Security Threats in the Model Context Protocol: A Comprehensive Survey and Trust Boundary Mitigation Framework for Agentic AI Systems

Sanjeeva Reddy Bora[1], Dr Srinivas Kishan Anapu[2]

Abstract: ***The Model Context Protocol (MCP), open-sourced by Anthropic in November 2024 and donated to the Linux Foundation's Agentic AI Foundation in December 2025, has rapidly emerged as the dominant interoperability standard for connecting AI agents to external tools, data sources, and enterprise systems. With over 97 million monthly SDK downloads and 10,000+ active servers by late 2025, MCP's adoption has dramatically outpaced its security maturity. This paper presents the first comprehensive academic survey of MCP security threats, synthesizing findings from seven disclosed CVEs, eleven major security incidents, and over a dozen demonstrated attack classes documented between April and December 2025. We propose a formal ten-class threat taxonomy spanning tool poisoning, indirect prompt injection via tool responses, cross-server data exfiltration, tool shadowing, supply-chain attacks, rug-pull exploits, credential theft, sampling abuse, terminal deception, and inter-agent trust exploitation. We map these threats against four emerging governance frameworks (OWASP Top 10 for Agentic Applications 2026, MITRE ATLAS, NIST IR 8596, and CSA MAESTRO), analyze domain-specific risk amplification in healthcare, financial services, and enterprise IT, and propose a defense-in-depth Trust Boundary Mitigation Framework (TBMF) combining MCP gateways, zero-trust identity, capability-based least privilege via OAuth scopes, runtime behavioral monitoring, and human-in-the-loop governance. Our analysis reveals that more capable models are paradoxically more susceptible to tool poisoning attacks (72.8% success rate on o1-mini), that 100% of tested LLMs execute malicious commands from peer agents, and that all 41 surveyed defense papers focus exclusively on integrity with zero availability protections- identifying critical research gaps for the community.***

Keywords: Model Context Protocol; MCP security; agentic AI; tool poisoning; prompt injection; trust boundaries; zero-trust architecture; AI governance; supply-chain security; large language models

1. Introduction

The transition from conversational AI to agentic AI- systems that autonomously discover, select, and execute tools to accomplish complex tasks- represents a fundamental paradigm shift in how artificial intelligence interfaces with enterprise systems. At the center of this transformation lies the Model Context Protocol (MCP), an open standard that provides a unified mechanism for AI agents to connect with external tools, data sources, and workflows [1]. Originally developed by Anthropic and released as an open-source specification in November 2024, MCP was donated to the Linux Foundation's Agentic AI Foundation in December 2025, cementing its position as the de facto interoperability layer for agentic AI [2].

The scale of MCP adoption is extraordinary. By the end of 2025, the protocol's TypeScript and Python SDKs had accumulated over 97 million monthly downloads, with more than 10,000 publicly available MCP servers registered across multiple ecosystems [3]. Major AI platforms including Claude, ChatGPT, Google Gemini, Microsoft Copilot, Cursor, VS Code, and GitHub Copilot have integrated MCP as their primary tool-use interface. This rapid adoption, however, has created a significant security deficit: the protocol's security mechanisms have consistently lagged behind its functional capabilities.

The security implications of MCP extend far beyond traditional API security concerns. When an AI agent operates under MCP, it dynamically discovers tools described in natural language, interprets those descriptions using a probabilistic language model, selects and sequences tool invocations at runtime, and acts on tool responses that may contain adversarial content. This creates what we term a *probabilistic execution surface*—an attack surface where the decision-making consumer is itself manipulable through the very data it processes. A December 2025 scan revealed approximately 1,000 MCP servers exposed on the public internet with no authentication whatsoever [4], while independent security assessments found command injection vulnerabilities in 43% of tested MCP implementations [5].

Despite these risks, the academic literature on MCP security remains nascent. While foundational work on prompt injection [6], tool-use safety [7], and LLM agent security [8] has established important theoretical groundwork, no comprehensive survey has synthesized the rapidly accumulating body of MCP-specific threat research, mapped it against emerging governance frameworks, analyzed domain-specific risk amplification, and proposed a unified mitigation architecture.

This paper makes four contributions. First, we present a formal ten-class threat taxonomy for MCP, grounded in demonstrated attacks rather than theoretical vulnerabilities (Section 3). Second, we provide the first systematic mapping of MCP threats to four major governance frameworks released in 2025–2026 (Section 5). Third, we analyze domain-specific threat amplification in healthcare, financial services, and enterprise IT environments (Section 6). Fourth, we propose a Trust Boundary Mitigation Framework (TBMF) that combines six reinforcing defense layers into a deployable architecture (Section 7). We conclude by identifying six key insights and open research challenges for the community (Section 8).

2. Background: MCP Architecture and Security Properties

2.1 Protocol Architecture

MCP implements a client-host-server architecture built on JSON-RPC 2.0 [1]. The *host* application (e.g., an IDE or AI assistant) manages multiple *MCP clients*, each maintaining a stateful one-to-one session with an *MCP server* that exposes three primitive types: **tools** (executable functions), **resources** (data endpoints), and **prompts** (templated interaction patterns). The protocol specification has evolved through four versions (2024-11-05, 2025-03-26, 2025-06-18, and 2025-11-25), progressively adding OAuth 2.1 authorization, Streamable HTTP transport, structured tool output, elicitation, and a tasks primitive for asynchronous operations [9].

Tool discovery occurs via the *tools/list* endpoint, which returns tool names, natural language descriptions, and JSON Schema parameter definitions. The LLM consumes these descriptions to decide which tools to invoke, what parameters to pass, and how to interpret responses. This design is intentional—it enables flexible, context-sensitive tool use without requiring hard-coded integrations- but it also introduces the fundamental security tension that this paper examines.

