CLOUD SECURITY ALLIANCE · IRIUSRISK · ACADEMIC RESEARCH

MAESTRO in Practice

Threats, mitigations, and outcomes across 12 case studies

A structured field study of the seven-layer agentic AI threat modeling framework, with comparative analysis and hardening recommendations for securing multi-agent systems.

12	7	60+	204
CASE STUDIES	MAESTRO LAYERS	THREATS CATALOGED	MITIGATIONS (TITO)

MAESTRO (Multi-Agent Environment, Security, Threat, Risk, and Outcome) was introduced by Ken Huang at the Cloud Security Alliance in February 2025 [Huang, 2025a]. This report compiles 12 documented applications across CSA, IriusRisk, and preprint research between 2025 and 2026 — extracting the threats identified, the mitigations applied, and the outcomes observed for each of the seven layers.

PREPARED FOR / Security Engineering · Content Strategy · Durham, NC

Author: Mazze LeCzzare Frazer · Compiled June 2026

Table of contents

AT A GLANCE · 8 sections · 12 case studies · jump links resolve to PDF bookmarks

Executive summary

CSA presents MAESTRO as a purpose-built threat-modeling framework for agentic AI [Huang, 2025a]. Unlike STRIDE, PASTA, or LINDDUN, it explicitly models adversarial machine learning, agent autonomy, dynamic agent-to-agent interactions, and AI supply chain risk. Its seven-layer decomposition has been adopted in published CSA guidance and commercial tooling, including IriusRisk v4.47 [IriusRisk, 2025] (October 2025), the open-source TITO CI/CD scanner [TITO repo, 2026] (February 2026), and Snyk Labs guidance [Snyk Labs, 2026] (February 2026).

Core findings

L3 is the most consistently exploited attack surface. Layer 3 (Agent Frameworks) appears in every case study reviewed. It is the orchestration hub linking foundation models (L1) to action execution (L4), and the most common architectural failure is collapsing L1→L3→L4 into a single implicit trust domain.

Trust boundary validation is missing by default. A TITO scan [Leath & Huang, 2026] of a representative agent surfaced 9 distinct threats and 204 mitigation recommendations — self-reported tool output, not independently reproduced here — with no validation point between user input and database access in the entire pipeline.

Memory and RAG are a persistence vector. CSA Labs [CSA Labs, 2026a] reports that 5 malicious documents can drive a 90% knowledge-base compromise rate, citing PoisonedRAG-style findings; treat as a CSA-reported figure pending the original study. Memory poisoning persists across sessions and cascades through L7.

MCP servers cannot be trusted by source. CSA Labs [CSA Labs, 2026b] cites a 72.8% tool-poisoning success rate against production LLM agents (attributed to Invariant Labs) using Unicode injection in MCP tool descriptions; see the original Invariant Labs research for methodology.

Agentic kill chains are real and growing. By 2026, CSA Labs' incident review [CSA Labs, 2026c] found 15 of 21 documented promptware incidents exhibited four or more kill-chain stages, with lateral movement in 8 — the LOTA ("Living Off the Agent") pattern, per CSA's stated corpus and coding rules.

Best decision — Adopt MAESTRO as the single threat modeling axis

Use MAESTRO as the structural threat model, MITRE ATLAS as the technique taxonomy, and NIST AI RMF as the governance wrapper. When not to use: single-call LLM products with no tool use or persistent memory — STRIDE remains sufficient. Alternative: OWASP Agentic Security Initiative (ASI) — complementary, but lacks MAESTRO's layered decomposition.

1 · Framework foundation

MAESTRO was authored by Ken Huang (CEO, DistributedApps.ai; Co-Chair, CSA AI Safety Working Groups) and published by the Cloud Security Alliance on February 6, 2025 [Huang, 2025a]. The framework's reference hub is maintained by CSA Labs [CSA Labs, MAESTRO Hub].

AT A GLANCE · 7 layers · 6-step methodology · supersedes STRIDE/PASTA/LINDDUN for agentic AI

How it differs from prior frameworks

Framework	Era	Models AI threats?	Models agent autonomy?	Layered?
STRIDE (Microsoft)	2002	No	No	Per-component
PASTA	2012	No	No	Linear pipeline
LINDDUN	2010s	Privacy only	No	Per-component
OWASP LLM Top 10	2023	Yes (LLM-centric)	Partial	Flat list
MAESTRO	2025	Yes — extends STRIDE/PASTA	Yes — agent + multi-agent	7 layers + cross-layer

The seven layers

Layer	Name	Scope
L1	Foundation Models	Model weights, training pipeline, inference, base alignment.
L2	Data Operations	Databases, vector stores, RAG pipelines, training datasets, embedding infrastructure.
L3	Agent Frameworks	LangChain, AutoGen, CrewAI, OpenAI Responses API, custom orchestrators, tool dispatch.
L4	Deployment & Infrastructure	Containers, Kubernetes, IaC, cloud services, networks, serverless functions.
L5	Evaluation & Observability	Logging, metrics, anomaly detection, evaluation benchmarks, drift monitoring.
L6	Security & Compliance	Vertical (cross-cutting): AuthN/Z, audit trails, regulatory mapping, key management.

Layer	Name	Scope
L7	Agent Ecosystem	Agent registries, A2A protocol, MCP servers, B2B integrations, multi-agent orchestration.

Six-step implementation methodology

1. System decomposition — break the system into components by layer; define agent capabilities, goals, tool registries, and interactions.
2. Layer-specific threat modeling — apply MAESTRO's threat catalogs to each layer.
3. Cross-layer threat identification — analyze how vulnerabilities in one layer cascade to others.
4. Risk assessment — likelihood × impact; prioritize via risk matrix.
5. Mitigation planning — select layer-specific, cross-layer, and AI-specific controls.
6. Implementation and monitoring — deploy controls; continuously update as agents and threats evolve.

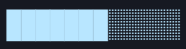
2 · Case studies at a glance

Twelve documented applications of MAESTRO across primary sources (CSA, IriusRisk, Snyk Labs) and preprint research between March 2025 and May 2026. Each is detailed in the next section.

AT A GLANCE · 12 case studies · 7 CSA articles/notes · 2 preprints · 3 vendor/journal sources

#	Subject	Source	Date	Evidence type	Primary layers
1	OpenAI Responses API	CSA / Huang	Mar 2025	First-party	L1·L2·L3·L5·L6·L7
2	Google A2A Protocol	CSA / Huang & Habler	Apr 2025	First-party	L3·L1·L2·L6·L7
3	MCP Secure Implementation	CSA / Huang & Chen	Jun 2025	First-party	L2·L3·L4·L6·L7
4	IriusRisk MAESTRO Library	IriusRisk v4.47	Oct 2025	Vendor release	All 7 + ATLAS
5	Network monitoring agent	arXiv 2508.10043	Aug 2025	Preprint	L2·L3·L4
6	A2A protocol	arXiv 2504.16902	Apr 2025	Preprint	All 7
7	TITO CI/CD scan of agent	CSA / Leath & Huang	Feb 2026	First-party	L1·L2·L3·L4
8	RAG agents as insider threats	CSA Labs	Mar 2026	Research note	L2·L3·L5·L7
9	MCP security crisis	CSA Labs	May 2026	Research note	L3·L4·L7
10	LOTA lateral movement	CSA Labs	May 2026	Research note	L3·L4·L7
11	Snyk Labs layered TM	Snyk Labs	Feb 2026	Vendor guidance	All 7
12	Enterprise lifecycle governance	IJAIDSML	2025	Journal article	All 7 + NIST

Layer coverage across case studies

Layer	Coverage (12 case studies)	Count
L1 — Foundation Models		8/12
L2 — Data Operations		11/12
L3 — Agent Frameworks		12/12
L4 — Deployment & Infra		9/12
L5 — Evaluation & Observability		6/12
L6 — Security & Compliance		7/12
L7 — Agent Ecosystem		10/12

3 · Case study deep dives

Full threat / mitigation / outcome breakdowns for each of the 12 case studies introduced in the previous section, in the same numbered order.

