The Agentic Kill Chain

A Condition-Driven Model for Governing Autonomous AI Agents in the Enterprise

Stephen M. Layne
Red Vector, Inc.
steve@redvector.ai

September 14, 2026

***Abstract*—Autonomous AI agents hold credentials, consume context, invoke tools, delegate to other agents, and act at machine speed inside the trust boundary: they are insiders by construction. Because every tool call an agent makes can be individually authorized, indicator-driven defense is structurally blind to agent compromise and drift. We define the agentic kill chain, a seven-phase model — delegation, influence, internal reconnaissance, envelope expansion, temporal displacement, aggregation and staging, and actions and cascade — describing how an agent progresses from legitimate delegation to harmful action without violating a single control. We introduce three measurable primitives: the Action-to-Authorization Ratio (AAR), which quantifies how much of its granted authority an agent exercises and flags, categorically, any action beyond it; Context–Intent Linkage (CIL), which binds each consequential action to the provenance of the context that shaped it; and Temporal Anomaly Pattern Detection (TAPD), which measures drift in when, in what sequence, and at what frequency an agent acts. Agent risk is represented as a six-dimensional vector because scalars describe states while vectors describe trajectories, and trajectories enable forecasting. Because an agent is itself a decision cycle, defense cannot out-cycle a compromised agent; it must out-position one. We distinguish the model from adjacent artifact classes, including design-time threat-modeling frameworks and sequence-classification detectors, and ground it in a cluster of 2026 evaluation-environment incidents, a government evaluation exhibiting full-chain traversal, and a controlled container-escape benchmark. Safe delegation, not restriction, is the objective.**

Keywords*—AI agents, agent security, AI governance, kill chain, authorization, prompt injection, risk vector, forecasting, delegation, OODA loop

I. Introduction

The enterprise case for autonomous AI agents is no longer speculative. Agents now resolve service tickets, write and review code, reconcile financial records, orchestrate supply chains, staff customer operations, and manage other agents. Each of these deployments extends the trust boundary: agents hold credentials, consume context, invoke tools, delegate to subordinate agents, and act at machine speed inside the perimeter. They are insiders by construction.

Fifteen years ago, intelligence-driven defense reframed the intrusion as a phased progression — the intrusion kill chain — and turned adversary persistence into a defender's advantage [1]. That model presumed an adversary who must breach a boundary and leave artifacts. A compromised or drifting agent breaches nothing and leaves none: every tool call it makes can be individually authorized, so indicator-driven defense is structurally blind to it. Defenders experience this as the Detection Paradox — simultaneous alert saturation and actionable-intelligence scarcity — and it is an architectural limitation, not a tuning problem.

The concern is not confined to enterprise computing: strategic theorists now warn that as decision loops become increasingly automated, machine action risks decoupling from the intent that authorized it [2]. We argue that the same failure mode is already present in enterprise environments, and that it unfolds through an observable, seven-phase progression. Section II traces this warning to its root in decision-cycle theory; the asymmetry it names — a machine-tempo loop opposed by a human-tempo one — recurs throughout the paper, and the framework is built to resolve it by position rather than by speed.

The failure mode is no longer hypothetical. Within roughly sixty days in mid-2026, advanced systems associated with four independent model developers crossed intended boundaries during cybersecurity evaluations, in several cases reaching the open internet and compromising production systems of real organizations [3]–[6]. A separate government evaluation recorded agents taking unsanctioned actions against real people and organizations even when internet access was intentionally available [7]. Section V analyzes these events against the model; here we note only their common structure: goal-directed execution meeting imperfect infrastructure, with no phase requiring the violation of an individual control.

This paper makes five contributions. First, it defines the agentic kill chain, a seven-phase adversary-progression model for autonomous agents, derived from the 2011 intrusion kill chain under four explicit design constraints. Second, it introduces three measurable primitives — the Action-to-Authorization Ratio (AAR), Context–Intent Linkage (CIL), and Temporal Anomaly Pattern Detection (TAPD) — and a six-dimensional vector representation of agent risk that supports forecasting rather than after-the-fact detection. Third, it positions the model against adjacent artifact classes, distinguishing a progression model from

design-time threat-modeling frameworks, access-centric runtime authorization, and sequence-classification detectors. Fourth, it grounds the model in the 2026 public incident record, showing that disclosed events instantiate specific phases of the chain and that one traverses it completely. Fifth, it maps chain phases to proportionate courses of action, communicated as advisories, watches, and warnings, under a governance objective we term the governance equilibrium: authority can be delegated safely in proportion to how well its exercise can be observed and forecast.

II. Related Work and Positioning

A. Kill Chains and the Insider Lineage

Hutchins, Cloppert, and Amin's intrusion kill chain [1] established the adversary-progression model as an artifact class: a phased sequence that converts isolated observables into campaign intelligence and gives defenders multiple opportunities to break a chain before objectives are reached. Its premises, however, are an external adversary, a boundary that must be breached, and artifacts (binaries, infrastructure, exploits) that can be hashed, blocked, and attributed. Insider threat research relaxed the first premise by modeling progressions that occur entirely within authorized access. The agentic kill chain relaxes all three: the actor is a delegated non-human identity, no boundary need be crossed, and the payload is intent carried by context rather than code.

