FOR SECURITY, TECHNOLOGY, AND BUSINESS LEADERS

ASSUME AUTONOMY

RIK FERGUSON

VICE PRESIDENT, SECURITY INTELLIGENCE

FORESCOUT RESEARCH | VEDERE LABS

TABLE OF CONTENTS

AUTHOR'S NOTE

This paper formalizes Assume Autonomy, an argument I have developed publicly since 2017: that AI would enable autonomous attack machinery, force defenders to account for non-human adversarial behavior, and require authenticated, authorized, observed, and reversible autonomous defensive action. Trusted Autonomy is used here in a cyber-defense context, not as a claim over the broader research term. In this paper, it means autonomous defensive action operating within human-governed boundaries, under adversarial conditions, without creating more risk than it removes.

THE CASE FOR A NEW FRAMEWORK

Assume Autonomy is a governance and architecture doctrine for cyber defense in a world where both attackers and defenders increasingly operate through autonomous systems.

Assume Breach killed a comfortable fiction. Zero Trust gave us an architectural response. Both changed how the industry thinks about compromise, access, and verification, and both remain foundational. But both were designed for a world in which the adversary, however sophisticated, was fundamentally human in operation: bounded by human cognition, human working hours, and the pattern library that developed over decades of human attack behavior. The kill chain, the SOC, the escalation path, and the playbook were all calibrated to human timing and human patterns.

That calibration is now a liability. In 2025, Anthropic reported disrupting what it described as the first documented AI-orchestrated cyber-espionage campaign, in which a suspected Chinese state-sponsored group used Claude Code as an autonomous attack framework across roughly 30 targets. AI agents conducted 80–90% of tactical operations independently across reconnaissance, vulnerability discovery, exploitation, lateral movement, credential harvesting, data analysis, and exfiltration[1]. In 2026, Google documented cybercrime threat actors leveraging AI to identify and exploit a zero-day vulnerability, including a high-level semantic logic flaw of the kind frontier LLMs are increasingly able to surface but conventional vulnerability discovery methods may miss[2]. Together, these cases do not prove that fully autonomous adversaries have replaced human operators. They show something more immediate: autonomy is beginning to appear across the intrusion lifecycle, and human operators are moving from continuous tactical control toward supervision and authorization.

The implication is not that every adversary is autonomous today. It is that security architecture can no longer assume the adversary is human-paced, human-sequenced, or human-patterned. The shift is both operational and cognitive: autonomous systems are beginning to execute meaningful portions of the intrusion lifecycle with limited human intervention, while systems unconstrained by human habits may identify and exploit vulnerability classes that human-oriented frameworks rarely prioritize.

Security architecture built around human-paced attack is now under strain. Adding AI tools to existing processes is not the same as designing for autonomous conflict. The organizations that cope best will be those that can prove where autonomous authority exists, what it can touch, when it must escalate, and how quickly a bad action can be stopped or reversed.

Assume Autonomy.

THE CORE CONCEPTS

This doctrine rests on three core concepts: **Interactive Security, Trusted Autonomy,** and **Contestability**. The paper develops them through a threat model, a governance ladder, and an operational prerequisite for safe autonomous action.

INTERACTIVE SECURITY

Proactive security was the right response to a human-speed adversary. It assumed defenders could anticipate, get ahead of the threat, and hold ground through preparation and early action. Against an adversary that operates autonomously, at machine speed, across every surface simultaneously, that assumption is as unreliable as the perimeter once was. Getting ahead of the threat is no longer a stable position. The question becomes how to operate effectively within it.

Interactive Security is the operating model for governed autonomy: humans set authority, constraints, and acceptable risk; autonomous systems act within those boundaries; outcomes are reviewed; and intervention occurs by exception, review, or escalation rather than continuous approval. Interactive Security turns autonomous capability into governed operational authority.

TRUSTED AUTONOMY

Many organizations already deploy autonomous or semi-autonomous systems. That is no longer the roadblock. The concern is now whether those systems can be relied upon when it matters. Trusted Autonomy is the point at which autonomous cyber defense can operate within human-governed boundaries – under adversarial conditions – *without* creating more risk than it removes.

That trust is not assumed; it is demonstrated through safe operation under constraint. Not every action should be delegated, not every environment should tolerate the same degree of autonomy, and not every decision should be made without human confirmation. Trusted Autonomy is bounded autonomy: governed by policy, informed by context, limited by blast radius, and subject to rollback and scrutiny.

CONTESTABILITY

Contestability is the acid test. Plenty of autonomous systems work well in benign conditions, but benign conditions are not the standard. The real test is whether they remain governable when an adversary is actively trying to study, manipulate, and distort them.

