CleanStart

2026

CISO Guide to Software Trust

How Verifiable Container Foundations Turn Software Trust into a Governance Metric



Cybersecurity INSIDERS

CleanStart

Executive Summary

Software trust has crossed into governance. It is now measured through inherited risk, verifiable provenance, and how quickly an organization can produce credible proof under regulatory, customer, and board scrutiny. That shift changes the CISO mandate. Boards now expect evidence that code was trustworthy before deployment, not merely scanned and patched after release.

This guide shows CISOs how to turn that expectation into an operating model built on verified container foundations, codified compliance, and measurable trust outcomes.

CleanStart is built for that shift. By starting with container images compiled from source under hermetic SLSA Level 3 controls, with signed SBOMs, cryptographic provenance, and FIPS-validated lineage, it gives CISOs a way to reduce inherited risk at the source and generate audit-ready evidence on demand. In benchmarked customer deployments, organizations have reduced inherited vulnerabilities, attack surface, and compliance preparation effort while improving software supply-chain proof generation.

The sections that follow turn that foundation into an operating model, measurable trust KPIs, and a board-level software trust program.

Key Points

- **Software trust is now a board-level mandate**
  AI transparency requirements, software assurance expectations, NIST SSDF, and broader accountability pressures increasingly demand proof that deployed code is verifiable, traceable, and compliant by design.

- **Containerization imports risk at scale**
  Independent 2025 analyses found 87% of public container images ship with high or critical vulnerabilities before any application code runs. Post-deployment patching cannot resolve risk that was inherited at the source.

- **Supply-chain compromise is now systemic**
  In the first five months of 2025, 22 of 24 industry sectors experienced documented supply-chain incidents, from poisoned base images to tampered registries. Software integrity has moved beyond engineering hygiene into enterprise risk territory.

- **CISOs must move from scan-and-patch to verify-and-minimize**
  Only verified, attested, and policy-compliant containers should reach production, with enforcement applied consistently at the admission gate. Gartner expects formal software supply-chain security programs to rise from approximately 60% of organizations in 2025 to 85% by 2028.

CleanStart

The Problem with Today's Containers

Containers delivered agility, portability, and consistency. They also introduced a category of risk that traditional security models were never built to control: vulnerabilities that exist before a single line of application code runs. Pull a base image, and you inherit whatever ships with it. That became the model.

The numbers leave little room for interpretation. In 2025, independent analyses found that 87% of public container images shipped with at least one high- or critical-severity vulnerability, including some of the most widely used enterprise base images. When development teams pull those images, they inherit whatever comes with them: known CVEs, outdated libraries, misconfigured defaults, and in some cases embedded malware. This is not risk the organization created. It is risk the organization imported. The governance question is no longer simply how many vulnerabilities exist. It is how much risk the organization accepted before writing a single line of application code.

For CISOs, that changes the accountability model. The question is no longer how efficiently teams remediate inherited risk. It is why unsafe, unverified software was allowed into production at all. Inherited risk is now a governance issue that boards expect leadership to manage upstream.

Supply-Chain Integrity: The Upstream Trust Problem

Supply-chain compromise is no longer a theoretical concern. It is now an operational reality with enterprise-wide consequences. The XZ Utils incident from March 2024 remains a defining example: a contributor spent two years building trust with the project maintainer, assumed co-maintainer status, and then introduced a backdoor into versions 5.6.0 and 5.6.1 of a compression library used across much of the Linux ecosystem.

In the first five months of 2025, 22 of 24 industry sectors experienced documented supply-chain incidents, including poisoned base images, compromised registries, and tampered dependencies. Attackers understand the leverage of upstream compromise. A single malicious image can spread across thousands of deployments before it is detected. The attack surface is no longer just the application. It is the trust relationships, dependencies, and build paths surrounding it.

The integrity problem extends beyond malicious tampering. Most public images still lack the provenance records needed to establish trust with confidence. Without signed SBOMs, reproducible build records, and attestation chains, CISOs cannot answer basic questions about what an image contains or where it came from. Unverifiable provenance is not managed trust. It is assumed trust.

Traditional images also ship with components that serve no production purpose: package managers, debug utilities, shell access, and sometimes default credentials. Each unnecessary component expands the attack surface and increases the burden of monitoring, patching, and control.