2.2 Architectural Properties Creating MCP’s Security Profile

Three properties distinguish MCP’s attack surface from traditional API architectures. First, **dynamic tool discovery** means the attack surface shifts between sessions—tools can appear, disappear, or change behavior without code deployment. Second, **natural language tool descriptions** are consumed directly by the LLM, blurring the boundary between data and control instructions- the same conflation that enables prompt injection [6]. Third, **multi-server composition** within a single host means all tool descriptions from all connected servers share the LLM’s context window, enabling cross-server attacks where a malicious server manipulates the agent’s interaction with legitimate servers.

The MCP specification acknowledges these risks, stating that “descriptions of tool behavior such as annotations should be considered untrusted, unless obtained from a trusted server” [9], yet the protocol provides no mechanism to enforce this at the transport level. Authentication was entirely absent in the initial specification and remains optional in current deployments.

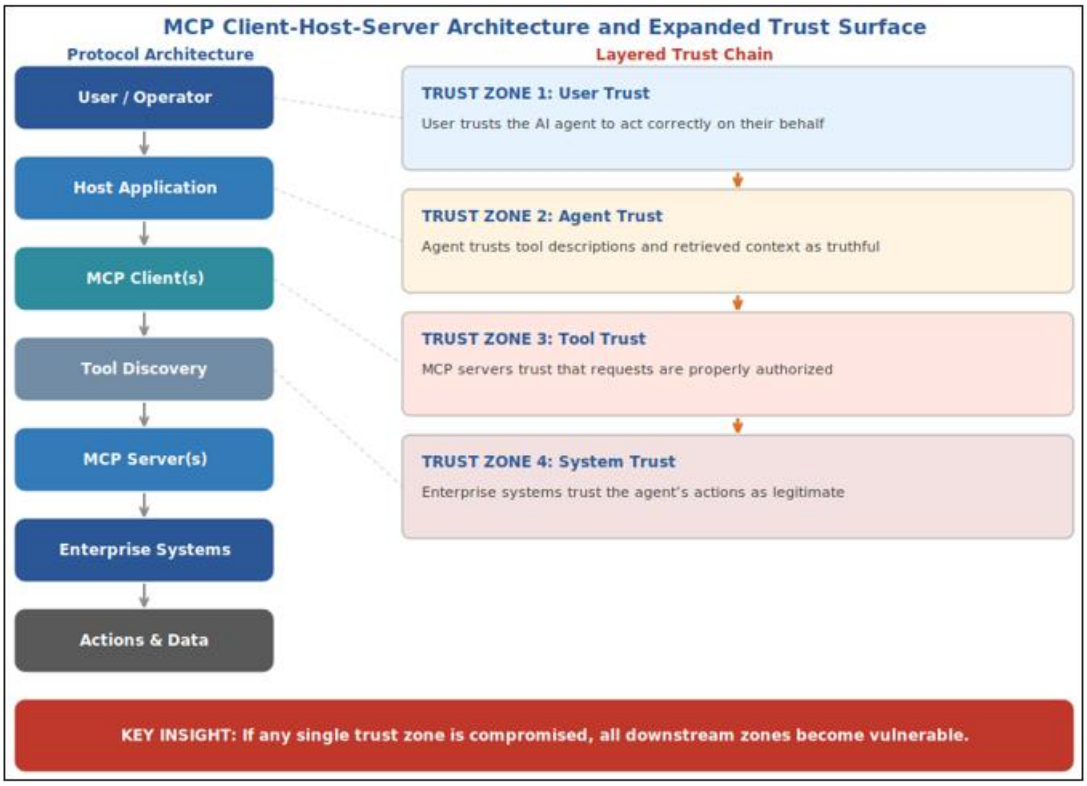

MCP Client-Host-Server Architecture and Expanded Trust Surface
Protocol Architecture
User / Operator
Host Application
MCP Client(s)
Tool Discovery
MCP Server(s)
Enterprise Systems
Actions & Data
Layered Trust Chain
TRUST ZONE 1: User Trust
User trusts the AI agent to act correctly on their behalf
TRUST ZONE 2: Agent Trust
Agent trusts tool descriptions and retrieved context as truthful
TRUST ZONE 3: Tool Trust
MCP servers trust that requests are properly authorized
TRUST ZONE 4: System Trust
Enterprise systems trust the agent’s actions as legitimate
KEY INSIGHT: If any single trust zone is compromised, all downstream zones become vulnerable.

Figure 1: MCP client-host-server architecture and the layered trust chain. Each trust zone depends on the integrity of all upstream zones; compromise at any layer propagates downward.

2.3 Comparison with Traditional API Security Models

Traditional API security operates on fundamentally different assumptions. REST and gRPC consumers are deterministic programs that follow pre-defined logic with static endpoint catalogs, schema-validated messages, and code-level access control enforcement. MCP replaces all three properties: deterministic consumers become probabilistic LLMs, static catalogs become dynamically discovered tools, and schema validation is supplemented by natural language descriptions that the LLM interprets subjectively. The emerging MCP gateway ecosystem is essentially reconstructing enterprise service bus (ESB) governance for the AI era [10], but with additional requirements for prompt-level inspection, behavioral analytics, and natural language policy enforcement that have no precedent in traditional enterprise integration.

3. A Ten-Class Threat Taxonomy

Research from Invariant Labs, Trail of Bits, CyberArk, Palo Alto Networks Unit 42, JFrog, and academic teams has established a comprehensive threat taxonomy. What distinguishes MCP threats from traditional API security is that every attack class has been demonstrated against production-grade implementations. We organize these into ten primary classes, summarized in Table 1.