AT A GLANCE · 12 deep dives · threats identified · mitigations applied · outcomes & lessons

CASE STUDY 1 March 24, 2025

OpenAI Responses API — third-party API threat surface

Ken Huang, CSA / DistributedApps.ai · [Huang, 2025b]

SYSTEM Stateful, multi-tool agentic API with web search, file search, code execution, and function calling.

LAYERS ANALYZED All 7 (primary L1·L2·L3·L5·L6; L4 vendor-managed; L7 for multi-agent).

Threats identified

Threat	Layer	Risk
T1.1 Adversarial examples via input parameter	L1	HIGH
T2.1 Vector store / file poisoning	L2	HIGH
T3.1 Tool misuse via unintended invocations	L3	HIGH
T3.2 Prompt injection overriding system instructions	L3	HIGH
T6.1 Stolen API key → full agent control	L6	HIGH
T7.1 Malicious agent interaction in ecosystem	L7	HIGH
C3.2 Cross-layer: prompt injection → arbitrary code execution	L3→L4	CRITICAL

Mitigations applied

- M1.1.2 Input validation; M1.1.3 output verification via Moderations API.
- M3.1.6 Least privilege for tool access; M3.1.4 confirmation prompts for high-risk tools.
- M3.2.1–M3.2.4 input sanitization, instruction separation, instruction hardening, output validation.
- M6.1.1–M6.1.5 HSM-stored keys, rotation, MFA, usage monitoring.
- Rate limiting (max_completion_tokens, max_prompt_tokens); loop detection; strict tool_choice enforcement.

Outcomes & lessons

The Responses API sits primarily at L3 but creates attack surfaces across all seven layers. L4 risks (infrastructure DoS) remain OpenAI's responsibility — consumers mitigate with retries, exponential backoff, and fallbacks. The exercise validated MAESTRO as a practical tool for mapping third-party API threat surfaces before production deployment.

CASE STUDY 2 April 30, 2025

Google A2A Protocol — agent-to-agent identity and trust

Ken Huang (CSA) & Dr. Idan Habler (Intuit AI Security) · [Huang & Habler, 2025]

SYSTEM HTTP-based A2A protocol — Agent Card discovery, task execution, and SSE streaming between autonomous agents.

LAYERS ANALYZED All 7 (primary L3; significant L1·L2·L6·L7).

Threats identified

Threat	Layer	Risk
T1.1 Adversarial message generation	L1	HIGH
T1.2 Model extraction via A2A query patterns	L1	MEDIUM
T2.1 Data poisoning of TextParts/FileParts/DataParts	L2	HIGH
T3.1 Unauthorized agent impersonation (spoofed Agent Cards)	L3	HIGH
T3.4 Malicious A2A server impersonating trusted company	L3	HIGH
T6.1 Stolen private keys → full agent impersonation	L6	HIGH
T7.1 Compromised agent attacking others in ecosystem	L7	HIGH

Mitigations applied

- M3.1.1 Decentralized Identifiers (DIDs) for agent identity, with periodic refresh.
- M3.2.1 Digital signatures (incl. multi-signature) for all A2A messages.
- M3.4.x: mTLS, DNSSEC, Certificate Transparency for Agent Cards, agent-registry verification, behavioral reputation, and honeypot servers.
- M4.1.3 Adaptive rate limiting for DoS defense.
- M6.1.1 HSM-backed credential storage; M6.1.2 regular key rotation; MFA.

Outcomes & lessons

Agent non-determinism amplifies A2A threats — identical inputs produce divergent outputs, complicating defense. Dynamic credential acquisition lets compromised agents rapidly escalate capability. Also available as a preprint, arXiv:2504.16902, recommending A2A+MCP synergy for interoperable secure orchestration.

CASE STUDY 3 June 23, 2025

MCP Secure Implementation — Grid Operations Assistant

Ken Huang (CSA) & Dr. Ying-Jung Chen (Georgia Tech) · [Huang & Chen, 2025]

SYSTEM MCP Server + Host (LLM orchestrator) + Client architecture for a Grid Operations Assistant.

LAYERS ANALYZED All 7 (deep focus on L2 SQL injection, L3 supply chain, L4 env disclosure, L7 tool misuse).

Threats identified

Threat	Layer	Risk
Agent tool misuse / goal manipulation via prompt injection	L7	HIGH
Evasion of security controls — weak input validation + poor logging	L6	MEDIUM
Information disclosure: get_env_variable() leaking API keys	L4	HIGH
SQL injection in insert_record() via f-string formatting	L2	CRITICAL
Arbitrary SQL via permissive execute_sql() tool	L2/L7	CRITICAL
Supply chain risk in mcp, FastAPI, uvicorn	L3	MEDIUM

Mitigations applied

- Parameterized queries — never f-string construction from LLM output.
- Tool allowlists; LLM output sanitization before execution.
- Human-in-the-loop for sensitive operations (fund transfers, deletions).
- Session integrity enforcement; comprehensive logging of all tool calls.
- Principle of least privilege per tool; remove broad-scope tools like execute_sql().

Outcomes & lessons

AI-callable tools are application inputs and demand the same validation as web inputs. MCP servers must be treated as untrusted third-party code. The CSA guide ships production-ready secure code patterns for each vulnerability class — a rare deliverable that bridges threat model to remediation snippet.

CASE STUDY 4 October 2, 2025

IriusRisk v4.47 — MAESTRO as a commercial threat library

Jorge Esperón, IriusRisk · [IriusRisk, 2025]

SYSTEM Automated multi-agent architecture threat modeling: questionnaire-driven MAESTRO layer selection with MITRE ATLAS countermeasure mapping.

LAYERS ANALYZED All 7 — selectable per-component, automatically imported.

Threats identified

Threat	Layer	Risk
Adversarial examples / evasion (foundation models)	L1	HIGH
Model extraction / inversion via API queries	L1	MEDIUM
Backdoor triggers in model weights	L1	HIGH
Data poisoning during training phase	L1/L2	HIGH
Membership inference attacks	L1	MEDIUM
Multi-agent architecture risks (orchestrator + sub-agents)	L3/L7	HIGH

Mitigations applied

- Layer-specific MAESTRO mitigations auto-imported per selected layer.
- Cross-mapping to MITRE ATLAS techniques and mitigations (dual-framework coverage).
- Single-platform governance: traditional software threats and agentic AI threats under one methodology.
- Multi-agent diagrams supporting orchestrators, sub-agents, and tool services.