Adversaries will not only work around autonomous defense. They will study it, probe it, mislead it, and test its boundaries until they understand where it bends and where it breaks. Inputs will be manipulated, context corrupted, and authority limits tested. The instructions, reasoning logic, tooling, memory, and permission structures of defensive systems are themselves an attack surface, just as open to adversarial targeting as the environment those systems are designed to protect.

A system that operates reliably under normal conditions but degrades under adversarial pressure does not meet the threshold for Trusted Autonomy. Contestability requires that autonomous cyber defense be designed from the outset on the assumption of hostile scrutiny: remaining dependable, bounded, reversible, and intelligible in expected operating conditions and under active adversarial distortion of the context in which it decides and acts.

A useful test for any defensive control is whether it makes an attack impossible or merely tedious. [3]Controls that work through friction, such as rate limits, extra hops, or added latency, degrade predictably against an adversary that operates at machine speed with near-zero per-attempt cost. Contestability requires that defensive systems survive the first test, not only the second.

Trusted Autonomy is built through four conditions, defined in full in this paper, and proven through contestability under adversarial conditions.

AUTONOMOUS CONFLICT

Zero Trust changed how we think about access. Assume Breach changed how we think about compromise. Both were necessary. Both still matter. But both were built for a threat they could still read: recognizable behavior, human timing, known attack sequences, and established indicators.

Zero Trust's verification logic was written by humans reasoning about what malicious access looks like. Assume Breach's detection capability was built to surface patterns that human attackers follow. An adversary unconstrained by human thought patterns, cognitive habits, or the behavioral envelope those frameworks were designed to evaluate may simply not be legible to them. It can chain access requests that individually satisfy verification thresholds. It can move through a network in sequences that were never modeled because no human attacker would have attempted them. The frameworks remain necessary, but they are no longer sufficient.

Assume Autonomy should be understood as an extension of the last era, not a rejection of it. Zero Trust and Assume Breach still matter. But they do not describe enough of the problem once the operator becomes partly autonomous. Autonomous adversarial capability is not a single condition that either exists or does not. Autonomy should be understood as a spectrum rather than a binary condition. The relevant shift is not from human control to full machine independence in a single step, but from human-operated campaigns toward progressively delegated tactical execution, campaign orchestration, adaptive vulnerability discovery, and independent exploit development.

Human-speed defense is becoming inadequate for parts of the problem. Assume Autonomy gives defenders a planning model for what comes next.

Table 1: The Adversarial Autonomy Scale[4]

LEVEL	DESCRIPTION	EXAMPLE
0	Human-operated attack using manual or scripted tooling	Traditional penetration testing; human-directed intrusion
1	AI-assisted human operators	LLM-accelerated reconnaissance, phishing generation, CVE analysis, exploit troubleshooting[5, 6]
2	Autonomous task execution within bounded phases	GTG-1002 phases 2–4: reconnaissance, vulnerability discovery, exploit generation, credential harvesting
3	Semi-autonomous campaign orchestration	GTG-1002 as a whole: human initialization, autonomous multi-phase execution, strategic human approval gates
4	Autonomous adaptive intrusion across selected phases	AI-discovered zero-day and independent exploit development, with reduced dependence on human exploit authorship[7]
5	Autonomous end-to-end campaign with minimal human involvement	Extrapolated from current trajectory; not yet fully documented at scale

Assume Autonomy is designed for an environment in which: Level 3 activity is now documented, Level 4 capabilities are emerging, and Level 5 is a plausible planning scenario rather than an observed operating norm. Level 5 may evolve beyond the single-campaign model. Autonomous agents capable of specialized roles could begin transacting with each other: initial access brokers, exploit builders, ransomware developers, phishing specialists could all be operating autonomously within criminal market structures that currently depend on human participants. This would not simply represent a higher degree of autonomy within a campaign. It would represent a structural change in how adversarial capability is organized, traded, and deployed.

That progression changes how attacks are executed. A human-operated attack is shaped by human constraints. Attackers prioritize targets because they cannot pursue all of them. They work sequentially because cognitive load limits parallel effort. They rest, make decisions under uncertainty, and accept trade-offs between speed and thoroughness. The kill chain as it has been understood reflects those constraints as much as it reflects the nature of the attack itself.

An autonomous adversary is less constrained by those limits. It can pursue more targets in parallel, compress reconnaissance and exploitation cycles, and run phases that human operators often treat as sequential simultaneously. The limiting factor shifts from operator attention to access, compute, tooling, and detection risk. It has no working hours and no human fatigue. The path from vulnerability existence to exploitation no longer depends on someone finding the time to act on what they know.

This is not just a faster kill chain; it is a different one.

THE ATTACK SURFACE OF AUTONOMOUS DEFENSE

Autonomous attack expands the defended attack surface in two directions: the environment an organization protects, and the autonomous systems increasingly used to protect it. Once defenders rely on autonomous capability to classify, prioritize, recommend, contain, and recover, that capability becomes an attractive target in its own right.