Table 1: Ten-class MCP threat taxonomy with demonstrated attack vectors

Threat Class	Description	Key Evidence
T1: Tool Poisoning	Malicious instructions in tool descriptions direct agent to exfiltrate data or perform unauthorized actions	MCPTox: 72.8% success on o1-mini [11]
T2: Indirect Prompt Injection	Adversarial instructions embedded in tool responses, documents, databases, or retrieved content	GitHub MCP exploit, May 2025 [12]
T3: Cross-Server Exfiltration	Malicious server contaminates shared context to redirect legitimate server actions	WhatsApp MCP exploit, Apr 2025 [13]
T4: Tool Shadowing	Rogue tool description overrides or redirects behavior of tools from other servers	MCPLib: 31 methods, >70% success [14]
T5: Supply-Chain Attacks	Compromised MCP packages, servers, or registries introduce trusted backdoors	CVE-2025-6514, CVSS 9.6 [15]
T6: Rug-Pull Attacks	Benign tool definitions changed to malicious versions post-approval	Invariant Labs disclosure [13]
T7: Credential Theft	Plaintext API keys/tokens extracted from MCP server configurations	Trail of Bits audit [16]
T8: Sampling Exploitation	MCP sampling primitive abused for resource theft and conversation hijacking	Unit 42 research [17]
T9: Terminal Deception	ANSI escape sequences hide malicious instructions from user while LLM processes them	Trail of Bits disclosure [16]
T10: Inter-Agent Trust	100% of tested LLMs execute malicious commands from peer agents that they resist from humans	Multi-agent study, Jul 2025 [18]

3.1 T1: Tool Poisoning- The Foundational MCP-Specific Attack

Tool poisoning exploits the fact that LLMs treat tool descriptions as trusted instructions. Malicious content embedded in tool descriptions- invisible to users viewing tool listings but fully processed by the LLM- can direct agents to exfiltrate SSH keys, configuration files, or credentials. CyberArk extended this to *Full-Schema Poisoning*, demonstrating that payloads can be injected into any part of the JSON schema including parameter descriptions, default values, and enum values [19]. They further demonstrated *Advanced Tool Poisoning* exploiting tool outputs rather than descriptions alone. The MCPTox benchmark evaluated 45 real-world MCP servers across 20 LLM agents, finding that more capable models are *more* susceptible: Claude-3.7-Sonnet refused attacks less than 3% of the time, and o1-mini exhibited a 72.8% attack success rate [11]. This inverts the conventional security assumption that more capable systems are more secure.

3.2 T2–T3: Prompt Injection and Cross-Server Exfiltration

Indirect prompt injection via tool responses represents the natural extension of Greshake et al.'s foundational work [6] into the MCP context. Demonstrated incidents include a GitHub MCP exploit (May 2025) where a crafted GitHub issue caused an agent to access private repositories and exfiltrate data through a public pull request [12]; a Supabase/Cursor exploit (June 2025) where a support ticket containing malicious SQL was executed by an agent with service-role database access; and zero-click attacks through Jira and Google Docs MCP servers where malicious content auto-executed without user interaction [5].

Cross-server data exfiltration exploits the shared context window. In the WhatsApp MCP exploit (April 2025), a malicious "random fact of the day" server injected instructions that caused the agent to use a legitimate WhatsApp server's *send_message* tool to exfiltrate the user's entire chat history to an attacker-controlled phone number [13]. The malicious server was never called directly- the attack operated purely through context contamination.

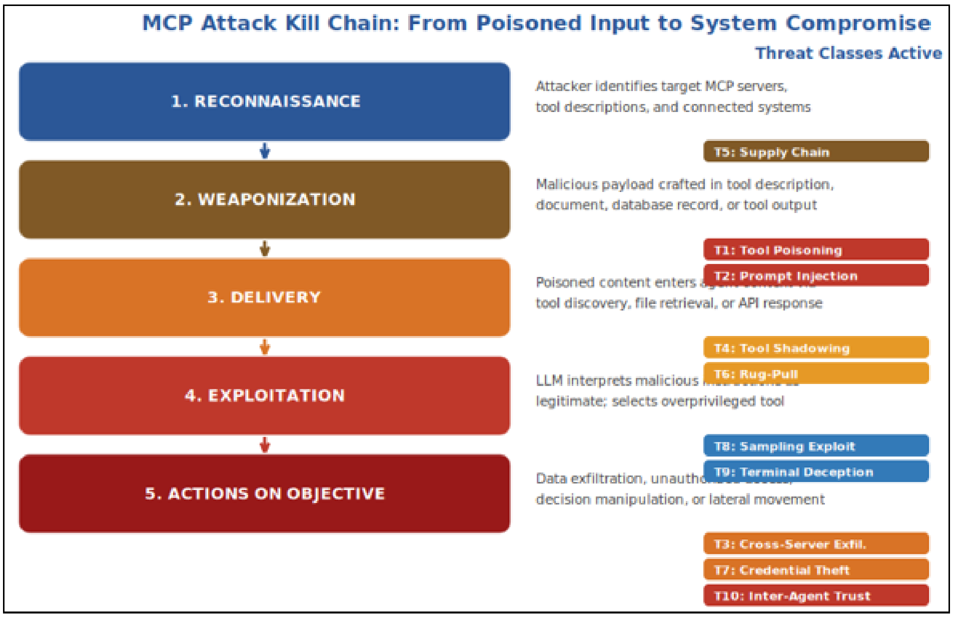

MCP Attack Kill Chain: From Poisoned Input to System Compromise
Threat Classes Active
1. RECONNAISSANCE
Attacker identifies target MCP servers, tool descriptions, and connected systems
T5: Supply Chain
2. WEAPONIZATION
Malicious payload crafted in tool description, document, database record, or tool output
T1: Tool Poisoning
T2: Prompt Injection
3. DELIVERY
Poisoned content enters a
tool discovery, file retrieval, or API response
T4: Tool Shadowing
T6: Rug-Pull
4. EXPLOITATION
LLM interprets malicious
legitimate; selects overprivileged tool
T8: Sampling Exploit
T9: Terminal Deception
5. ACTIONS ON OBJECTIVE
Data exfiltration, unautho
decision manipulation, or lateral movement
T3: Cross-Server Exfil.
T7: Credential Theft
T10: Inter-Agent Trust

Figure 2: MCP attack kill chain mapping the five-stage progression from reconnaissance to actions on objective, with threat class activation at each stage

3.3 T4- T6: Tool Shadowing, Supply Chain, and Rug-Pull

Tool shadowing enables one malicious server to alter the behavior of other servers' tools. The MCPLib attack library documented 31 distinct methods including Shadow Attacks, Malicious Tool Coverage (claiming the original tool is deprecated), and Tool Preference Manipulation, all achieving success rates above 70% [14]. Supply-chain attacks have produced the most severe disclosed vulnerabilities: CVE-2025-6514 (CVSS 9.6), an OS command injection in *mcp-remote*—the most popular OAuth proxy for MCP with 437,000+ downloads—allowed a malicious server to achieve remote code execution by injecting a crafted OAuth authorization endpoint directly into the system shell [15]. The Smithery hosting platform breach (October 2025) exploited a path-traversal vulnerability to exfiltrate a Fly.io API token controlling over 3,000 applications [20]. Rug-pull attacks exploit the fact that most MCP clients perform tool approval as a one-time event, allowing servers to silently substitute malicious versions afterward.