Outcomes & lessons

IriusRisk's v4.47 release is a commercial threat-modeling platform that integrates MAESTRO natively. Lowers the AI-security expertise barrier for security teams. Critically, threats are not just identified — each is bound to an actionable, standards-mapped countermeasure. The release marks MAESTRO's transition from CSA reference to operational tool.

CASE STUDY 5 August 12, 2025

Network monitoring agentic AI — an empirical MAESTRO prototype study

Zambare, Thanikella, Liu (arXiv preprint — venue not yet confirmed) · [Zambare et al., 2025 (preprint)]

SYSTEM Prototype LLM-based network monitoring agent (Python + LangChain + WebSocket telemetry) with inference, memory, parameter tuning, and anomaly detection.

LAYERS ANALYZED All 7 — empirical focus on L2 (memory), L3 (framework), L4 (infrastructure).

Threats identified

Threat	Layer	Risk
Resource DoS via traffic replay → telemetry delays	L4	HIGH
Memory poisoning via historical log tampering	L2	CRITICAL
Cross-layer communication vulnerabilities (unvalidated flows)	L1↔L4	HIGH
Adaptation logic corruption — anomaly thresholds skewed	L2/L3	HIGH

Mitigations applied

- Memory isolation and integrity validation.
- Planner validation — sanity checks on agent plan outputs before execution.
- Real-time anomaly response systems.
- Multilayered defense-in-depth architecture aligned to MAESTRO layers.

Outcomes & lessons

An empirical prototype study applying MAESTRO to a network-monitoring agent. Both attack scenarios were confirmed with measurable performance impact, demonstrating MAESTRO is not purely theoretical, though this is one prototype, not a field-wide validation. The authors conclude memory integrity is the most critical gap in deployed agent systems — memory and RAG systems require the same integrity controls as any other data store.

CASE STUDY 6 April 23 / May 2, 2025

A2A protocol — preprint security analysis

Habler, Huang, Narajala, Kulkarni · [Habler et al., 2025 (preprint)]

SYSTEM Google's A2A protocol — Agent Card management, task execution integrity, authentication, and A2A+MCP interoperability.

LAYERS ANALYZED All 7 — detailed per-layer threat IDs and M-code mitigations.

Threats identified

Threat	Layer	Risk
Agent Card spoofing enabling full impersonation	L3	HIGH
Message injection causing malicious agent behavior	L3	HIGH
Data poisoning via message parts (T2.1)	L2	HIGH
Malicious A2A server impersonation (T3.4)	L3	HIGH
Unauthorized credential access (T6.1)	L6	HIGH

Mitigations applied

- DIDs + mTLS + Certificate Transparency for agent cards.
- Multi-signature message schemes.
- Behavioral reputation systems resistant to identity change.
- HSMs for credential storage; time-stamped signatures preventing replay.

Outcomes & lessons

An early preprint applying MAESTRO to a specific agentic protocol. Argues that A2A+MCP synergy improves secure interoperability. Co-authored by industry (Intuit) and CSA leadership, bridging academic and practitioner perspectives — not yet confirmed as peer-reviewed at a venue.

CASE STUDY 7 February 11, 2026

TITO CI/CD scan — MAESTRO embedded in the pipeline

Steven Leath (TITO) & Ken Huang (CSA) · [Leath & Huang, 2026]

SYSTEM Representative LLM agent with 5 tools (shell, file R/W, browser, DB, webhooks) + PostgreSQL + RAG memory.

LAYERS ANALYZED L1·L2·L3·L4 — STRIDE-LM + MAESTRO + MITRE ATT&CK; mapping.

Threats identified

Threat	Layer	Risk
OpenAI API key leak: env config → HTTP headers	L4/L6	CRITICAL
L1→L3→L4 attack chain: user input → LLM → tool dispatch → shell/DB	L1→L4	CRITICAL
Unvalidated trust boundary crossings between layers (×4)	Cross-layer	HIGH
Implicit single trust domain spanning entire pipeline	Architectural	CRITICAL

Mitigations applied

- Block PRs introducing critical threats: tito scan --maestro --fail-on critical.
- Threat diffing — surface only deltas per PR to reduce alert fatigue.
- Trust boundary enforcement at every L1↔L3↔L4 transition.
- Per-tool RBAC; scoped, expiring tokens for sensitive operations.

Outcomes & lessons

This TITO run reportedly surfaced 9 distinct threats and 204 mitigation recommendations across 24 assets and 169 data flows for a moderately complex agent — self-reported scanner output from this one case study, not independently reproduced here. The critical insight, per this account: every agent with tool access has critical findings, framed as a structural property rather than a code-quality issue. The L1→L3→L4 collapsed trust domain recurs as an architectural failure across this case study set.

CASE STUDY 8 March 25, 2026

RAG agents as enterprise insider threats

CSA AI Safety Initiative · [CSA Labs, 2026a]

SYSTEM Enterprise AI agents with broad file/API/communication credentials and RAG-based knowledge systems.

LAYERS ANALYZED L2·L3·L5·L7.

Threats identified

Threat	Layer	Risk
Memory poisoning for persistence (5 docs → 90% RAG compromise)	L2	CRITICAL
Lateral movement via legitimate credential harvesting	L4/L7	HIGH
Silent data exfiltration over authorized connections	L2/L7	HIGH
Indirect prompt injection via docs/emails/DB records	L1/L2	HIGH
Agent collusion across L7 channels	L7	MEDIUM

Mitigations applied

- Cryptographic provenance for RAG content (agents cannot write to their own memory from untrusted sources).
- Authenticated, validated sources only for KB population.
- Memory auditing and routine clearing of poisoned state.
- TITO MAESTRO scans mapped to MITRE ATT&CK; for architecture-specific attack-path analysis.

Outcomes & lessons

Traditional enterprise security tooling is blind to agent-mediated insider threats because the agent operates within its authorized permissions. AI agents fundamentally alter the kill chain. Organizations should adopt MAESTRO's five core threat families as the taxonomy for agent insider-threat red-teaming.

CASE STUDY 9 May 4, 2026

MCP security crisis — systemic design flaws

CSA AI Safety Initiative · [CSA Labs, 2026b]

SYSTEM MCP server ecosystem — STDIO command injection, tool poisoning ("rug pull"), and registry-level supply chain compromise.

LAYERS ANALYZED L3·L4·L7.

Threats identified

Threat	Layer	Risk
STDIO command injection in MCP transport	L3/L4	CRITICAL
Tool poisoning / "rug pull" via Unicode injection (72.8% success)	L3/L7	CRITICAL
Supply chain compromise of MCP registries	L7	HIGH
Privilege escalation via tool chaining beyond intended scope	L3	HIGH

Mitigations applied

- Promote MCP server trust boundaries to first-class threat modeling primitives.
- Cryptographic provenance verification for MCP servers before invocation.
- Behavioral sandboxing + runtime monitoring for tool invocations.
- Zero-trust: no MCP server trusted by source alone.
- Hash-based baselines for tool call signatures; micro-segmentation policy controls.

Outcomes & lessons

MCP servers should be treated with the same skepticism as third-party web inputs. The 72.8% attack success rate against production agents (Invariant Labs) demonstrates critical urgency. CSA now recommends MAESTRO threat models for agentic deployments explicitly include MCP server trust boundaries with assigned scenarios for STDIO injection, tool poisoning, and rug pull.