Operational Drag: The Downstream Consequences

Inherited risk does not remain confined to the build stage. It cascades through operations. Security teams spend time triaging CVEs they did not introduce and often cannot fully control. The pattern is predictable. The advisory lands, known base-image issues resurface, and engineering, platform, and security teams are pulled back into urgent rebuild coordination.

This is the CVE treadmill in practice. In 2025, 70% of IT teams reported spending more than six hours each week patching container and open-source components. That is time diverted from higher-value security work such as threat modeling, application security, and architectural risk reduction.

When vulnerable images reach production, runtime detection becomes the fallback control. It operates at a disadvantage.

A 2023 Vectra AI study found that 83% of SOC analysts admit to ignoring alerts they suspect are false positives. That matters because when trust is not established before runtime, security programs become more dependent on runtime detection to catch what should have been blocked earlier.

Detection is necessary, but it is not a substitute for upstream trust.

Policy Enforcement Gap: The Governance Failure

Most organizations have container security policies. Far fewer enforce them consistently. Mutable tags still pass admission controls. Unsigned artifacts still make it into production because enforcement remains in audit mode.

If a control does not block unsafe artifacts at admission, it is not reducing risk. It is documenting intent. Boards are not asking what the organization documented. They are asking what it can prove it prevented.

Compliance Drag: Evidence Assembled Under Pressure

Compliance obligations add another layer to the problem. Requirements under EO 14028, NIST 800-53 and 800-171, FedRAMP, and FIPS all intersect with container security, yet most compliance processes were not designed around containerized software delivery.

When auditors ask for SBOMs, provenance records, or attestation evidence, many organizations are forced to assemble that proof manually and under time pressure. That is evidence that should have been generated during the build and carried forward with the artifact.

This is where compliance drag becomes visible. The burden is not just meeting the control. It is proving, often retroactively, that the control was met. The stronger model is to make that evidence a built-in property of the software lifecycle rather than a last-minute audit exercise.

CleanStart

The CISO Shift:
From Reactive Patching to Provable Trust

The failures described in the previous section are not isolated problems. They are the predictable result of a security model that tries to identify and manage risk only after unsafe artifacts have already entered the software lifecycle. Scanning, patching, triage, and manual evidence collection are downstream compensating mechanisms. They do not establish trust. They compensate for the absence of it.

CISOs need to change that model at its foundation. The goal is not simply to patch faster. It is to reduce inherited vulnerabilities, lower dependence on runtime detection, and make compliance evidence native to the build. The objective is not to make security debt easier to manage. It is to prevent avoidable security debt from entering the software lifecycle in the first place. That shift can be understood through a simple trust maturity progression.

Trust Maturity Model

REACTIVE	PREVENTIVE	PROVABLE
Scan and patch after deployment	Verified, minimized images before deployment	End-to-end attestation across the lifecycle
Inherited CVEs untracked	Enforced signing and provenance	Real-time SBOMs and transparency logs
Manual compliance evidence	Reproducible, policy-gated builds	Board-reportable trust KPIs

Most organizations still operate in the Reactive tier. The Preventive tier introduces verification gates so that only signed, minimized, and policy-compliant images can reach production. The Provable tier makes trust measurable and continuous: every artifact carries attestations, every dependency is tracked through live software metadata, and boards receive quantified trust metrics instead of qualitative assurance.

Principles for the Shift

Trust must be established before runtime
The admission gate is the last reliable control point. Once a container is running, the organization is dependent on detection, and detection fails often enough to matter. Only images that are signed, attested, and verified against policy should be admitted. Unsigned artifacts and mutable tags must be blocked at the gate.

Provenance is non-negotiable
Every image must carry a verifiable SBOM, a reproducible build record, and a transparency log entry. An organization that cannot prove where an image came from and what it contains cannot assert that it is trustworthy.

Compliance should be codified into the build
SBOMs, attestations, and FIPS validation should be generated at build time and travel with the artifact. When auditors request evidence, it should export in minutes through automated pipelines that already hold the proof.

Trust must be measurable at the governance level
Software trust needs metrics a CISO can track and leadership can challenge: inherited vulnerabilities, attack-surface exposure, attestation coverage, remediation speed, and time to produce evidence. These measures turn software trust from a security aspiration into a governance discipline.

Gartner projects that by 2028,

85% of enterprise software teams

will have formalized software supply-chain security programs, up from approximately 60% in 2025.

The organizations moving first are not simply adding controls around existing images.

They are changing the trust foundation itself.

