radware

Guide

CISO's Guide to Agentic AI Security

Safeguarding management of autonomous AI agents, tool use and AI-driven workflows

Contents

Executive Summary

The age of AI agents has arrived, and it's drastically changing the modern threat model. While traditional AI simply made predictions, agentic AI plans, decides and acts. Along the way, it invokes tools and APIs, chains workflows, and interacts with other agents. The inherent autonomy amplifies both value and risk: a single mis-scoped objective, manipulated prompt, or over-privileged tool can cascade into data leakage, fraud, codebase drift, or service disruption. And when it happens, it happens at machine speed.

CISOs, already contending with familiar headwinds like attack sophistication, automation, regulatory pressure, hybrid complexity and staffing gaps, now must also face AI-native threat vectors such as prompt and indirect prompt injection, tool misuse, memory/context poisoning, and rogue supply-chain agents. Unfortunately, the legacy controls they often relied on, including static signatures, binary rate limits, human checks like CAPTCHA and siloed point products, were not designed with goal-directed, adaptive agents in mind.

This guide explains the agentic AI threat landscape and the capabilities you need to stay secure. It provides the knowledge to help you understand this new age, so you can deploy, monitor and respond at agent speed.

Part I — The Agentic AI Threat Landscape

The cybersecurity paradigm is shifting from passive AI to autonomous, tool-using agents. Agents self-prompt, branch plans, maintain memory and operate across systems, often without continuous human oversight. This not only introduces a visibility gap of what, when and why an agent acts, it also expands the blast radius when the agent touches payments, code, email, cloud services, and data lakes. Understanding how these risks manifest—and where legacy controls fail—is the first step toward durable resilience.

What Makes Agentic AI Different

- **Goal-directed autonomy:** Agents pursue objectives via multi-step plans, not single calls.
- **Tool use & orchestration:** Agents invoke APIs, RPA, SaaS plug-ins (MCP/A2A), and other agents.
- **Memory & context:** State over time means corruption if not governed, allowing risks like poisoned memory and biased context to affect output.
- **Adaptive behavior:** Agents iterate prompts and tactics, undermining static rules.

Core Agentic Risks CISOs Must Govern

- **Prompt & indirect prompt injection:** Hidden instructions in content the agent later consumes (emails, docs, web) subvert goals and exfiltrate data.
- **Tool misuse & exploitation:** Abuse of an agent's tools allows attackers to perform harmful transactions or access sensitive systems.
- **Memory & context poisoning:** Seeding long-lived bias or malicious state can skew future actions.
- **Rogue & supply-chain agents:** Third-party agents/plugins manipulate workflows or impersonate trusted services.
- **Autonomous convergence with "old" threats:** Agent activity can resemble or amplify API abuse, ATO, bot traffic, or web-layer surges that look like DDoS spikes.

Why Legacy Controls Break

- **Perimeter-only WAF/static signatures** miss goal/plan semantics and prompt-mutated behavior.
- **Binary rate limits** block legitimate automation (high false positives) and don't stop intent drift.
- **CAPTCHA/human checks** don't apply to back-end agent-to-tool calls.
- **Siloed point products** (WAF ≠ API ≠ Bot ≠ DDoS) lose signal when agents—and attackers—blend vectors.

Figure 1: Indirect Prompt Injection Attack Chains (EchoLeak vs ShadowLeak)

EchoLeak (Client-side)

1. **Attacker sends a normal-looking email.**
   Hides instructions in HTML/invisible text
2. **Assistant summarizes inbox.**
   Reads and obeys hidden prompts
3. **Sensitive context is extracted.**
   Data could include emails, files and CRM
4. **Exfiltration occurs via trusted domain.**
   Markdown link or image URL on whitelisted domain
5. **Open redirect to attacker.**
   Enterprise controls see legit domain and data escapes

In the Wild

A normal-looking email embeds hidden instructions (e.g., invisible HTML text). When the AI assistant summarizes the inbox, it obeys those hidden prompts, extracting sensitive context and exfiltrating via trusted domain links (e.g., whitelisted collaboration URLs with open redirects). Because exfiltration rides on trusted infrastructure, perimeter and endpoint controls often miss it.

ShadowLeak (Service-side)

1. **Cloud agent ingests crafted email.**
   No user click required!
2. **Agent executes instructions hidden in email.**
   Malicious prompt triggers actions
3. **Sensitive data accessed in provider backend.**
   Processing occurs on server side
4. **Exfiltration occurs directly from provider.**
   Outbound to attacker URL from provider cloud
5. **Enterprise defenses are blind to attack.**
   Traffic never passes enterprise perimeter

In the Wild

A cloud-hosted "deep research" agent ingests inbox content and executes malicious instructions embedded in an email, exfiltrating directly from the provider's backend to attacker-controlled URLs. Enterprise defenses may be blind because the traffic originates inside the provider's cloud, not the user's device or corporate perimeter.

