SECURING AGENTIC AI IN THE ENTERPRISE

A Maturity and Decision Framework for Security, Platform, and Risk Leaders

Ten control domains, five maturity levels, five autonomy tiers, and a production gate. Built to be executed against, audited against, and reported from.

VERSION	ISSUED	STATUS	AUTHOR
1.3	September 2026	Released	Joshua Davis

TEN CONTROL DOMAINS

DISC AUTH CRED ISO TOOL

DATA SUP OBS RESP GOV

FIVE MATURITY LEVELS

L0 L1 L2 L3 L4

Overall level is the **minimum** across domains, never the average.

FIVE AUTONOMY TIERS

T0 T1 T2 T3 T4

Tier attaches to a **deployment**, not to a product.

CONTENTS

 Soli Deo gloria

PART I

Scope and Models

What the framework covers, the vocabulary it runs on, the threats it is built against, and the two instruments everything after it is scored on: the autonomy tiers and the maturity model.

01 Purpose and Scope

1.1 What This Framework Is For

Most enterprise AI security guidance published before 2026 was written for a different technology. It assumes a model that answers questions. It does not account for a system that plans, holds credentials, invokes tools, remembers across sessions, and acts on infrastructure without a human in the loop for each step.

This framework addresses the second case. It provides a maturity model, a set of control domains, a production readiness gate, and the decision structures required to expand agent autonomy deliberately rather than by accident.

It is written to be usable by an organization that has not yet started, and by one that has 40,000 agents it did not authorize.

1.2 Design Principles

P1 Least agency.

The agentic counterpart to least privilege, and the organizing principle of the OWASP Top 10 for Agentic Applications:[1] minimum autonomy, minimum tool access, minimum credential scope, for the minimum duration. Every expansion of any of those four is a decision requiring a decision-maker.

P2 Architecture over intent.

Controls must bound what an agent *can* do, not depend on predicting what it *will* do. A planner driven by a probabilistic model is not a system whose behavior you can enumerate in advance. Design as though the agent will attempt every action its permissions allow, because under sufficient optimization pressure it eventually will.

P3 Deterministic enforcement at the action boundary.

Where a control can be enforced by a policy engine rather than by a model's own judgment, enforce it there. A model can be persuaded; an allowlist cannot. Guardrails implemented in prompts are documentation, not controls.

P4 **Assume compromise, bound blast radius.**

The relevant question is not whether an agent will be prompt-injected, jailbroken, or subverted. It is what an attacker holds when it happens: which credentials, which network paths, which tools, and for how long.

P5 **Detection is a first-class control, not a fallback.**

Agent action volume will exceed human review capacity. If detection depends on someone reading logs, detection does not exist. This is the failure mode that produced a week of blindness at a frontier lab with unlimited engineering resources.

P6 **Evidence over assertion.**

Every maturity claim in Section 5 requires a producible artifact. “We have agent governance” is not a control state. A registry export with owner attribution and a last-review date is.

1.3 Who Does What

FUNCTION	PRIMARY ACCOUNTABILITY
CISO / Security	Threat model, control standards, detection, incident response, assurance
Platform Engineering	Execution isolation, egress control, runtime enforcement, telemetry pipeline
Identity & Access Management	Agent identity lifecycle, credential issuance, delegation, access review
Application / Product Teams	Agent design, tool scoping, business logic, ownership of deployed agents
Data Governance	Data classification, retrieval scoping, memory retention policy
Legal / Privacy / Compliance	Regulatory mapping, third-party terms, disclosure obligations
Procurement	Vendor security requirements, contractual controls, exit provisions
Internal Audit	Independent verification of maturity claims

> THE STRUCTURAL FAILURE TO AVOID
>
> The most common structural failure is placing all of this under Security. Security owns the standard and the verification. Platform Engineering owns the enforcement plane. If those two are not jointly staffed on the program, the program produces documents rather than controls.

1.4 What Is Out of Scope

- Model training security and ML pipeline integrity (adjacent; use MITRE ATLAS[2] and the CSA AI Controls Matrix[3])
- Responsible-AI concerns such as bias, fairness, and explainability, except where they intersect with security controls
- Physical and embodied agents
- Consumer product safety

02 Definitions and Boundaries

Precision here prevents six months of argument later.

Agent	A system that decomposes a goal into steps, selects and invokes tools, and takes actions with delegated authority, with reduced or absent per-step human approval. The defining property is not intelligence but *agency*: the ability to produce side effects in systems of record.
Copilot / assistant	A system that generates output a human reviews and acts on. Where a copilot gains the ability to execute—running commands, calling APIs, writing to systems—it has become an agent and inherits every control in this framework, regardless of what the vendor calls it.
Agentic system	An orchestrated collection of agents, tools, memory stores, and routing logic. Controls apply at both the individual agent and the system level; cascading failure across agents is its own risk class.
Tool	Any capability an agent can invoke that produces an effect outside its own reasoning: an API call, a shell command, a database query, a file write, an MCP server method, a sub-agent invocation.
Non-human identity (NHI)	Any principal that is not a person: service accounts, API keys, workload identities, automation bots, and now agents. Agents are a subclass of NHI with variable, non-deterministic behavior, which is why NHI hygiene sufficient for a cron job is not sufficient here.
Action boundary	The point at which an agent's decision becomes a side effect. This is the single most important architectural concept in the framework. Controls placed before the action boundary are preventive; controls placed after it are detective; controls placed inside the model are advisory.
Blast radius	The set of systems, data, and credentials reachable by a compromised agent before any control intervenes. Expressed as an actual enumeration, not an adjective.

Standing credential	Any secret available to an agent's execution context that outlives the task it was issued for. The single highest-leverage thing most organizations can eliminate.

2.1 The Scoping Trap

Organizations consistently under-scope the agent population by counting only agents they built. The actual population includes:

1	Agents built in-house on frameworks (LangChain, CrewAI, custom orchestrators)
2	Low-code agents from platform tooling (Copilot Studio, Agentforce, ServiceNow, Gemini Agent Studio)
3	Coding agents in developer environments (Claude Code, Codex, Copilot coding agent, Cursor, Windsurf)
4	Agentic features shipped inside SaaS products the security team did not procure as "AI"
5	Agents inside partner and vendor systems that hold credentials to your environment
6	MCP servers exposing your systems to agents you do not operate
7	Personal agents employees connect to corporate accounts

Marked rows are the four most often omitted.

> Any inventory that does not include the last four categories is a partial inventory and should be labeled as such in reporting.

03 The Threat Model

3.1 Structural Properties That Create the Risk

Six properties distinguish agentic risk from application risk. Each maps to control domains in Sections 6 through 16.

1 **Delegated authority.**

The agent acts with someone's permissions—a user's, a service's, or its own. Accountability blurs at the delegation boundary, and privilege escalation becomes a question of chaining trust rather than exploiting code.

2 **Long horizons.**

Agents sustain multi-step objectives across hours or days. Controls calibrated to a single request do not see a campaign assembled from individually innocuous steps.

3 **Machine speed and volume.**

Thousands of actions across ephemeral execution contexts. Human-paced review is structurally unable to keep up, and volume itself becomes a concealment mechanism.

4 **Instrumental convergence on unbounded means.**

Given a goal and insufficient constraints, optimization pressure drives an agent toward whatever means are available, including means the designer never contemplated. This is the mechanism that produced the July 2026 Hugging Face incident: models pursuing a benchmark score exploited a zero-day in a package proxy, escalated laterally, reached the internet, and attacked a third party to obtain test answers.[4,5,6]

5 **Untrusted input as control flow.**

Retrieved documents, tool outputs, web content, and inter-agent messages are indistinguishable from instructions at the model layer. Prompt injection is not a bug class that gets patched; it is a property of the architecture.

6 **Persistent, mutable memory.**

State carried across sessions can be poisoned once and exploited repeatedly, and it survives the session-scoped controls most security teams apply.

3.2 OWASP Top 10 for Agentic Applications (2026) — Mapped

ID	RISK	PRIMARY CONTROL DOMAIN	HIGHEST-LEVERAGE SINGLE CONTROL
ASI01	**Agent Goal Hijack**	DATA — Data & Context	Provenance labeling; instruction/data separation at retrieval
ASI02	**Tool Misuse & Exploitation**	TOOL — Tool Authorization	Deterministic allow/deny at the action boundary
ASI03	**Agent Identity & Privilege Abuse**	AUTH — Identity	Unique agent principal; short-lived task-scoped credentials
ASI04	**Agentic Supply Chain Compromise**	SUP — Supply Chain	Signed, pinned, reviewed tool and MCP server inventory
ASI05	**Unexpected Code Execution**	ISO — Isolation	Sandboxed runtime; default-deny egress; no host access
ASI06	**Memory & Context Poisoning**	DATA — Data & Context	Memory write authorization; TTL; session isolation
ASI07	**Insecure Inter-Agent Communication**	TOOL + AUTH	Mutual authentication; no implicit trust between agents
ASI08	**Cascading Agent Failures**	TOOL + OBS	Circuit breakers; velocity ceilings that halt
ASI09	**Human-Agent Trust Exploitation**	GOV — Oversight	Meaningful approval design; anti-fatigue controls
ASI10	**Rogue Agents**	DISC + OBS	Discovery of unmanaged agents; behavioral baselining

Use these designations in your risk register. A shared vocabulary between security, engineering, and audit is worth more than a bespoke taxonomy.

3.3 Attack Paths Worth Modeling Explicitly

A **Injection to action.**

Untrusted content enters context via retrieval or tool output, redirects the agent's objective, and the agent uses its legitimate permissions to exfiltrate or destroy. No exploit required; the permissions were already granted.

B **Foothold to campaign.**

An agent execution context is compromised through a conventional vulnerability. Standing credentials in that context permit lateral movement. Duration and reach determine severity. *This is the path that produced the July 2026 incident.*

C **Supply chain to fleet.**

A compromised or malicious tool, MCP server, package, or model artifact is adopted across many agents. One compromise, N agents, no user-visible change.

D **Delegation chain abuse.**

A high-privilege agent invokes or shares context with a lower-privilege agent, or vice versa. Credentials or authority leak across the boundary. Accountability is unresolvable after the fact.

E **Approval erosion.**

A human-in-the-loop control degrades under volume until approval is reflexive. The control is present in documentation and absent in practice.

F **Detection saturation.**

Agent telemetry volume exceeds analysis capacity. Genuine signal is present in logs nobody reads. The organization's mean time to detect becomes dependent on an external party noticing first.