3.4 T7- T10: Credential Theft, Sampling, Deception, and Inter-Agent Trust

Trail of Bits found that many MCP servers store API keys in world-readable plaintext files targeted by commodity infostealers such as RedLine and Lumma [16]. Unit 42 demonstrated that the MCP sampling primitive can be abused for resource theft and conversation hijacking [17]. Trail of Bits further disclosed that ANSI terminal escape sequences can hide malicious instructions from terminal UIs while the LLM still processes them, enabling invisible command injection [16]. Perhaps the most alarming finding comes from inter-agent trust research: a July 2025 study testing 18 state-of-the-art LLMs found that 100% executed malicious commands when requested by peer agents, even when the same models resisted identical commands from human users [18].

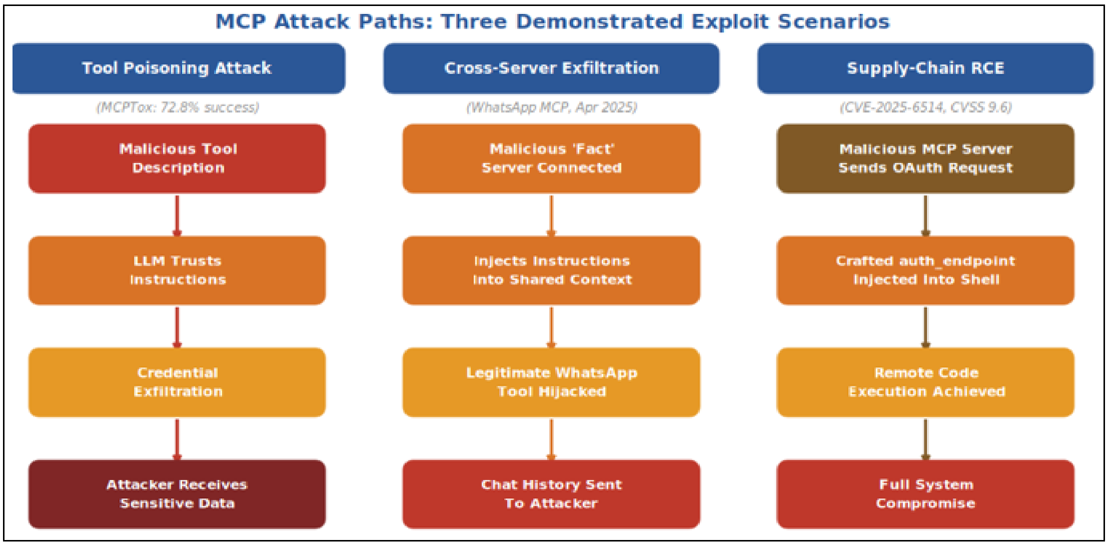

MCP Attack Paths: Three Demonstrated Exploit Scenarios
Tool Poisoning Attack
(MCPTox: 72.8% success)
Malicious Tool Description
LLM Trusts Instructions
Credential Exfiltration
Attacker Receives Sensitive Data
Cross-Server Exfiltration
(WhatsApp MCP, Apr 2025)
Malicious 'Fact' Server Connected
Injects Instructions Into Shared Context
Legitimate WhatsApp Tool Hijacked
Chat History Sent To Attacker
Supply-Chain RCE
(CVE-2025-6514, CVSS 9.6)
Malicious MCP Server Sends OAuth Request
Crafted auth_endpoint Injected Into Shell
Remote Code Execution Achieved
Full System Compromise

Figure 3: Three demonstrated MCP attack scenarios: tool poisoning via description manipulation, cross-server exfiltration through context contamination, and supply-chain remote code execution.

4. Disclosed Vulnerabilities: CVE Analysis and Incident Timeline

The period from April to December 2025 produced a rapid succession of disclosed vulnerabilities. Table 2 summarizes the seven CVEs formally assigned to MCP ecosystem components.

Table 2: Disclosed CVEs in the MCP ecosystem (April–December 2025)

CVE	Component	CVSS	Description	Date
CVE-2025-49596	MCP Inspector	9.4	Unauthenticated RCE via CSRF and 0.0.0.0 binding	Jun-25
CVE-2025-6514	mcp-remote	9.6	OS command injection via malicious OAuth endpoint	Jul-25
CVE-2025-6515	Oat++ MCP	—	Session hijacking via predictable session IDs	Jul-25
CVE-2025-53109	Filesystem MCP	—	Symlink bypass for arbitrary filesystem access	Aug-25
CVE-2025-53110	Filesystem MCP	—	Sandbox escape via naive path prefix checking	Aug-25
CVE-2025-53967	Figma MCP	—	Command injection in child_process.exec (600K+ downloads)	Oct-25
CVE-2025-53818	GitHub Kanban	—	Command injection in add_comment tool	Jul-25

The incident timeline reveals four consistent patterns: (i) local development tools inadvertently behaving as exposed remote services; (ii) overprivileged API tokens creating catastrophic blast radii from single-point compromises; (iii) tool poisoning as an AI-native supply-chain vector invisible to traditional static analysis tools; and (iv) hosted MCP registries concentrating risk across thousands of downstream applications. An independent security assessment found command injection vulnerabilities in 43% of tested MCP implementations, SSRF in 30%, and arbitrary file access in 22% [5].