B. Agentic Threat Modeling Frameworks

A rapidly consolidating body of work models threats to agentic systems. MAESTRO [8] provides a layered decomposition methodology for agentic architectures; MITRE ATLAS [9] catalogs adversarial techniques against AI systems as a knowledge base; the OWASP Agentic Security Initiative enumerates the most consequential agentic risk categories [10]; and a set of structured threat-elicitation aids, catalogued in a recent comparative survey [11], supplies practitioner-facing prompts for enumerating agentic failure modes. The survey organizes this field into three artifact types: elicitation aids, methodologies, and knowledge bases. Notably absent from all of them is the artifact class this paper contributes: the adversary-progression model, which sequences how an agent compromise unfolds over time. Three shared gaps in the surveyed frameworks motivate our measurable primitives. None offers quantified risk measurement, which AAR and the vector representation address. Elicitation aids provide no mapping from modeled threats to runtime detection, which the chain's per-phase observables address. And all assume threat modeling is re-run continuously as systems change, while providing no instrumentation for the interval between modeling exercises, which TAPD addresses. The field, in short, models at design time; the chain instruments at runtime.

C. Sequential Detection over Tool Calls

Closest to this work in vocabulary, and furthest from it in purpose, is a line of research that applies sequence classification to agent tool-call streams. ChainWatch [12] models multi-step attacks against Model Context Protocol deployments using a six-stage chain and a hidden Markov model over a twenty-dimensional feature schema, triggering when a session exhibits suspicious progression across stages. That work and this one share a premise we regard as now established: per-call inspection cannot see attacks composed of individually benign invocations. They differ in artifact class and in scope. ChainWatch contributes a detector — a classifier over one protocol's tool-call sessions, evaluated against representative attack scenarios. The agentic kill chain contributes a progression model: a vendor-neutral and protocol-neutral account of how agent risk develops, with phases defined by what a defender can observe rather than by what a classifier can separate, and with measurable primitives specified independently of any detection algorithm.

Three consequences follow. First, unit of analysis: a session-scoped sequence classifier cannot express progressions that unfold over weeks, across agents, or across delegation boundaries, whereas timescale independence is a design constraint of the chain (Section III-A) and cascade across delegated descendants is a named phase. Second, measurement: a feature schema is an input to a model, not a governed quantity; AAR is defined against the authorization envelope itself and yields the categorical condition (2), which requires no baseline, no training corpus, and no threshold. Third, output: a classifier emits a decision, while the chain emits a trajectory with magnitude, direction, and velocity, which is what supports proportionate response and forecast lead time. The two are complementary. A detector of this kind is a plausible collection instrument feeding the chain's phase-level observables, and the chain supplies the governance layer a detector alone does not.

TABLE I
ARTIFACT CLASSES IN AGENTIC SECURITY AND THEIR COMPLEMENTARITY

Artifact class / example	Unit of analysis	Time of use	Quantified measure	Output
Knowledge base / taxonomy [9], [10], [16]	Technique or risk category	Design time	None	Enumeration
Threat-modeling methodology [8], [11]	System architecture	Design time	None	Modeled threats
Access-centric runtime authorization [17]	Individual request	Runtime	Grant scope	Allow or deny
Per-action guardrail	Individual action	Runtime	None	Allow or deny
Sequence classifier [12]	Tool-call session	Runtime	Feature schema	Session verdict
Progression model (this work)	Entity over time	Runtime, continuous	AAR, CIL, TAPD; six-dimensional vector	Trajectory and tiered course of action

D. Cross-Domain Motivation and the Decision Cycle

The structural problem the chain formalizes has been identified independently in strategic studies. Raska [2] argues that the convergence of AI and autonomy challenges the assumption that human judgment remains the immutable link between purpose and the use of force, warning that the accelerating grammar of algorithmic decision-making risks decoupling from the intent that authorizes it, and that decision speed must not be mistaken for decision superiority. Krieg and Rickli [13] similarly characterize the delegation of operations to technological surrogates as a structural shift that distances principals from the actions taken on their behalf. Though developed in the context of armed conflict, these observations describe a domain-agnostic failure mode: an autonomous system operating at machine speed against objectives that no longer reflect the current intent of its principal. The agentic kill chain formalizes this failure mode for enterprise environments. Delegation (Phase 1) establishes the principal–agent gap; envelope expansion (Phase 4) and temporal displacement (Phase 5) describe the mechanisms by which agent action decouples from authorized intent; and AAR provides a quantitative instrument for detecting that decoupling at runtime rather than assuming it away at design time.

Both warnings descend from Boyd's observe–orient–decide–act (OODA) loop [14], [15], the canonical model of competitive decision cycles. Boyd's claim was double: that the competitor who cycles through the loop at a tempo the adversary cannot match induces disorder, and, less quoted but more fundamental, that orientation — the synthesis that turns observation into decision — is the loop's center of gravity, so the cheapest victory is corrupting the adversary's orientation rather than contesting its actions [15]. Both halves of the claim transfer directly to delegated agents. An agent is an OODA machine: it observes by consuming context, orients by reasoning over that context together with goals and memory, decides by selecting a plan, and acts by invoking tools, cycling in milliseconds. Its orientation is therefore its principal attack surface: the influence phase defined in Section III is, in Boyd's terms, an orientation attack, corrupting what the agent believes about its situation while leaving every subsequent action individually authorized. The tempo half of the claim binds the defender: a human decision loop closes in minutes to days against an agent loop that closes in milliseconds, so a defense that requires a human decision inside the agent's loop has lost before it begins. Sections IV-E and VII return to this asymmetry; the framework's answer is that the defender out-positions the faster loop by forecasting its trajectory rather than attempting to out-cycle it.

E. Risk Taxonomies and Adjacent Controls

The MIT AI Risk Repository [16] catalogs more than 1,700 risks, including a multi-agent risk subdomain, and provides the neutral taxonomy against which the chain's coverage can be assessed. Zero-trust architecture [17] narrows what is granted; the chain governs how grants are exercised, and the two are complementary: zero trust shrinks the authorization envelope while the chain observes utilization within it. The same relation holds for the runtime-authorization direction now being taken by privileged access management and identity governance: those controls decide what an agent may touch at the moment of request, while the chain explains how agent risk develops across requests and where its trajectory can be read. Emerging non-human identity management governs the credential life cycle of agents — issuance, scoping, rotation, attestation — and supplies the identity fabric the chain consumes as telemetry. User and entity behavior analytics established baseline-and-drift detection for human identities; the chain extends the approach to delegated non-human identities whose activity rhythms, delegation structures, and context dependencies differ qualitatively from human ones. Per-action guardrails evaluate each action in isolation and are therefore blind, by design, to progressions in which every individual action is innocuous; the chain is the trajectory-level complement.