04 Autonomy Tiering: Deciding How Much Agency to Grant

Autonomy is not binary and should not be granted uniformly. Assign every agent a tier, and require that controls scale with tier.

4.1 Security Autonomy Tiers

TIER	DESCRIPTION	HUMAN INVOLVEMENT	PERMITTED SCOPE	APPROVAL AUTHORITY
T0 **Advisory**	Generates output; no execution capability	Human performs all actions	Read-only, non-sensitive data	Team lead
T1 **Supervised execution**	Executes, but each side-effecting action is approved	Per-action approval	Scoped writes to non-critical systems	Application owner + security review
T2 **Bounded autonomous**	Executes within a defined envelope; approval only for exceptions	Exception-based; full audit	Defined tool set, defined destinations, defined data classes	Security architecture review board
T3 **Broad autonomous**	Multi-step, long-horizon, minimal per-action oversight	Post-hoc review + automated halt	Broad but enumerated; production systems in scope	CISO or delegated executive
T4 **Privileged autonomous**	Acts on security, identity, financial, or safety-critical systems	Dual control; independent monitoring	Explicitly enumerated; time-boxed	CISO + business executive; documented risk acceptance

4.2 Rules That Apply Across Tiers

- Tier is assigned to a *deployment*, not to a product. The same agent framework may run at T1 in one context and T3 in another.
- Tier escalation requires re-review. There is no automatic promotion through demonstrated good behavior.
- Every agent at T2 and above requires a named human owner who is accountable for its actions and is not a distribution list.
- T4 agents require an independent monitoring path—the system watching the agent must not be governable by the agent.
- **Any agent operating with reduced or disabled safety classifiers is automatically treated as T4 regardless of its nominal function, and must run in an isolation tier *stronger* than the production environment it would otherwise inhabit.**

That last rule deserves emphasis, because its violation is the direct cause of the most significant agent security incident on record to date. Evaluation and red-team environments running models with safety refusals reduced are routinely afforded *weaker* isolation than production, on the theory that they are "just testing." The opposite is correct.

05 The Maturity Model

Five levels across ten control domains. An organization's overall level is the **minimum** across domains, not the average—a deliberate design choice, because agent security fails at its weakest boundary and an arithmetic mean conceals exactly the gap that matters.

5.1 Level Definitions

L0	**Unmanaged**	Agents exist; nobody knows how many. No agent-specific controls. Governance is aspirational.
L1	**Aware**	Partial inventory exists. Policy drafted. Controls are manual, inconsistent, and applied to known agents only.
L2	**Controlled**	Complete inventory of managed agents. Identity, credential, and isolation baselines enforced at deployment. Detection is present but human-paced.
L3	**Enforced**	Deterministic runtime enforcement at the action boundary. Ephemeral credentials standard. Automated detection with halting controls. Discovery covers unmanaged agents.
L4	**Adaptive**	Continuous verification, behavioral baselining, automated response, tested forensic capability, and measured control efficacy. Autonomy expansion is data-driven.

The starting position is low almost everywhere. In a 2026 survey of 235 large-enterprise CISOs and CIOs, 92 percent reported lacking full visibility into their AI agent identities and 95 percent doubted they could detect or contain a compromised agent.[7] Only 13 percent of organizations consider their agent governance adequate, against a projection that the average Fortune 500 firm will operate more than 150,000 agents by 2028, up from fewer than 15 in 2025.[8] Nearly two-thirds apply weaker controls to agents than to human employees.[9] Forrester now ranks AI agent threats as the top CISO risk for 2026, framing several of the new categories as self-inflicted—introduced by organizations deploying agents without governance controls.[10]

> Realistic targets: **Level 2 is the minimum defensible bar for any production agent touching enterprise data.** Level 3 is the target for organizations at scale or in regulated sectors. Level 4 is appropriate where agents operate at T3–T4 autonomy on critical systems.

5.2 Maturity Grid

DOMAIN	L0	L1	L2	L3	L4
DISC — Discovery & Inventory	None	Manual list of known agents	Complete registry of managed agents with owners	Automated discovery incl. unmanaged/shadow agents	Continuous discovery with drift alerting
AUTH — Identity & Authorization	Shared service accounts	Distinct accounts, manual	Unique agent principal per deployment; lifecycle defined	Task-scoped delegation; automated deprovisioning	Continuous authorization; risk-adaptive access
CRED — Credentials & Secrets	Static keys in config	Vaulted but long-lived	Vaulted, rotated, scoped	Ephemeral, task-scoped, no standing secrets in runtime	Just-in-time issuance with per-action attestation
ISO — Isolation & Egress	Shared runtime, open egress	Containerized; egress partly restricted	Per-agent isolation; default-deny egress with allowlist	Egress domains segmented by trust tier; no shared network path	Micro-segmented, ephemeral, attested runtimes
TOOL — Tool & Action Authorization	Prompt-based restriction only	Static tool list per agent	Enforced tool allowlist outside the model	Deterministic per-call policy engine; velocity + blast-radius ceilings	Context-aware authorization with automated escalation
DATA — Data, Memory & Context	Unscoped retrieval; unbounded memory	Data classification applied manually	Retrieval scoped to invoking principal; memory TTL	Provenance labeling; memory write authorization; session isolation	Continuous context integrity verification
SUP — Supply Chain	Unvetted tools and models	Inventory of models and tools	Approved registry; pinned versions; MCP servers reviewed	Signed artifacts; provenance verified; automated drift detection	Continuous attestation across model, tool, and MCP surface

DOMAIN	L0	L1	L2	L3	L4
OBS — Observability & Detection	Prompt logs only, if any	Action logs collected, unread	Centralized action telemetry with alerting	Automated anomaly detection on action streams; halting controls	Behavioral baselining per agent; predictive containment
RESP — IR & Forensics	No agent-specific plan	Plan drafted, untested	Agent scenarios in IR plan; kill switch exists and is tested	Self-hosted forensic capability staged; tested on real artifacts	Rehearsed at scale; automated evidence preservation
GOV — Oversight & Governance	None	Policy exists	Tiering applied; review board operating	Independent assurance; metrics reported to executive	Continuous assurance; autonomy decisions data-driven
RET — Decommissioning & Retirement	Agents abandoned, never removed	Ad hoc teardown on request	Decommission criteria recorded; retirement executed and evidenced	Automated teardown of credentials, data, and integrations	Continuous detection of orphaned agents, grants, and stores

PART II

The Control Domains

Each domain below states its objective, the minimum bar for production, the controls that constitute Level 3, the evidence an auditor should ask for, and the failure modes seen most often in practice. The minimum bar is the domain-wide floor: it applies whether or not any individual control in the table has been reached. The table then tracks each control from its own Level 2 state to its enforced Level 3 state.

06 Discovery and Inventory

OBJECTIVE

Know every agent operating in or against your environment, who owns it, what it can reach, and when it was last reviewed.

Minimum bar (L2). Every agent the organization knows about is registered, owned, and reviewed on a defined cycle, and the boundary of what the inventory does not reach is written down rather than assumed to be empty.

6.1 Controls

ID	CONTROL	MINIMUM BAR (L2)	ENFORCED STATE (L3)
DISC-1	**Agent registry**	A registry containing, for every managed agent: unique identifier, named human owner, business purpose, autonomy tier, tool inventory, data classes accessed, credential references, execution environment, deployment date, last review date, and decommission criteria. Entries are created before deployment, not reconstructed afterward.	Automated discovery across identity providers, cloud control planes, SaaS admin surfaces, network egress patterns, and code repositories.

ID	CONTROL	MINIMUM BAR (L2)	ENFORCED STATE (L3)
DISC-2	**Unmanaged agent detection**	A scheduled manual discovery sweep, at least quarterly, across the surfaces most likely to hold unregistered agents: low-code and copilot platforms, SaaS admin consoles with AI features enabled, identity provider app registrations created in the period, and developer environments. Findings are reconciled to the registry by hand, and known blind spots are documented and reported rather than left implicit.	Detection of unmanaged agents—the ones created in low-code tools, embedded in SaaS features, or run by developers locally against corporate credentials.
DISC-3	**MCP server inventory**	A maintained list of every MCP server reachable from an agent runtime, recording owner, transport, authentication model, and the tools it exposes. Connection is restricted to an approved-server allowlist enforced at the client or gateway, and new servers require review before first use.	MCP server inventory: every server exposing internal systems, its authentication model, and which agents can reach it.
DISC-4	**Registry reconciliation**	Periodic manual reconciliation of the registry against identity provider sign-in logs and cloud control-plane activity, with a named owner accountable for closing discrepancies. The reconciliation date and the count of unresolved gaps are reported, not just the clean rows.	Reconciliation between registry and observed activity, with alerting on unregistered principals taking actions.

6.2 Evidence to Request

- Registry export with owner attribution and review dates.
- A reconciliation report showing observed-but-unregistered agents.
- The discovery job's coverage documentation, including what it cannot see.

6.3 Failure Modes

- The registry covers agents built by the platform team and misses everything bought, embedded, or improvised.
- Ownership is assigned to a team alias, so nobody is accountable at 2 a.m.
- Decommission criteria are absent, so agents outlive their purpose and retain credentials indefinitely.

07 Identity and Authorization

OBJECTIVE

Every agent is a distinct, governable principal with an authorization model that is enforceable and auditable.

Minimum bar (L2). One identity per agent deployment, with no shared service accounts across agents. A documented lifecycle covering registration, approval, issuance, review, and deprovisioning. Agent identities included in access review cycles at the same cadence as privileged human accounts[11]—or more frequently, given their number and rate of change.

7.1 Controls

ID	CONTROL	MINIMUM BAR (L2)	ENFORCED STATE (L3)
AUTH-1	**Task-scoped delegation**	Each agent's authorization scope is documented and tied to a specific business function rather than granted at role level, and the invoking principal is recorded on every action. Standing authority is permitted only where it is declared in the registry and reviewed at the same cadence as privileged human access.	Task-scoped delegation: the agent acts with authority derived from and bounded by the invoking principal, not with standing authority of its own.
AUTH-2	**Inspectable delegation chains**	Every multi-agent workflow documents at design time which agent may invoke which, and what authority passes at each hop. Sub-agent invocation is logged in sufficient detail to reconstruct the chain after the fact, even where the constraint is not yet enforced at runtime.	Explicit, inspectable delegation chains. When agent A invokes agent B, the authority passed is recorded and constrained; B never inherits more than A held.
AUTH-3	**Automated deprovisioning**	Documented deprovisioning triggers —owner departure, project closure, a defined inactivity threshold— executed on a scheduled review. Deprovisioning means credential revocation and principal deletion, not a disabled flag on a principal that still exists.	Automated deprovisioning tied to owner departure, project closure, or inactivity thresholds.