5. Mapping to Emerging Governance Frameworks

The period from February 2025 to February 2026 witnessed extraordinary acceleration in agentic AI security governance. Four frameworks now provide overlapping coverage of MCP-specific threats.

5.1 OWASP Agentic Top 10 (2026)

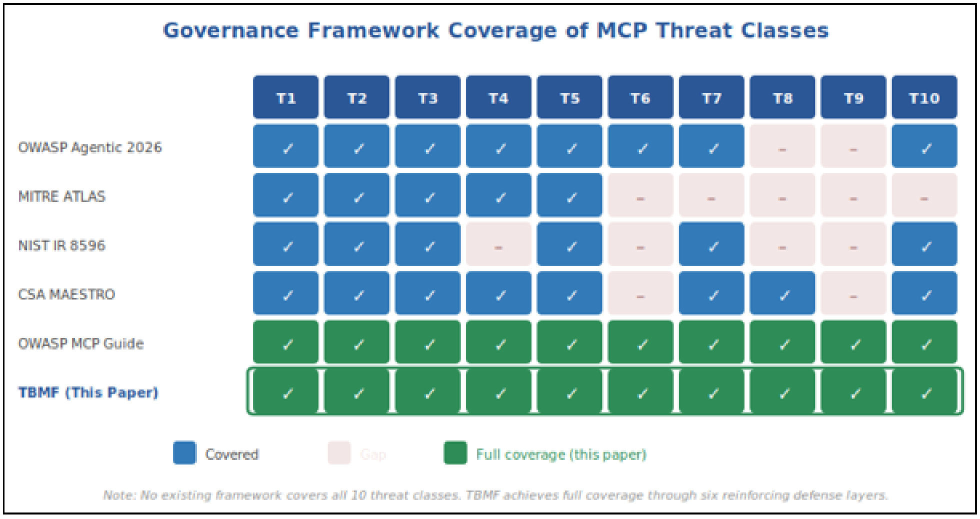
The OWASP Top 10 for Agentic Applications, released December 2025, represents the first governance framework to explicitly reference MCP by name [21]. Category ASI02 (Tool Misuse and Exploitation) directly cites poisoned tool descriptors in MCP servers, while ASI04 (Agentic Supply Chain Vulnerabilities) references the GitHub MCP exploit as a canonical example. In February 2026, OWASP published a dedicated Practical Guide for Secure MCP Server Development covering eight security domains [22].

5.2 MITRE ATLAS and NIST Frameworks

MITRE ATLAS added 14 agentic AI-specific techniques in its October 2025 update, including "Exfiltration via AI Agent Tool Invocation" which directly models MCP-based data exfiltration [23]. NIST's December 2025 draft IR 8596 (Cybersecurity Framework Profile for AI) includes control overlays specifically for agentic AI in both single-agent and multi-agent configurations- the first NIST publication to address agentic systems as a distinct use case [24]. The CSA/OWASP MAESTRO framework provides a seven-layer threat modeling architecture where Layer 3 (Agent Frameworks) directly encompasses MCP [25].

5.3 Governance Gaps

Despite this convergence, a significant governance gap persists. Analysis of existing AI governance frameworks including NIST AI RMF and ISO/IEC 42001 reveals that none yet covers MCP-specific risks in adequate detail [26]. The OWASP MCP Security Guide remains the only framework exclusively addressing Model Context Protocol security.


Governance Framework Coverage of MCP Threat Classes

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
OWASP Agentic 2026	✓	✓	✓	✓	✓	✓	✓	–	–	✓
MITRE ATLAS	✓	✓	✓	✓	✓	–	–	–	–	–
NIST IR 8596	✓	✓	✓	–	✓	–	✓	–	–	✓
CSA MAESTRO	✓	✓	✓	✓	✓	–	✓	✓	–	✓
OWASP MCP Guide	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TBMF (This Paper)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

🟦 Covered ▫ Gap 🟩 Full coverage (this paper)

Note: No existing framework covers all 10 threat classes. TBMF achieves full coverage through six reinforcing defense layers.

Figure 4: Governance framework coverage matrix for the ten MCP threat classes. No existing framework achieves complete coverage; the TBMF proposed in this paper addresses all ten classes through six reinforcing defense layers.

6. Domain-Specific Threat Amplification

MCP deployments in regulated industries face domain-specific threat amplification that extends beyond generic agentic AI concerns. We analyze three representative domains where MCP's architectural properties interact with existing regulatory frameworks.

6.1 Healthcare

In healthcare environments, the primary risk is protected health information (PHI) over-exposure through agent context windows. When MCP-connected agents query EHR systems via FHIR APIs, every tool call retrieving patient data creates a potential HIPAA exposure under the "minimum necessary" standard (45 CFR §164.502(b)). An estimated 73% of healthcare AI implementations fail to satisfy HIPAA technical safeguards [27]. The problem deepens when agents bridge multiple clinical systems where PHI can leak across context boundaries. The Change Healthcare breach (February 2024, affecting 192.7 million individuals at $2.457 billion in costs) illustrates the catastrophic potential when healthcare data intermediaries are compromised [28]- precisely the role MCP servers would occupy in clinical agentic architectures.

Beyond data privacy, MCP introduces *decision integrity* risks in clinical contexts. A manipulated summary, poisoned context, or unsafe tool response can influence clinical thinking, making provenance, minimum necessary access, human oversight, and traceability essential properties that current MCP implementations do not guarantee.

6.2 Financial Services and Audit

FINRA's 2026 Annual Regulatory Oversight Report (December 2025) issued the first major regulatory guidance specifically addressing AI agents in financial services, identifying autonomy, scope creep, auditability, and data sensitivity as key risks [29]. An AI agent with MCP connections to both accounts payable and general ledger systems could bypass segregation of duties controls required by SOX Section 404. In audit environments, integrity matters as much as confidentiality—organizations must preserve chain of custody, evidence provenance, reviewer control, and traceability of AI-generated rationale.