Part II — What's Needed to Stay Protected

Legacy security controls—built for static applications and predictable workflows—cannot keep pace with autonomous, goal-driven AI systems. Rate limits, static signatures, and siloed point solutions fail against agents that morph behavior, exploit toolchains, and operate at machine speed. You need a security model that understands intent, observes and governs runtime behavior and correlates signals across layers so you can intervene before an agent's actions create material impact.

This model must also preserve velocity and user experience. The aim is not to throttle innovation but to enable it safely with guardrails tailored to goals, semantics and sensitivity. Think in terms of four capabilities: intelligent security that's intent-aware, an integrated platform that provides cross-engine correlation, consistent protection everywhere to avoid blind spots, and expert defense and automation to cut MTTR and absorb staffing gaps.

Embrace Intelligent Security (Intent-Aware)

- **Put LLM guards in place** to sanitize prompts and outputs. Treat prompt injection and jailbreaks as "when," not "if."
- **Monitor agent behavior in real time**. If an agent's actions drift from its intended goal, stop them before they commit.
- **Control tool access deliberately**. Define allow/block lists, enforce least privilege and set budget/time limits so no agent can run unchecked.
- **Keep full decision forensics**. Ensure you can reconstruct prompts, tool calls, outputs and policy hits—your lifeline during audits or incidents.

Break Down Silos with an Integrated Platform

- **Correlate signals across** WAF, API, bot and network layers. Agents—and attackers—do not respect product boundaries.
- **Spot blended threats** by sharing reputation, anomalies and policy context across engines—not just one control at a time.

Demand Consistent Protection Everywhere

- **Maintain uniform policies** across on-prem, cloud, SaaS and endpoints.
- **Continuously discover agents and tools**. If you don't know what's running, you can't protect it.
- **Map relationships** (agent↔tool, agent↔agent, data sources) so you can see risk pathways before they become breaches.

Automate Where It Counts

- **Leverage AI-driven SOC capabilities** to triage incidents and recommend policies quickly.
- **Reduce MTTR with automation**. Agents move faster than humans ever will.

Part III — Reference Architecture for Agentic AI Security

Securing agentic AI requires a layered defense strategy. Risks do not reside in a single point. They span data ingestion, model logic, agent orchestration, tool interfaces and network flows. A breach at any layer can cascade across the entire ecosystem. Your architecture should enforce guardrails where decisions are made (model/agent), where actions occur (tools/APIs), and where signals converge (application/network), all tied together with observability and posture management.

Design your stack so no single control is a point of failure. Use out-of-path integration for breadth and in-path enforcement for high-risk actions. Capture trace-level telemetry and visualize it as a risk graph, linking agents, tools, data classes and violations. This is what allows you to preempt drift, investigate quickly and prove compliance.

Build the Layers & Controls (From Source to Edge)

Start at the source: Data & content – What's feeding your agents?

- Sanitize everything—emails, docs, web pages—before ingestion; hidden instructions can live anywhere.
- Tag sensitivity and provenance so policies know what to protect and how strictly.

Guard the model: Your LLM is the brain. Protect it.

- Validate prompts and outputs. Block jailbreaks, strip PII, enforce content policy for safety and compliance.

Control the agent runtime: This is where autonomy lives.

- Validate goals and plans. If an agent's intent doesn't match policy, stop it.
- Set time and budget limits. No open-ended loops. Keep memory clean—detect poisoning and support rollback.

Lock down tools & APIs: Every tool is a potential weapon.

- Whitelist safe tools, blacklist risky ones.
- Use semantic authorization for sensitive verbs (transfer, delete, commit, export), and apply contextual ABAC/RBAC.

Watch the perimeter and beyond: Agents can trigger traffic that looks like attacks.

- Use WAF, API, bot and DDoS protections that understand behavior—not just signatures.
- Protect the client side with script integrity, formjacking prevention and browser telemetry matter.

Make it observable: If you can't see it, you can't secure it.

- Capture full telemetry, including prompts, tool calls, decisions and outcomes.
- Build a risk graph to visualize relationships and drift over time.