CASE STUDY 10 March / May 2026

"Living Off the Agent" (LOTA) — AI agents as lateral movement vectors

CSA AI Safety Initiative · [CSA Labs, 2026c]

SYSTEM Enterprise AI agents with authenticated connections to shared memory, MCP tool registries, and inter-agent channels.

LAYERS ANALYZED L3·L4·L7.

Threats identified

Threat	Layer	Risk
LOTA: repurposing authenticated connections via prompt injection	L7	CRITICAL
Shared memory pivot — poisoning all agents reading from the store	L2/L7	HIGH
MCP tool registry abuse — malicious metadata propagation	L3/L7	HIGH
Seven-stage kill chain: access → escalation → recon → persistence → C2 → lateral → objective	Cross-layer	CRITICAL
Morris II worm: self-replicating prompts in RAG email environments	L2/L7	HIGH

Mitigations applied

- L4 microsegmentation preventing container bridges to vector DBs and secret stores.
- L7 static (not runtime) trust boundary definitions; enforced A2A authentication.
- Behavioral identity: continuous intent verification against declared agent persona.
- Immutable audit logs and agent gateways enforcing Zero Trust.

Outcomes & lessons

Cross-agent propagation rose from 0% of documented promptware incidents (2023) to 38% (8 of 21 incidents, 2025–26). By 2026, 15 of 21 incidents exhibited 4+ kill-chain stages, with lateral movement in 8. MAESTRO L4 and L7 provide the most relevant vocabulary for LOTA defense design.

CASE STUDY 11 February 10, 2026

Snyk Labs — layered threat modeling guidance

Snyk Labs · [Snyk Labs, 2026]

SYSTEM General agentic AI ecosystems using vector DBs, RAG pipelines, and LangGraph / AutoGen / Claude-based agent frameworks.

LAYERS ANALYZED All 7 — emphasis on L2, L3, L4, L7.

Threats identified

Threat	Layer	Risk
Vector DB unauthorized access and query manipulation	L2	HIGH
Workflow integrity violations and execution path manipulation	L3	HIGH
Container/host attacks against the agent runtime	L4	HIGH
Inter-agent authentication gaps	L7	HIGH

Mitigations applied

- L1 model robustness training; L2 vector DB access controls + query validation.
- L3 workflow integrity verification + execution sandboxing.
- L4 container hardening + network segmentation; L5 behavioral monitoring + anomaly detection.
- L6 dynamic policy enforcement; L7 agent authentication + communication encryption.
- Multi-layer anomaly correlation for sophisticated cross-layer attacks.

Outcomes & lessons

Snyk Labs positions MAESTRO as the operational checklist for developers integrating OWASP Agentic Security Initiative (ASI) taxonomy with architectural layer analysis: MAESTRO provides the layer structure, OWASP ASI provides the threat taxonomy. The two are complementary, not competing.

CASE STUDY 12 2025

Enterprise agentic AI lifecycle governance

Sandeep Kumar Anuguthala, IJAIDSML · [Anuguthala, 2025]

SYSTEM Enterprise agentic AI across full lifecycle — planning, design, development, deployment, monitoring, decommissioning.

LAYERS ANALYZED All 7 — MAESTRO integrated with NIST AI RMF + MITRE ATLAS.

Threats identified

Threat	Layer	Risk
Inherent agentic risk classification gaps	Governance	HIGH
Control drift between design and production	L5/L6	HIGH
Compliance violations from inadequate threat modeling	L6	HIGH
Insufficient decommissioning controls (data + key residue)	L2/L6	MEDIUM

Mitigations applied

- Inherent risk assessment + risk tiering at design.
- Continuous control validation across lifecycle phases.
- MAESTRO for agent-specific threat modeling; MITRE ATLAS for technique taxonomy; NIST AI RMF for governance.
- Formal handoff gates between lifecycle stages anchored to MAESTRO layers.

Outcomes & lessons

MAESTRO + NIST AI RMF + MITRE ATLAS forms the emerging governance triad for enterprise agentic AI: MAESTRO handles agentic threat modeling, NIST provides risk management structure, ATLAS supplies adversarial ML technique vocabulary. Together they deliver lifecycle-wide control coverage that no single framework provides.

4 · Per-layer threat catalog

Canonical threats and mitigations distilled from the CSA primary publication and the subsequent application articles. Threat IDs are normalized to L[n]-T[n] for cross-study consistency.

AT A GLANCE · 7 layers · 60+ threats cataloged · 40+ canonical mitigations

L1 · Foundation Models

ID	Threat	Description
L1-T1	Adversarial Examples / Evasion	Crafted inputs fool the model into incorrect predictions.
L1-T2	Model Stealing / Extraction	API queries reconstruct model behavior — IP theft.
L1-T3	Backdoor Attacks	Hidden triggers in weights cause malicious behavior.
L1-T4	Membership Inference	Determine whether specific data was in the training set.
L1-T5	Data Poisoning (Training)	Malicious training data corrupts model behavior.
L1-T6	Reprogramming Attacks	Repurposing the model for an unintended malicious task.
L1-T7	DoS / Sponge Attacks	Expensive adversarial queries exhaust resources.
L1-T8	Prompt Leaking / Extraction	Tricking the model into revealing the system prompt.

Canonical mitigations

- Adversarial training; differential privacy in training and inference.
- Rate limiting + anomaly detection on suspicious query patterns.
- Rigorous training data validation, sanitization, and provenance.
- Input validation + output verification (Moderations API, ensembles).
- Model provenance verification and system-fingerprint tracking.

L2 · Data Operations

ID	Threat	Description
L2-T1	Data Poisoning	Training/operational data tampering corrupts agent behavior.
L2-T2	Data Exfiltration	Theft of sensitive data from DBs, vector stores, or RAG.
L2-T3	DoS on Data Infrastructure	Disrupting data access required by agents.
L2-T4	Data Tampering	Modifying data in transit or at rest.
L2-T5	Compromised RAG Pipelines	Malicious code/data injected into RAG processing.
L2-T6	Vector Store Leakage	Embeddings exposing sensitive source documents.
L2-T7	SQL/NoSQL Injection via Tools	LLM-generated queries carrying attacker payloads.

Canonical mitigations

- Strong access controls and least privilege per agent task.
- End-to-end data lineage; signed datasets for regulatory reporting.
- Data integrity monitoring (checksums, digital signatures).
- Cryptographic provenance for RAG; authenticated-only ingestion.
- Parameterized queries — never f-string SQL from LLM output.
- Versioning and regular audits of vector store contents.

L3 · Agent Frameworks

ID	Threat	Description
L3-T1	Prompt Injection	Natural language overriding system instructions.
L3-T2	Tool Misuse	Tools used in unintended or harmful ways.
L3-T3	Compromised Framework Components	Malicious code in libraries/modules.
L3-T4	Framework Backdoors	Hidden vulnerabilities enabling unauthorized access.
L3-T5	Supply Chain Attacks	Targeting framework dependencies.
L3-T6	Input Validation Failures	Code injection via weak framework input handling.
L3-T7	Framework Evasion	Agents bypassing in-framework security controls.
L3-T8	Resource Exhaustion	Tool loops or complex queries draining compute/budget.
L3-T9	State Manipulation	Tampering with conversation state references.
L3-T10	Unauthorized Tool Access	Bypassing tool access controls.

