Snyk Finds Vulnerabilities Before Deployment. Governance After Deployment Is Still Manual.

by Nick Clark | Published March 27, 2026 | PDF

Snyk integrated security scanning into the developer workflow, finding vulnerabilities in code, dependencies, containers, and infrastructure-as-code before deployment. Shift-left security is real, and Snyk executes it well. But Snyk's governance is pre-deployment: it identifies risks in artifacts before they run. Runtime governance of what deployed systems actually do, which operations they perform and whether those operations comply with policy, is not part of the scanning model. The gap is between finding vulnerabilities and governing operations.

Snyk's developer-first approach to security is effective. Finding and fixing vulnerabilities in the IDE, in pull requests, and in CI pipelines catches problems early. The gap described here is about what happens after deployment.

Scanning is pre-deployment, governance is runtime

Snyk scans artifacts: source code, package manifests, container images, Terraform configurations. It identifies known vulnerabilities, suggests fixes, and tracks remediation. This is pre-deployment security.

But a system that passes all Snyk scans can still perform unauthorized operations at runtime. A container with no known vulnerabilities can still access data it should not. An application with no code vulnerabilities can still violate compliance policies through its runtime behavior. Pre-deployment scanning verifies what the artifact is. It does not govern what the artifact does.

Supply chain security is artifact governance, not operation governance

Snyk's supply chain security features verify the integrity and safety of dependencies. This is artifact governance: ensuring the components that make up a system are safe. But artifact governance and operation governance are different concerns. A safe artifact can still be used to perform unsafe operations.

What cryptographic governance provides

Cryptographic governance operates at runtime. Every operation is gated by a signed policy reference validated at execution time. The governance does not check what the code looks like before deployment. It checks what the system is doing at the moment of execution, against cryptographically signed policy.

Pre-deployment scanning and runtime cryptographic governance are complementary. Snyk verifies artifacts before deployment. Cryptographic governance verifies operations during execution. Together, they cover the full lifecycle. Separately, each leaves the other's gap open.

The remaining gap

Snyk made pre-deployment security scanning accessible. The remaining gap is in runtime governance: whether every operation is cryptographically validated against signed policy at the moment of execution, not just whether the code was safe when it was deployed.

Cryptographic Governance All 21 steps →

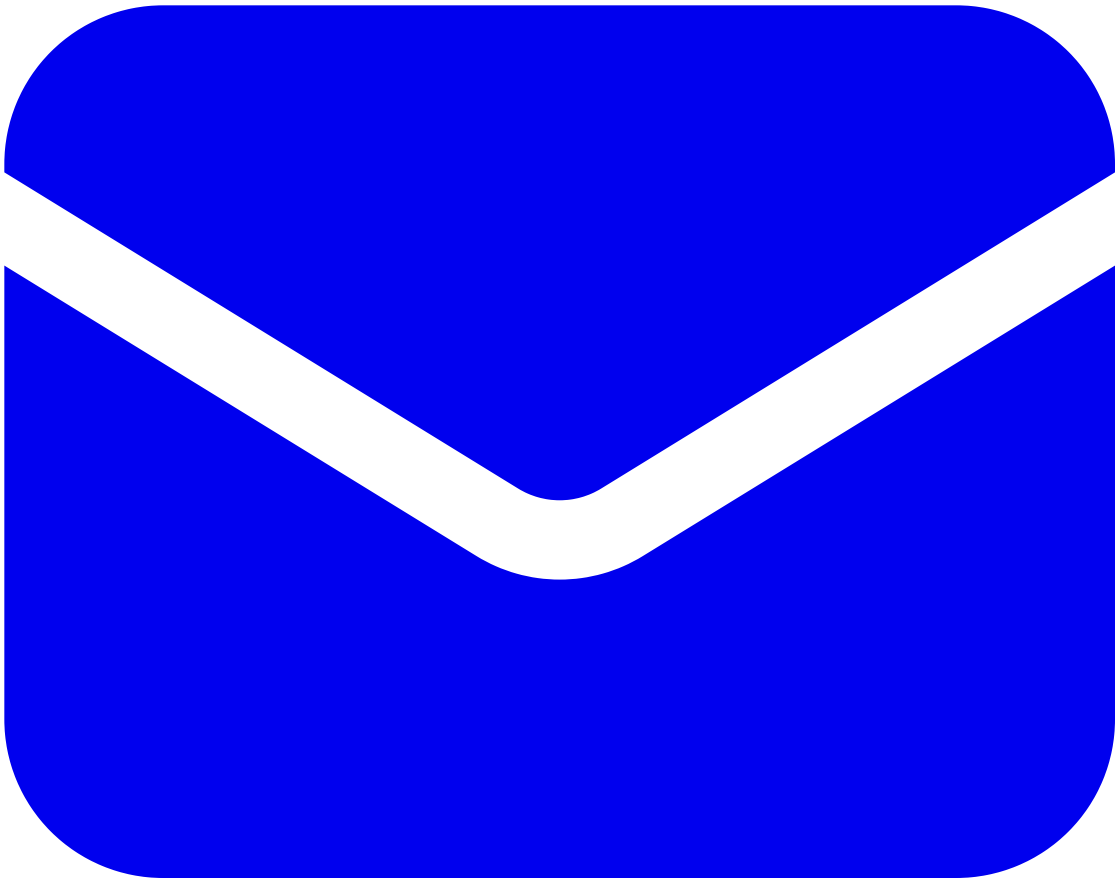
Policy that binds cryptographically — not by convention.