CleanStart

Operationalizing Provable Trust:
Six Imperatives

Provable trust starts at the container foundation. Rather than compensating for inherited risk after deployment, CleanStart moves control upstream and gives CISOs six enforceable conditions for production: reduce inherited vulnerabilities, minimize attack surface, rebuild quickly when exposure changes, verify provenance cryptographically, block non-compliant artifacts at admission, and generate compliance evidence as part of the build. Together, these six turn software trust from an advisory concept into an enforceable production standard.

CleanStart is built on a compile-from-source foundation with hermetic SLSA Level 3 builds and FIPS-validated cryptographic lineage. The six imperatives below translate the trust principles introduced in Section III into enforceable production requirements.

1

Eliminate Inherited Risk at the Source

CISO requirement: Production software should not inherit avoidable vulnerabilities from its container foundation. Measure: inherited vulnerabilities per image. Prove: establish a baseline and demonstrate measurable reduction before broader production adoption. CleanStart starts with a glibc-compatible minimal base designed to reduce known inherited vulnerabilities at release. Public images routinely carry dozens or hundreds of CVEs before any application code is added. CleanStart components are vetted against multiple vulnerability sources, and security fixes are applied before release. The result is that inherited risk is reduced upstream, before it enters the environment. In customer benchmarks, this has translated into major reductions in inherited vulnerabilities per image.

2

Minimize the Attack Surface

CISO requirement: Production images should contain only the components required to run the workload. Measure: image size and unnecessary components. Prove: demonstrate measurable reduction in attack surface across in-scope workloads. CleanStart reduces attack surface by removing unnecessary packages, debug tools, default credentials, and nonessential services. Images run with least privilege and tightly constrained permissions. In production deployments, that has translated into smaller images, faster startup, and less runtime monitoring overhead. The principle is simple: the less unnecessary code present, the less there is to exploit, monitor, or patch.

3

Rebuild Faster Than Exposure Accumulates

CISO requirement: New vulnerability exposure should trigger a measurable, time-bound rebuild response. Measure: CVE rebuild mean time to remediation. Prove: demonstrate that critical fixes can move from disclosure to verified publication within the defined SLO. Automated workflows monitor upstream advisories and rebuild affected images within hours of disclosure. Each update is signed and published with an advisory that documents impact and remediation. That compresses rebuild time from weeks to hours and gives CISOs verifiable evidence of response speed. It also breaks the recurring cycle in which the same inherited issues resurface across multiple workloads while teams wait for patches.

4

Verify Provenance Cryptographically

CISO requirement: Production software should be able to prove its origin, build identity, and integrity. Measure: signing and attestation coverage across production workloads. Prove: every in-scope artifact carries verifiable provenance. CleanStart images carry signed provenance, including Sigstore signatures, Rekor transparency records, OIDC identity binding, and SLSA or in-toto attestations. Builds are reproducible and pinned to verified commits and dependencies, making provenance cryptographically verifiable by design. That matters because supply-chain attacks increasingly exploit weak trust assumptions around versioning, build identity, and artifact origin. Cryptographic provenance turns those assumptions into verifiable controls.