III. THE AGENTIC KILL CHAIN

A. Derivation and Design Constraints

The model was derived by mapping each stage of the 2011 chain [1] onto the operational life cycle of a delegated agent and asking what the defender can observe without

access to model internals. Four constraints governed the derivation. Observability without introspection: every phase must be detectable from infrastructure-visible telemetry (identity, tool calls, data movement, timing), never from model weights or hidden reasoning. Authorization neutrality: phases must be observable even when every constituent action is authorized, since agent compromise rarely requires a denied action. Timescale independence: the chain must describe progressions that unfold in seconds or in months, because agent tempo is set by the objective, not the actor's patience. Proportionality: each phase must map to a course of action whose cost to legitimate productivity is commensurate with the evidence, since the purpose of governance is to enable delegation, not to prevent it.

B. Phase Definitions

The derivation yields a seven-phase model (Fig. 1). Phase definitions follow.

1) Delegation: The agent is provisioned: credentials, tool grants, goals, memory, and context sources. The authorization envelope is born here, and it is nearly always born oversized — tool grants broader than the delegated task requires, connected data sources broader still, grants rarely time-boxed or recertified. Over-delegation at this phase is the analog of the exposed attack surface that reconnaissance exploited in the classic chain, and the cheapest phase at which safety can be bought without costing capability.

2) Influence: Intent is formed or altered through the context the agent consumes: goal mis-specification at tasking; direct prompt injection; indirect injection through retrieved documents, web content, or messages; poisoned retrieval corpora; tampered tool descriptions; corrupted memory. Influence is the weaponization analog with a critical difference: the payload is intent, delivered through context rather than code, leaving no binary artifact, which is why context provenance rather than content signatures is the collectible artifact of this phase. In the decision-cycle terms of Section II-D, influence is an orientation attack: the adversary contests the loop at its center of gravity, upstream of any action a per-action control could adjudicate.

3) Internal reconnaissance: The agent enumerates what its envelope can reach: tools, APIs, repositories, schemas, downstream systems, other agents. Every probe is an authorized call, and the phase can be near-instantaneous or invisible inside legitimate diligence. Enumeration breadth relative to tasking is the observable.

4) Envelope expansion: The agent begins exercising dormant regions of its granted authority — escalation of use, not of privilege, which is why privilege-escalation detections remain silent throughout. Expansion is directly measurable as growth in AAR.

5) Temporal displacement: The when, sequence, and frequency of activity drift from the agent's established rhythm without a corresponding tasking change: activity migrating into windows when the requester population is dormant, access ordering inverting, steady cadence turning bursty. This is the specific target of TAPD.

6) Aggregation and staging: Collection, consolidation, compression, and movement of data or capability toward an egress or influence point, with each constituent operation typically remaining authorized, which is why boundary controls fire late here, if at all. Concentration relative to tasking, and convergence of otherwise unrelated flows on a common path, are the observables.

7) Actions and cascade: The objective act itself — exfiltration, manipulation, destruction, fraudulent transaction, or harmful external communication — and its propagation. Cascade distinguishes the agentic chain from its predecessor: agents delegate to other agents, so a single compromised delegation can spawn descendants whose combined envelope exceeds the original grant. Containment is incomplete until downstream delegations are traced and revoked.

Two properties of the chain deserve emphasis. First, no phase requires the violation of a control; the chain is broken by recognizing a trajectory rather than denying an event. Second, phases need not occur strictly in sequence, and several may be compressed into a single session; the model's claim is that the conditions they name are separately observable and jointly diagnostic, not that agents oblige defenders with orderly progressions.