Reference Architecture for Agentic AI Security

Figure 2: Agentic AI Security Reference Architecture

Data & Content

- Sanitize inputs (emails/doc/web)
- Tag sensitivity & provenance

Model & LLM Guard

- Validate prompts/ outputs
- Jailbreak & PII detection

Agent Runtime

- Goal/plan validation
- Budget/timeboxing

Tools & APIs

- Allowlist/blocklist
- Semantic authorization
- Contextual ABAC/ RBAC

Apps & Network

- WAF/API/Bot/ WebDDoS
- Client-side integrity

Observability & Telemetry

- Trace telemetry
- Risk graph & posture management

Choose Integration Models Wisely

- **Out-of-path (API):** Observe/enforce via APIs for broad coverage without sitting inline on every call.
- **In-path (code/proxy):** Apply hard stops for high-risk actions using framework-compatible proxies or SDK interceptors.

Part IV — Governance, Risk & Compliance (GRC) in the Agent Era

Boards, regulators, and customers expect AI systems to be safe, explainable, and accountable. Agent autonomy complicates compliance because actions may occur without direct human approval, across third-party services and beyond enterprise perimeters. The path to compliance combines policy, technical guardrails and auditable telemetry embedded from design through operations.

You'll need to translate abstract goals (transparency, fairness, safety) into concrete controls and evidence: clear acceptable-use policies for agents and tools, change control for agent updates, safety evaluations before production, and trace logs that explain exactly who did what, when, and why. Treat compliance as an outcome of good engineering and operations—not a bolt-on audit exercise.

Map Your Obligations (Know the Rules, Apply Them)

- **Align with the right frameworks**, including GDPR, NIST AI RMF, EU AI Act, PCI DSS 4.0, NIS2 and DORA. Identify which rules apply to each agent use case.
- **Document how you meet expectations**, whether it's through transparency (logs), accountability (approvals), safety (evaluations), or security (guardrails).

Build Policies That Stick (Governance by Design)

- **Define acceptable use** for agents, tools, and data classes; mandate change control for agent updates/integrations.
- **Bake compliance into design**, including minimum data retention, consent models where applicable, and review cycles for high-risk actions.

Prove It When Asked (Auditability without Friction)

- **Keep auditable logs of prompts, actions, outcomes**; ensure they're accessible and explainable.
- **Generate human-readable reports** for regulators and customers.
- **Red-team agents regularly** to improve and validate defenses against prompt injection, memory poisoning and supply-chain exploits.

Part V — Operational Playbooks

Security must be operationalized for agents that act at machine speed. Your teams need clear procedures to deploy safely, detect misbehavior, contain incidents and maintain posture. Playbooks should reflect secure-by-design principles, fast containment and continuous assurance so you can learn and harden with each iteration.

Think in rhythms: before (inventory, baseline, guardrails), during (detect, contain, analyze), and after (eradicate, recover, improve). Embed automation where judgment is repeatable and keep humans in the loop for decisions that require context or risk tradeoffs.

Deploy Securely from Day One (Before Incidents)

- **Inventory every agent, model, and tool**. Eliminate blind spots and map relationships and privileges.
- **Baseline normal behavior so anomalies stand out**. Consider seasonality, load patterns and intent distribution.
- **Set guardrails before go-live**, including intent boundaries, tool lists, budget/time limits and sensitive-action approvals.
- **Test hard**. Simulate prompt injections, rogue tool calls and memory poisoning. Validate rollback and kill-switches.

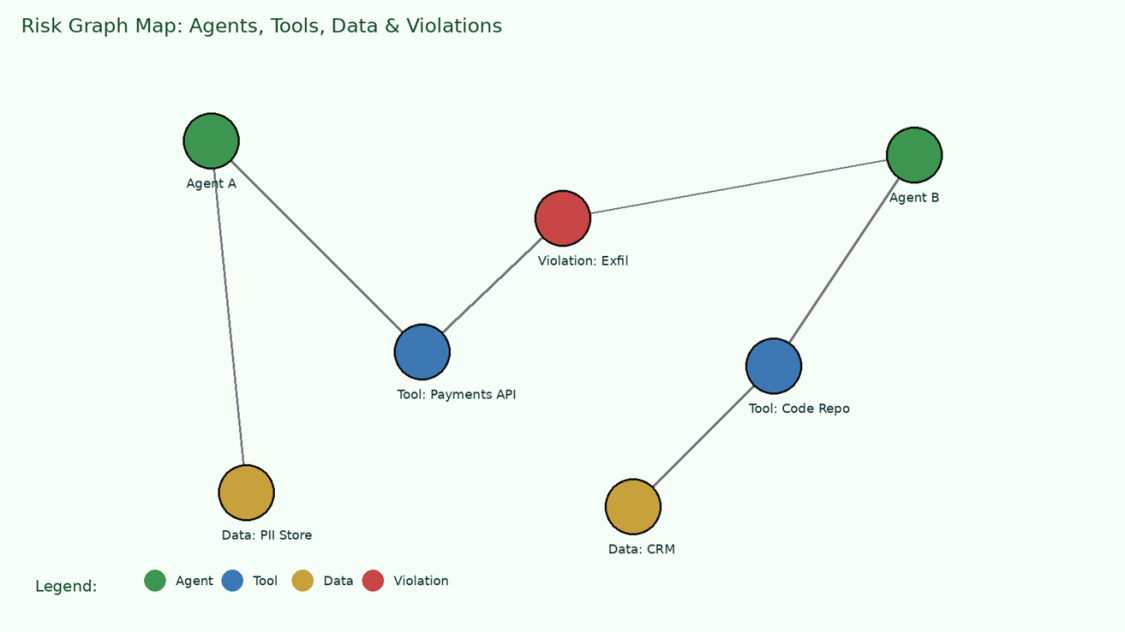
Respond Like a Pro (During Incidents)