5

Enforce Trust at Admission

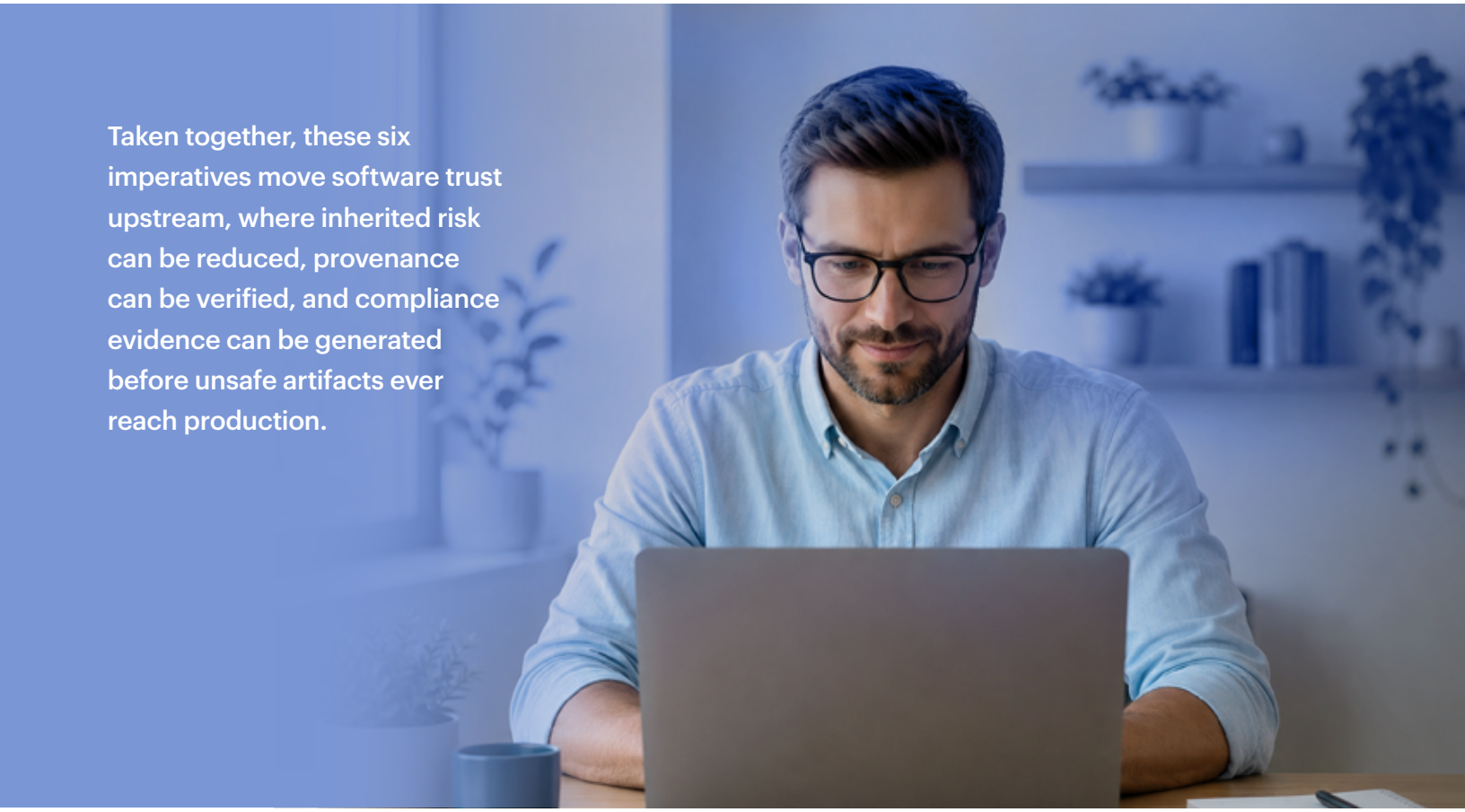
CISO requirement: Production environments should reject software that cannot prove its identity, provenance, and policy compliance. Measure: percentage of in-scope production workloads protected by admission enforcement. Prove: demonstrate that unsigned, unverifiable, or non-compliant artifacts fail at the gate. CleanStart integrates with CI/CD and runtime platforms to enforce these policies before deployment. Admission policies ensure that only signed, attested, policy-compliant containers reach production. This is the point where software trust becomes enforceable. Unsafe artifacts do not generate exceptions after the fact. They fail at the gate.

6

Generate Compliance Evidence at Build

CISO requirement: Compliance evidence should be generated as part of software delivery, not reconstructed during an audit. Measure: evidence generation/export time and manual preparation effort. Prove: produce the required evidence directly from the build and artifact lifecycle. CleanStart generates compliance evidence as part of the build, including SBOMs, build attestations, provenance logs, and support for FIPS-validated image variants. That gives security and compliance teams traceable proof of software origin, cryptographic lineage, and control support without relying on manual collection after deployment. The result is faster evidence production and a more defensible compliance posture. Section V explains the framework alignment in more detail.

Taken together, these six imperatives move software trust upstream, where inherited risk can be reduced, provenance can be verified, and compliance evidence can be generated before unsafe artifacts ever reach production.



The CISO Outcome

For boards, software trust becomes visible only when it is measured. For CISOs, that means replacing qualitative assurances with reportable outcomes: fewer inherited vulnerabilities, smaller attack surfaces, faster evidence generation, and verified provenance coverage across production workloads. In benchmarked customer environments, these improvements have translated into lower inherited exposure, reduced compliance preparation effort, and stronger proof of software supply-chain trust. These metrics, detailed in Section VI, become the governance language of software trust.