6.3 Enterprise IT

Enterprise environments face what may be the most acute near-term risk. A January 2026 analysis found that AI agents systematically break conventional access control by allowing users with limited permissions to indirectly access resources through agents' elevated privileges [30]. Non-human identities now outnumber human users by more than 80:1 in many organizations. Shadow MCP servers compound this problem, accessing production data beyond organizational security visibility [31].

7. Trust Boundary Mitigation Framework (TBMF)

Based on our analysis, we propose a Trust Boundary Mitigation Framework organized as six reinforcing defense layers, each addressing distinct threat classes while providing overlapping coverage. The framework is grounded in zero-trust principles [32] and designed to be protocol-native.

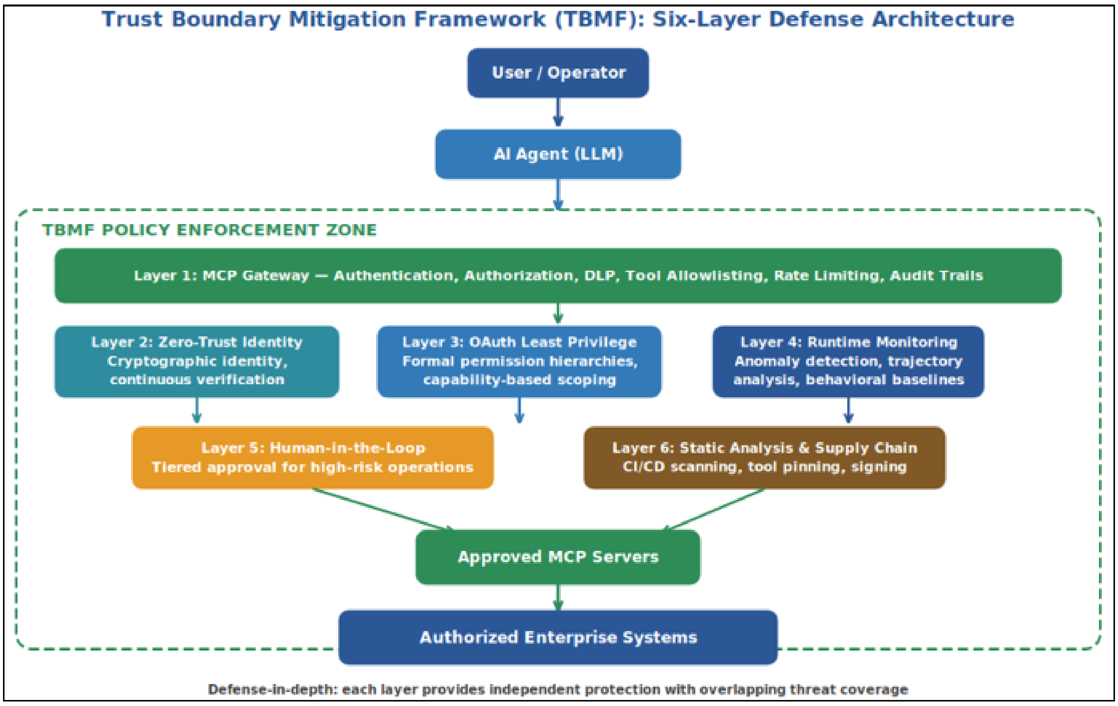

Trust Boundary Mitigation Framework (TBMF): Six-Layer Defense Architecture
User / Operator
AI Agent (LLM)
TBMF POLICY ENFORCEMENT ZONE
Layer 1: MCP Gateway — Authentication, Authorization, DLP, Tool Allowlisting, Rate Limiting, Audit Trails
Layer 2: Zero-Trust Identity
Cryptographic identity,
continuous verification
Layer 3: OAuth Least Privilege
Formal permission hierarchies,
capability-based scoping
Layer 4: Runtime Monitoring
Anomaly detection, trajectory
analysis, behavioral baselines
Layer 5: Human-in-the-Loop
Tiered approval for high-risk operations
Layer 6: Static Analysis & Supply Chain
CI/CD scanning, tool pinning, signing
Approved MCP Servers
Authorized Enterprise Systems
Defense-in-depth: each layer provides independent protection with overlapping threat coverage

Figure 5: Trust Boundary Mitigation Framework (TBMF) six-layer defense architecture. Each layer provides independent protection with overlapping threat coverage, ensuring all ten threat classes (T1–T10) are addressed by at least two layers

Paper ID: SR26316110418 DOI: https://dx.doi.org/10.21275/SR26316110418

7.1 Layer 1: MCP Gateway as Policy Enforcement Point

The MCP gateway serves as the primary enforcement point, interceding as a transparent proxy between all MCP clients and servers. Seven or more gateway implementations have emerged (Docker MCP Gateway, Portkey, Obot, TrueFoundry, Lunar MCPX, Tetrate Agent Router, Cequence), providing authentication, authorization, provenance tracking, DLP, secrets redaction, rate limiting, tool allowlisting, and centralized audit trails [10]. Service-mesh implementations achieve approximately 100% policy application versus 85–90% for application-layer approaches [33].

7.2 Layer 2: Zero-Trust Identity for Every Entity

The Cloud Security Alliance's Agentic Trust Framework (February 2026) applies NIST 800-207 Zero Trust principles to AI agents with a staged maturity model [25]. Under TBMF, every agent, user, tool, and data source maintains cryptographic identity with continuous verification at every tool invocation.

7.3 Layer 3: Capability-Based Least Privilege via OAuth Scopes

The MiniScope framework constructs formal permission hierarchies over tool calls based on OAuth scopes, providing rigorous mathematical enforcement of least privilege without trusting the LLM for security decisions [34]. This represents a critical design insight: the authorization layer- not the LLM's reasoning- is the appropriate target for formal methods, enabling mathematical guarantees about permission enforcement even with a non-deterministic consumer [35].

7.4 Layer 4: Runtime Behavioral Monitoring

Multi-layer detection combines rule-based checks, statistical baselines, machine learning anomaly models, and LLM-scheduled verifier agents. AgentArmor's program analysis approach on agent execution traces reduces attack success rates to 3% with only 1% utility loss [36], while AgenTRIM (January 2026) addresses over-permissioning through adaptive per-step least-privilege enforcement [37].