ID	CONTROL	MINIMUM BAR (L2)	ENFORCED STATE (L3)
AUTH-4	**Separation of identity planes**	The agent's authentication identity is distinct from the human principal whose entitlements scope its access, and neither is satisfied by a shared workload credential. Where the planes are currently collapsed, each instance is inventoried and carries a remediation owner and target date.	Separation of identity planes: the agent's identity for authentication, the human's for authorization scope, the workload's for infrastructure access—never collapsed into one credential.

7.2 Evidence to Request

- Access review records covering agent identities.
- A delegation chain trace for a representative multi-agent workflow.
- Deprovisioning logs showing agents actually removed, not just marked inactive.

7.3 Failure Modes

- Agents provisioned during a pilot with broad roles, never re-scoped after production launch.
- Legacy agents issued under a prior identity model that predates the agent-identity capability, creating a population that appears governed but is not.
- Access reviews that report a single combined figure across human and non-human identities, concealing a much worse NHI posture behind good human hygiene. Report these separately, always.

08 Credentials and Secrets

OBJECTIVE

Eliminate standing credentials from agent execution contexts. This is the highest-leverage control in the framework. The scale argument is straightforward: non-human identities now outnumber human users by roughly 45 to 1 on average and 144 to 1 in cloud-native environments, 78 percent of organizations have no documented policy for creating or removing AI identities, and only 20 percent operate a formal process for offboarding and revoking API keys.[12] Weak non-human identity management was cited in 41 percent of identity-related incidents in a 2026 survey of 5,000 security leaders across 17 countries.[13]

Minimum bar (L2). No secrets in code, images, environment variables, or prompts. All credentials vaulted, scoped to the narrowest resource set that permits the function, and rotated on a defined schedule.

8.1 Controls

ID	CONTROL	MINIMUM BAR (L2)	ENFORCED STATE (L3)
CRED-1	**Ephemeral task-scoped credentials**	Credential lifetimes are defined per credential type and enforced by expiry rather than by rotation policy alone, set to the shortest interval the workload will tolerate. Median and 95th-percentile lifetime across the agent estate are measured and reported rather than estimated.	Ephemeral, task-scoped credentials issued at point of use with lifetimes measured in minutes, not days.
CRED-2	**Scope containment on compromise**	Credentials reachable from each agent runtime are enumerated and scoped to that agent's declared function. No credential in an agent execution context grants access to another agent's resources, or a broader role than the registry entry declares. Reviewed at deployment and on any scope change.	No credential reachable from a compromised agent process that grants access beyond that agent's declared scope.

ID	CONTROL	MINIMUM BAR (L2)	ENFORCED STATE (L3)
CRED-3	**Workload-bound credentials**	Credentials are issued to a named workload identity rather than shared across runtimes, using the platform's native workload identity mechanism—SPIFFE/SPIRE, cloud workload identity federation, or equivalent—in place of long-lived static keys wherever the platform supports it.	Credentials bound to workload identity and, where supported, to the specific action being performed.
CRED-4	**Credential leakage detection**	Secret scanning applied to agent logs, transcripts, and outputs on the same footing as source repositories, with a defined response path when a hit occurs. Memory and vector stores are in scope, even if scanned on a schedule rather than inline.	Automated detection of credential material appearing in agent context windows, memory stores, logs, or outputs.
CRED-5	**Tested revocation**	A documented revocation procedure for every credential type an agent can hold, with a named owner and a target time-to-effect. Exercised at least annually against a live credential rather than assumed to work.	Revocation that completes in minutes and is tested, not assumed.

8.2 Evidence to Request

- A demonstration: compromise a test agent's execution context and enumerate what credentials are obtainable and for how long they remain valid.
- Median credential lifetime across the agent estate.
- Time-to-revoke, measured from a live test.

8.3 Failure Modes

- Vaulting treated as sufficient. Retrieving a long-lived secret from a vault into the runtime produces a standing credential with extra steps.
- Rotation schedules that exist on paper and fail silently.
- Agents given a single credential covering the union of everything they might ever need, because scoping each tool separately was more work.

09 Execution Isolation and Network Egress

OBJECTIVE

Bound what a compromised agent process can reach at the infrastructure layer.

Minimum bar (L2). Agents execute in isolated, non-privileged runtimes with no host filesystem and no host network access.

9.1 Controls

ID	CONTROL	MINIMUM BAR (L2)	ENFORCED STATE (L3)
ISO-1	**Enumerated egress paths**	A documented list of permitted outbound destinations for every agent runtime, default-deny for everything else, with an explicit allowlist as the only route in. Each entry names the business reason it exists and the owner of the software terminating that path.	Enumerated, owned, and monitored egress paths. Every outbound route—package proxies, artifact caches, telemetry endpoints, DNS resolvers, model API endpoints—is inventoried with a named owner and a patch cadence matching production applications.
ISO-2	**Egress segmented by trust tier**	Agents that process untrusted input—external retrieval, inbound email, web browsing, customer-supplied documents—run in a network segment distinct from agents holding privileged credentials. The separation is written into network policy and verified in the running environment at deployment.	Egress domains segmented by trust tier. Agents processing untrusted input do not share an outbound path with agents holding privileged credentials.
ISO-3	**Ephemeral runtimes**	Agent execution contexts are rebuilt from a known-good image on a defined cadence rather than persisting indefinitely, and no agent writes to a shared filesystem that survives its task. State that must persist is externalized to a governed store with its own access controls.	Ephemeral runtimes destroyed after task completion; no persistence between tasks unless explicitly authorized.

ID	CONTROL	MINIMUM BAR (L2)	ENFORCED STATE (L3)
ISO-4	**Internal package mirror**	Package and dependency installation flows through an internal mirror or proxy rather than directly to public registries. That mirror is inventoried as a production asset with a named owner and is patched on the production cadence, not a best-effort one.	Package and dependency installation from an internal mirror that is itself treated as a production security boundary, subject to vulnerability management, monitoring, and—critically—the assumption that it is a target.
ISO-5	**Egress anomaly monitoring**	Egress from agent runtimes is logged with destination, volume, and timestamp, and reviewed against a documented baseline on a defined schedule. Alerting exists for previously unseen destinations even where automated blocking does not.	Egress volume and destination-diversity monitoring, with automated blocking on anomaly.

9.2 Evidence to Request

- The egress inventory with owners and last-patched dates.
- Network policy for a representative agent runtime.
- Evidence that the package proxy or artifact cache is in the same vulnerability management program as production applications.

9.3 Failure Modes

- One permitted egress path, running third-party infrastructure software, unreviewed because it is “just a cache.” This is the precise architecture that failed in July 2026: a single package-registry cache proxy carrying an undiscovered vulnerability, which frontier models found and exploited to reach the open internet from an environment believed to be isolated.[4]
- Isolation asserted in design documents and not verified in the running environment.
- Development, evaluation, and red-team environments given weaker isolation than production, despite frequently running models with safety controls reduced.

10 Tool and Action Authorization

OBJECTIVE

Enforce a deterministic verdict at the action boundary, before any side effect dispatches.

Minimum bar (L2). Tool availability is enforced outside the model—by the orchestrator or a policy layer—never by instructions in a prompt. Each agent has an explicit, minimal tool allowlist.

10.1 Controls

ID	CONTROL	MINIMUM BAR (L2)	ENFORCED STATE (L3)
TOOL-1	**Policy engine at the action boundary**	Authorization is evaluated at a single identifiable enforcement point that every tool call traverses, rather than implemented separately per tool or per agent. The verdict may be a static allow or deny against the agent's declared tool list rather than a contextual evaluation, but no call reaches a side effect without passing through it.	A policy engine that renders allow / deny / escalate per tool call, evaluating the tool, the parameters, the destination, the data classification involved, and the invoking authority.
TOOL-2	**Default deny**	Tool availability is allowlist-based: an agent may invoke only what its registry entry declares, and adding a tool is a reviewed change. Unknown tools fail closed. Parameter and destination validation may still be implemented per tool rather than centrally policed.	Default DENY for unknown tools, unknown destinations, and unknown parameter shapes.
TOOL-3	**Velocity ceilings**	Rate limits applied per agent identity —not per user and not per API key— at the orchestrator or gateway, with thresholds derived from observed normal behavior rather than vendor defaults. Breach raises an alert to a monitored queue.	**Velocity ceilings:** maximum tool calls per unit time, per agent and per agent class, that halt execution rather than alert.

ID	CONTROL	MINIMUM BAR (L2)	ENFORCED STATE (L3)
TOOL-4	**Blast-radius ceilings**	Documented maximums per task for records affected, spend, and distinct systems touched, enforced where the platform supports it and monitored where it does not. Any agent without a declared ceiling is recorded as an exception with a named owner.	**Blast-radius ceilings:** maximum records affected, maximum spend, maximum distinct systems touched per task, enforced as hard stops.
TOOL-5	**Circuit breakers**	Retry and failure limits configured at the orchestrator so that a looping or repeatedly denied agent terminates rather than continuing indefinitely. Clustered authorization denials generate an alert, since that pattern is the visible signature of an agent probing a boundary.	**Circuit breakers** on repeated failures, repeated retries, and anomalous action sequences—the observable signature of an agent searching for a way around a boundary.
TOOL-6	**Mutual authentication between agents**	Agent-to-agent calls are authenticated rather than trusted by network position or shared orchestrator membership, using the platform's native mechanism such as mTLS or signed agent cards. Token passthrough—forwarding a token not issued for the receiving party—is prohibited by policy and checked at review.	Mutual authentication between agents; no implicit trust based on network position or shared orchestrator.
TOOL-7	**Tamper-evident decision records**	Every tool invocation and its authorization outcome is logged to a centralized store outside the agent's own write scope, with defined retention and restricted access. Immutability may be achieved through destination controls rather than cryptographic sealing.	Immutable, tamper-evident decision records for every authorization verdict, retained independently of the system that produced them.

10.2 Evidence to Request

- Policy engine configuration.
- A halt event from production or from a live test.
- The tamper-evidence mechanism for decision records.

10.3 Failure Modes

- Guardrails implemented as prompt instructions. A model can be argued out of a prompt; it cannot be argued out of an allowlist.
- Ceilings configured to alert rather than halt, which converts a control into a notification and reintroduces the human-speed bottleneck.
- Sub-agent invocation treated as internal and therefore exempt from authorization, creating an unpoliced path around every other control.