Canonical mitigations

- Architectural separation of user input from system instructions.
- Instruction hardening and output validation before tool execution.
- Per-agent RBAC tied to tool type; tool allowlists.
- Execution sandboxing; scoped, time-bounded tokens.
- max_completion_tokens + max_prompt_tokens + loop detection.
- SBOM tracking + automated dependency scanning.
- Human-in-the-loop for high-risk tool calls (deletion, transfers).

L4 · Deployment & Infrastructure

ID	Threat	Description
L4-T1	Compromised Container Images	Malicious code in agent OCI images.
L4-T2	Orchestration Attacks	Exploiting Kubernetes/orchestrators for access.
L4-T3	IaC Manipulation	Tampering with Terraform/CloudFormation.
L4-T4	Denial of Service	Overwhelming infrastructure for agents.
L4-T5	Resource Hijacking	Compromised infra used for cryptomining, etc.
L4-T6	Lateral Movement	Pivoting from one container to other resources.
L4-T7	Information Disclosure	Env vars and secrets leaked via tool calls.
L4-T8	Privilege Escalation / Sandbox Escape	Breaking out of operational context.

Canonical mitigations

- Container image signing + verification; hardened base images.
- IaC compliance scanning (PCI-DSS, SOC 2, HIPAA).
- Robust DDoS protection; adaptive rate limiting.
- Egress restrictions and network segmentation.
- No direct env-var exposure via agent-callable tools.
- Microsegmentation between agent containers and data stores.

L5 · Evaluation & Observability

ID	Threat	Description
L5-T1	Manipulated Evaluation Metrics	Poisoning benchmarks to favor malicious agents.
L5-T2	Compromised Observability Tools	Malicious code in monitoring systems.
L5-T3	DoS on Evaluation Infrastructure	Preventing proper detection.
L5-T4	Evasion of Detection	Agents designed to avoid triggering alerts.
L5-T5	Data Leakage via Observability	Sensitive data in logs/dashboards.
L5-T6	Observability Data Poisoning	Manipulating logs to mask activity.

Canonical mitigations

- Secure logging infrastructure with strong access controls.
- Log integrity monitoring (checksums, digital signatures).
- Anomaly detection accounting for agent non-determinism.
- Data minimization — avoid logging sensitive inputs/outputs in plaintext.
- Prompt injection test suites + drift monitoring in production.
- Isolated logging systems resistant to agent manipulation.

L6 · Security & Compliance (vertical)

ID	Threat	Description
L6-T1	Security Agent Data Poisoning	False positives / missed threats.
L6-T2	Evasion of Security AI Agents	Bypassing AI-based controls.
L6-T3	Compromised Security AI Agents	Disabling defenses from within.
L6-T4	Regulatory Non-Compliance	Misconfiguration violates privacy law.
L6-T5	Bias in Security AI Agents	Discriminatory security decisions.
L6-T6	Lack of Explainability	Opaque decisions preventing audit.
L6-T7	Model Extraction of Security Agents	Crafting effective bypasses.
L6-T8	Credential Compromise	Stolen keys / service accounts.
L6-T9	Compliance Violations	Processing PII/PHI/financial without safeguards.

Canonical mitigations

- Immutable audit trails for all agent actions and tool calls.
- Regulatory mapping: GDPR, HIPAA, PCI-DSS, Basel III, EU AI Act.
- DLP on all model inputs and outputs.
- MFA for credential access; HSMs; regular key rotation.
- Explainable AI (XAI) for security-critical decisions.
- Red-teaming the security agents themselves.

L7 · Agent Ecosystem

ID	Threat	Description
L7-T1	Compromised Agents	Malicious agents posing as legitimate services.
L7-T2	Agent Impersonation	Deceiving users or other agents.
L7-T3	Agent Identity Attack	Compromising AuthN/Z mechanisms.
L7-T4	Agent Tool Misuse	Tools used in harmful, unintended ways.
L7-T5	Agent Goal Manipulation	Diverting agent objectives.
L7-T6	Marketplace Manipulation	False ratings/reviews promoting malicious agents.
L7-T7	Integration Risks	Vulnerabilities in APIs/SDKs.
L7-T8	Repudiation	Agents denying actions they performed.
L7-T9	Compromised Agent Registry	Malicious listings in discovery systems.
L7-T10	Multi-Agent Collusion	Secret coordination toward malicious goals.
L7-T11	Sybil Attacks	Fake identities for disproportionate influence.
L7-T12	Pricing Model Manipulation	Exploiting agent pricing structures.

Canonical mitigations

- Secure inter-agent communication: mTLS, digital signatures, message integrity.
- DIDs for ecosystem-level identity; CT logs for agent cards.
- Behavioral reputation systems resistant to identity change.
- Agent registry with trusted authority governance.
- Sandboxing agents to limit blast radius.
- Circuit breakers in critical workflows (e.g., payments).
- Multi-agent red-teaming before ecosystem deployment.

Cross-layer threats

MAESTRO explicitly catalogs threats that span multiple layers, exploiting interactions between them. These are the most damaging in observed case studies because they bypass single-layer controls.

Threat	Layers	Example scenario
Supply chain attack	L3 → L7 → L1	Malicious library at L3 compromises all downstream agent behaviors including ecosystem interactions.
Lateral movement	L4 → L2 → L1	Container access at L4 → vector store poisoning at L2 → model behavior corruption at L1.
Privilege escalation	Any → adjacent	Agent gains admin permissions at L3, abuses them at L2.
Data leakage cascade	L2 → L5	Sensitive training data at L2 exposed through observability logs at L5.
Goal misalignment cascade	L2 → L7	Training data poisoning corrupts agent goals; propagates through multi-agent comms in L7.
Prompt injection kill chain	L1 → L3 → L4	User prompt → model selects malicious tool call → shell execution on host. Observed in 7/12 case studies.

5 · Comparative insights

Cross-case patterns drawn from the 12 case studies above: which layers are attacked most often, which mitigations recur, and where MAESTRO's current coverage still has gaps.

AT A GLANCE · layer attack frequency · recurring mitigations · coverage gaps

Layer attack frequency

Layer	Frequency	Primary attack vectors
L3 — Agent Frameworks	HIGHEST (12/12)	Prompt injection, tool misuse, supply chain — in every case study.
L2 — Data Operations	HIGH (11/12)	RAG/vector store poisoning, SQL injection via tools, exfiltration.
L7 — Agent Ecosystem	HIGH (10/12)	Agent impersonation, MCP tool poisoning, marketplace manipulation, LOTA.
L4 — Infrastructure	MEDIUM-HIGH (9/12)	Lateral movement, privilege escalation, container escape, credential leakage.
L1 — Foundation Models	MEDIUM (8/12)	Adversarial examples, backdoor triggers, prompt leaking.
L6 — Security & Compliance	MEDIUM (7/12)	Credential theft, compliance violations — the vertical layer that bypasses all others.
L5 — Observability	MEDIUM (6/12)	Log poisoning, detection evasion — often the enabling condition for other attacks.

Key structural finding

L3 is the most consistently exploited attack surface in this set of case studies — it is the orchestration hub linking the model (L1) to action execution (L4). Collapsing L1→L3→L4 into a single implicit trust domain is among the most commonly observed architectural failures, present in 7 of 12 case studies. This is the top fix.

