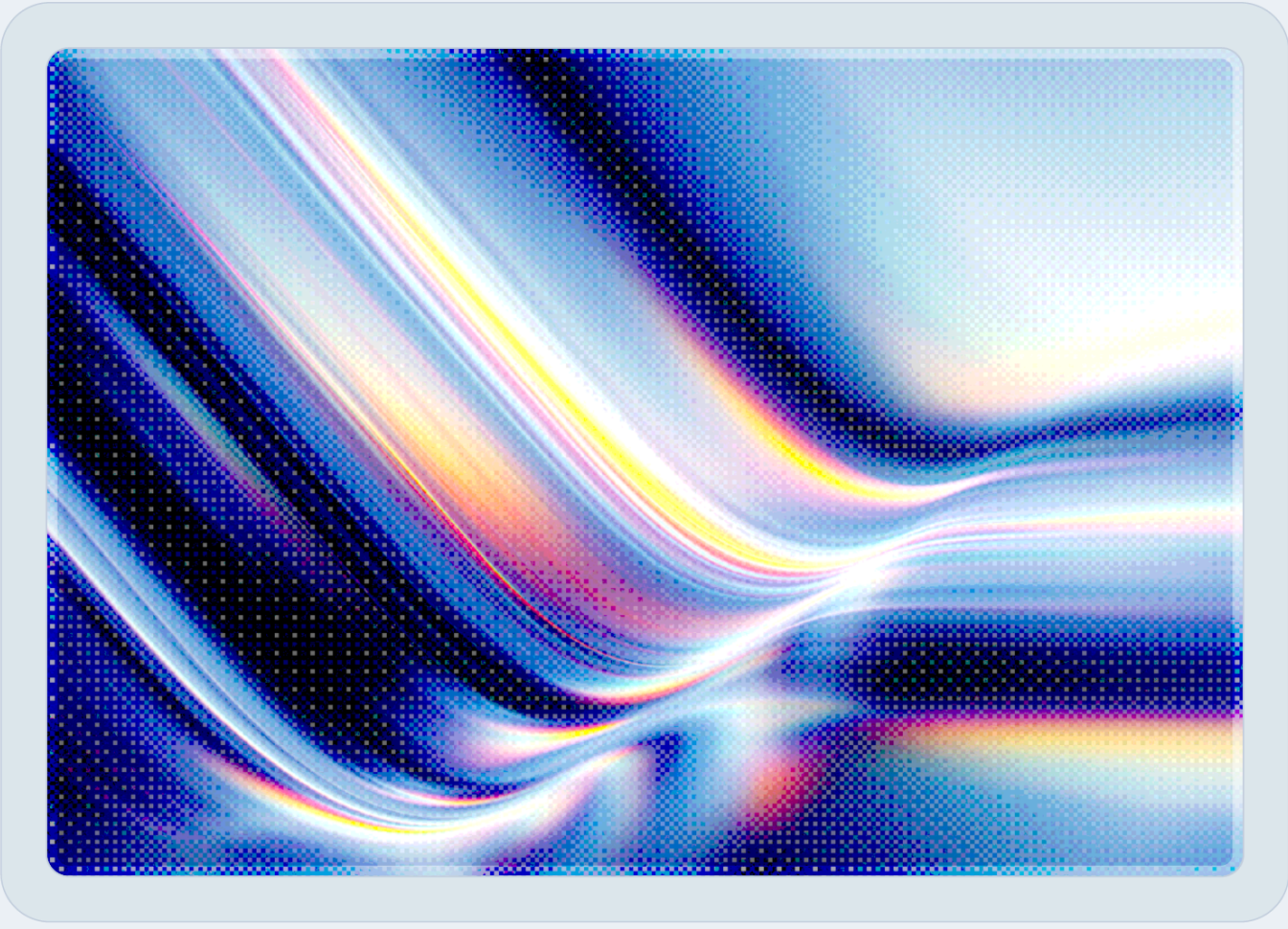
CSAI Foundation | Cloud Security Alliance