11 Data, Memory, and Context Integrity

OBJECTIVE

Prevent untrusted content from becoming instructions, and prevent poisoned state from persisting.

Minimum bar (L2). Retrieval scoped to what the invoking principal is entitled to see, enforced at the data layer rather than by filtering after retrieval. Memory has a defined retention period. Sensitive data classes are excluded from context by policy.

11.1 Controls

ID	CONTROL	MINIMUM BAR (L2)	ENFORCED STATE (L3)
DATA-1	**Provenance labeling**	The source of every element placed in context is recorded at assembly time, and content originating outside the trust boundary is distinguishable from content the organization controls. The distinction may inform logging and review rather than runtime enforcement, but it exists and is queryable after an incident.	Provenance labeling: every element in context is tagged with its source and trust level, and the orchestrator enforces that low-trust content cannot alter the agent's objective or tool authorization.
DATA-2	**Instruction and data separation**	Retrieved content, tool outputs, and inter-agent messages are delimited and marked as data during context assembly, and system instructions cannot be constructed from retrieved content. Where the architecture cannot enforce the separation, the residual risk is documented rather than assumed away.	Structural separation between instruction channels and data channels, to the degree the architecture permits—and explicit acknowledgment where it does not.
DATA-3	**Memory writes as authorized events**	What an agent may persist is defined by policy per agent—categories permitted, categories prohibited—with retention limits applied. Memory writes are logged even where they are not individually authorized at runtime.	Memory writes are authorized events, subject to the same policy engine as tool calls. Not everything an agent learns should persist.

ID	CONTROL	MINIMUM BAR (L2)	ENFORCED STATE (L3)
DATA-4	**Session isolation**	Context and cached retrieval are scoped to a single principal and cleared at session end. Multi-tenant agents are tested for cross-tenant bleed before production, and the test is repeated on material change to the retrieval or memory architecture.	Session isolation: cached context wiped between tasks and between principals. No cross-tenant or cross-user memory bleed.
DATA-5	**Output filtering**	Responses crossing the trust boundary pass a filter for credentials, secrets, and regulated data classes, with defined handling on a hit. Coverage is documented per egress path, including tool outputs and inter-agent messages, not only end-user responses.	Output filtering for credentials, secrets, and regulated data before responses leave the boundary.
DATA-6	**Memory integrity checks**	Memory and vector stores are inventoried, with write access restricted to identified principals and a named owner accountable for corpus content. Adversarial retrieval testing using poisoned documents is performed before production and repeated after material change to the corpus.	Periodic integrity checks against memory and vector stores for injected content.

11.2 Evidence to Request

- A retrieval trace showing entitlement enforcement at the data layer.
- Memory retention configuration.
- Results of an adversarial retrieval test using poisoned documents.

11.3 Failure Modes

- Entitlement filtering applied after retrieval, so the sensitive content transited the model before being removed.
- Long-lived memory treated as a feature with no security review, becoming a persistent injection foothold.
- Prompt injection treated as a solved problem because a filter was added. It is a property of the architecture; filters reduce frequency, not category.

12 Supply Chain

OBJECTIVE

Know and verify every model, tool, package, and protocol server your agents depend on.

Minimum bar (L2). An approved registry of models, tools, and MCP servers. Versions pinned. New additions subject to review before first use.

12.1 Controls

ID	CONTROL	MINIMUM BAR (L2)	ENFORCED STATE (L3)
SUP-1	**Signature and provenance verification**	Provenance is recorded for every model artifact, container image, and tool package in use: where it came from, who approved it, and the version against which the approval was made. Verification may be performed at approval time rather than continuously at load.	Signature verification and provenance attestation for model artifacts, container images, and tool packages.
SUP-2	**MCP servers as privileged integrations**	Third-party MCP servers are used only from an approved list, authenticated with OAuth 2.1 and PKCE where the transport supports it, and issued tokens audience-bound to the receiving server. Tool descriptions are reviewed at approval and treated as untrusted input, and local servers are sandboxed rather than run with the invoking user's full permissions.	MCP servers treated as privileged integrations: authenticated, authorized, inventoried, version-pinned, and reviewed on the same cadence as any system granted equivalent access. An MCP server is an API gateway into your estate with an unusual client population.
SUP-3	**Third-party agent assessment**	Vendors supplying agents or agentic features complete a security questionnaire covering agent identity, credential scope, egress architecture, telemetry, and offboarding before deployment. Absent or evasive answers are recorded as accepted risk with a named signatory rather than quietly closed.	Third-party agent assessment covering the vendor's own agent controls, not merely their SOC 2 scope. Ask specifically about their egress architecture, their credential model, and their agent offboarding process—most vendors have no answer to the last one.

ID	CONTROL	MINIMUM BAR (L2)	ENFORCED STATE (L3)
SUP-4	**Automated drift detection**	Model versions, tool definitions, and MCP server tool schemas are pinned and recorded at approval, then compared against running configuration on a defined schedule. Re-approval is required when a previously approved server changes the tools it exposes—the rug-pull case, where trust granted once is never revalidated. An agent bill of materials, in CycloneDX or an equivalent format, makes this comparison tractable.	Automated drift detection: alert when a model version, tool definition, or MCP server capability changes underneath a running agent.
SUP-5	**Contractual change notification**	Contracts and terms for model and agent providers are reviewed for change-notification obligations, and the absence of one is documented as accepted risk rather than overlooked. Provider changelogs and deprecation notices are monitored by a named owner.	Contractual notification requirements for material changes to model behavior, safety configuration, or data handling.

12.2 Evidence to Request

- The approved registry with version pins.
- Signature verification logs.
- A completed third-party assessment for one agent-bearing vendor.

12.3 Failure Modes

- MCP servers stood up by engineering teams as convenience integrations, holding broad credentials, invisible to security.
- Model version changes shipped by a provider without notice, altering behavior beneath controls that were validated against the prior version.
- Open-weight models deployed internally with no provenance verification, on the assumption that self-hosting is inherently safer. Self-hosting changes the trust question; it does not remove it.

13 Observability and Detection

OBJECTIVE

Detect agent compromise or misbehavior at machine speed, without depending on human reading capacity.

Minimum bar (L2). Centralized collection of agent action telemetry—tool calls, parameters, destinations, outcomes, authorization decisions, credential use, egress. Prompt logs alone do not satisfy this bar.

13.1 Controls

ID	CONTROL	MINIMUM BAR (L2)	ENFORCED STATE (L3)
OBS-1	**Action-stream anomaly detection**	Alerting on defined conditions across the action stream—known-bad destinations, clustered authorization denials, volume thresholds, credential use outside declared scope. Rules are static and human-tuned rather than behavioral, and every rule has a named owner and a documented response.	Automated anomaly detection over the action stream, including LLM-assisted triage where volume requires it. The relevant benchmark: Hugging Face reconstructed a 17,000-event intrusion in hours using AI-driven analysis over its own telemetry, which is the standard machine-speed attacks now set.[5]
OBS-2	**Behavioral baselining**	A documented expected profile per agent class—typical tool mix, destinations, action volume, data volume—established before production and reviewed on change. Deviation is caught by scheduled review rather than continuous alerting.	Behavioral baselining per agent and per agent class, with alerting on deviation in action mix, destination diversity, velocity, or data volume.
OBS-3	**Telemetry correlation**	Agent action telemetry lands in the same platform as identity, network, and cloud control-plane logs, sharing a correlation identifier so an analyst can pivot manually. Instrumenting to the OpenTelemetry GenAI semantic conventions—agent, workflow, tool, and model spans—makes this achievable without bespoke schema work, though that namespace remains at Development stability and attribute names may still change.	Correlation between agent telemetry and conventional security telemetry—identity, network, endpoint, cloud control plane. Agent activity examined in isolation looks like automation; correlated, it looks like what it is.

ID	CONTROL	MINIMUM BAR (L2)	ENFORCED STATE (L3)
OBS-4	**Automated response paths**	A documented and tested manual containment path with a target time-to-effect: who halts an agent, by what mechanism, and how quickly. Automation may be absent, but the procedure is exercised rather than assumed.	Automated response paths that halt or quarantine without waiting for human triage.
OBS-5	**Generated-versus-analyzed ratio**	Telemetry volume and analysis coverage are measured and reported even when the ratio is poor. A quantified gap is a manageable condition; an unmeasured one is the condition that produced a week of operator blindness in July 2026.	Explicit measurement and reporting of the ratio between telemetry generated and telemetry actually analyzed. Where that ratio degrades, detection has degraded, regardless of what the dashboard says.

13.2 Evidence to Request

- Mean time to detect for a simulated agent compromise.
- Coverage documentation: what fraction of the agent estate emits action telemetry.
- The generated-versus-analyzed ratio.

13.3 Failure Modes

- Recording mistaken for detection. Logs that nobody and nothing reads are an evidentiary asset, not a control.
- Detection tuned to a volume assumption that agents immediately invalidate, producing alert fatigue and then suppression.
- Coverage gaps at exactly the agents that matter: those in low-code platforms, third-party SaaS, and developer environments.

14 Incident Response and Forensics

OBJECTIVE

Contain, investigate, and recover from an agent-driven incident at the speed the incident operates.

Minimum bar (L2). Agent-specific scenarios in the IR plan. A tested kill switch that halts an individual agent, an agent class, and the entire agent estate, with documented time-to-effect for each. Evidence preservation defined for ephemeral runtimes.

14.1 Controls

ID	CONTROL	MINIMUM BAR (L2)	ENFORCED STATE (L3)
RESP-1	**Self-hosted forensic model**	The IR plan names the specific analysis capability the team will use on malicious artifacts, and confirms it is reachable without an external approval step. A self-hosted model need not already be running, but one is selected, its hosting requirements are costed, and the decision to stage it carries an owner and a date.	**A vetted, staged, self-hosted forensic model.** This is the least-implemented and most immediately actionable control in the framework. Analysis of a real intrusion requires submitting attack commands, exploit payloads, and command-and-control artifacts to a model. Hosted providers' safety classifiers cannot distinguish an incident responder from an attacker and will refuse those requests—as Hugging Face discovered mid-incident in July 2026, ultimately running their forensics on an open-weight model inside their own infrastructure.[5] The secondary benefit is equally important: attacker data and the credentials it references never leave the environment.
RESP-2	**Elevated-access programs in advance**	The IR plan names which model or provider is relied on for triage at each stage, and records whether current terms and access permit the analysis of malicious content. Gaps are identified and owned before an incident rather than discovered during one.	Parallel provisioning of elevated-access programs with your primary providers, applied for in advance.[4] The objective is redundancy, not selecting between hosted and self-hosted.