Recurring mitigation patterns

Mitigation	Cited in	Layer focus
Input validation at every layer transition	11 / 12	L1↔L2↔L3↔L4

Mitigation	Cited in	Layer focus
Least-privilege tool access / allowlists	9 / 12	L3 + L7
Continuous logging + immutable audit trails	8 / 12	L5 + L6
Human-in-the-loop for high-risk calls	7 / 12	L3 + L7
Cryptographic provenance for RAG / memory	6 / 12	L2
mTLS + DID-based agent authentication	5 / 12	L6 + L7
MITRE ATLAS technique mapping	5 / 12	All layers
Container microsegmentation	4 / 12	L4

Coverage gaps

- Externally-operated LLM agents as post-exploitation tools on compromised hosts have no dedicated MAESTRO category or AICM control yet — flagged in CSA Labs' CISO Daily Briefing (June 2, 2026) [CSA Labs, 2026g]. A Marimo RCE attack is reported to have demonstrated an LLM autonomously conducting lateral movement, credential harvesting, and full exfiltration in under 2 minutes — per that briefing, not independently verified here.
- L5 (Observability) is the least operationalized layer. Despite being cataloged, monitoring for adversarial agent behavior remains largely theoretical in deployed systems.
- Behavioral identity — continuous intent verification against declared persona — is not yet a formal MAESTRO control. Identified by Cequence as the required third identity layer beyond cryptographic identity and authorization.
- Agentic Secure Development Lifecycle (ASDL) extends MAESTRO into full SDLC but is still v1 [CSA Labs, 2026e].

6 · Recommendations for securing multi-agent systems

Eight prioritized recommendations drawn from the case study evidence above. Each binds to specific MAESTRO layers and references the strongest supporting evidence. Priorities reflect frequency of exploitation × ease of mitigation.

AT A GLANCE · 8 recommendations · ranked by exploitation frequency × mitigation ease

01 Enforce explicit trust boundary validation at every MAESTRO layer transition

CRITICAL · Layers: L1 ↔ L3 ↔ L4

One of the most commonly observed architectural failures across the case studies is collapsing L1→L3→L4 into a single implicit trust domain. Add validation checkpoints at every boundary (L1→L2, L2→L3, L3→L4). Require explicit authorization before any LLM output becomes executable intent. Treat LLM outputs as untrusted inputs — never blindly execute model-generated tool calls. Run TITO [TITO repo, 2026] or an equivalent scanner against existing codebases to quantify current trust boundary gaps — record the tool version, ruleset, and run date alongside any results you cite (see the TITO evidence-record template in Appendix B).

Evidence	Owner & next action
7 / 12 case studies. A CSA-documented TITO scan reported 204 mitigation recommendations tied to this pattern — self-reported tool output, not independently reproduced here.	Platform / Agent Engineering — Run a trust-boundary scan against the top 3 production agents this quarter.

02 Apply least privilege to all agent tool access

CRITICAL · Layers: L3 + L7

Direct mitigation for L3-T2 (Tool Misuse) and L7-T4 — the two most frequently exploited threats. Create tool allowlists per agent role; agents see only tools relevant to their declared function. Use scoped, time-bounded tokens for sensitive operations. Eliminate overly broad tools (e.g., execute_sql() accepting arbitrary queries). Apply per-agent RBAC tied to transaction type and data classification.

Evidence	Owner & next action
9 / 12 case studies — recurring mitigation across CSA, IriusRisk, and Snyk Labs.	AppSec / Agent Platform Team — Publish a tool-allowlist template; require sign-off for any grant broader than single-purpose.

03 Protect RAG and memory with cryptographic integrity controls

HIGH · Layers: L2

RAG and agent memory are a primary persistence vector for compromised agents. CSA Labs [CSA Labs, 2026a] reports a 90% knowledge-base compromise rate from 5 poisoned documents, citing PoisonedRAG-style findings — verify against the original study before treating this as an established figure. Treat memory and RAG as security-critical data stores regardless. Populate KBs only from authenticated, validated sources with cryptographic provenance. Never permit agents to write directly to their persistent memory from untrusted content. Daily memory state verification and anomaly detection on retrieval patterns. Write-through integrity logging on all vector store modifications.

Evidence
Case Studies 5, 8, 10. 90% compromise figure is CSA-reported, pending primary-source verification.

Owner & next action
Data Security / ML Platform — Gate KB ingestion behind authenticated-source and provenance checks before the next RAG deployment.

04 Integrate MAESTRO threat modeling into CI/CD

HIGH · Layers: L3 + L4

Static threat models become stale immediately in agentic systems. Integrate TITO [TITO repo, 2026] or an equivalent MAESTRO-aware scanner into GitHub Actions / GitLab CI. Block PRs introducing critical agentic threats: tito scan --maestro --fail-on critical. Use threat diffing — show only deltas per PR to reduce alert fatigue. Require a MAESTRO threat model as a gating artifact in the design phase per ASDL methodology.

Evidence
Case Study 7 (TITO scanner reference deployment).

Owner & next action
DevSecOps / Platform Engineering — Add a MAESTRO-aware scan gate to the agent build pipeline; block merges on critical findings.

05

Apply Zero Trust to all MCP servers and A2A agent cards

HIGH · Layers: L7

CSA Labs [CSA Labs, 2026b] cites a 72.8% MCP tool-poisoning success rate against production agents, attributed to Invariant Labs (April 2025) — check the original Invariant Labs artifact for methodology and sample size before treating this as a general base rate. Treat MCP tool descriptions as application inputs subject to validation regardless. Hash-based baselines for expected tool call signatures; alert on deviation. Verify MCP server identity via cryptographic provenance before use. Micro-segmentation to detect and block suspicious invocations. For A2A: require mTLS + DID-based identity for all agent communications.

Evidence
Case Studies 2, 3, 6, 9. 72.8% figure is CSA-reported, attributed to Invariant Labs.

Owner & next action
Security Architecture — Require cryptographic provenance verification before any new MCP server is approved for use.

06

Deploy agent-specific observability with MAESTRO-aware telemetry

MEDIUM-HIGH · Layers: L5

Standard infrastructure monitoring is blind to agent-mediated attacks operating within authorized permissions (LOTA pattern). Capture full reasoning chain tracing — tools called, order, inputs, and the agent's stated rationale at each step. Behavioral baselines per agent role; alert on persona deviation. Runtime behavioral guardrails per L5 catalog. Immutable, tamper-resistant log stores — agents cannot modify their own audit logs. Anomaly detection accounting for non-determinism.

Evidence
Case Study 10 (LOTA pattern): CSA Labs' incident review reports 15/21 incidents with ≥4 kill-chain stages in 2026, per its stated corpus.

Owner & next action
SRE / Detection Engineering — Stand up reasoning-chain logging and behavioral baselines for at least one production agent this quarter.

07

Conduct MAESTRO-structured red team exercises

MEDIUM · Layers: All

Red teaming surfaces cross-layer chains that automated scanning misses. Use MAESTRO as the red team vocabulary: assign teams to specific layers (L2 data poisoning, L7 ecosystem). Explicitly test cross-layer chains: L1 prompt injection → L3 tool dispatch → L4 execution. Include indirect prompt injection via RAG content, emails, docs, and API responses. Use MITRE ATLAS as the technique taxonomy in reports. Document findings in MAESTRO notation for direct mapping to controls.