"Living Off the Agent": AI Agents as Lateral Movement

The LOTA Threat Pattern and Enterprise Defense Guidance

2026-05-19

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

Key Takeaways

- A new attack pattern termed "Living Off the Agent" (LOTA) repurposes AI agents' own legitimate, authenticated connections as the vector for lateral movement — replacing the binaries and scripts of traditional Living Off the Land (LOTL) attacks with natural language instructions injected into agent-processed content.
- Analysis of 21 documented multi-stage agentic AI incidents in 2025–2026 found that lateral movement appeared in eight of those cases, up from three of twelve incidents in 2024 and none in 2023, tracking the rapid proliferation of enterprise agentic deployments [1].
- A second-order prompt injection vulnerability in ServiceNow's Now Assist platform, disclosed in November 2025 by security researchers at AppOmni, provides a concrete real-world example: a low-privilege agent could be manipulated into causing a higher-privilege peer to export sensitive case files and escalate administrative roles, without any compromise of underlying infrastructure [2].
- Current multi-agent architectures impose no consistent standard for inter-agent authentication or message integrity verification, meaning a single compromised or manipulated agent can propagate adversarial instructions downstream as if they were trusted operational directives [3].
- Shared memory stores, MCP tool registries, and agent-to-agent communication channels have all been demonstrated as viable entry points for persistent LOTA-style attacks; organizations should treat each surface as a trust boundary requiring explicit access controls.

Background

From LOTL to LOTA

Living Off the Land (LOTL) is a well-established tradecraft principle in which attackers avoid introducing foreign binaries by weaponizing trusted system utilities — PowerShell, WMI, certutil, and similar tools that defenders expect to see operating legitimately. The detection challenge is inherent: the attacker's

footprint is the footprint of normal administration. The emergence of AI agents in enterprise environments has introduced a structurally equivalent problem at a new layer of the stack, one for which most security organizations have yet to develop the specialized instrumentation required to detect.

Modern enterprise AI agents are built to be useful precisely because they are trusted. A productivity agent integrated with Microsoft 365, Google Workspace, or Salesforce holds OAuth tokens, calendar access, document permissions, and API credentials on behalf of its users. It reads emails, populates CRM records, triggers workflows, and coordinates with other agents to complete multi-step tasks. These capabilities are the same ones that make a compromised agent an ideal pivot point: when an attacker's instructions travel inside the agent's normal activity stream, they inherit the agent's access rights and produce the agent's telemetry. This is the essence of LOTA — living off the agent rather than the land.

Security researchers studying agentic AI attack patterns describe the core mechanism directly: attackers adopt an AI agent's legitimate, authenticated connections to pivot between systems by injecting malicious instructions into content the agent processes, with natural language serving as the attack vector [1][4]. Where LOTL abuses binaries, LOTA abuses context windows. The instruction set is the same English the agent uses to perform legitimate work, making signature-based detection largely ineffective against novel or obfuscated injection payloads.

The Promptware Kill Chain

Academic and practitioner research has begun to formalize how these attacks unfold. Researchers publishing the Promptware Kill Chain framework in January 2026 analyzed 36 prominent studies and real-world incidents affecting production LLM systems and identified 21 multi-stage attacks that traversed four or more stages [1]. Their framework maps a seven-stage progression: Initial Access via prompt injection, Privilege Escalation, Reconnaissance, Persistence through memory and retrieval poisoning, Command and Control, Lateral Movement, and Actions on Objective. The lateral movement stage appeared in eight of the 21 documented incidents in 2025–2026 — up from three of twelve in 2024 and none in 2023. This growth trajectory is consistent with the rapid proliferation of enterprise agentic deployments over the same period.

CSA's own research note on promptware C2 infrastructure, published in March 2026, documented how the Reprompt incident demonstrated native command-and-control capabilities embedded within a prompt injection chain — one of six attacks in the dataset that achieved five or more kill chain stages [5]. LOTA represents the lateral movement phase of that same progression, realized at enterprise scale and with the full authentication context of the agent's provisioned identity.

Security Analysis

The Anatomy of a LOTA Attack

A LOTA attack begins where an AI agent receives input it should not trust. In indirect prompt injection, the attacker does not interact with the agent directly; instead, malicious instructions are embedded in documents, emails, web pages, calendar invitations, or database records that the agent processes as part of its normal workflow. When the agent encounters this poisoned content, it interprets the embedded instructions as legitimate directives. Because the agent acts with the user's credentials and trust context, the resulting actions — reading sensitive files, sending outbound communications, calling APIs, modifying group memberships — appear as normal agent activity to downstream systems and security tools alike.