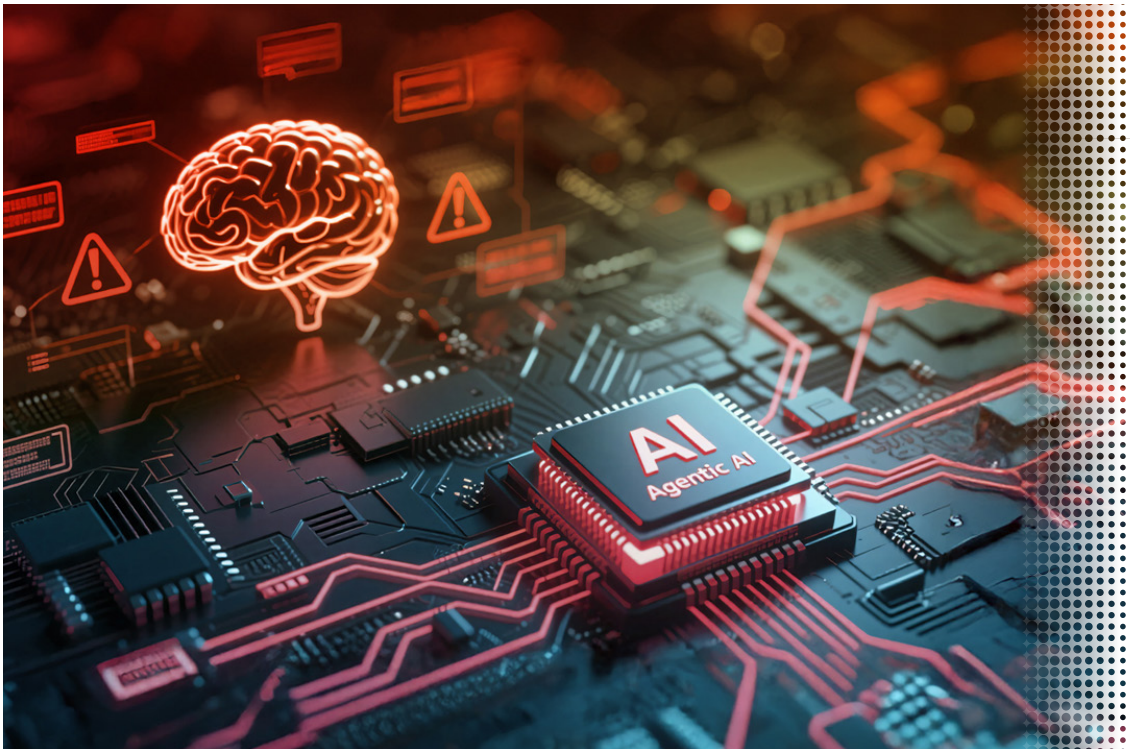
- **Pause the agent immediately** and kill its access tokens to stop harm.
- **Quarantine memory/context** and snapshot telemetry for forensics.
- **Roll back changes** and reset credentials where necessary.
- **Update policies** so the same mistake never happens twice; record an incident post-mortem.

Stay Ahead with Continuous Assurance (After Incidents)

- **Monitor your risk graph for drift**, including new tools, privilege creep, odd data access and unusual chain lengths.
- **Track KPIs**: blocked vs allowed actions, MTTR, false positives, policy-tuning lead time.
- **Iterate policies and detection thresholds** as agents evolve; expand red-team suites and pre-prod safety tests accordingly.