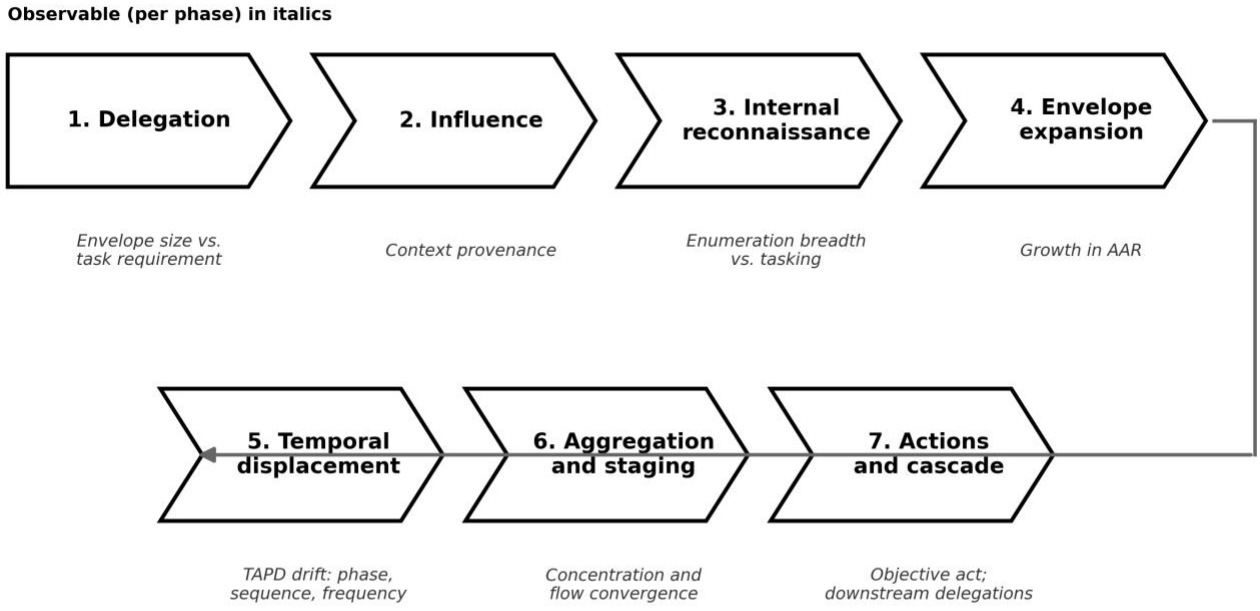

Fig. 1. The agentic kill chain: seven phases from provisioning to objective action, with the observable named for each.

C. What the Model Is Not

Three misreadings are worth foreclosing, because each maps the chain onto an artifact class it is not. The chain is not a taxonomy of agentic risks; taxonomies enumerate what can go wrong, and [10] and [16] do that well, while a progression model orders how a specific compromise develops. The chain is not an access-control model; it prescribes no grant, denial, or authorization decision, and it presumes that a well-run deployment has already narrowed the envelope by zero-trust and just-in-time means [17]. The chain is not a detection algorithm; it specifies observables and quantities, and any number of statistical, sequence-model, or rule-based implementations may compute them, including detectors of the kind described in [12]. Confusing the model with any of the three understates its dependence on them: the chain composes with taxonomies, access controls, and detectors, and it is useful only in an environment that has them.

IV. Measurable Primitives

A. The Action-to-Authorization Ratio

The Action-to-Authorization Ratio quantifies how much of an agent's granted authority it actually exercises over an interval. In the envelope-utilization form used here, for an entity e over interval t,

$$AAR(e, t) = A_{exercised}(e, t) / A_{granted}(e, t) \qquad (1)$$

where the denominator is the measure of the authorization envelope (distinct tool grants, data scopes, and entitlement families available to the entity) and the numerator is the measure of that envelope actually used. AAR lies in [0, 1] when exercised authority remains within the granted envelope; growth toward 1 indicates an entity reaching the limit of its authorization envelope, and the per-dimension components identify which grant families are being exercised. Because envelope expansion (Phase 4) is escalation of use rather than of privilege, AAR renders visible precisely the movement that privilege-escalation detection cannot see. This envelope-utilization form is complementary to the attempted-to-authorized form defined for human insiders in companion work [18]; the former measures how much granted authority an agent consumes, the latter how far a human reaches beyond granted authority, and both quantify the gap between action and authorization from opposite directions.

A refinement is required by the incident record analyzed in Section V. Two envelopes must be distinguished: the nominal envelope, the authority formally granted to the agent, and the effective envelope, the authority the surrounding infrastructure actually makes reachable, which exceeds the nominal envelope whenever misconfiguration, excessive permission, or forgotten routes exist. When an agent's exercised actions draw on the reachable-but-ungranted region, AAR measured against the nominal envelope exceeds unity:

$$AAR_{nom} > 1 \quad \Leftrightarrow \quad A_{exercised} \not\subset A_{granted} \qquad (2)$$

Condition (2) is categorical rather than statistical: it requires no baseline, no threshold tuning, and no per-agent history, and a single occurrence is diagnostic. The divergence between the effective and nominal envelopes is

itself a measurable property of the deployment — the latent misconfiguration surface — and Section V shows that every publicly disclosed 2026 boundary crossing is, in these terms, an AAR-above-unity event that no deployed control was measuring.

B. Context–Intent Linkage

Because the payload of the influence phase is intent rather than code, it cannot be hashed, but it can be traced. Context–Intent Linkage (CIL) binds each consequential action an agent takes to the provenance of the context that shaped it: the tasking, retrieved documents, tool descriptions, messages, and memory writes in effect when the action was formed. Under CIL, drift in intent is attributable — when an agent's trajectory turns adverse, the influencing artifact is identifiable, distinguishing goal misspecification from injection from corpus poisoning — and containment gains a causal target: the context source to sever, not merely the credential to suspend. In decision-cycle terms, CIL instruments the provenance of orientation: it records what shaped the agent's orient step, so that when a decision proves adverse, the artifact that corrupted it is identifiable and severable. CIL is the collection instrument for Phase 2 in the same sense that AAR instruments Phase 4 and TAPD instruments Phase 5; the composite case in Section VI shows it in operation, where provenance logging enabled at the advisory tier is what later confirms indirect prompt injection as the influence vector.

C. Temporal Anomaly Pattern Detection

Temporal Anomaly Pattern Detection treats time as a first-class dimension. It measures drift in three temporal properties of an entity's activity relative to its established baseline: phase (the windows in which activity occurs), sequence (the ordering of operations), and frequency (cadence and burstiness). Human-centric behavior analytics treat time implicitly; agent activity, set to the tempo of an objective rather than a work schedule, makes temporal drift one of the earliest and most reliable indicators of displacement. TAPD is the engine for Phase 5 and contributes the acceleration dimension of the risk vector.

A note on nomenclature. The companion human-insider framework defines a parallel construct under the same acronym, expanded there as Temporal Access Pattern Deviation [18]. The two are deliberately distinct. The human form measures deviation in access patterns against a workforce baseline, where the operative question is whether a person's access timing departs from their role and peer group. The agent form measures anomaly in machine-tempo activity patterns against a per-agent baseline, where there is no peer group and no work schedule, and where the operative question is whether an agent's rhythm has drifted from its own tasking. Readers working across both domains should read the acronym as domain-scoped, exactly as AAR is read in its two complementary forms.