Adoption Without Friction

Security controls that create excessive friction get bypassed. That is why developer adoption has to be treated as part of the security design, not as a secondary concern. CleanStart supports multiple image profiles for different operating requirements, including hardened production bases, developer-ready non-production builds, minimal edge images, FIPS-oriented variants, workload-optimized runtime images, and AI/ML-oriented images. Developers interact with these through familiar registry workflows. What changes is the trust quality of what enters the environment. The move from reactive security to provable trust does not require a wholesale transformation program. It requires a stronger starting point.

Securing AI Workloads

AI expands the software trust problem by increasing the number of artifacts, runtime dependencies, and execution environments that have to be verified and governed. Model-serving stacks, GPU-enabled runtimes, training pipelines, and supporting orchestration layers all introduce additional software components that can carry inherited risk. CleanStart extends the same provenance, vulnerability reduction, and evidence-generation model to those workloads, allowing AI infrastructure to be governed under the same trust standard as traditional application workloads. As AI assurance expectations mature, infrastructure verifiability will increasingly become part of the compliance burden rather than a separate engineering concern.

Organizations ahead of this curve treat these six imperatives as preconditions for production. The compliance and assurance posture that follows from those preconditions is the subject of the next section.

CleanStart

Compliance & Assurance by Design

CISOs will be judged less by whether their programs declare compliance than by whether they can produce cryptographically defensible proof on demand. CleanStart is built for that test. Compliance is a property of the build, not a deliverable assembled under audit pressure. The governance model is simple:

CONTROL WHAT ENTERS THE LIFECYCLE → **GENERATE EVIDENCE AS SOFTWARE IS BUILT** → **PRODUCE DEFENSIBLE PROOF WHEN REQUIRED**

FIPS Validation: Built In, Not Bolted On

FIPS compliance cannot be retrofitted casually. The cryptographic module has to be validated, the integration has to be documented, and the lineage from certified component to deployed image has to remain traceable. CleanStart supports FIPS-validated image variants with CMVP-certified cryptographic modules integrated natively. That allows encryption paths to align with FIPS 140-2 and 140-3 requirements without relying on workarounds, compensating controls, or downstream exceptions.

Framework Alignment

CleanStart supports compliance across the control frameworks CISOs are most often required to satisfy:

FRAMEWORK	HOW CLEANSTART ALIGNS
EO 14028	SBOMs are generated during the build and signed with each image. Attestations document provenance, dependencies, and build environment, supporting software supply-chain transparency requirements.
NIST 800-53 / 800-171	Capabilities align most directly to Configuration Management, System and Information Integrity, and System Acquisition control families. Exportable evidence supports more consistent authorization and audit preparation.
FedRAMP	Mapped controls and exportable audit logs support ATO evidence assembly. Reducing inherited CVEs also lowers associated POA&M burden.
FIPS 140-2/3	CMVP-certified cryptographic modules support validated encryption paths without the need for bolt-on retrofitting.
DoD STIGs	Hardened configurations, reduced attack surface, and enforced least privilege strengthen alignment with STIG-oriented deployment expectations.
EU CRA (emerging)	Provenance documentation, SBOM generation, and stronger vulnerability transparency position organizations for emerging supply-chain assurance requirements under the CRA.



Automated Evidence Generation

Every CleanStart image carries its compliance documentation: Git commit lineage, dependency pinning, reproducible build records, and signed SBOM attestations. When auditors request evidence, it exports in minutes through the build pipeline that already holds it. Customer benchmarks show audit evidence packages exporting in under 15 minutes.

Audit Evidence Package Includes

- SBOM with signed attestation
- Build provenance and transparency log entry
- Dependency manifest with pinned versions
- FIPS validation documentation
- Control mapping to applicable frameworks

CleanStart attestations include exploitability context via VEX (Vulnerability Exploitability eXchange), so security teams can distinguish vulnerabilities that require immediate action from those that are present but unreachable in the deployed configuration. This reduces false-positive noise and focuses remediation effort where it matters.

Future-Proofing Compliance

Regulatory expectations are converging around software supply-chain transparency. The EU Cyber Resilience Act will increase expectations for provenance, vulnerability handling, and software accountability. OMB memoranda continue to tighten federal software assurance requirements.

AI governance frameworks are also beginning to extend accountability beyond model behavior to the software environments those systems run on.