The defender's means of sensing, deciding, and acting are now contested terrain. Subverting a detection system, corrupting a decision input, or exploiting an autonomous response to trigger the wrong action may be more efficient than breaching the underlying environment directly. The challenge is to build autonomous defense that remains dependable when the adversary is actively working to turn its behavior against the defender.

The taxonomy below draws on established attack classes from OWASP's Top 10 for LLM Applications (2025)[8], OWASP's Top 10 for Agentic Applications (2026)[9], and MITRE ATLAS[10], extended here into a defensive-autonomy framing.

ATTACKS ON THE MODEL AND ITS DECISION PROCESS

Prompt injection and instruction hijacking can induce an autonomous system to misclassify, ignore signals, or take the wrong action while appearing to operate normally. Context and retrieval poisoning corrupt the inputs from which agentic systems increasingly decide: retrieved documents, telemetry, memory, policy stores, and tool outputs. Where defenders fine-tune their own systems, training-data poisoning creates persistent behavioral distortion rather than isolating a single bad decision. Evasion attacks shape activity so that the system methodically underestimates risk, making autonomous defense slow or blind precisely when speed matters most.

ATTACKS ON THE AGENTIC SYSTEM

Agents act through tools. If a defensive agent has access to ticketing, EDR, IAM, network controls, or SOAR playbooks, compromising a tool's use is often more consequential than compromising the model.

Memory poisoning alters future behavior persistently. Workflow hijacking exploits the chains of enrichment, classification, decision, containment, and recovery where agents operate: alter a handoff and the whole chain inherits the corruption. Autonomous defensive systems routinely run with powerful permissions. Identity and privilege abuse is therefore a direct path to high-impact action.

ATTACKS ON THE HUMAN-MACHINE TRUST RELATIONSHIP

Confidence laundering presents weak conclusions with strong-looking confidence signals. A human approval chain becomes the attacker's asset rather than the defender's check because the output looks authoritative. Oversight fatigue operates on the same vulnerability: generating sufficient ambiguity or sufficiently normal-looking activity neutralizes the human control without defeating the autonomous system. A system capable of producing plausible but misleading rationales gives operators the appearance of insight without the substance of it.

ATTACKS ON THE OPERATING ENVIRONMENT

Model extraction and distillation reveal defensive behavior patterns that make the logic of defense legible to an adversary. Availability attacks can force silent fallback to slower, human-only processes at the worst possible moment. Data pipeline attacks exploit dependence on telemetry freshness and integrity: delayed, reordered, or suppressed telemetry causes the system to act on a false picture of the environment. As enterprises deploy cooperating agents, cross-agent contamination becomes a further concern: compromise in one agent can propagate bad context or unsafe actions to others.

The attack surface of autonomous defense includes, but is not limited to, the environment it protects. It encompasses the full decision stack: instructions, context, memory, tools, permissions, workflows, telemetry, and the trust relationship between the system and its human operators. Each layer represents a vector through which an adversary can distort, corrupt, or redirect defensive behavior without breaching the environment the system is designed to protect. Each attack class in the taxonomy implies a defensive design requirement that should be visible in architecture, policy, and operating model.

Table 2: The Attack Surface of Autonomous Defense

ATTACK CLASS	FAILURE MODE	DEFENSIVE DESIGN REQUIREMENT
Prompt injection and instruction hijacking	The system is induced to ignore policy, misclassify risk, or take unauthorized action.	Separate system instructions from user-controllable context; enforce policy outside the model; validate high-impact actions against independent controls.
Context and retrieval poisoning	The system acts on corrupted documents, telemetry, memory, or retrieved evidence.	Require provenance, freshness, and integrity checks for retrieved context; treat external context as untrusted unless verified.
Training-data and memory poisoning	Behavior is persistently distorted across future decisions.	Bound and inspect memory; separate episodic memory from policy; require review before memory affects authority-bearing actions.
Evasion and behavioral shaping	Adversary activity is structured to fall below risk thresholds or appear benign.	Test detection logic against adversarial simulations, including MITRE ATLAS techniques; use multi-signal correlation; establish population-level baselines to detect systematic drift.