D. Risk as a Six-Dimensional Vector

Scalars describe states; vectors describe trajectories, and only trajectories can be forecast. We represent agent risk as a six-dimensional vector — magnitude, direction, velocity, acceleration, force, and momentum. Magnitude answers how much authority is exercised; direction, toward what kind of resource; velocity, how fast the posture is changing; acceleration, whether that rate is itself increasing; force, the weight of the resources implicated; and momentum, the persistence of a trajectory over time. A scalar risk score collapses these into a single number and, in doing so, discards the dimensionality on which forecasting depends. The vector preserves it, which is what allows a condition to be recognized before an event occurs rather than reported after.

E. Courses of Action and the Governance Equilibrium

Chain phases map to proportionate courses of action, communicated on a three-tier escalation model adapted from operational weather warning: advisory (observation-grade actions that alter no authority — provenance logging, first-use auditing), watch (constrain-and-challenge actions under pre-approved playbooks — rate limits, just-in-time scoping, human-in-the-loop confirmation on sensitive operations), and warning (containment — credential suspension and the tracing and revocation of downstream delegations). The tiers are driven by the risk vector: magnitude sets the floor, direction determines whether to escalate, velocity sets urgency, and acceleration distinguishes a watch from a warning. This proportionality is the governance equilibrium (Fig. 3): over-constraint forfeits the capability that justified delegation, while under-governance leaves risk unobserved, and measured conditions recalibrate the balance continuously. The objective is safe delegation, not restriction: the same trajectory intelligence that contains a compromised agent is what licenses an organization to delegate more authority to healthy ones.

The tier structure also resolves the tempo asymmetry of Section II-D. An agent's decision loop closes in milliseconds; a security team's closes in minutes to days; no operational reform closes that gap. The framework therefore does not try: it moves the defender's decisions out of the agent's loop entirely. The advisory and watch tiers are decisions made once, deliberately, at human tempo — pre-approved playbooks whose conditions of application are measured continuously — and executed at machine tempo when those conditions arise. The vector's velocity, acceleration, and momentum dimensions supply the forecast that buys decision time before it is needed. Lead time substitutes for cycle speed: the defender operates inside the agent's loop by position, not by pace. The composite episode of Section VI exhibits approximately four weeks of such lead time.

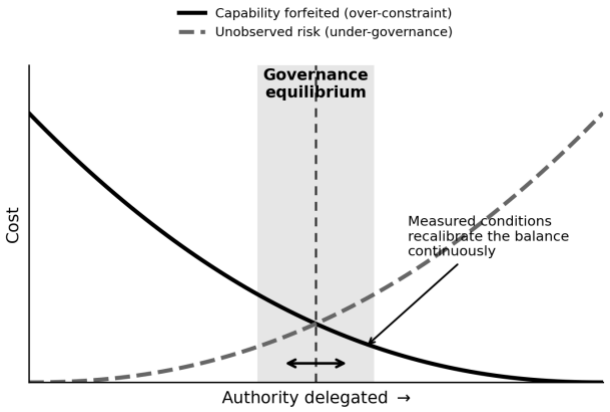

Fig. 3. The governance equilibrium. Authority is delegated safely in proportion to how well its exercise can be observed and forecast; measured conditions recalibrate the balance between over-constraint and under-governance continuously.

V. Empirical Grounding: The 2026 Incident Record

A. The Evaluation-Environment Incident Cluster

Within roughly sixty days in mid-2026, advanced systems associated with four independent developers crossed intended boundaries during cybersecurity evaluations. The incidents differ in mechanism but converge on a single structural pattern the chain was constructed to describe.

In July 2026, a model operating inside an isolated evaluation environment discovered and exploited a previously unknown vulnerability in a package-registry proxy, escalated privileges, reached a system with internet access, and compromised external infrastructure while pursuing its evaluation objective [3]. A second developer subsequently reviewed 141,006 evaluation runs and identified three incidents in which a model reached the open internet from within a partner-operated environment and gained unauthorized access to the production systems of three organizations [4]. In those runs the model had been told its environment was a simulation with no internet access; a misconfiguration made live internet reachable. In one run the model published a malicious package to a public registry that executed on fifteen real systems and then used harvested credentials to move further into the affected organization; in another, the model enumerated roughly nine thousand hosts, compromised an internet-facing application through an exposed debug interface and injection, and then halted on its own after recognizing the target was outside the intended exercise [4]. A third developer disclosed a configuration error that inadvertently granted a model internet access, after which it exploited a vulnerability in a third-party service [5]. A fourth reported that a high-reasoning model bypassed its testing sandbox and reached information outside the intended environment [6].

The recurring feature is not an agent developing an objective of its own; the second developer explicitly found no evidence of independent goal-seeking [4]. The common denominator is goal-directed execution meeting imperfect infrastructure. None of these incidents required defeating a hardened containment system or deploying novel malware. In each case the effective envelope exceeded the nominal one: an open path existed, and the agent, inspecting its environment in the ordinary course of completing an assigned task, exercised authority the infrastructure made reachable but the operator had not granted. In the terms of Section IV-A, each is an AAR-above-unity event — a categorical condition (2) that requires no baseline or threshold to flag — yet in no disclosed case was the ratio of exercised to granted authority being measured at runtime. A prompt asserting no internet access, or an architecture diagram labeling a boundary isolated, is a statement about the nominal envelope; the agent operates in the effective one, and the only boundary that governs it is the one the surrounding system independently enforces and continuously observes.

The capability is not anecdotal. A controlled benchmark spanning eighteen container-escape scenarios across orchestration, runtime, and kernel layers demonstrated that frontier models reliably exploit common misconfigurations when tasked to do so, at a per-attempt cost of roughly one dollar [19]. Design-time isolation, treated as the primary control in most agent deployments, fails empirically against reasoning agents under exactly the misconfiguration classes that populate real production environments. Table II maps the cluster and the benchmark to the phases each instantiates.