The second variant, second-order prompt injection, exploits the trust hierarchy in multi-agent systems. In this pattern, an attacker with low-privilege access embeds malicious instructions in data that will later be processed by a higher-privilege agent. Security researchers at AppOmni disclosed precisely this vulnerability in ServiceNow's Now Assist platform in November 2025. The platform's agent discovery feature — which enables agents to communicate with each other to complete complex tasks — could be weaponized: a low-privileged user embedding instructions in a service case description could cause a higher-privilege agent operating on that case to export data to an external URL and escalate account permissions [2]. ServiceNow subsequently addressed the vulnerability through platform updates. Crucially, the researchers characterized the root cause not as a software bug but as expected behavior under certain default configuration options — a distinction with significant implications for how organizations audit agentic deployments.

Shared Memory and Persistent Compromise

A distinct subset of LOTA attacks targets the shared memory and retrieval systems that enable agents to learn from prior interactions. Unlike a prompt injection that terminates when a conversation closes, memory poisoning writes the attacker's payload into the agent's long-term knowledge store, where it can be recalled in future sessions and influence agent behavior days or weeks after the initial compromise event [6].

In multi-agent architectures, this problem compounds. Where agents share a common episodic memory store, a single poisoned episode — recorded as a "successful resolution" of a fabricated task — can propagate as learned policy to any agent that subsequently retrieves and replicates that sequence. Research on multi-agent security architectures published in early 2026 identifies this as a cross-system

risk with no current technical standard to mitigate it: there is presently no established protocol requiring agents to authenticate the provenance of memory entries or messages received from peer agents, and frameworks that rely on implicit network-layer trust — a common default in current production deployments — treat instructions from peer agents as trusted without verifying their origin [3].

MCP Tool Registries as Attack Infrastructure

The Model Context Protocol (MCP), which has become a widely adopted standard for connecting AI assistants to external tools and data sources, introduces a further lateral movement surface. Tool poisoning attacks — first publicly disclosed by Invariant Labs in April 2025 — embed adversarial instructions inside tool descriptions and metadata that the agent model reads but the human user cannot easily inspect [7]. An agent that processes a malicious tool description may subsequently exfiltrate data to attacker-controlled infrastructure or trigger unauthorized actions in connected services, all while appearing to the user to be performing its assigned task.

Researchers have since identified malicious MCP servers deployed in the wild — including compromised packages modified to embed exfiltration instructions within their tool metadata — confirming that the tool poisoning threat has moved beyond proof-of-concept into operational attack infrastructure [7]. Empirical testing of production LLM agents against real-world MCP servers has found broad susceptibility to this class of attack [7]. Because MCP servers are commonly installed through package managers without the formal security review applied to other production dependencies, and because they can receive silent updates, they represent a supply chain risk that feeds directly into LOTA-style lateral movement — a pathway that requires no user interaction beyond the initial server installation [8].

The Detection Gap

The fundamental detection challenge in LOTA is that the attack traffic is legitimate agent activity. When a compromised productivity agent reads an inbox, accesses a document folder, and posts a message to a collaboration platform, those events register as three routine agent operations — not as three stages of a lateral movement chain. Traditional endpoint detection and response tools and SIEM rules were designed to identify anomalous process execution, network connections, and file access patterns. An agent operating entirely within its sanctioned perimeter, using its provisioned credentials, and invoking only its authorized tools produces none of those signals — even while executing an attacker's instructions.

Security researcher Christian Schneider's analysis of AI agents as attack pivots describes this gap in terms of the security model's implicit assumptions: the agent acts with legitimate credentials in trusted SaaS environments, leaving no malicious binary, no exploited CVE, and no anomalous authentication

event — only a sequence of authorized API calls that happens to serve attacker objectives [9]. OWASP's Top 10 for LLM Applications identifies Excessive Agency — over-privileged tool access and unbounded autonomy — as one of the primary structural enablers of this class of attack, because the attack surface is proportional to the permissions granted [10]. Purpose-built agentic AI behavioral monitoring, capable of reasoning about intent and action sequences rather than just individual events, is required to close the detection gap that LOTA exploits.

Recommendations

Immediate Actions

Organizations deploying AI agents with access to sensitive enterprise systems should first verify the patch status of any agentic AI platform for known prompt injection and privilege escalation vulnerabilities. ServiceNow customers should confirm that relevant security patches addressing agent discovery vulnerabilities are applied and audit agent discovery groupings for unintended trust relationships [2]. More broadly, any production agent with write access to data later processed by a higher-privilege agent should be reviewed for second-order injection exposure in its default configuration.