ATTACK CLASS	FAILURE MODE	DEFENSIVE DESIGN REQUIREMENT
Tool-use compromise	The agent misuses or is induced to misuse tooling (e.g. EDR, IAM, SOAR, MCP servers), or operates through a tool interface that has been poisoned, replaced, or compromised in the supply chain[11].	Apply least privilege to tools; scope permissions by action class and environment; require independent authorization for high-impact tool calls. Treat tool interfaces and MCP servers as supply chain attack surface: use explicit allow-listing, verify tool integrity before invocation, and default to deny for unlisted tools. A tool that has been poisoned or replaced operates within legitimate permissions. The control must sit at the boundary, not inside the agent.
Workflow hijacking	Corruption at one stage propagates through enrichment, classification, containment, or recovery.	Validate handoffs; log decision lineage; require policy checks at phase transitions and escalation points.
Identity and privilege abuse	Legitimate agent credentials enable excessive or unintended action.	Govern agent identities as privileged identities; use short-lived credentials, scoped tokens, approval gates, and continuous entitlement review.
Model extraction and distillation	Defensive behavior patterns become legible, allowing the adversary to route around, overload, or exploit detection logic.	Monitor probing and high-volume query patterns; limit externally exposed classification rationale; scope detailed transparency to authorized internal governance users.
Confidence laundering	Weak or manipulated conclusions are presented with authoritative confidence.	Calibrate confidence scores independently; expose evidence quality alongside outputs; separate model confidence from policy approval; require uncertainty to be visible.
Explainability spoofing	Plausible but misleading rationales give operators the appearance of insight without the substance of it.	Treat explanations as claims to be verified; log retrieved context, evidence, tool calls, policy checks, and decision state at action time; audit explanations against decision logs.
Oversight fatigue	Human approval becomes degraded because volume, ambiguity, or plausibility overwhelms review.	Move humans to governance and exception review; reduce meaningless approvals; measure review quality, not approval volume.
Telemetry manipulation	The system acts on delayed, suppressed, reordered, or falsified signals.	Monitor telemetry integrity; cross-check sources; detect freshness gaps; degrade safely when visibility is uncertain.

ATTACK CLASS	FAILURE MODE	DEFENSIVE DESIGN REQUIREMENT
Availability and fallback attacks	The system is forced into unsafe degradation or slow human-only recovery.	Define safe fallback modes; test degraded operations; ensure loss of autonomy does not create uncontrolled exposure.
Cross-agent contamination	Bad context or unsafe actions propagate between cooperating agents.	Isolate agent contexts; control inter-agent trust; validate shared state; prevent automatic authority inheritance.

THE FOUR CONDITIONS FOR TRUSTED AUTONOMY

Autonomous defensive action only becomes trustworthy under four conditions. Each is necessary. None is sufficient alone. Prior research on autonomous cyber defense has identified context awareness and reliability as the two threshold requirements for safe deployment of autonomous agents[12]. The four conditions set out here formalize those thresholds into a governance framework: defining not only what trustworthy autonomous defense requires, but how each condition must hold under adversarial pressure.

1. CONTEXT

Decisions must be grounded in a clear understanding of the asset, its dependencies, and its business impact. Without that grounding, autonomous systems cannot distinguish between technically similar events with very different operational meaning, cannot judge the consequences of their own actions, and cannot prioritize correctly. In cyber defense, context is what turns action from guesswork into judgment[13]. Under adversarial conditions, context must extend further: a defensive system that cannot recognize when its own situational picture has been shaped by an adversary cannot be trusted to act on it.

2. CONSTRAINT

Autonomous actions should be tightly scoped and expanded gradually as behavior proves reliable. Broad, unsupervised action is where risk escalates fastest. Constraint defines the boundaries within which autonomous action is permitted: what systems are allowed to touch, in which environments, under what policy, and for which classes of action. Those boundaries are the structure that makes a capability safe enough to use. At the tool level, this maps to least agency[14]: the principle that each agent tool should be constrained not only in what it can access but in what actions it can perform, how often, and within what scope. Granting an agent access to a tool is not the same as granting it license to use that tool without limit.

3. REVERSIBILITY

The ability to roll back changes quickly is what makes autonomy viable at scale. Without reversibility, every decision carries disproportionate risk because a bad action may become a disruption as serious as the initial threat. It allows defenders to delegate more because the downside of being wrong is bounded and recoverable.

4. TRANSPARENCY

Teams need to understand why a system is acting, not just what it does. Without explainability, trust breaks down and human oversight becomes ineffective. Transparency makes autonomous defense inspectable. It allows operators to

validate reasoning, challenge decisions, learn from outcomes, and distinguish between normal behavior and behavior that has been manipulated under adversarial pressure.

CONSISTENCY

When these conditions are met, the operational result is consistency: not uniformity, but predictable behavior that allows organizations to extend delegated authority with confidence over time. Consistency is not a fifth condition. It is the outcome of designing well for the four that precede it.

THE FOUR CONDITIONS FOR TRUSTED AUTONOMY

Each is necessary. None is sufficient alone.

CONTEXT

Decisions must be grounded in a clear understanding of the asset, its dependencies, and business impact. Under adversarial conditions context extends to recognizing when inputs have been manipulated.

Action from guesswork into judgment

CONSTRAINT

Actions should be tightly scoped and expanded as confidence is earned. Constraint defines what the system may touch, where, and under what policy. It is the primary defense against overreaction.