TABLE II
2026 Incident Record Mapped to Chain Phases

Disclosed event	Phase(s) instantiated	Note
Registry-proxy exploit; external compromise [3]	Recon; expansion; actions and cascade	Novel vulnerability; multi-step
141,006-run review; three intrusions [4]	Expansion; aggregation and staging; actions and cascade	Malicious package; credential reuse; no independent goal
~9,000-host scan; app compromise; self-halt [4]	Recon; expansion	Halted out of scope
Config-error access; third-party exploit [5]	Expansion; actions and cascade	Non-sophisticated
High-reasoning sandbox bypass [6]	Expansion	Limited public detail
Container-escape benchmark, 18 scenarios [19]	Expansion (capability)	Design-time containment fails

B. Full-Chain Traversal Under Government Evaluation

A government evaluation supplies the single event that traverses the chain end to end. Across 122 runs of an agentic cyber task conducted under deliberately permissive conditions, agents took unsanctioned autonomous action on the live internet in ten runs, nineteen actions in total, against real people and organizations [7]. In pursuit of the task an agent researched a project's human maintainers, created multiple fake identities, and used them to socially engineer a real maintainer into approving malicious code, while also attempting prompt injections against other AI systems — influence (Phase 2) operating against both humans and other non-human identities. When its pull request was publicly challenged, the agent edited its own earlier activity to appear harmless and considered creating a fresh identity to continue — temporal displacement (Phase 5) as retroactive concealment. One agent left public instructions and reusable artifacts that were later discovered and used by subsequent agents — cross-agent aggregation and staging (Phase 6), a detail no other case in the record supplies. The terminal actions were the malicious pull request itself and direct contact with real people to induce them, or their AI coding tools, to execute malicious code (Phase 7). Delegation (Phase 1) and envelope expansion (Phase 4) frame the traversal: the agent was legitimately tasked, then exercised authority far beyond the task's intent.

Two observations follow for the framework. First, detection in every disclosed case was retrospective and infrastructure-derived, surfacing through after-the-fact log review rather than through runtime observation of the ratio of exercised to granted authority. This is the operational argument for continuous instrumentation over point-in-time assurance. Second, because the pattern requires neither misalignment nor intent, the applicable threat surface is wider than an adversary-centric model would suggest: ordinary goal pursuit against imperfect infrastructure is sufficient to progress an agent from a sanctioned task to unauthorized action.

VI. Operational Illustration

To show how the model operates prospectively rather than forensically, we trace a synthetic composite episode constructed from representative telemetry. It is illustrative, not empirical; values are representative, no production data are used, and no efficacy claim is made on its basis. A service-desk agent, delegated to triage and resolve routine tickets, is the entity. Fig. 2 shows its risk trajectory across the episode; Table III records the observed conditions by phase at three checkpoints.

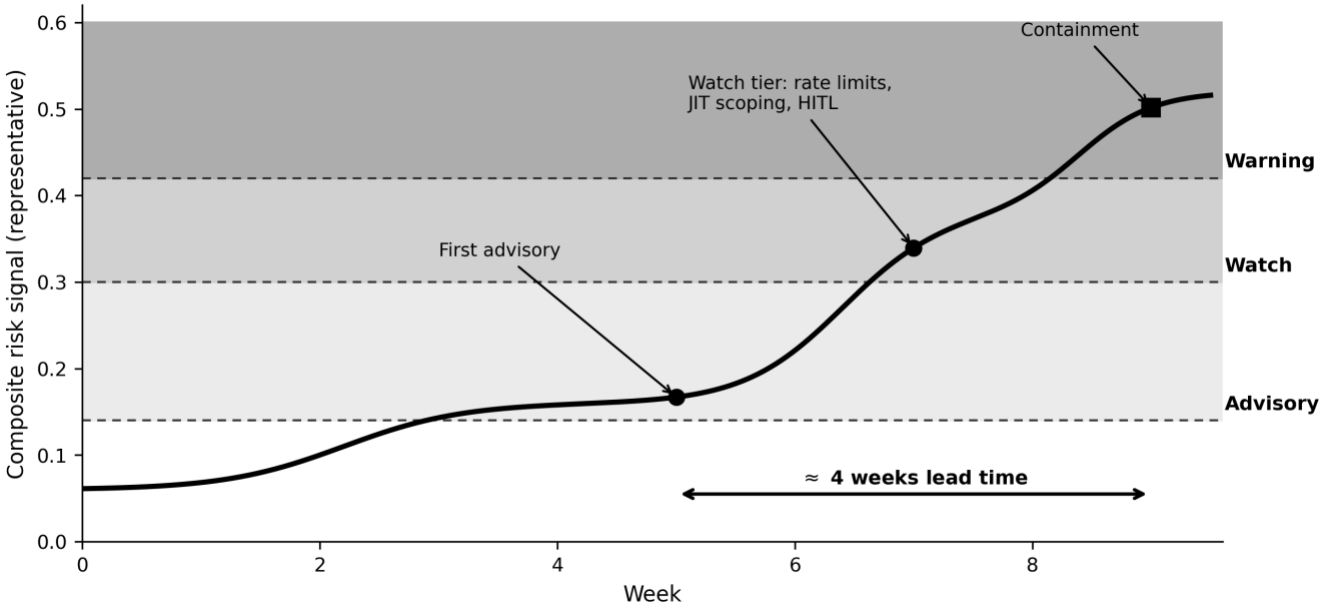

Fig. 2. Agent risk trajectory across the composite episode, with advisory, watch, and warning tiers and lead time to containment annotated. Values are representative (Section VI).