Evidence
Case Studies 8, 11, 12 — recommended by CSA, Snyk Labs, and IJAIDSML.

Owner & next action
Red Team / AppSec — Schedule the first MAESTRO-structured exercise, assigning a layer owner per team.

08

Align MAESTRO to regulatory and standards frameworks

MEDIUM · Layers: L6 + Governance

MAESTRO does not replace compliance — it informs control selection. Map L6 controls to GDPR (data minimization, right to erasure), HIPAA (PHI by diagnostic agents), PCI-DSS (payment agents), Basel III (financial models). Use MAESTRO as the threat model feeding NIST AI RMF risk assessments. Cross-reference against MITRE ATLAS for dual-standard coverage. For IriusRisk users, activate the MAESTRO library in v4.47+ for automated standards-mapped countermeasure generation.

Evidence
Case Studies 4, 12 — the emerging MAESTRO + NIST + ATLAS governance triad.

Owner & next action
GRC / Compliance — Map existing L6 controls to applicable regulations and flag gaps to legal/compliance.

7 · Implementation checklist

A condensed operational checklist suitable for issue trackers or design reviews. Order reflects sequencing — discovery before enforcement before monitoring.

AT A GLANCE · 5 phases · discovery → enforcement → pipeline → observability → governance

Discovery

- Inventory all agents, tool registries, MCP servers, and A2A endpoints.
- Map every component to its MAESTRO layer(s).
- Run a baseline TITO scan: tito scan --repo . --maestro --mitre --attack-paths.
- Document trust boundaries explicitly — every layer transition gets a name.

Enforcement

- Replace overly broad tools (e.g., execute_sql()) with constrained alternatives.
- Enforce per-agent RBAC and tool allowlists; scoped, expiring tokens for sensitive ops.
- Add human-in-the-loop confirmation for deletes, transfers, external comms.
- Parameterized queries only — no f-string SQL from LLM outputs.
- mTLS + DIDs for A2A; cryptographic provenance for MCP servers.
- Cryptographic integrity controls on RAG and memory stores.

Pipeline

- Integrate tito scan --maestro --fail-on critical into CI.
- Block PRs that introduce critical threats; surface diffs for non-blocking findings.
- Make MAESTRO threat models a design-phase gating artifact.
- SBOM tracking + dependency scanning for all agent framework libs.

Observability

- Capture full reasoning chains: tools, order, inputs, rationale.
- Behavioral baselines per agent role; alert on persona deviation.
- Immutable audit logs; agents cannot edit their own trail.
- Prompt-injection test suites and drift monitoring in production.

Governance

- Map L6 controls to GDPR, HIPAA, PCI-DSS, Basel III, EU AI Act.

Use NIST AI RMF as the governance wrapper; MITRE ATLAS for technique taxonomy.

Schedule MAESTRO-structured red team exercises quarterly.

Re-evaluate the threat model whenever tools, prompts, or integrations change.

8 · Annotated references & evidence notes

Claim-level citations in this report use compact author-year keys, e.g. [Huang, 2025a] — not raw inline URLs. Each key resolves below, grouped by evidence type so readers can judge evidentiary weight at a glance: framework specifications, research preprints, first-party implementation guidance, CSA Labs practitioner research notes, and vendor/governance sources. arXiv items are labeled preprint throughout — none is described as peer-reviewed unless a venue record is independently confirmed.

Framework and standards

[Huang, 2025a] Huang, K. (2025, Feb 6). Agentic AI threat modeling framework: MAESTRO. Cloud Security Alliance.

First-party framework specification

Canonical public introduction to MAESTRO, defining its layer-specific analysis, cross-layer threat identification, risk assessment, and mitigation planning. Use as the authoritative citation for MAESTRO's purpose and methodology — not as independent evidence that individual controls work in production.

[CSA Labs, MAESTRO Hub] Cloud Security Alliance Labs. MAESTRO reference hub.

First-party reference hub

Living index of MAESTRO materials; useful as a pointer, not a dated primary source in its own right.

[MITRE ATLAS] MITRE. (n.d.). MITRE ATLAS: Adversarial Threat Landscape for Artificial-Intelligence Systems.

Maintained adversarial-ML knowledge base

Technique taxonomy referenced throughout this report alongside MAESTRO's architectural layers; complements but does not substitute for system-specific trust-boundary analysis.

[NIST AI RMF] National Institute of Standards and Technology. (2023). Artificial Intelligence Risk Management Framework (AI RMF 1.0). NIST AI 100-1.

U.S. government risk-management standard

Governance wrapper for this report's lifecycle and accountability recommendations — a risk-management framework, not a technical threat catalog.

Direct MAESTRO applications (preprints)

[Habler et al., 2025 (preprint)] Habler, I., Huang, K., Narajala, P., & Kulkarni, A. (2025). Building a secure agentic AI application leveraging A2A protocol. arXiv:2504.16902.

Research preprint

Applies MAESTRO to A2A deployments — Agent Card management, task-execution integrity, authentication. Cite as a preprint; do not describe as peer-reviewed without a verified venue publication.

[Zambare et al., 2025 (preprint)] Zambare, P., Thanikella, V. N., & Liu, Y. (2025). Securing agentic AI: Threat modeling and risk analysis for a network-monitoring agentic AI system. arXiv:2508.10043.

Research preprint

Applies MAESTRO's seven-layer model to a network-monitoring prototype. Supports that MAESTRO can be applied to operational agent architectures; does not alone establish field-wide effectiveness.

[arXiv:2508.09815, 2025 (preprint)] OWASP multi-agent-system threat-modeling extension. arXiv:2508.09815.

Research preprint

Extends threat-modeling coverage for multi-agent systems; referenced here as supplementary preprint context, not independently verified beyond the abstract.

[Huang & Habler, 2025] Huang, K., & Habler, I. (2025, Apr 30). Threat modeling Google's A2A protocol with the MAESTRO framework. Cloud Security Alliance.

First-party practitioner analysis

Practical MAESTRO walkthrough of the A2A protocol. Use for implementation guidance and illustrative scenarios; pair with the Habler et al. preprint for stronger academic grounding.

Tooling and implementation

[Huang & Chen, 2025] Cloud Security Alliance. (2025, Jun 23). A primer on Model Context Protocol (MCP) secure implementation.

First-party practitioner guidance

Implementation-oriented source for MCP server hardening — parameterized access, constrained tools, session integrity, least privilege. Cite specific recommendations precisely; do not generalize its examples into prevalence statistics.

[IriusRisk, 2025] IriusRisk. (2025, Oct 2). Product release 4.47.

Vendor release documentation

Supports the narrow claim that IriusRisk shipped a MAESTRO-related library in this release. Appropriate for product capabilities, not for adoption or efficacy claims.

[IriusRisk, 2026] IriusRisk. (2026, Jan 21). MAESTRO threat modeling.

Vendor blog

Follow-up vendor post on the same integration; same evidentiary caveats as the 2025 release note.

[Snyk Labs, 2026] Snyk Labs. (2026, Feb 10). MAESTRO threat modeling.

Vendor security guidance

Practitioner cross-check for layered threat modeling. Treat as expert guidance, not empirical evidence — supports control patterns, not numeric effectiveness claims.