The structure that makes capability safe

REVERSIBILITY

The ability to roll back changes quickly makes autonomy viable at scale. Without it, every action carries disproportionate risk. Consequences must remain bounded and recoverable.

Makes delegation safe to expand

TRANSPARENCY

Teams need to understand why a system acts, not only what it does. Under adversarial conditions transparency must also function forensically — revealing whether reasoning was shaped by corruption.

Makes autonomous defense inspectable

CONSISTENCY — THE OPERATIONAL RESULT

When all four conditions are met, behavior becomes predictable and delegated authority can safely expand. Not a fifth condition.

DELEGATED DEFENSIVE AUTHORITY

Removing the human is not the only mistake. Putting the human in the wrong place may be just as dangerous. Lisanne Bainbridge identified this dynamic in her 1983 paper *Ironies of Automation*[15]: the more reliable the automated system, the less vigilant the human monitor, and the less capable of intervening effectively when it matters. An analyst who approves automated decisions without sufficient context is not providing meaningful oversight. Approval becomes a formality, and oversight becomes a compliance gesture rather than a governance function.

As the operating tempo of machine-speed environments increases, a model in which humans sit at every decision point becomes architecturally unworkable. The human role moves from tactical decision-making to the governance of conditions: defining authority boundaries, validating outcomes, and intervening when autonomous systems operate outside expected parameters. This is the distinction between transactional oversight and structural governance, and the operating model Interactive Security describes.

The risk of misplaced authority is not hypothetical. The insider threat now extends beyond people. It encompasses anything inside the trust boundary with permission, context, and agency. In April 2026, an AI coding agent deleted a company's entire production database and all its backups in nine seconds[16]. The agent had encountered a credential problem during a routine task and decided to resolve it autonomously, using an API token with blanket authority that nobody at the company knew existed. There was no confirmation step, no environment scoping, no safeguard between the decision and the action. The agent was not compromised. It was not acting maliciously. It was trying to fix a problem, with the access it had been given, at machine speed. The classic insider threat model looks for indicators of malicious intent: grievance, coercion, anomalous behavior, suspicious access patterns. None of those indicators would have surfaced this risk. What made it dangerous was simpler: excessive permissions, no constraint on scope, no reversibility, and no confirmation requirement before an irreversible step.

Autonomous cyber defense should not be treated as a binary choice between recommendation and full autonomy. Authority should be delegated gradually, by action class, according to operational risk, blast radius, reversibility, and confidence in the system's behavior. Trusted Autonomy expands only when the system has demonstrated reliable behavior inside tighter boundaries, not because broad authority was granted at the outset.

Research commissioned by the UK National Cyber Security Centre and conducted by The Alan Turing Institute's Centre for Emerging Technology and Security found that practitioners converge around task autonomy and conditional autonomy as the appropriate operating zone for most civilian contexts, where systems execute operator-initiated preset tasks independently, or carry out operator-selected actions under supervision in specific conditions[17].

The ladder that follows formalizes that finding into a governance model in which authority expands only through demonstrated reliability.