Patent
US 19/561,229 · filed
Primary Technical Disclosure
○ Ethical Enforcement as Infrastructure: Cryptographic Governance for Autonomous Systems
Secondary Technical
○ Governance Gate as Deterministic Precondition: No Verification, No Execution○ Canonical Alias to External Policy Indirection: Policy Evolution Without Agent Mutation○ Immutable-by-Default Policy Objects: Governance Changes Through Successor Issuance○ Runtime Policy Resolution Pipeline: Mandatory Verification Before Every Execution○ Freshness, Revocation, and Anti-Rollback Controls: Preventing Stale Authority○ Memory-Derived Eligibility Conditioning: Past Violations Constrain Future Authorization○ Intent-Independent Authorization: Governance Without Alignment Scoring○ Execution Feedback as Enforcement Signals: Operational Outcomes Shaping Future Authorization○ Trust Degradation as State Transition: Policy-Defined Narrowing of Permitted Actions○ Structural Quarantine: Execution Prevention Until Authorized Remediation○ Lineage-Constrained Governance Inheritance: Constraints That Persist Across Generations○ Unauthorized Fork Prevention: Lineage Continuity as Anti-Cloning Mechanism○ Meta-Policy Objects: Higher-Order Constraints Across System Behavior Categories○ Quorum-Based Governance Override: Multi-Party Approval With Signature-Chain Continuity○ Distributed Alias Publication: Policy Dissemination Through Federated Registries○ Fallback Enforcement Agents: Distributed Monitors as Defense-in-Depth○ Append-Only Governance Audit Ledger: Tamper-Evident Records of Every Authorization○ Governance Without Persistent Keypairs: Trust-Slope Authorization Replacing Static Keys○ Execution Eligibility Indicator: Dynamic Computation From Policy, Memory, and Lineage
Applications (General)
○ EU AI Act Compliance Through Structural Governance○ Financial Services Audit Trails Without Trusted Intermediaries○ Healthcare Compliance Through Structural Governance○ Defense Data Classification Enforcement○ Environmental Monitoring With Tamper-Proof Governance○ Pharmaceutical Supply Chain Governance○ Nuclear Facility Operational Governance○ Child Safety Content Enforcement
Applications (Specific)
○ HashiCorp Vault Manages Secrets. It Does Not Make Policy Cryptographically Binding.○ AWS KMS Manages Encryption Keys. The Keys Do Not Carry Governance.○ Open Policy Agent Decoupled Policy From Code. The Policy Is Not Cryptographically Bound.○ Styra Made OPA Enterprise-Ready. The Governance Model Did Not Change.● Snyk Finds Vulnerabilities Before Deployment. Governance After Deployment Is Still Manual.○ Palo Alto Networks Inspects Traffic. It Does Not Govern the Operations That Generate It.○ SPIFFE/SPIRE Provides Workload Identity. The Identity Has No Cryptographic Governance Binding.○ cert-manager Automates Certificate Lifecycle. The Certificates Carry No Governance Policy.○ Keycloak Provides Open-Source Identity Management. The Tokens It Issues Carry No Governance Binding.○ HashiCorp Boundary Provides Zero-Trust Access. The Access Sessions Have No Cryptographic Governance.○ Teleport Provides Unified Infrastructure Access. Access Control Is Not Cryptographic Governance.○ BeyondTrust Manages Privileged Access. Privilege Is Not Cryptographic Governance.○ CyberArk Pioneered Privileged Access Security. The Privilege Model Has No Cryptographic Governance Layer.○ 1Password Made Password Management Accessible. The Credentials It Manages Are Still Credentials.
Cryptographic Governance overview →
AQ
deterministic
autonomy

Legal

Last updated: 2026-03-03

-
- Inventive Steps
- Licensing
- Patents
- Articles
- Legal
- Opportunities
- Sitemap



-
- nick@qu3ry.net
- 72 28 14 36 01

Invented by Nick Clark | Founding Investors: Devin Wilkie