CleanStart's combination of provenance documentation, signed SBOM generation, reproducible builds, and exportable attestation records aligns not only to current requirements but to the regulatory direction that is already visible. Organizations moving early are not simply reducing audit pain. They are positioning for assurance obligations that will become harder to satisfy through manual evidence collection alone.

CleanStart

Execution & Proof

The case for software trust as a board-level concern is now established. What matters next is execution: how to prove the model works in practice, how to sequence adoption without disruption, and how to measure outcomes in terms a board, auditor, and security leader can all understand. This section turns provable trust into a phased rollout and a governance model.

The 100-Day Blueprint: From Pilot to Proof

The blueprint below is designed as a lighthouse rollout for priority workloads. The goal is not enterprise-wide transformation in 100 days. It is to produce credible proof of capability, establish the operating model, and create the internal case for broader enforcement. In most environments, wider rollout follows over the next four to six months as teams phase in policy and coverage service by service.

PHASE 1 → PHASE 2 → PHASE 3 → PHASE 4

Baseline
Days 0–15

Inventory current images, measure inherited vulnerabilities and attack surface, and enable CleanStart policies in audit mode to quantify exposure without interrupting delivery. The objective is to establish the starting condition clearly enough that improvement can be demonstrated later.

TRUST OUTCOME
Quantify inherited exposure

Verify
Days 16–45

Migrate 2-3 priority workloads to CleanStart bases. Integrate signed SBOMs and provenance into CI/CD pipelines. Validate developer adoption with minimal friction. The objective is to prove that stronger trust controls are operationally practical, not just theoretically sound.

TRUST OUTCOME
Prove adoption and workflow fit

Prove
Days 46–75

Enable provenance and signature enforcement in non-production. Validate CVE rebuild SLO by demonstrating a critical patch rebuilt and published within hours. The objective is to prove that the trust infrastructure holds under real operating conditions.

TRUST OUTCOME
Validate provenance and rebuild responsiveness

Enforce
Days 76–100

Enable admission policies for pilot workloads in production. Block non-compliant images, export SBOM and attestation bundles, and demonstrate that trust controls can operate as enforceable production standards rather than advisory checks. The objective is to show that audit readiness and runtime enforcement can be achieved together.

TRUST OUTCOME
Demonstrate production policy and audit readiness

Each phase builds on the one before it. The baseline phase quantifies inherited exposure. The verification phase proves adoption. The proof phase validates the trust infrastructure itself. The enforcement phase demonstrates that production controls and audit evidence can operate in the same model.

Board-Level Governance Metrics

Boards do not want a tour of the security stack. They want measurable outcomes that show whether risk is falling, trust controls are working, and evidence can be produced on demand. That is the purpose of the governance layer: to translate software trust into metrics that can be reviewed, challenged, and tracked over time.

METRIC	TARGET	WHAT IT TELLS LEADERSHIP
Inherited vulnerabilities per image	85%+ reduction from baseline	How much avoidable risk is entering the lifecycle
Attack surface (image size and components)	70-80% reduction	How much unnecessary exposure is being eliminated
CVE rebuild mean time to remediation	Hours, not weeks	How quickly the organization responds to new exposure
Workload signing and attestation coverage	100% of in-scope production workloads	How much production software is verifiably sourced
Audit evidence export time	Under 15 minutes	How quickly the organization can prove its controls
Compliance preparation effort	70% reduction in staff hours	How much assurance still depends on manual work

When these metrics are measured and reviewed consistently, software trust stops being a technical aspiration and becomes a governance discipline. CISOs gain a way to show not just that controls exist, but that inherited exposure is falling, provenance coverage is increasing, and audit readiness is becoming measurable rather than anecdotal.

From Reactive to Provable

The old model treats inherited risk as a cleanup problem. The stronger model treats it as an admission problem, a provenance problem, and a governance problem. CleanStart gives CISOs a way to change that equation. What begins as a stronger container foundation becomes something more durable: a repeatable trust model that reduces inherited exposure, accelerates proof, and gives leadership measurable evidence that software integrity is being enforced rather than assumed.

CISOs that want to benchmark their current container trust posture should start with the governance metrics in this guide: inherited vulnerabilities, provenance coverage, admission enforcement, and time to proof. Those measures reveal quickly whether software trust is still being managed downstream or has become an enforceable production standard.

Take the next step

Explore the potential impact of hardened container images on vulnerability exposure, remediation effort, and software delivery.

Explore the Impact

Book a Demo