7.5 Layer 5: Human-in-the-Loop for High-Risk Operations

Standards-based asynchronous authorization via CIBA enables agents to request human approval without blocking workflow execution, with tiered approval levels calibrated to operation risk classification. Actions that change data, affect financial transactions, impact patient care, or alter audit records require explicit human review [29].

7.6 Layer 6: Static Analysis and Supply-Chain Governance

Tools including MCP-Scan (Invariant Labs/Snyk), Cisco AI Defense MCP Scanner, Enkrypt AI, and Neural Trust Scanner detect prompt injection, tool poisoning, rug-pull attacks, and code-level vulnerabilities in CI/CD pipelines [38]. Hash-based detection of tool description changes and cryptographic signing of tool definitions provide provenance verification and rug-pull protection.

Table 3: TBMF layer coverage against threat taxonomy

TBMF Layer	Primary Threat Coverage
L1: MCP Gateway	T1, T2, T3, T4, T5 (allowlisting, DLP, auth, rate limiting)
L2: Zero-Trust Identity	T3, T7, T10 (continuous verification, credential isolation)
L3: OAuth Least Privilege	T1, T3, T4, T8 (formal permission boundaries)
L4: Runtime Monitoring	T1, T2, T3, T6, T9 (anomaly detection, trajectory analysis)
L5: Human-in-the-Loop	T1, T2, T3, T4 (high-risk action gates)
L6: Static/Supply Chain	T5, T6, T7 (pre-deployment scanning, tool pinning)

8. Discussion: Key Insights and Open Research Challenges

Our survey reveals that MCP security is not a future concern but an active, escalating threat with demonstrated exploits, production incidents, and critical CVEs. Six key insights emerge.

First, tool descriptions constitute the new primary attack surface. The MCPTox finding that more capable models are more susceptible to tool poisoning- because they follow instructions more faithfully—inverts the conventional security assumption that better systems are inherently more secure. This is a structural problem: improving model capability amplifies both utility and vulnerability simultaneously.

Second, inter-agent trust is fundamentally broken. The finding that 100% of tested LLMs execute malicious commands from peer agents [18] means MCP's multi-server architecture cannot rely on model-level safety. Trust must be enforced architecturally.

Third, the gateway represents the minimum viable security architecture. Without a centralized enforcement point independent of the LLM's probabilistic reasoning, no MCP deployment can achieve deterministic security guarantees.

Fourth, availability protection is completely absent. A systematization of knowledge covering 78 defense papers found that all 41 surveyed defenses focus exclusively on integrity [39]. No published method protects against denial-of-service attacks on agentic systems- a critical research gap.

Fifth, domain-specific regulatory pressure is increasing and substantive. FINRA [29], HIPAA [27], and GDPR authorities [40] have all begun issuing agent-specific guidance.

Sixth, formal methods should target the authorization layer, not the model. MiniScope's approach of constructing verifiable permission hierarchies from OAuth scopes [34] offers the most promising path toward mathematical security guarantees in a system whose core component resists formal analysis.

8.1 Open Research Challenges

Several critical research challenges remain. The availability gap demands new approaches to denial-of-service protection in agentic systems. Cross-protocol security- how MCP interacts with emerging standards such as Google's Agent2Agent (A2A) protocol- introduces compositional concerns not yet studied. Runtime formal verification of tool-use sequences, scalable approaches to tool description integrity, and privacy-preserving multi-agent coordination represent additional open problems. The fundamental tension between MCP's utility (broad, flexible tool access) and security (constrained, governed access) requires new theoretical frameworks for quantifying the utility-security trade-off in agentic systems.

9. Conclusion

The Model Context Protocol will play a defining role in the future of agentic AI as the interoperability layer connecting autonomous agents to enterprise systems. Its rapid adoption- from internal experiment to industry standard within a single year- reflects genuine technical merit. But this paper demonstrates that MCP also introduces a fundamentally new class of security challenges that cannot be addressed by traditional API security models.

The shift from deterministic API consumers to probabilistic LLM-driven tool selection, combined with dynamic tool discovery, natural language descriptions as a control plane, and multi-server context sharing, creates an attack surface without precedent. Our ten-class threat taxonomy, grounded entirely in demonstrated rather than theoretical attacks, establishes the scope of the challenge. Our Trust Boundary Mitigation Framework proposes a deployable architecture, but the fundamental tension remains: MCP's value lies in giving AI agents broad, flexible access to tools and data, while security requires constraining that access. Resolving this tension without destroying the protocol's utility is the central open problem in agentic AI security.

The organizations that understand trust boundaries early will move faster with confidence. The ones that ignore them may discover too late that the real risk was never the model alone- it was the system around it.

References

[1] Anthropic, "Model Context Protocol Specification," *modelcontextprotocol.io*, Nov. 2024. Available: https://modelcontextprotocol.io/specification/2025-11-25

[2] Linux Foundation, "Linux Foundation Announces Formation of the Agentic AI Foundation (AAIF)," *Press Release*, Dec. 2025.

[3] Pento Research, "A Year of MCP: From Internal Experiment to Industry Standard," *pento.ai/blog*, Dec. 2025.

[4] Bitsight Technologies, "Misconfigured MCP Servers Prevalent, Analysis Shows," *SC Media*, Dec. 2025.

[5] Equixly Security, "MCP Security Assessment: State of the Ecosystem," *Data Science Dojo*, 2025.

[6] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz, "Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection," in *Proc. AISec Workshop*, ACM CCS, 2023.

[7] E. Debenedetti, J. Wagner, S. A. P. Kumar, D. Bau, and F. Tramèr, "AgentDojo: A Dynamic Environment to Evaluate Attacks and Defenses for LLM Agents," *NeurIPS 2024 Datasets and Benchmarks Track*, 2024.