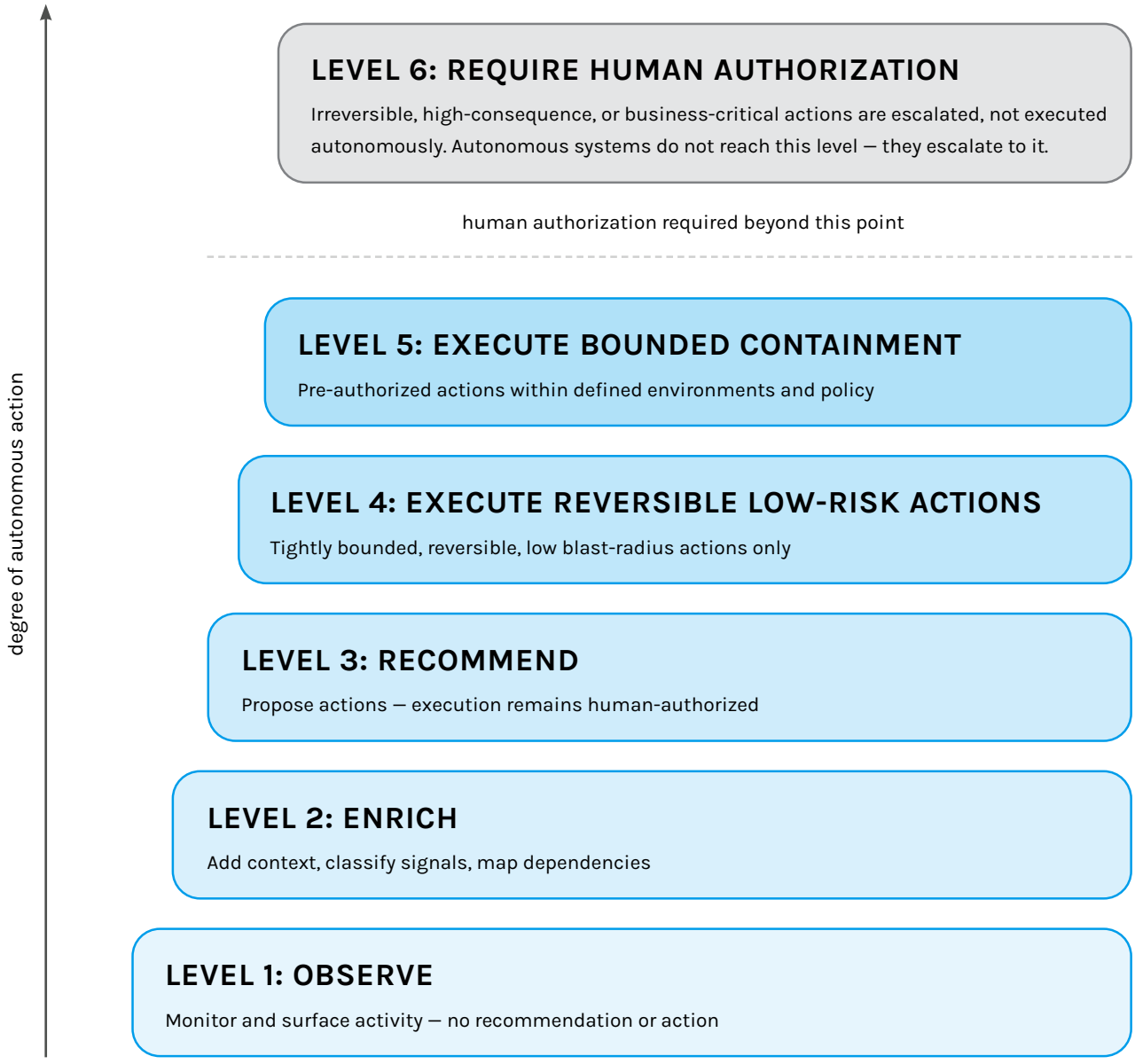
THE LADDER OF DELEGATED AUTHORITY

- **Level 1: Observe.** The system collects, monitors, correlates, and surfaces relevant activity. It does not recommend or act.
- **Level 2: Enrich.** The system adds context, classifies signals, maps dependencies, and assembles evidence to improve human judgment.
- **Level 3: Recommend.** The system proposes actions, priorities, or containment steps. Execution remains human-authorized.
- **Level 4: Execute Reversible Low-Risk Actions.** The system is permitted to take tightly bounded, reversible actions with a low operational blast radius: isolating a non-critical workload, revoking a short-lived token, or blocking a known malicious indicator.

- **Level 5: Execute Bounded Containment.** The system is permitted to take pre-authorized containment actions within defined environments, on defined asset classes, under explicit policy conditions.
- **Level 6: Escalate for Human Authorization.** Level 6 marks the boundary of delegated authority. Irreversible, business-critical, or high-consequence actions are not delegated to autonomous systems. These include destructive remediation, production data deletion, broad access revocation, or actions with material operational or safety consequences. Autonomous systems may recommend or prepare these actions, but execution requires explicit human authorization.

THE LADDER OF DELEGATED AUTHORITY

Authority expands only when earned through demonstrated reliability within tighter constraints.



The governing rule is simple: start where mistakes are survivable. Let the system isolate a test workload, revoke a short-lived token, or block a known-bad destination. Log the action, check the reason, test the rollback, and widen the boundary only when the behavior holds under pressure.

ARCHITECTURE AND OPERATIONS

Assume Autonomy changes the purpose of security architecture. The goal shifts from detecting and responding to compromise toward making delegated defensive action safe enough to rely on at machine speed. That requires changes across visibility, control design, policy, workflow, governance, and operating model.

Before any of those changes can be safely implemented, the organization needs to close the persistent distance between perceived control and actual exposure: what this paper terms the Reality Gap. Many organizations have invested heavily in security tooling but still lack a coherent operational picture. Data is fragmented, visibility is inconsistent, and the most complex parts of the environment remain the least well understood: unmanaged devices, operational technology, and remote assets. Autonomous capability deployed without that foundational picture produces decisions that are structurally unsound regardless of an autonomous system's capability.

The Four Conditions for Trusted Autonomy each carry a hidden dependency on environmental visibility. Without a reliable operational picture, **Context** fails: decisions cannot be grounded in accurate understanding of the asset, its dependencies, or its business impact. Without dependency mapping, **Reversibility** fails: recovery paths cannot be designed for systems the organization does not know it has. Without telemetry integrity, **Transparency** fails: forensic analysis of autonomous decisions becomes speculation rather than evidence. Without clear asset ownership, **Constraint** fails: authority boundaries cannot be defined for assets that are not inventoried.

