fall into several categories:

- hardware-rooted security and system integrity
- supply chain intelligence and provenance validation
- cryptographic platforms supporting post-quantum algorithms
- identity and certificate lifecycle management systems
- data visibility and governance platforms
- AI governance and monitoring tools
- recovery and resilience infrastructure

Organizations should evaluate their current technology stack against these categories and identify gaps.

The objective is not to adopt specific vendors, but to ensure that the necessary capabilities are present and integrated.

15.13 Sequence of Execution

The order of operations matters.

A practical sequence is:

1. establish cryptographic and identity visibility
2. classify data based on long-term sensitivity

3. map trust relationships and dependencies
4. validate supply chain and system origin
5. introduce cryptographic agility
6. begin identity and system migration
7. implement data-centric controls
8. build recovery and reconstitution capabilities
9. enforce governance and track progress

This sequence aligns visibility, prioritization, and execution.

15.14 Final Observation

The organizations that act now will operate with increasing clarity and control.

The organizations that delay will operate under increasing uncertainty.

The difference is not incremental.

It is structural.

16. CONCLUSION

The security models that organizations rely on today were built on a set of assumptions that are no longer stable.

Encryption was assumed to provide durable confidentiality.
Identity systems were assumed to validate trust relationships reliably.
Infrastructure was assumed to operate within controlled and observable boundaries.

Those assumptions are now under pressure from forces that are not aligned in time.

Data is being collected continuously.
Regulatory expectations are accelerating.
Quantum capability is advancing toward thresholds that will invalidate widely deployed cryptographic systems.

These conditions do not represent a future disruption.

They define the current operating environment.

The result is a shift from immediate risk to deferred exposure. Systems may operate securely in the present while accumulating vulnerabilities that will only become visible under future conditions. Data that appears protected today may already exist outside of organizational control. Trust relationships that appear valid may depend on mechanisms that will not hold over time.

This is not a failure of individual technologies.

It is a failure of how trust has been defined.

Trust cannot be established through encryption alone.
It cannot be enforced at a single layer.
It cannot be verified once and assumed indefinitely.

It must be engineered across the system.

The Trusted Systems Framework presented in this paper provides a structure for doing so. It defines how trust can be established from origin through operation, validated continuously, and maintained under changing technical conditions. It recognizes that hardware, software, data, identity, supply chain, and autonomous systems are not independent domains, but components of a single trust surface.

Organizations that approach post-quantum transition as a cryptographic upgrade will address only part of the problem.

Organizations that treat trust as a system property will address the problem at its source.

The difference between these approaches will determine not only how systems are secured, but whether they remain trustworthy over time.

The transition to post-quantum security is not a future planning exercise.

It is a response to exposure that already exists.

The organizations that survive the post-quantum transition will not be the ones who encrypted faster.

They will be the ones who built systems where trust was never assumed in the first place.

References and Source Materials

This paper integrates publicly available research, government guidance, and industry developments with original analysis from SecureFi Institute.

- National Institute of Standards and Technology (NIST), *Post-Quantum Cryptography Standards (FIPS 203, FIPS 204, FIPS 205)*, 2024
- National Security Agency (NSA), *Commercial National Security Algorithm Suite 2.0 (CNSA 2.0)*, 2022
- Office of Management and Budget (OMB), *Memorandum M-23-02: Migrating to Post-Quantum Cryptography*, 2023
- Cybersecurity and Infrastructure Security Agency (CISA), *Post-Quantum Cryptography Initiative and Guidance*, 2024
- European Union Agency for Cybersecurity (ENISA), *Post-Quantum Cryptography: Current State and Quantum Mitigation*, 2023
- The White House, *National Cybersecurity Strategy*, 2023
- Global Risk Institute, *Quantum Threat Timeline Report*, 2023
- Hewlett Packard Enterprise (HPE), *HPE Introduces Security Advancements to Secure AI Adoption and Strengthen Enterprise Resiliency*, 2026
- Hewlett Packard Enterprise (HPE), *Silicon Root of Trust and Secure Supply Chain Architecture*, various publications
- Post-Quantum Ltd., *PQC and Quantum AI (QAI): Emerging Intersections*, 2024
- IBM Research, *Quantum Computing Roadmap and Cryptographic Implications*, various publications
- Google Quantum AI, *Quantum Computing Progress and Error Correction Research*, various publications
- National Institute of Standards and Technology (NIST), *Transition to Post-Quantum Cryptography Guidance*, 2024

Additional sources include SecureFi Institute research briefs and publicly available industry analyses on hybrid computing, AI-enabled systems, and emerging cyber risk.

About SecureFi Institute

SecureFi Institute focuses on leadership awareness and governance readiness across emerging computing technologies, including artificial intelligence, cybersecurity, high-performance computing, and quantum systems.

The Institute works to help government and institutional leaders understand the security and strategic implications of these technologies before they become deeply embedded in critical infrastructure.

SecureFi Institute Special Brief No. 002

Trusted Systems in an Autonomous, Post-Quantum World

Why Encryption Alone Will Not Protect Your Organization

April 2026

Figures and Analytical Models

All figures, diagrams, and analytical models presented in this research brief were developed by SecureFi Institute as part of its research on emerging computing architectures and cybersecurity implications.

Research Disclaimer

This research brief is provided for informational and educational purposes and reflects analysis from SecureFi Institute on emerging computing technologies and cybersecurity trends. The views expressed are intended to support awareness and discussion of technology and infrastructure challenges and do not represent official policy positions.