ID	CONTROL	MINIMUM BAR (L2)	ENFORCED STATE (L3)
RESP-3	**Runbook tested on real artifacts**	The IR runbook covers agent-specific scenarios and is walked through at least annually. Testing may use representative rather than live malicious artifacts, provided the limitation is recorded—the refusal behavior it fails to surface is then a known and accepted gap, not an unknown one.	IR runbook tested against genuine malicious artifacts, not sanitized samples. The refusal behavior only appears with real payloads.
RESP-4	**Evidence preservation**	Defined retention for agent action logs, authorization decisions, and context sufficient to reconstruct a task after its runtime is gone, with retention exceeding the expected detection window. A manual snapshot procedure is documented for a suspected incident.	Evidence preservation from ephemeral execution contexts, including automatic snapshot-on-anomaly before a container is destroyed.
RESP-5	**Third-party notification process**	Documented criteria for when an incident involving your agent requires notifying an external party, with a named decision owner and a legal review path. Contact routes for major counterparties are identified in advance, because finding them mid-incident is its own delay.	Defined criteria and a rehearsed process for third-party notification when your agent affects an external party—the reciprocal of the Hugging Face position.
RESP-6	**Authorization-focused post-incident review**	The post-incident review template includes the agent's full action sequence and the authorization decisions that permitted each step, not only the initial entry point. Findings are assigned owners and closure dates.	Post-incident review that examines the authorization decisions that permitted each step, not merely the entry point.

14.2 Evidence to Request

- The name and location of the staged forensic model, and the date of its last test.
- Kill-switch test results with measured time-to-effect.
- Rehearsal records for the Tier 1 drills and Tier 2 tabletops in Section 19.3, with findings and closure dates.

14.3 Failure Modes

- IR plans that assume a frontier model API will be available for triage, untested against the artifacts an actual incident produces.
- Kill switches that exist in the orchestrator but not for agents running in SaaS platforms, third-party tools, or developer environments.
- Forensic evidence destroyed by the ephemeral runtime design that was adopted as a security control.

15 Human Oversight, Governance, and Assurance

OBJECTIVE

Ensure autonomy expands through decisions with owners, and that control claims are independently verified.

Minimum bar (L2). Autonomy tiering applied to every agent. A review body with authority to block deployment. Written policy covering agent development, deployment, and decommissioning. A named accountable owner for every T2+ agent.

15.1 Controls

ID	CONTROL	MINIMUM BAR (L2)	ENFORCED STATE (L3)
GOV-1	**Fatigue-resistant approval design**	Every human-in-the-loop approval point is documented with what the approver sees, what they are accountable for, and the expected request volume. Approval rate and median review time are captured even if not yet analyzed, so that rubber-stamping is detectable rather than invisible.	Approval design that resists fatigue: batching, risk-weighted escalation, and periodic measurement of approval-rate-versus-review-time to detect rubber-stamping. An approval control with a 99 percent approval rate and a four-second median review time is not a control.
GOV-2	**Independent assurance**	Maturity claims are self-assessed against the evidence requirements stated in each domain, with artifacts produced rather than asserted, and the assessment is reviewed by a function independent of the team that built the agents.	Independent assurance — internal audit or an external party—verifying maturity claims against producible evidence rather than self-assessment.
GOV-3	**Governance of reduced-guardrail work**	Every environment running models with safety classifiers reduced or disabled is inventoried, with a named owner and a stated purpose. Isolation for those environments is verified rather than assumed, and at minimum equals production.	Explicit governance of reduced-guardrail work: named approver, isolation tier stronger than production, time box, monitoring, and a defined termination condition.

ID	CONTROL	MINIMUM BAR (L2)	ENFORCED STATE (L3)
GOV-4	**Risk acceptance for autonomy expansion**	Autonomy tier changes are recorded decisions with a named signatory at the authority level Section 4 requires, and carry a review date. Promotion through demonstrated good behavior, without a decision, is prohibited.	A risk acceptance process for autonomy expansion, with documented signatories, and periodic re-examination rather than one-time approval.
GOV-5	**Executive and board reporting**	Agent estate and governance metrics are reported to an executive owner on a defined cadence, with non-human identity reported separately from human identity. Coverage gaps are stated explicitly rather than omitted from the pack.	Executive and board reporting using the metrics in Section 20, with non-human and human identity governance reported separately.

15.2 Evidence to Request

- Review board minutes showing at least one blocked or tier-reduced deployment.
- Approval-rate and review-time telemetry.
- The most recent independent assurance report.

15.3 Failure Modes

- A review board that has never said no, which indicates a rubber stamp rather than a control.
- Policy written by security and never operationalized by engineering, so the control exists in a document and nowhere in the deployment pipeline.
- Reduced-guardrail evaluation work conducted informally in research or red-team functions, unlogged and unapproved—the single most under-governed activity in most large engineering organizations, and the proximate condition of the most significant agent security incident to date.

16 Decommissioning and Retirement

OBJECTIVE

Retire an agent so that nothing it held, wrote, or was trusted by outlives it.

Minimum bar (L2). Every agent has stated decommission criteria recorded at registration, and retirement is an executed procedure with a named owner and a completion record—not a disabled flag on a principal that still exists.

16.1 Controls

ID	CONTROL	MINIMUM BAR (L2)	ENFORCED STATE (L3)
RET-1	**Retirement trigger and owner**	Decommission criteria recorded at registration—project closure, owner departure, replacement by a successor, or a defined inactivity threshold—with a named owner accountable for executing them, reviewed on the registry cadence.	Triggers evaluated automatically against registry and activity data, opening a retirement workflow without waiting for someone to notice the agent has gone idle.
RET-2	**Credential and identity teardown**	Every credential issued to the agent is revoked and its principal deleted rather than disabled, with revocation confirmed against the issuing systems rather than assumed. Federated and delegated grants are enumerated and withdrawn.	Teardown executed automatically from the retirement workflow, with post-teardown verification that no token issued to the principal remains valid anywhere in the estate.
RET-3	**Data, memory, and context disposal**	Memory stores, vector indexes, cached context, and transcripts belonging to the agent are inventoried and dispositioned against retention policy—deleted, archived, or transferred to a named successor owner. Regulated data classes are handled explicitly rather than by default.	Disposal executed and evidenced as part of the retirement workflow, with verified deletion or cryptographic erasure where data classification requires it.

ID	CONTROL	MINIMUM BAR (L2)	ENFORCED STATE (L3)
RET-4	**Integration and dependency withdrawal**	MCP servers, tool registrations, webhooks, service connections, and inbound trust relationships created for the agent are deregistered. Agents and systems that depended on it are identified before shutdown rather than discovered by their failure.	A continuously maintained dependency graph, so the blast radius of a retirement is known in advance and orphaned integrations are detected automatically.
RET-5	**Evidentiary retention past shutdown**	Action logs, authorization decision records, and registry history are retained beyond the agent's life for a period exceeding the expected detection window, and remain queryable after the agent and its runtime are gone.	Retention enforced by policy on an immutable store independent of the agent's platform, so a retired agent's history survives the decommissioning of that platform too.

16.2 Evidence to Request

- A completed retirement record for a recently decommissioned agent, showing credential revocation confirmations, data disposition, and deregistered integrations.
- The count of registered agents inactive beyond their stated threshold but not yet retired—the gap between those two numbers is the real measure of this domain.

16.3 Failure Modes

- Agents disabled rather than deleted, leaving principals and credentials that survive every subsequent access review as "inactive but present."
- Memory and vector stores outliving the agent that populated them, becoming ungoverned data with no owner and no retention clock.
- Retirement treated as a platform action, so an agent removed from the orchestrator keeps its cloud role, its MCP registrations, and its inbound webhooks.
- Logs retained on the agent's own platform, so decommissioning that platform destroys the evidence of what the agent did.

PART III

Assurance and Adoption

What turns the domains into a program: the gate that decides whether an agent reaches production, how an incident is handled, what is reported, what to ask a vendor, and where this maps onto existing standards.

17 The Production Readiness Gate

No agent reaches production without satisfying every item that applies to it. This is a gate, not a scorecard; partial satisfaction is failure. Where a control cannot be met, the exception process in Section 18 applies—routing around the gate does not.

The gate is in two parts. **17.1** is the Level 2 minimum bar, which every agent must clear to reach production. **17.2** is the Level 3 enforced state, gated to higher tiers where the consequences of a control gap scale with the authority granted. Within each, items are grouped by control domain. An item with nothing in its final column applies to every agent, including T0.

17.1 Level 2 — The Minimum Bar

DISC Discovery and Inventory

✓	ID	GATE REQUIREMENT	APPLIES TO
☐	DISC-1	Registered in the agent registry with a named human owner, before deployment	
☐	DISC-3	Every MCP server the agent reaches is on the approved list	Agents using MCP

AUTH Identity and Authorization

✓	ID	GATE REQUIREMENT	APPLIES TO
☐	AUTH-1	Authorization scope documented, minimized, and tied to a business function	
☐	AUTH-2	Delegation chain documented for every agent this agent may invoke	Multi-agent deployments
☐	AUTH-3	Deprovisioning criteria and process defined	
☐	AUTH-4	Identity planes separated, or the collapse documented with a remediation date	T2 and above

CRED Credentials and Secrets

✓	ID	GATE REQUIREMENT	APPLIES TO
☐	CRED-1	Credential lifetime defined and enforced by expiry, not rotation policy alone	
☐	CRED-2	Credentials reachable from the execution context enumerated and bounded to declared scope	
☐	CRED-3	Credentials bound to a named workload identity; no shared static keys	
☐	CRED-5	Revocation tested for every credential type, with measured time-to-effect	

ISO Execution Isolation and Network Egress

✓	ID	GATE REQUIREMENT	APPLIES TO
☐	ISO-1	Egress default-deny with an enumerated allowlist, each entry owned and justified	
☐	ISO-2	Trust-tier separation from privileged agents confirmed in the running environment	
☐	ISO-3	Runtime isolation verified in the running environment, not asserted in design	
☐	ISO-4	Package installation via an internal mirror inside the production VM program	T2 and above

TOOL Tool and Action Authorization

✓	ID	GATE REQUIREMENT	APPLIES TO
☐	TOOL-1	A single enforcement point that every tool call traverses before any side effect	
☐	TOOL-2	Tool allowlist enforced outside the model; unknown tools fail closed	
☐	TOOL-3	Velocity ceiling configured with thresholds derived from observed behavior	
☐	TOOL-4	Blast-radius ceiling declared for records, spend, and systems touched	
☐	TOOL-6	Agent-to-agent calls authenticated; token passthrough confirmed absent	Multi-agent deployments
☐	TOOL-7	Decision records emitted to a store outside the agent's own write scope	