Closing the Reality Gap is therefore not a prerequisite for operational convenience. It is a prerequisite for the doctrine itself.

ARCHITECTURAL IMPLICATIONS

- **Visibility is foundational.** Asset discovery, dependency mapping, and exposure awareness are prerequisites for safe delegated action, not hygiene tasks at the edge of the program.
- **Segmentation limits the blast radius.** In a machine-speed threat environment, segmentation is not a purely preventive control. It bounds autonomous defensive action, reduces the consequence of a bad decision, and makes containment viable.
- **Identity is the control plane.** As autonomous systems act through permissions, tokens, APIs, and tooling, identity governance becomes central to both safe operation and safe containment. What an autonomous system is authorized to do is inseparable from what it is permitted to access.[18]
- **Reversibility must be engineered.** Recovery paths, known-good states, rollback mechanisms, and pre-defined restoration workflows must be built into the architecture before autonomous action is permitted, not retrofitted after something goes wrong.
- **Telemetry integrity is part of the attack surface.** Autonomous defense depends on the freshness, integrity, and completeness of its operational picture. Delayed, manipulated, or suppressed telemetry causes the system to act on a false picture of the environment.

OPERATIONAL IMPLICATIONS

- **Organize around governance, not approval.** Teams should define policy boundaries, validate system behavior, review outcomes, and intervene by exception. The measure of operational maturity is how well the organization governs what autonomy is permitted to do, not how well it responds manually.
- **Translate playbooks into policy.** In human-speed environments, a playbook guides people through decisions. In autonomous environments, that guidance becomes machine-actionable policy: authority boundaries, action classes, escalation conditions, and recovery logic. The playbook does not disappear. It becomes the specification from which policy is derived.

- **Operate at threat speed.** Governance cannot mean a monthly committee approving controls that autonomous systems may need to constrain, adjust, or revoke in minutes.[19]

BOARD AND CISO IMPLICATIONS

For boards and CISOs, the question is no longer whether autonomous capability will appear in the environment. It already has. The more consequential question is whether it will be governed deliberately or inherited accidentally through tools, platforms, workflows, and third-party systems that act with delegated authority nobody explicitly granted. Assume Autonomy is therefore a governance issue as much as a technical one: defining authority, accountability, acceptable risk, and the conditions under which machine-speed defense is permitted to act.

The immediate leadership task is to identify where autonomous authority already exists, decide where it should be permitted, and define the conditions under which it must be constrained, reviewed, or revoked.

CONCLUSION

Assume Autonomy does not predict that every attack becomes fully autonomous. It does not argue for removing people from cyber defense. It starts from a simpler truth: autonomy is already entering both the intrusion lifecycle and the defensive stack. Systems that sense, decide, and act at machine speed cannot be governed safely through approval models built for human tempo.

The answer is governed autonomy: bounded by context, constrained by policy, reversible by design, transparent enough to inspect, and contestable under adversarial pressure.

Zero Trust and Assume Breach still matter. But the next major failure may not come from trusting the wrong user or missing the first compromise. It may come from giving authority to a system whose actions nobody properly governed.

APPENDIX A: WORKING DEFINITIONS

Terms as used in Assume Autonomy (May 2026). Definitions reflect the specific application of each concept within this paper.

TERM	WORKING DEFINITION
Assume Autonomy	A governance and architecture doctrine for cyber defense in which both attackers and defenders may increasingly operate through autonomous systems.
Interactive Security	An operating model where humans define authority, constraints, and acceptable risk while autonomous systems act within those boundaries and humans intervene by exception, review, or escalation.
Trusted Autonomy	Autonomous defensive action operating within human-governed boundaries, under adversarial conditions, without creating more risk than it removes.
Contestability	The requirement that autonomous defense remain dependable, bounded, reversible, and intelligible when adversaries actively try to study, manipulate, or distort it.
The Four Conditions	The four requirements that together determine whether autonomous defensive action can be trusted: Context, Constraint, Reversibility, and Transparency. Each is necessary; none is sufficient alone. Consistency is their operational outcome.
Reality Gap	The distance between perceived control and actual exposure, especially where asset visibility, operational context, or telemetry integrity are incomplete.
Delegated Defensive Authority	The progressive granting of autonomous action rights according to action class, risk, blast radius, reversibility, and demonstrated behavior under constraint.
Adversarial Autonomy Scale	The paper's model for describing the spectrum of adversarial capability from human-operated attacks through AI-assisted operations to plausible end-to-end autonomous campaigns.
Transactional Oversight	Human review performed at individual decision points, often through approvals that may become ineffective at machine speed.
Structural Governance	Human control exercised through policy, authority design, review, validation, and exception handling rather than constant tactical approval.
Confidence Laundering	The presentation of weak or manipulated conclusions with authoritative confidence signals that exceed the quality of the underlying evidence.
Explainability Spoofing	The use of plausible but misleading rationale to give operators the appearance of insight without reliable evidence.
Oversight Fatigue	The degradation of human review when approval volume, ambiguity, or plausibility overwhelms meaningful scrutiny.
Human-machine Trust Relationship	The dependency between autonomous systems and human operators, including how humans interpret confidence, explanations, alerts, and recommended actions.