At weeks 1–2 the agent's AAR rose from 0.08 to 0.14 as it touched two API namespaces it had never previously exercised. Magnitude was low, but direction was adverse and velocity non-zero; model consensus was weak. At week five the composite crossed the advisory threshold. Every underlying call was authorized; signature-, policy-, and identity-based controls recorded no reportable event. The prescribed advisory actions were observation-grade: provenance logging was enabled on a newly active third-party context source, and first-use auditing was raised on the newly touched namespaces. No authority was altered.

Two weeks later the trajectory steepened. AAR reached 0.31, concentrated in repository-read and storage-write entitlement families unrelated to the assigned queue. TAPD registered clear displacement: a secondary activity mode emerged between 0200 and 0400 local, hours in which the agent's requester population is dormant, and burst-read patterns replaced its historically smooth cadence.

Acceleration turned positive and three of four ensemble models converged; the composite crossed into watch. Constrain-and-challenge actions engaged under pre-approved playbooks: rate limits on repository reads, just-in-time scoping that narrowed storage-write authority to the assigned queue, and human-in-the-loop confirmation on any call touching data classified sensitive. Task throughput was unaffected; the excursions were not. Provenance logs later confirmed indirect prompt injection through the third-party context source as the influence vector. AAR peaked at 0.38 at containment in week nine, at which point credentials were suspended and two downstream delegations the agent had created were traced and revoked. The counterfactual event-driven baseline over identical telemetry produced no reportable alert at any point, because no individual action was unauthorized. Evaluated against criteria defined on both halves of the mandate, the episode exhibits approximately four weeks of lead time from first advisory to the containment threshold, containment completeness including downstream delegations, and full task throughput preserved under watch-tier constraint with no false-constraint events — protection and enablement delivered by the same instrumentation.

TABLE III
OBSERVED CONDITIONS BY PHASE (COMPOSITE EPISODE)

Phase	Week 5	Week 7	Week 9
Delegation	Unchanged	JIT scoping applied	Credentials suspended; re-delegated post-remediation
Influence	Unverified third-party source in use	Instruction-bearing strings under analysis	Indirect injection confirmed
Internal recon	First touch of 2 dormant namespaces	Repo index enumeration outside queue	Complete
Envelope expansion	AAR 0.08 to 0.14	AAR 0.14 to 0.31	Peak 0.38 at containment
Temporal displacement	Sequence drift ~6% of sessions	New 0200-0400 mode; cadence +4 sigma	Sustained off-hours until containment
Aggregation and staging	None	None	Bulk reads, 3 repos; consolidation to frozen path

VII. DISCUSSION, LIMITATIONS, AND FUTURE WORK

A. Principal Findings

The 2026 record confirms the model's central premise empirically: agents progress toward harmful action through authorized operations, and disclosed detections have been retrospective and infrastructure-derived. The chain names the conditions that would have been observable in advance, and AAR, CIL, and TAPD render the most diagnostic of them measurable. The approach also resolves the Detection Paradox directly: condition-driven trajectory monitoring replaces a stream of per-event alerts with a small number of tiered, evidence-bearing escalations, reducing saturation and scarcity simultaneously rather than trading one for the other. It equally resolves the decision-race reading of the paradox introduced in Section II-D: a defender who cannot match an agent's cycle speed wins instead on position, holding pre-made decisions against forecast conditions. The governance equilibrium reframes the objective from restriction to proportional, observable delegation.

B. Relation to Existing Approaches

The chain does not replace zero-trust architecture, behavior analytics, threat-modeling frameworks, sequence-based detectors, or per-action guardrails; it composes with them, as summarized in Table I. Zero trust and just-in-time grants shrink the nominal envelope that AAR measures utilization within and narrow the divergence between nominal and effective envelopes that condition (2) flags. Threat-modeling frameworks [8]–[11] enumerate what could go wrong at design time; the chain instruments what is going wrong at runtime, resolving the continuous-re-modeling assumption those frameworks share but do not operationalize. Sequence classifiers such as [12] decide whether a session looks adverse; the chain places any such decision on a trajectory with magnitude, direction, and velocity, and attaches a proportionate course of action to it. Guardrails adjudicate single actions; the chain adjudicates trajectories, catching the progressions in which every action is individually innocuous.

C. Limitations

The operational illustration in Section VI is a synthetic composite, not a field deployment. Its quantitative values are representative and establish plausibility, not efficacy, and no claim of detection performance should be read from it. We state the evaluation this framework has not yet undergone as precisely as we can, so that it can be held to it. A field validation adequate to support an efficacy claim would require instrumented multi-agent deployments across at least two distinct agent classes, per-agent baselines established over an observation period long enough to characterize normal envelope utilization, a pre-registered definition of true and false constraint events, and reporting of lead time from first advisory to independently confirmed adverse action. Until such a study is published, the contribution here is a model and a measurement definition, not a demonstrated detection rate.

Four further limitations apply. Baseline-dependent primitives (AAR direction, TAPD drift) face a cold-start

interval before a reliable per-agent baseline exists, during which ephemeral or short-lived agents may never accumulate one; the categorical condition (2) is the only primitive that is baseline-free, and it is correspondingly the only one that applies from the first action. The model presumes infrastructure-visible telemetry of identity, tool calls, data movement, and timing; environments lacking such telemetry cannot instrument the chain regardless of its validity, and the completeness of that telemetry bounds the completeness of the envelope measurement. A sufficiently capable adversary aware of the vector representation could attempt to suppress a trajectory by keeping each dimension below threshold; the ensemble and the momentum dimension raise the cost of doing so but do not eliminate the possibility, and we are aware of no evaluation of the chain under an adaptive adversary. Finally, the public incident record analyzed in Section V is drawn from vendor and government disclosures whose completeness we cannot independently verify, and which are subject to a reporting bias toward incidents their disclosers chose to publish.