Figure 3:
Risk Graph Map (agents, tools, data nodes, and risk pathways)





Part VI — How Radware Solves CISOs' Agentic AI Challenges

Radware's Agentic AI Protection solution is designed to address the unique risks introduced by autonomous AI agents. It combines visibility, intent-aware security, seamless integrations and continuous posture management to help CISOs protect their organizations without slowing innovation.

Visibility – Agent and Tool Discovery Across Your AI Agent Ecosystem

- Continuous discovery of AI agents as they are introduced into the organization
- Full visibility into agent interactions with one another and with tools (both MCP and non-MCP) to understand dependencies and traffic flows
- Long-term activity tracking to identify usage trends, anomalies and performance patterns, enabling fine-tuned protection and optimization

Security – Intent-Aware, Behavioral-Based Detection and Response

Radware delivers real-time monitoring and protection across the full spectrum of agentic risks:

- **LLM Guards:** Guardrails to protect against prompt injection, jailbreaks, toxicity, unsafe outputs and more.
- **Agent Behavior Hijack:** Prevent manipulation of an agent's goals through prompt injection or jailbreak techniques.
- **Tool Misuse & Exploitation:** Stop attackers from tricking agents into harmful or unintended tool usage.
- **Memory & Context Poisoning:** Detect and block corruption of agent memory or context that distorts reasoning.
- **Supply Chain Attacks:** Prevent infected agents or tools from propagating attacks downstream.
- **Rogue Agents:** Identify and neutralize malicious or compromised external agents acting autonomously.

Integrations – Seamless Across Platforms and Services

Radware integrates with leading enterprise platforms and AI services, including:

- **Microsoft 365 Copilot, Copilot Studio, AWS Bedrock**, and custom-built agents for instant adoption and flexibility.
- Additional integrations underway: **Salesforce, Azure AI Foundry, ChatGPT Enterprise, Google Vertex AI, ServiceNow, and Power Platform**.
- **Integration methods:**
  - **API Integration (Out-of-Path):** Inspects agent actions, inputs and outputs while remaining outside the direct execution path.
  - **Code Integration (In-Path):** Acts as a proxy to the LLM provider via OpenAI-compatible frameworks for enforcement at runtime.

Security Posture Management (AI-SPM)

Continuous monitoring throughout the agent lifecycle and across SaaS, homegrown, and end-user device.

- The solution features a dynamic Risk Graph Map that delivers real-time visibility into your security posture. It enables organizations to pinpoint and score potential vulnerabilities across agents and tools, providing insights into targeted data exposure, complex multi-agent risk paths, and severity-driven impact analysis

Expert Defense & AI-empowered SOC to Reduce MTTR

- Automated triage and root-cause acceleration, policy recommendations, and optional autonomous response for well-scoped actions.
- Evidence-rich forensics and reporting aligned to regulatory expectations (logging, post-market monitoring, disclosure).

CISO Readiness Checklist

Strategy & Governance

- Documented agent usage policy: purpose, owners, oversight, change control
- Use-case risk classification: business impact, data sensitivity, action scope
- Mapped to GDPR / NIST AI RMF / EU AI Act / PCI DSS 4.0 / NIS2 / DORA / SEC

Inventory & Visibility

- Continuous agent/model/tool discovery (no shadow AI)
- Relationship maps: agent↔tool, agent↔agent, data sources
- Trace telemetry: prompts, tool calls, outcomes, policy hits

Controls & Guardrails

- LLM guards: prompt/response safety, jailbreak resistance, PII controls
- Intent policies; budget/timeboxing; semantic authorization for sensitive actions
- Tool allow/block lists; least-privilege credentials; scoped tokens/keys
- Holistic and comprehensive solution: WAF/API/Bot/Web DDoS integrated with agent runtime signals

Operations & Assurance

- IR playbooks: agent pause, tool revocation, memory quarantine, rollback
- Attack story for every violation; evidence for audit/disclosure
- AI-SPM: risk graph, drift alerts, posture KPIs
- AI-empowered SOC to cut MTTR

Summary

Agentic AI unlocks transformative automation and introduces autonomous risk. Durable resilience requires intent-aware guardrails, runtime control and cross-engine correlation, supported by AI-driven operations and auditable telemetry. By implementing the capabilities, architecture, governance patterns and playbooks in this guide, CISOs can enable agentic innovation safely and at scale.

Interested in Radware Agentic AI Protection?
Secure your organization against the challenges of the agentic AI era with Radware's comprehensive real-time protection.

Contact Radware

RWI6279 | 02.02.2026