A FIELD STUDY OF AGENTIC AI THREAT MODELING

MAESTRO in Practice

Threats, mitigations, and outcomes across 12 case studies

Twelve documented applications of the seven-layer threat-modeling framework for agentic AI — what actually broke, what fixed it, and how confidently each claim is sourced.

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

Built for a threat model that didn’t exist yet

STRIDE, PASTA, and LINDDUN all predate agentic AI. None has a native concept of a system that calls tools, holds memory across sessions, or negotiates with other agents on your behalf. MAESTRO’s answer is a seven-layer decomposition, plus explicit modeling of the threats that cross layer boundaries.

L1	Foundation Models	Weights, training, inference, alignment
L2	Data Operations	Databases, vector stores, RAG pipelines
L3	Agent Frameworks	Orchestrators, tool dispatch
L4	Deployment & Infrastructure	Containers, IaC, cloud, networks
L5	Evaluation & Observability	Logging, metrics, drift monitoring
L6	Security & Compliance	AuthN/Z, audit trails, regulation
L7	Agent Ecosystem	Registries, A2A, MCP, multi-agent

LAYER ATTACK FREQUENCY

12/12

CASE STUDIES SHOW L3 EXPLOITED

Agent Frameworks is the layer every attack passes through

L3 sits between the model (L1) and action execution (L4) — the orchestration hub. The failure that recurs isn't one vulnerability class, it's an architectural habit: collapsing L1→L3→L4 into a single implicit trust domain. Seven of twelve case studies show that exact chain.

Memory and RAG are a persistence vector, not just a knowledge store

Three case studies treat memory poisoning as how a single compromise survives past the session that created it — agents that write to their own memory from untrusted content carry that compromise forward indefinitely.

90% knowledge-base compromise rate from just five poisoned documents, per CSA Labs' citation of PoisonedRAG-style findings.

SOURCING: CSA-REPORTED, VERIFY PRIMARY STUDY

MCP servers can’t be trusted by source alone

Tool-poisoning and STDIO-injection findings recur across four case studies. The lesson is one web security learned decades ago: anything the model can be induced to call is an input, not a trusted internal surface.

72.8% tool-poisoning success rate against production agents, attributed to Invariant Labs’ Unicode-injection research.

SOURCING: ATTRIBUTED, METHODOLOGY UNVERIFIED HERE

Agentic kill chains are real, and growing

0% → 38%

CROSS-AGENT PROPAGATION, 2023 TO 2025–26

The “Living Off the Agent” pattern: an agent’s own authenticated connections become the attacker’s lateral-movement infrastructure — invisible to tooling that only watches for unauthorized access.

Per CSA Labs’ incident review: **15 of 21** documented promptware incidents in 2026 showed four or more kill-chain stages, with lateral movement in **8**.

CRITICAL · L1 ↔ L3 ↔ L4

Enforce trust-boundary validation at every layer transition

- Add a validation checkpoint at every layer boundary: L1 → L2, L2 → L3, L3 → L4.
- Require explicit authorization before any model output becomes executable intent.
- Treat LLM output as an untrusted input — never execute a model-generated tool call directly.

Recommendation 1 of 8 in the full report

Every headline number is graded by how confidently it’s sourced

The striking stats in this space travel well on their own. Most are someone else’s research, reframed. The report carries an explicit evidence ledger so you know how far back to go before citing one.

CSA MAESTRO framework, Feb 2025	VERIFIED
5 docs → 90% RAG compromise	CSA-REPORTED
72.8% MCP tool-poisoning success	NEEDS PRIMARY SOURCE

Read all 12 case studies

42 pages: every case study deep dive, the per-layer threat catalog, all eight recommendations, the implementation checklist, and the complete evidence ledger.

FULL POST & PDF

mazzeleczzare.com/blog/maestro-in-practice

Link in the post · comments if LinkedIn strips it