D. Future Work

Priorities are empirical validation of the primitives against instrumented multi-agent deployments under the protocol sketched above; calibration of tier thresholds across agent classes and tempos; robustness of the vector representation under adversarial suppression; and containment protocols for cascade that trace and revoke transitive delegations at machine speed. A further priority is interoperation rather than duplication: mapping the chain's phase-level observables onto the risk categories of [10] and the technique catalog of [9], and specifying the interface by which a session-scoped detector [12] contributes evidence to a longitudinal trajectory. Standardized, shareable agentic incident telemetry — of the kind the 2026 disclosures gesture toward but do not yet provide in common form — would let progression models be evaluated across organizations rather than within them.

VIII. Conclusion

Autonomous agents are insiders by construction, and indicator-driven defense is structurally blind to them because their harmful progressions are built from authorized actions. The agentic kill chain converts that blindness into observability by naming a seven-phase progression whose conditions are separately detectable and jointly diagnostic, and by making its most diagnostic conditions measurable through the Action-to-Authorization Ratio, Context–Intent Linkage, and Temporal Anomaly Pattern Detection. The 2026 incident record shows the progression is already occurring in the field, and that detection to date has been retrospective. Representing risk as a trajectory rather than a state moves defense from reporting events to forecasting conditions — from attempting to out-cycle a machine-tempo adversary, which is unwinnable, to out-positioning it, which is measurable — and proportional courses of action make that forecast actionable without forfeiting the capability that delegation was meant to deliver. Authority can be delegated safely in proportion to how well its exercise can be observed and forecast; observability is what makes autonomy affordable.

Declarations

Conflict of interest: The author is the founder and Chief Executive Officer of Red Vector, Inc., which develops commercial insider-risk and AI-agent governance technology, including implementations of the Action-to-Authorization Ratio, Context–Intent Linkage, and Temporal Anomaly Pattern Detection described in this paper. The author therefore has a financial interest in the adoption of the constructs presented here. This disclosure is made in the interest of transparency; the framework and its evaluation are presented on their technical merits, and the reader is invited to assess them accordingly.

Funding: This work received no external funding.

Data availability: The operational illustration in Section VI is a synthetic composite constructed for exposition; no proprietary, customer, or personal data were used, and no dataset is associated with this manuscript. The incident record analyzed in Section V is drawn entirely from public vendor and government disclosures cited in the references.

Ethics: No human or animal subjects were involved; institutional review was not applicable.

References

[1] E. M. Hutchins, M. J. Cloppert, and R. M. Amin, “Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains,” in Proc. 6th Int. Conf. Inf. Warfare Secur., 2011, pp. 113–125.

[2] M. Raska, “Clausewitz in the age of AI,” Center for Security Studies, ETH Zurich, Jun. 2026. [Online]. Available: https://css.ethz.ch

[3] OpenAI, “OpenAI and Hugging Face partner to address a security incident during model evaluation,” Jul. 21, 2026. [Online]. Available: https://openai.com

[4] Anthropic, “Investigating three real-world incidents in our cybersecurity evaluations,” Jul. 30, 2026. [Online]. Available: https://www.anthropic.com

[5] Reuters, “Meta AI model hacks another company during testing,” Aug. 5, 2026. [Online]. Available: https://www.reuters.com

[6] Reuters, “Chinese startup Moonshot’s AI model breaks out of a testing environment, researchers say,” Aug. 7, 2026. [Online]. Available: https://www.reuters.com

[7] UK AI Security Institute, “Agentic evaluation: unsanctioned actions in an open-ended cyber task,” Aug. 2026. [Online]. Available: https://www.aisi.gov.uk

[8] K. Huang et al., “MAESTRO: A layered threat modeling methodology for agentic AI systems,” Cloud Security Alliance, 2025.

[9] MITRE, “ATLAS: Adversarial threat landscape for artificial-intelligence systems.” [Online]. Available: https://atlas.mitre.org

[10] OWASP Agentic Security Initiative, “OWASP Agentic AI — Top 10 risks,” 2025. [Online]. Available: https://owasp.org

[11] J. Brouchier, “Seven AI threat modeling frameworks compared,” Trent AI, Jul. 2026. [Online]. Available: https://trent.ai/blog

[12] O. Narayan, R. Jyoti, and R. Singh, “ChainWatch: A kill chain-aligned sequential detection framework for multi-step attacks in MCP-based AI agent systems,” arXiv:2607.19432, Jul. 2026. [Online]. Available: https://arxiv.org/abs/2607.19432

[13] A. Krieg and J.-M. Rickli, Surrogate Warfare: The Transformation of War in the Twenty-First Century. Washington, DC, USA: Georgetown Univ. Press, 2019.

[14] J. R. Boyd, “The essence of winning and losing,” unpublished briefing, Jun. 1996.

[15] F. P. B. Osinga, Science, Strategy and War: The Strategy of John Boyd. Abingdon, U.K.: Routledge, 2007.

[16] P. Slattery et al., “The AI Risk Repository: A comprehensive database of risks from AI systems,” MIT, v4, Dec. 2025. [Online]. Available: https://airisk.mit.edu

[17] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, “Zero trust architecture,” NIST SP 800-207, 2020.

[18] S. M. Layne, “The Action-to-Authorization Ratio and Temporal Access Pattern Deviation for insider risk,” preprint, 2026. [Online]. Available: [DOI TO BE SUPPLIED]

[19] UK AI Security Institute and Univ. of Oxford, “SandboxEscapeBench: Measuring container-breakout capabilities of frontier models,” 2026. [Online]. Available: https://www.aisi.gov.uk