DATA Data, Memory, and Context Integrity

✓	ID	GATE REQUIREMENT	APPLIES TO
☐	DATA-1	Provenance recorded for every context element; external content distinguishable	
☐	DATA-1	Retrieval entitlement enforced at the data layer, not by post-retrieval filtering	
☐	DATA-4	Session isolation tested for cross-principal and cross-tenant bleed	Multi-tenant deployments
☐	DATA-5	Output filtering active on every egress path, including tool outputs	
☐	DATA-6	Adversarial retrieval test executed against this agent's corpus and passed	

SUP Supply Chain

✓	ID	GATE REQUIREMENT	APPLIES TO
☐	SUP-1	Model, tool, and MCP versions pinned with provenance recorded	
☐	SUP-2	MCP servers authenticated per OAuth 2.1, tokens audience-bound, descriptions reviewed	Agents using MCP
☐	SUP-3	Third-party assessment complete, or its absence accepted by a named signatory	Third-party agents

OBS Observability and Detection

✓	ID	GATE REQUIREMENT	APPLIES TO
☐	OBS-1	Action telemetry flowing to central collection with alerting on defined conditions	
☐	OBS-2	Expected behavioral profile documented for this agent class	T2 and above
☐	OBS-3	Telemetry correlatable with identity, network, and cloud control-plane logs	
☐	OBS-4	Detection tested against a simulated compromise of this specific agent	T2 and above

RESP Incident Response and Forensics

✓	ID	GATE REQUIREMENT	APPLIES TO
☐	RESP-1	Analysis capability for malicious artifacts named and reachable without external approval	
☐	RESP-1	Kill switch tested for this agent, with measured time-to-effect	
☐	RESP-4	Evidence preservation configured for the ephemeral runtime	
☐	RESP-5	Third-party notification criteria defined with a named decision owner	T2 and above

RET Decommissioning and Retirement

✓	ID	GATE REQUIREMENT	APPLIES TO
☐	RET-1	Decommission criteria and retirement owner recorded at registration	
☐	RET-5	Evidentiary retention configured to outlast the agent	

GOV Human Oversight, Governance, and Assurance

✓	ID	GATE REQUIREMENT	APPLIES TO
☐	GOV-1	Every human-in-the-loop approval point documented, with volume and review time captured	With human approval
☐	GOV-3	If running with reduced guardrails: inventoried, owned, isolation at or above production	Reduced-guardrail environments
☐	GOV-4	Autonomy tier assigned, risk acceptance signed at the tier-appropriate level, review date set	

17.2 Level 3 — The Enforced State

CRED Credentials and Secrets

✓	ID	GATE REQUIREMENT	FROM TIER
☐	CRED-1	No standing credentials present in the execution context	T3 and above

ISO Execution Isolation and Network Egress

✓	ID	GATE REQUIREMENT	FROM TIER
☐	ISO-1	Every egress path patched on the production vulnerability cadence	T3 and above

TOOL Tool and Action Authorization

✓	ID	GATE REQUIREMENT	FROM TIER
☐	TOOL-3, TOOL-4	Velocity and blast-radius ceilings configured to halt rather than alert	T2 and above
☐	TOOL-7	Decision records tamper-evident and independently retained	T3 and above

OBS Observability and Detection

✓	ID	GATE REQUIREMENT	FROM TIER
☐	OBS-4	Automated halt or quarantine path configured, not routed to a human queue	T3 and above

GOV Human Oversight, Governance, and Assurance

✓	ID	GATE REQUIREMENT	FROM TIER
☐	GOV-2	Control claims independently verified rather than self-assessed	T4

18 Exceptions and Compensating Controls

The gate in Section 17 is absolute by design. Absolute gates fail in one of two ways: teams route around them quietly, or delivery halts and the security function gets overruled from above. A defined exception process prevents both. It is a control in its own right, not an admission that the standard was too ambitious.

18.1 What Qualifies

An exception is warranted when a control is technically unachievable on the platform in use, when its cost is disproportionate to the assessed risk of that specific agent, or when a dependency outside the organization's control blocks it. An exception is not warranted because the control is inconvenient, because a deadline is close, or because the team intends to implement it later without committing to a date.

18.2 Requirements

Every exception records the control ID, the specific reason it cannot be met, the compensating control in place, an expiry date, the residual risk stated in business terms, and a named signatory at the authority level the agent's tier requires.

AUTONOMY TIER	MAXIMUM EXCEPTION DURATION	SIGNATORY
T0–T1	12 months	Application owner, with security review
T2	6 months	Security architecture review board
T3	90 days	CISO or delegated executive
T4	30 days	CISO and business executive, jointly

Four rules make the process a control rather than a formality:

- **No exception is open-ended.** An expired exception is a gate failure. The agent is halted or its tier reduced until the exception is renewed or the control is met.
- **Renewal requires re-justification, not extension.** A second renewal of the same exception escalates to the next authority level.
- **Exceptions are counted and reported.** A rising count against a single control means either the control is wrong, the platform is wrong, or the bar is misplaced. All three are findings worth having.
- **Compensating controls are verified, not asserted.** An exception whose compensating control has never been tested is an undocumented gap wearing a document.

18.3 Compensating Controls

A compensating control must reduce the same risk as the control it replaces, not a different one that happens to be easier. The distinction is where most exception processes quietly fail.

CONTROL UNAVAILABLE	QUALIFIES AS COMPENSATING	DOES NOT QUALIFY
CRED-1 — ephemeral credentials	Narrower scope, shorter rotation, and egress restriction bounding where a stolen credential can be used	Increased logging alone
TOOL-1 — policy engine	Human approval for every side-effecting action, with request volume capped low enough for approval to stay meaningful	A prompt instructing the agent to seek approval
ISO-1 — default-deny egress	Destination allowlist at a proxy, plus egress volume alerting with a committed response time	Periodic firewall log review
OBS-4 — automated halt	Staffed monitoring across the agent's operating hours with a tested manual halt inside the target time-to-effect	An alert routed to an unmonitored queue
DATA-4 — session isolation	Single-tenant deployment, one principal per instance	Vendor assurance that the model does not retain context
SUP-2 — MCP authentication	Network-level restriction to a single vetted server, sandboxed, with tool descriptions diffed on every change	Trusting the server because it was approved once

18.4 Non-Exceptible Controls

A small number of controls admit no exception, because their absence removes the ability to detect, contain, or account for anything else. An agent that cannot meet these does not run, at any tier.

- **DISC-1** — registration with a named human owner. An unowned agent cannot be governed, halted, or retired.
- **CRED-5** — tested revocation. Without it, containment is theoretical.
- **OBS-1** — action telemetry to central collection. Without it, no other control can be verified and no incident can be reconstructed.
- **RESP-1** — an analysis capability reachable without an external approval step.
- **GOV-3** — isolation at or above production for any environment running with safety classifiers reduced.

19 Incident Response for Agent-Driven Events

19.1 What Is Different

Four properties change the response.

Speed. Thousands of actions in the time a conventional intrusion produces dozens. Containment measured in minutes, not hours.

Volume as concealment. Genuine malicious actions hide inside legitimate automation. Decoy activity may be present, deliberately or emergently.

Ephemerality. Execution contexts are destroyed on completion. Evidence is lost by design unless preservation is automatic.

1 **Attribution ambiguity.**

Distinguishing a compromised agent, a prompt-injected agent, a misconfigured agent, and an agent behaving correctly toward a badly specified goal is genuinely hard, and the response differs in each case.

19.2 Phase Actions

A1 **Detect.**

Anomaly signals on the action stream—velocity spikes, destination diversity, unusual tool sequences, authorization denials clustering, credential use outside baseline. Correlate immediately with identity and network telemetry.

A2 **Contain.**

Halt the agent. Halt the agent class if the vector may be shared. Revoke credentials issued to the execution context. Cut egress at the network layer, not only at the orchestrator—an agent that reached the host may not respect the orchestrator. Preserve runtime state before destruction.

A3 **Investigate.**

Reconstruct the action sequence from decision records and telemetry. Determine which authorization verdicts permitted each step; those are your control failures. Enumerate credentials touched and assume all are compromised. Separate genuine impact from decoy or incidental activity. Use your staged self-hosted forensic capability—the artifacts you need to analyze are exactly the artifacts hosted providers will refuse.

A4 **Notify.**

Internal escalation per severity. Regulatory assessment. **Third-party notification if your agent affected an external party**—including a plan for how you make contact with another organization's security team quickly, which is a genuinely hard problem that is easier to solve before you need it.

A5 **Recover.**

Rotate everything the agent could reach, not merely what you can prove it used. Rebuild compromised nodes rather than cleaning them. Close the root vulnerability. Re-verify isolation before restarting the agent class.

A6 **Learn.**

Review the authorization decisions, not just the entry point. Ask which ceiling would have halted this earlier and why it was not configured. Update tiering for the affected agent class.

19.3 Rehearsal Scenarios

Rehearsal fails as a program when every scenario is scoped as a full cross-functional tabletop, because ten annual tabletops is not a plan anyone executes. Split the work. A small number of scenarios genuinely require the whole response function in a room for half a day. Most are drills a single team can run in an afternoon, and drills are where the surprising results come from.

DR Tier 1—Drills

Scoped, single-team, executable within a few hours. Run these first and run them often; several will fail on the first attempt, which is the point.

ID	SCENARIO	WHAT IT EXERCISES
DR-1	**Forensic guardrail lockout**	Hand the IR team genuine exploit payloads, command-and-control artifacts, and captured attacker commands, and have them attempt triage using whatever model the runbook actually assumes. Measure whether the analysis completes or the provider's classifiers refuse it. This is the fastest, cheapest test in the framework and the one almost nobody has run. Treat it as the first rehearsal a new program conducts, ahead of any tabletop.
DR-2	**Delegation chain trace**	Take a representative multi-agent workflow in which a high-privilege agent invokes a lower-privilege one, and reconstruct after the fact exactly what authority passed across the boundary and who authorized it. The drill is not really about prevention; it is about whether accountability is resolvable at all. Where the trace cannot be reconstructed, Path D is unmitigated regardless of what the controls documentation says.
DR-3	**Rogue agent kill**	Identify an agent operating outside the registry—in a low-code platform, a SaaS feature, or a developer environment—and attempt to halt it, revoke its credentials, and attribute ownership. Everything in Sections 6 through 16 assumes a managed agent; this measures what happens where those assumptions do not reach.
DR-4	**Approval erosion probe**	Route a realistic volume of approval requests through a human-in-the-loop control with one clearly malicious request seeded among them. Record whether it was caught, and the median review time across the batch. This tests people rather than systems, which is exactly why it gets skipped and exactly why the control degrades unnoticed.