[TITO repo, 2026] Leath, S., & Huang, K. (2026). TITO: Threat-modeling and CI/CD scanner for agentic systems [Source code repository]. GitHub.

Open-source implementation artifact

Source for scanner installation, commands, and version-specific behavior. Any scan counts cited from this report (assets, data flows, threats, mitigations) should be treated as self-reported tool output — see the evidence-record template in §8.3 before citing them elsewhere.

Supplementary CSA Labs research notes

[Leath & Huang, 2026] Cloud Security Alliance. (2026, Feb 11). Applying MAESTRO to real-world agentic AI threat models: From framework to CI/CD pipeline.

First-party implementation guidance

Supports treating threat modeling as a CI/CD artifact. Pair with a versioned TITO evidence record whenever citing its scan results elsewhere.

[CSA Labs, 2026a] Cloud Security Alliance AI Safety Initiative. (2026, Mar 25). AI-agent insider threat: Autonomous compromise.

Practitioner research note

Threat framing for agents with broad enterprise permissions, including memory-based persistence. Numerical claims (e.g. RAG-compromise rate) should link to the original experiment before being treated as established.

[CSA Labs, 2026b] Cloud Security Alliance AI Safety Initiative. (2026, May 4). MCP security crisis [Research note].

Practitioner research note

Synthesizes MCP transport, tool-description, and registry risk. Use as current threat context; cite the underlying Invariant Labs study directly for success-rate claims and methodology.

[CSA Labs, 2026c] Cloud Security Alliance AI Safety Initiative. (2026, May 19). Living off the agent (LOTA): Lateral movement [Research note].

Practitioner research note

Articulates agent-mediated lateral movement and shared-memory pivots. Incident-frequency claims are attributed to CSA's own corpus and retained here only as CSA's reported analysis.

[CSA Labs, 2026d] Cloud Security Alliance. (2026, Mar 11). AI-induced lateral movement [Research note].

Practitioner research note

Earlier note in the same lateral-movement research line as CSA Labs, 2026c; same evidentiary caveats apply.

[CSA Labs, 2026e] Cloud Security Alliance Labs. (2026, Apr 2). Agentic Secure Development Lifecycle (ASDL) v1.

First-party methodology (v1)

Extends MAESTRO into a full SDLC methodology; explicitly a v1, cited here as an emerging rather than settled practice.

[CSA Labs, 2026g] Cloud Security Alliance Labs. (2026, Jun 2). CISO daily briefing.

Practitioner briefing note

Source for the Marimo RCE coverage-gap item in §5; incident details are per this briefing and not independently verified here.

[CSA Labs, 2026h] Cloud Security Alliance Labs. (2026, Apr 2). Agentic master framework alignment matrix v1.

First-party reference matrix

Cross-framework alignment reference (MAESTRO / NIST / ATLAS); background context for §6, recommendation 8.

Governance and ecosystem context

[CSA Labs, 2026f] Cloud Security Alliance Labs. (2026, May). Five Eyes agentic AI guidance analysis [Whitepaper].

Practitioner summary of government guidance

Useful context relating MAESTRO's layers to government-scoped risk categories. Cite the original Five Eyes/CISA guidance for normative recommendations; use this source for its MAESTRO-specific interpretation only.

[CSA, 2026 (ORCHIDEAS)] Cloud Security Alliance. (2026, Jun 5). Designing agentic AI systems with the ORCHIDEAS framework.

First-party guidance

Adjacent design-framework post; referenced for context, not relied on for quantitative claims in this report.

[CSO Online, 2025] CSO Online. (2025, Oct 15). Introducing MAESTRO: A framework for securing generative and agentic AI.

Trade press

General-audience trade coverage of MAESTRO; background only, not a primary or evaluative source.

[Cequence, 2026] Cequence. (2026, May). Agentic Zero Trust research [Whitepaper].

Vendor whitepaper

Source for the 'behavioral identity as a third identity layer' framing in §5's coverage gaps; vendor-authored, not independently verified here.

[AIGL, 2025] AIGL. (2025, Apr). Agentic AI MAS threat modelling guide v1 [Whitepaper].

Practitioner guide

Independent practitioner guide covering similar ground to MAESTRO; referenced for cross-context, not cited for specific claims in this report.

[Anuguthala, 2025] Anuguthala, S. K. (2025). Enterprise agentic AI lifecycle governance. IJAIDSML.

Journal article

Basis for Case Study 12's lifecycle-governance framing (MAESTRO + NIST AI RMF + MITRE ATLAS). Venue's peer-review status not independently confirmed in this report — treat as a published journal article pending that check.

8.2 · Evidence ledger — flagged quantitative claims

High-impact numeric claims in this report, their sourcing status, and the editorial decision applied. "CSA-reported" means the figure traces to a CSA Labs research note rather than the original primary study; treat it as CSA's reported analysis unless you independently verify the underlying source.

Claim	Status	Editorial decision
CSA MAESTRO framework, Feb 2025	Verified, first-party	Retained as canonical framework source.
A2A threat-modeling paper, arXiv:2504.16902	Verified, primary preprint	Retained; labeled preprint, not peer-reviewed.
Network-monitoring paper, arXiv:2508.10043	Verified, primary preprint	Retained; labeled preprint, venue unconfirmed.
CSA article on Google A2A + MAESTRO	Verified, first-party	Retained as implementation case study, not independent validation.
“First threat-modeling framework purpose-built for agentic AI”	Overstated	Reworded to “CSA presents MAESTRO as…”
“Rapidly became the lingua franca”	Unsupported / promotional	Replaced with “adopted in published CSA guidance and commercial tooling.”
“First empirical validation” (Case Study 5)	Overstated	Reworded to “an empirical prototype study.”
“First major commercial platform” (Case Study 4)	Needs proof	Attributed explicitly to IriusRisk; “first major” removed.
5 docs → 90% RAG compromise	Needs primary-study citation	Attributed to CSA Labs, flagged pending the original PoisonedRAG-line study.
72.8% MCP tool-poisoning success	Needs original Invariant Labs citation	Attributed to CSA Labs' citation of Invariant Labs; original methodology not independently verified.
15/21 incidents, 8 lateral-movement, 38% propagation	Needs transparent methodology	Framed as CSA Labs' reported analysis per its stated corpus.
TITO: 9 threats, 204 mitigations, 24 assets, 169 flows	Self-reported / reproducibility required	Framed as self-reported scanner output; evidence-record template added (§8.3).

8.3 · TITO evidence-record template

Any TITO (or equivalent MAESTRO-aware scanner) result cited as evidence — in this report or in downstream work — should carry a record in this shape rather than a bare headline number. The fields below are a fill-in template, not a completed record for this report's own claims.

```
TITO evidence record
Repository: <public URL or commit hash>
TITO version: <release / tag>
Ruleset version: <hash>
Scan command: tito scan --repo . --maestro --mitre --attack-paths
Configuration hash: <sha256>
Run date: <ISO 8601>
Result artifact: <JSON/SARIF URL or checksum>
```

Compiled June 2026 by Mazze LeCzzare Frazer. Where information is drawn from secondary summaries, this is noted above rather than presented as direct observation. Threat IDs are normalized to L[n]-T[n] notation; original sources sometimes use T[layer].[number] — the mapping is direct.