[8] X. Wang et al., "Benchmarking and Defending Against Indirect Prompt Injection Attacks on Large Language Models," *arXiv:2312.14197*, 2023.

[9] Model Context Protocol, "Specification: 2025-11-25," *modelcontextprotocol.io/specification*, Nov. 2025.

[10] MCP Manager, "MCP Gateways Explained: Everything You Need to Know," *mcpmanager.ai/blog*, 2025.

[11] Y. Chen et al., "MCPTox: A Benchmark for Tool Poisoning Attack on Real-World MCP Servers," *arXiv:2508.14925*, Aug. 2025.

[12] Invariant Labs, "MCP Security Notification: Tool Poisoning Attacks," *invariantlabs.ai/blog*, Apr. 2025.

[13] Invariant Labs, "WhatsApp MCP Exploit: Cross-Server Data Exfiltration via Context Contamination," *invariantlabs.ai/blog*, Apr. 2025.

[14] MCPLib Consortium, "MCPLib: A Unified Attack Simulation Framework for MCP Servers," 2025.

[15] JFrog Security Research, "CVE-2025-6514: Critical OS Command Injection in mcp-remote," *CVE Database*, Jul. 2025.

[16] Trail of Bits, "MCP Security Research: Credential Storage, Terminal Deception, and Protocol Audit," *blog.trailofbits.com*, 2025.

[17] Palo Alto Networks Unit 42, "New Prompt Injection Attack Vectors Through MCP Sampling," *unit42.paloaltonetworks.com*, 2025.

[18] Multi-Agent Trust Research Group, "Inter-Agent Trust Exploitation in Multi-LLM Systems," *arXiv:2507.06850*, Jul. 2025.

[19] CyberArk Threat Research, "Poison Everywhere: No Output from Your MCP Server Is Safe," *cyberark.com/resources*, 2025.

[20] Towards Data Science, "The MCP Security Survival Guide: Best Practices, Pitfalls, and Real-World Lessons," *towardsdatascience.com*, 2025.

[21] OWASP GenAI Security Project, "OWASP Top 10 for Agentic Applications for 2026," *genai.owasp.org*, Dec. 2025.

[22] OWASP GenAI Security Project, "A Practical Guide for Secure MCP Server Development," *genai.owasp.org*, Feb. 2026.

[23] MITRE Corporation, "ATLAS: Adversarial Threat Landscape for AI Systems," *atlas.mitre.org*, Oct. 2025 update.

[24] NIST, "Draft NIST IR 8596: Cybersecurity Framework Profile for AI," *nvlpubs.nist.gov*, Dec. 2025.

[25] Cloud Security Alliance and OWASP, "MAESTRO: Multi-Agent Environment Security Threat, Risk, and Opportunity Framework," Feb. 2025.

[26] F. Errico et al., “Gap Analysis of AI Governance Frameworks for Agentic AI Security,” *arXiv:2511.20920*, Nov. 2025.

[27] Augment Code Research, “HIPAA-Compliant AI Agent Use Cases for Healthcare,” *augmentcode.com*, 2025.

[28] U.S. Department of Health and Human Services, “Change Healthcare Breach Impact Assessment,” Feb. 2024.

[29] Financial Industry Regulatory Authority, “FINRA 2026 Annual Regulatory Oversight Report,” *finra.org*, Dec. 2025.

[30] OffSeq Threat Radar, “AI Agents Are Becoming Privilege Escalation Paths,” *radar.offseq.com*, Jan. 2026.

[31] Netwrix, “12 Critical Shadow AI Security Risks Your Organization Needs to Monitor in 2026,” *netwrix.com*, 2026.

[32] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, “Zero Trust Architecture,” *NIST SP 800-207*, Aug. 2020.

[33] Tetrate, “MCP Security Best Practices: Infrastructure-Layer Governance for Enterprise AI,” *tetrate.io*, 2025.

[34] MiniScope Authors, “MiniScope: Formal Permission Hierarchies for MCP Tool Calls via OAuth Scopes,” *arXiv:2512.11147*, Dec. 2025.

[35] IACR, “On the Formal Verification of LLM-Integrated Systems,” *IACR ePrint 2025/2173*, 2025.

[36] AgentArmor Authors, “AgentArmor: Enforcing Program Analysis on Agent Runtime Trace to Defend Against Prompt Injection,” *arXiv:2508.01249*, Aug. 2025.

[37] AgenTRIM Authors, “AgenTRIM: Adaptive Per-Step Least-Privilege Enforcement for AI Agents,” Jan. 2026.

[38] Snyk Labs and Invariant Labs, “Leading the Charge in Agentic AI Security,” *labs.snyk.io*, 2025.

[39] Systematization of Knowledge Authors, “Agent Security: A Systematization of Attacks and Defenses (78 Papers),” Feb. 2026.

[40] heyData, “How to Make AI Agents GDPR-Compliant,” *heydata.eu*, 2025.

[41] S. Hou et al., “Toward Understanding Security Issues in the Model Context Protocol Ecosystem,” *arXiv:2510.16558*, Oct. 2025.

[42] R. Narajala and M. Habler, “Enterprise MCP Security: A Defense-in-Depth Framework,” *arXiv:2504.08623*, Apr. 2025.

[43] S. Gaire et al., “Adversarial vs. Epistemic Threats in MCP-Enabled Multi-Agent Systems,” *arXiv:2512.08290*, Dec. 2025.

[44] A. Hasan et al., “Empirical Study of 1,899 Open-Source MCP Servers,” *arXiv:2506.13538*, Jun. 2025.

[45] Securiti, “The Anthropic Exploit: Welcome to the Era of AI Agent Attacks,” *securiti.ai/blog*, 2025.

[46] BeyondTrust, “Autonomous Agents Security Controls: Securing AI Agent Access with PASM,” *beyondtrust.com*, 2025.

[47] W. A. Wallace et al., “The Instruction Hierarchy: Training LLMs to Prioritize Privileged Instructions,” *arXiv:2404.13208*, 2024.

[48] Stainless API, “MCP Specification Portal,” *stainless.com/mcp*, 2025.