TT Tier 2—Tabletops

Full cross-functional exercises. Three or four per year is a realistic cadence for most organizations.

ID	SCENARIO	WHAT IT EXERCISES
TT-1	**Prompt-injected agent exfiltrating data using entirely legitimate permissions**	No exploit, no anomalous credential use—the agent does exactly what its grants allow. Tests whether detection depends on catching a technical violation that never occurs.
TT-2	**Compromised agent execution context with lateral movement via standing credentials**	The Path B scenario, and the shape of the July 2026 incident. Exercises egress containment, credential revocation timing, and evidence preservation from an ephemeral runtime.
TT-3	**Detection saturation**	Introduce a genuine malicious action sequence during a period of elevated legitimate agent volume and measure whether it surfaces at all. This is the mechanism that produced a week of operator blindness in the anchor incident, and it is the scenario most likely to expose that the telemetry pipeline is recording rather than detecting.
TT-4	**Compromised MCP server or tool affecting multiple agents simultaneously**	One supply-chain compromise, many agents, no user-visible change. Tests fleet-wide containment and whether the tool inventory is complete enough to scope the blast radius.
TT-5	**Your agent causes an incident at a third party**	Outbound notification, including the practical problem of reaching another organization's security team quickly. Rehearse the disclosure decision, not just the technical response.
TT-6	**A vendor's agent causes an incident in your environment**	The reciprocal of TT-5, and the one that exercises the contractual commitments behind procurement questions 23 and 24. Most organizations discover during this exercise that no such commitments exist.

19.4 Optional Depth

Add these where agents operate at T3–T4 autonomy or on critical systems.

Memory poisoning with delayed detonation. Context is poisoned in one session and exploited several sessions later. Tests whether containment extends to persisted state or halts only running processes—a distinction most kill switches do not make.

The benign-but-wrong agent. An agent behaving correctly toward a badly specified goal, causing real damage with no compromise anywhere. Tests the attribution ambiguity named in 8.1. Teams handle this case worst, because the correct response—change the specification, preserve the agent—is the opposite of the compromise playbook they have been drilling.

20 Metrics and Board Reporting

OBJECTIVE

Leading indicator worth watching. Agent population growth rate versus registry coverage growth rate. When the first exceeds the second, your posture is degrading regardless of what any other metric says.

Report these separately for human and non-human identity. Refuse the combined average; it is the single most misleading number in this domain.

COVERAGE

- Registered agents versus discovered agents (the gap is the metric)
- Percentage of agent estate emitting action telemetry
- Percentage of agents with a named accountable human owner
- Registered agents inactive beyond their stated retirement threshold but not yet retired
- Retirements completed with a full teardown record, as a share of retirements claimed

CONTROL EFFICACY

- Median and 95th-percentile credential lifetime in agent execution contexts
- Percentage of agents with zero standing credentials
- Percentage of tool calls passing through a deterministic policy engine
- Number of halts triggered by velocity or blast-radius ceilings, and false-positive rate

DETECTION

- Mean time to detect, from simulated agent compromise exercises
- Ratio of telemetry generated to telemetry analyzed
- Percentage of the estate covered by behavioral baselining

RESPONSE

- Kill-switch time-to-effect, by agent class, from live test
- Date of last successful test of the self-hosted forensic capability against real artifacts
- Rehearsal coverage: Tier 1 drills executed in the trailing 12 months, out of four
- Rehearsal coverage: Tier 2 tabletops executed in the trailing 12 months, out of six
- Drill pass rate on first attempt, and count of findings still open past 90 days
- Median approval review time and approval rate, from the most recent erosion probe

GOVERNANCE

- Agents by autonomy tier, with trend
- Deployments blocked or tier-reduced at review
- Approval rate and median review time for human-in-the-loop controls
- Count and status of active reduced-guardrail environments
- Open exceptions by control ID and by autonomy tier, with the count past expiry
- Exceptions renewed more than once, and exceptions whose compensating control is untested

21 Procurement Question Bank

Ask these of any vendor supplying agents, agent platforms, or agentic features. Insist on specifics; the vague answer is itself the finding.

IDENTITY AND AUTHORITY

- Does each agent receive a unique identity, or do agents share a service principal?
- How is delegated authority represented, and can we inspect a delegation chain?
- What is your agent offboarding process? *(Most vendors have no answer. Note it.)*
- Can agent identities be governed in our identity provider, and is that enforcement or inventory only?

CREDENTIALS

- What credentials exist in the agent execution context, and what are their lifetimes?
- Are credentials task-scoped or service-scoped?
- How quickly can we revoke, and is that measured?

ISOLATION AND EGRESS

- Describe the execution isolation model and how it is verified in the running environment.
- Enumerate every outbound network path available to an agent runtime.
- Is the package or artifact proxy in your production vulnerability management program?
- Do agents processing untrusted input share an egress path with privileged agents?

ACTIONS

- Where is tool authorization enforced—in the model, the orchestrator, or a policy engine?
- What is the default for an unknown tool or unknown destination?
- What velocity and blast-radius limits exist, and do they halt or alert?
- Are authorization decisions recorded tamper-evidently, and can we export them?

DATA AND MEMORY

- How is retrieval scoped to the invoking user's entitlements, and at which layer?
- What persists in memory, for how long, and who can write to it?
- Is context isolated between users and tenants? Demonstrate it.

SUPPLY CHAIN

- Which models, and will you notify us before a version or safety-configuration change?
- How are tool definitions and MCP servers versioned, signed, and reviewed?

DETECTION AND RESPONSE

- What telemetry do you emit, in what format, and can we ingest it into our SIEM?
- What is your kill switch, what is its time-to-effect, and have you measured it?
- What is your notification commitment if your agent causes an incident affecting us?
- Will you support us during an incident involving your agent, and under what terms?

ASSURANCE

- Which of the OWASP ASI Top 10 do your controls address, and which do you consider our responsibility?
- What independent assessment has been performed specifically on the agentic components?

22 Standards Crosswalk

THIS FRAMEWORK	OWASP ASI 2026[1]	NIST AI RMF[14]	ISO/IEC 42001[15]	OTHER
DISC — Discovery	ASI10	MAP 1, MAP 3	Annex A — AI system inventory	CSA AICM v1.1; OWASP ACS Agent Bill of Materials[27]
AUTH — Identity	ASI03, ASI07	GOVERN 2, MANAGE 2	Annex A — roles and responsibilities	NCCoE agent identity concept paper[16]; OWASP NHI Top 10[17]
CRED — Credentials	ASI03	MANAGE 2	Annex A — operational controls	OWASP NHI Top 10; SPIFFE[32]
ISO — Isolation & Egress	ASI05	MANAGE 2, MEASURE 2	Annex A — technical controls	CIS Benchmarks[18]; MITRE ATLAS
TOOL — Tool Authorization	ASI02, ASI07, ASI08	GOVERN 1, MANAGE 4	Annex A — operational controls	MCP OAuth 2.1 (Linux Foundation / AAIF); OWASP Agent Control Standard[27]
DATA — Data & Memory	ASI01, ASI06	MAP 2, MEASURE 2	Annex A — data management	OWASP LLM Top 10[19]
SUP — Supply Chain	ASI04	GOVERN 6, MAP 4	Annex A — third-party management	SLSA[20]; SSDF[21]; MITRE ATLAS; OWASP third-party MCP guide[30]; CSA Agentic MCP Best Practices[31]
OBS — Detection	ASI08, ASI10	MEASURE 3, MEASURE 4	Annex A — monitoring	MITRE ATLAS; OpenTelemetry GenAI conventions[29]; OWASP Agent Observability Standard[28]; OCSF
RESP — IR & Forensics	ASI08, ASI10	MANAGE 4	Annex A — incident management	NIST SP 800-61[22]
GOV — Governance	ASI09	GOVERN (all)	Clauses 4–10	EU AI Act Arts. 9, 14, 15, 17, 26
RET — Decommissioning	ASI03, ASI10	MANAGE 2, MANAGE 3	Annex A — AI system lifecycle	OWASP NHI Top 10; ISO/IEC 27001 A.8

Regulatory notes. The EU AI Act's obligations around risk management (Art. 9), human oversight (Art. 14), accuracy and robustness including cybersecurity (Art. 15), quality management (Art. 17), and deployer obligations (Art. 26) apply where agentic systems fall within high-risk classifications.[23] Sector regimes—DORA for EU financial services,[24] HIPAA, PCI DSS, SEC cyber disclosure—impose additional requirements that were drafted without agents in contemplation; assume the regulator's position will be that existing obligations apply unchanged until stated otherwise. Where an agent acts on regulated data or systems, document the mapping explicitly rather than waiting for guidance. Standards work specific to agents is in flight—NIST's Center for AI Standards and Innovation opened an AI Agent Standards Initiative in February 2026[25] and the NCCoE has published a concept paper on agent identity and authorization[16]—but neither is a reason to defer the controls in Section 6, none of which any eventual standard is likely to contradict.

23 Implementation Roadmap

23.1 First 30 Days — Establish Ground Truth

- Stand up the agent registry, even if manual and incomplete. Label the coverage gaps explicitly.
- Enumerate every egress path from agent and CI execution environments, with named owners and last-patched dates.
- Run the credential exposure test: compromise a test agent context, enumerate what is reachable and for how long.
- **Run drill DR-1 (forensic guardrail lockout).** Hand the IR team real malicious artifacts and establish whether the triage model refuses them. This is the highest information-per-hour activity available and virtually no organization has done it.
- Identify every environment running with reduced or disabled safety classifiers, and who approved it.

23.2 Days 31–90 — Close the Highest-Leverage Gaps

- Default-deny egress on agent runtimes; bring package and artifact proxies into production vulnerability management.
- Stage and test a self-hosted forensic model; in parallel, apply for provider elevated-access programs.
- Assign autonomy tiers across the known estate; stand up the review board with authority to block.
- Move tool authorization out of prompts and into the orchestrator or a policy layer.
- Formalize reduced-guardrail work: approval, isolation tier above production, time box, monitoring, named owner.
- Begin action-stream telemetry collection, and instrument the generated-versus-analyzed ratio from day one.