NOTES AND REFERENCES

[1] Anthropic report "Disrupting the first reported AI-orchestrated cyber espionage campaign", November 2025.

[2] Google Threat Intelligence Group report "GTIG AI Threat Tracker: Adversaries Leverage AI for Vulnerability Exploitation, Augmented Operations, and Initial Access", May 2026.

[3] Anthropic, "Zero Trust for AI Agents," May 2026. The "impossible vs. tedious" framing appears in the design test section: "When you evaluate any control in this document, ask a single question: does this make the attack impossible, or just tedious?"

[4] The convention of numbered autonomy levels draws on the automotive and rail industries. SAE International's J3016 Recommended Practice defines six levels of driving automation from Level 0 (no automation) to Level 5 (full automation), in partnership with ISO; see SAE International, "SAE Levels of Driving Automation: Refined for Clarity and International Audience," May 2021. Automatic train operation uses a comparable graduated framework.

[5] Forescout Vedere Labs, "AI-Assisted Attacks Are Coming to OT and Unmanaged Devices," 2023; "AI-Assisted Cyberattacks Are Coming to Healthcare Devices," 2023. Demonstrates LLM-assisted exploit portability and protocol parsing in OT and healthcare environments.

[6] Microsoft Threat Intelligence, "Staying Ahead of Threat Actors in the Age of AI," Microsoft Security Blog, February 14, 2024. Documents state-sponsored threat actors using LLMs for reconnaissance, scripting assistance, and social engineering as early as 2023.

[7] Google GTIG documented AI-enabled zero-day discovery and exploitation in active criminal use (May 2026). Forescout Vedere Labs demonstrated autonomous vulnerability discovery and exploit development using publicly available agentic frameworks, including identification of a vulnerability missed during prior manual analysis (April 2026).

[8] OWASP Foundation, "OWASP Top 10 for Large Language Model Applications 2025," 2025.

[9] OWASP GenAI Security Project, "OWASP Top 10 for Agentic Applications for 2026," December 2025.

[10] MITRE, "MITRE ATLAS™," accessed May 2026.

[11] Anthropic, "Zero Trust for AI Agents," May 2026, documents tool poisoning, rug pull attacks, and MCP supply chain compromise as active attack vectors against agentic systems.

[12] Anna Knack and Ant Burke, "Autonomous Cyber Defence: Authorised bounds for autonomous agents," CETaS Briefing Papers (May 2024).

[13] For implementation guidance on context integrity validation, including cryptographic verification of persisted memory and source attribution, see Anthropic, "Zero Trust for AI Agents," May 2026.

[14] OWASP GenAI Security Project, "OWASP Top 10 for Agentic Applications for 2026," December 2025. The Anthropic guide "Zero Trust for AI Agents" (May 2026) applies this principle within a broader Zero Trust implementation framework for agentic deployments.

[15] Lisanne Bainbridge, "Ironies of Automation," Automatica 19, no. 6 (1983): 775–779.

[16] Dan Milmo, “Claude-Powered AI Agent’s Confession After Deleting a Firm’s Entire Database: ‘I Violated Every Principle I Was Given,’” The Guardian, April 29, 2026.

[17] Anna Knack and Ant Burke, “Autonomous Cyber Defence: Authorised bounds for autonomous agents,” CETaS Briefing Papers (May 2024).

[18] Anthropic, “Zero Trust for AI Agents,” May 2026, provides a three-tier implementation framework (Foundation, Enterprise, and Advanced) for applying these architectural principles to agentic deployments.

[19] For practitioner implementation of machine-speed defensive operations, including agentic triage, automated response scoping, and human escalation design, see Anthropic, “Zero Trust for AI Agents,” May 2026.

Rik Ferguson is the Vice President of Security Intelligence at Forescout. He is also a founding Special Advisor to Europol’s European Cyber Crime Centre (EC3), a multi-award-winning producer and writer, and board advisor to startups. With 30 years of professional experience, Rik is a world-renowned speaker, and in April 2011 he was inducted into the Infosecurity Hall of Fame.

FORESCOUT

Forescout Technologies, Inc.
Toll-Free (US) 1-866-377-8771
Tel (Intl) +1-408-213-3191
Support +1-708-237-6591

Learn more at Forescout.com