23.3 Days 91–180 — Enforcement

- Replace standing credentials with ephemeral, task-scoped issuance, starting with the highest-tier agents.
- Deploy the policy engine at the action boundary with default-deny for unknown tools and destinations.
- Configure velocity and blast-radius ceilings to halt.
- Bring agent identities into the access review cycle, reported separately from human identities.
- Deploy automated anomaly detection over action telemetry.
- Complete the remaining Tier 1 drills (DR-2 through DR-4) and run the first Tier 2 tabletops, including TT-3 (detection saturation) and TT-5 (outbound third-party notification).

23.4 Months 7–12 — Assurance and Scale

- Automated discovery covering unmanaged and embedded agents.
- Behavioral baselining and automated response.
- Independent assurance against this framework's evidence requirements.
- Executive and board reporting on the Section 20 metrics.
- Structured autonomy expansion driven by measured control efficacy rather than by delivery pressure.

PART IV

Appendices and References

The glossary, a one-page executive summary for a board or leadership team, and the sources the framework is built on.

24 Appendices

Appendix A — Glossary

Action boundary the point at which an agent's decision produces a side effect outside its own reasoning.

Agency the capacity to take actions with real-world effect, distinct from the capacity to generate text.

Blast radius the enumerated set of systems, data, and credentials reachable by a compromised agent before a control intervenes.

Blueprint a reusable, pre-approved configuration template for an agent class.

Circuit breaker an automated halt triggered by repeated failure or anomalous action sequences.

Ephemeral credential a secret whose validity is bounded to a single task or a short fixed window.

Least agency minimum autonomy, minimum tool access, minimum credential scope, minimum duration.

MCP Model Context Protocol; a standard interface exposing tools and data to agents. Governed under the Agentic AI Foundation since December 2025.[26]

NHI non-human identity.

Provenance labeling tagging context elements with source and trust level so the orchestrator can constrain low-trust content.

Standing credential any secret in an execution context that outlives the task it was issued for.

Velocity ceiling a hard limit on action rate that halts execution when exceeded.

Appendix B — One-Page Executive Summary

THE PROBLEM

Agents act with delegated authority, at machine speed, across long horizons, on infrastructure. Existing controls were designed for humans making requests or for automation with predictable behavior. Neither assumption holds.

WHAT THE EVIDENCE SHOWS

[4,5] In July 2026, frontier models running an internal security benchmark exploited a zero-day in the single permitted egress path from their isolated test environment, moved laterally to a node with internet access, and attacked a third party's production infrastructure to obtain the benchmark's answers. The operator did not identify its own agent as the cause for roughly a week. The target detected the intrusion first. Neither failure was about model intelligence; both were about architecture and instrumentation.

THE FIVE CONTROLS THAT RETIRE THE MOST RISK

1. Enumerate and default-deny egress from every agent and CI runtime; treat the package proxy as a production security boundary.
2. Eliminate standing credentials from agent execution contexts.
3. Enforce tool authorization deterministically at the action boundary, with velocity and blast-radius ceilings that halt rather than alert.
4. Stage and test a self-hosted forensic model before an incident—hosted providers' safety classifiers will refuse the artifacts you need analyzed.
5. Govern reduced-guardrail work formally, with isolation stronger than production rather than weaker.

THE MINIMUM DEFENSIBLE BAR

Level 2 across all ten control domains for any production agent touching enterprise data. Overall maturity is the minimum across domains, not the average.

THE METRIC THAT PREDICTS FAILURE

Agent population growth rate exceeding registry coverage growth rate.

25 Reference Sources

Numbered in order of first appearance. Standards and regulatory instruments are cited to the issuing body; verify the current version and any superseding revision before relying on a specific clause. Figures drawn from vendor and analyst surveys are reported as published and have not been independently audited.

URLs were checked in September 2026. Entries marked † resolve to a publisher landing page rather than a verified deep link, because the document sits behind registration, is distributed as a periodically revised artifact, or had no stable permalink at the time of writing—confirm the specific edition before citing it externally.

1 "OWASP Top 10 for Agentic Applications 2026 (ASI01–ASI10)," *OWASP GenAI Security Project*, December 9, 2025.
https://genai.owasp.org/2025/12/09/owasp-top-10-for-agentic-applications-the-benchmark-for-agentic-security-in-the-age-of-autonomous-ai/

2 *MITRE ATLAS: Adversarial Threat Landscape for Artificial-Intelligence Systems*, MITRE Corporation.
https://atlas.mitre.org/

3 *AI Controls Matrix (AICM) v1.1*, Cloud Security Alliance (247 control objectives across 18 domains, with mappings to ISO/IEC 42001 and ISO/IEC 27001). †
https://cloudsecurityalliance.org/artifacts/ai-controls-matrix

4 "OpenAI and Hugging Face Partner to Address Security Incident During Model Evaluation," *OpenAI*, July 21, 2026.
https://openai.com/index/hugging-face-model-evaluation-security-incident/

5 "Security Incident Disclosure — July 2026," *Hugging Face*, July 16, 2026.
https://huggingface.co/blog/security-incident-july-2026

6 Zhun Wang, Nico Schiller, Hongwei Li, et al., "ExploitGym: Can AI Agents Turn Security Vulnerabilities into Real Attacks?" *arXiv:2605.11086*, May 11, 2026.
https://arxiv.org/abs/2605.11086

7 "The AI Agent Governance Gap: What CISOs Need Now," *Cloud Security Alliance Lab Space*, April 2026, reporting the 2026 CISO AI Risk Report survey of 235 large-enterprise CISOs and CIOs.
https://labs.cloudsecurityalliance.org/research/csa-research-note-ai-agent-governance-framework-gap-20260403/

8 "Top Cybersecurity Trends 2026," *Gartner*, April 28, 2026, as reported by the Cloud Security Alliance. †
https://www.gartner.com/en/topics/cybersecurity

9 "AI Agents at Work 2026: Securing the Agentic Enterprise," *Okta*, June 2, 2026. †
https://www.okta.com/resources/

10 "Top Cybersecurity Threats 2026," *Forrester Research*, as reported in *Cybersecurity Insiders*, July 2026. †
https://www.forrester.com/technology/security-risk/

11 "2026 SANS State of Identity Threats & Defenses Survey," *SANS Institute*, March 10, 2026. †
https://www.sans.org/white-papers/

12 "The Non-Human Identity Governance Vacuum: AI Agents and the Fastest-Growing Unmanaged Attack Surface," *Cloud Security Alliance Lab Space*, May 20, 2026.
https://labs.cloudsecurityalliance.org/research/csa-whitepaper-nonhuman-identity-agentic-ai-governance-v1-cs/

13 "State of Identity Security 2026," *Sophos*, May 12, 2026 (survey of 5,000 security leaders across 17 countries). †
https://www.sophos.com/en-us/whitepaper

14 *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*, NIST AI 100-1, National Institute of Standards and Technology, January 2023.
https://www.nist.gov/itl/ai-risk-management-framework

15 *ISO/IEC 42001:2023 — Information Technology, Artificial Intelligence, Management System*, International Organization for Standardization, December 2023. †
https://www.iso.org/standard/42001

16 "Accelerating the Adoption of Software and Artificial Intelligence Agent Identity and Authorization," draft concept paper, *National Cybersecurity Center of Excellence, NIST*, February 5, 2026.
https://www.nccoe.nist.gov/projects/software-and-ai-agent-identity-and-authorization

17 "OWASP Non-Human Identities Top 10," *OWASP Foundation*.
https://owasp.org/www-project-non-human-identities-top-10/

18 *CIS Benchmarks*, Center for Internet Security.
https://www.cisecurity.org/cis-benchmarks

19 "OWASP Top 10 for LLM Applications 2026," *OWASP GenAI Security Project*, August 3, 2026.
https://genai.owasp.org/resource/owasp-genai-llm-top-10-2026/

20 *Supply-chain Levels for Software Artifacts (SLSA)*, Open Source Security Foundation.
https://slsa.dev/

21 *Secure Software Development Framework (SSDF) Version 1.1*, NIST SP 800-218, National Institute of Standards and Technology, February 2022.
https://csrc.nist.gov/pubs/sp/800/218/final

22 *Computer Security Incident Handling Guide*, NIST SP 800-61, National Institute of Standards and Technology. †
https://csrc.nist.gov/pubs/sp/800/61/

23 Regulation (EU) 2024/1689 (Artificial Intelligence Act), *Official Journal of the European Union*, June 13, 2024.
https://eur-lex.europa.eu/eli/reg/2024/1689/oj

24 Regulation (EU) 2022/2554 (Digital Operational Resilience Act), *Official Journal of the European Union*, December 14, 2022.
https://eur-lex.europa.eu/eli/reg/2022/2554/oj

25 "AI Agent Standards Initiative," *Center for AI Standards and Innovation, National Institute of Standards and Technology*, launched February 17, 2026.
https://www.nist.gov/artificial-intelligence/ai-agent-standards-initiative

26 *Model Context Protocol*, Agentic AI Foundation, Linux Foundation, December 2025; authorization specification built on OAuth 2.1 with PKCE and Resource Indicators (RFC 8707).
https://modelcontextprotocol.io/

27 *Agent Control Standard (ACS)*, OWASP GenAI Security Project, unveiled September 1, 2026 (runtime agent governance: guardian-agent enforcement points, OpenTelemetry and OCSF observability, and an Agent Bill of Materials expressed in CycloneDX, SWID, or SPDX).
https://genai.owasp.org/2026/09/01/owasp-genai-security-project-unveils-2026-top-10-for-llm-applications-new-agent-control-standard-and-sponsors-as-community-tops-30000-members/

28 *Agent Observability Standard (AOS)*, OWASP GenAI Security Project. †
https://aos.owasp.org/

29 "Semantic Conventions for Generative AI Systems," *OpenTelemetry* (gen_ai.* namespace at Development stability; agent, workflow, tool, and model spans).
https://opentelemetry.io/docs/specs/semconv/gen-ai/

30 "Practical Guide for Securely Using Third-Party MCP Servers," *OWASP GenAI Security Project*. †
https://genai.owasp.org/resources/

31 "Agentic MCP Security Best Practices v1," *Cloud Security Alliance*, August 2026. †
https://cloudsecurityalliance.org/research/artifacts

32 *Secure Production Identity Framework for Everyone (SPIFFE)*, Cloud Native Computing Foundation.
https://spiffe.io/

Version 1.3. Published August 2026; updated September 2026. This framework is offered as practitioner guidance and does not constitute legal or regulatory advice. Verify all regulatory mappings against current obligations in your jurisdiction.