The principle of least privilege must be applied specifically to agent credentials and tool authorizations. Agents should hold the minimum permissions required for each discrete task, and permission grants should be scoped and time-limited rather than persistent. Organizations should inventory every agent's tool set and permission scope against the tasks that agent is designed to perform, treating any excess as an attack surface.

MCP server installations should be subject to the same supply chain security review applied to other third-party software dependencies. Tool descriptions should be treated as untrusted input, and organizations should maintain an inventory of installed MCP servers with defined review cadences to catch compromised or updated packages.

Short-Term Mitigations

Near-term mitigations center on introducing trust boundaries that the current generation of agentic frameworks does not enforce by default. Structured validation of data inputs before they reach agent processing — and output filtering before agent-initiated API calls — reduces the probability of successful

indirect injection by creating checkpoints between the attacker's payload and the agent's execution context.

Multi-agent architectures should adopt explicit authentication for inter-agent messages rather than relying on implicit network-layer trust. Research on multi-agent security architectures finds that no single topology is universally safer — risk profiles differ substantially by deployment context and task type [3]. The primary protective factor is not architectural topology but message provenance control: explicitly authenticating the origin of inter-agent messages reduces the probability of cross-agent compromise regardless of whether the system uses centralized or decentralized orchestration. Agent orchestration frameworks that support message signing or cryptographic attestation of agent identity should therefore be preferred over configurations that lack message provenance controls entirely.

Human-in-the-loop controls for high-consequence agent actions provide a reliable near-term mitigation available today. Actions with irreversible or externally visible effects — sending outbound communications, deleting records, modifying access control lists, triggering financial transactions — should require explicit human confirmation before execution. This does not eliminate the LOTA attack surface, but it breaks the kill chain at the Actions on Objective stage, which is where adversarial objectives are realized.

Strategic Considerations

At the strategic level, organizations should integrate agent threat modeling into their security architecture review process before agentic systems enter production. CSA's MAESTRO framework provides a seven-layer model specifically designed for agentic AI that maps lateral movement risks at the infrastructure layer — where a compromised agent container can bridge to vector databases and secret stores — and at the agent ecosystem layer, where trust boundaries are defined dynamically at runtime rather than by static policy [11][12]. MAESTRO's cross-layer threat propagation analysis is directly applicable to the second-order injection and memory poisoning vectors described above.

Agent behavioral monitoring — purpose-built observability that captures reasoning traces, tool call sequences, and semantic intent rather than just discrete event logs — should be treated as a first-class security control, not merely an operational convenience. The current generation of SIEM integrations is not designed to reason about the meaning of agent actions in context. Organizations deploying agentic AI at scale should evaluate monitoring and runtime guardrail solutions specifically designed for agentic behavior, and should conduct adversarial red-team exercises against production agents to validate detection coverage [4].

Finally, organizations should establish clear agent identity and lifecycle governance. Every deployed agent should carry a defined identity, a documented scope of access, a structured review cadence for its permissions, and a clear decommissioning path. Without this foundation, the growing fleet of enterprise agents risks replicating the unmanaged service account sprawl that has enabled lateral movement in traditional IT environments for years — with the added complication that agentic service accounts can receive and act on instructions in natural language, across every connected system, without the predictable execution patterns that make service account abuse detectable today.

CSA Resource Alignment

LOTA attack patterns engage several layers of CSA's existing security frameworks. Within MAESTRO, lateral movement maps primarily to Layer 4 (Infrastructure), where agent compromise propagates to shared platform components such as vector databases and secrets managers, and to Layer 7 (Agent Ecosystem and Trust Boundaries), where dynamic runtime trust relationships create the conditions for cross-agent exploitation [11]. CSA's guidance on applying MAESTRO to real-world agentic AI threats, published in February 2026, explicitly addresses cross-layer attack propagation and provides baseline mitigations applicable to lateral movement risks of the kind described in this note [12].

The AI Controls Matrix (AICM) [13] addresses the governance structures most relevant to LOTA in its domains covering AI supply chain security, access management, and orchestration security. The AICM Orchestrated Service Provider guidelines provide specific controls for organizations operating multi-agent pipelines, including requirements for trust boundary documentation and least-privilege tool configuration. Organizations conducting STAR for AI [14] assessments should include agent identity management, inter-agent authentication, and MCP server supply chain review as explicit assessment criteria.

CSA's March 2026 research note on promptware and agent C2 infrastructure examined the kill chain stage immediately preceding lateral movement and provides complementary guidance on agent isolation and behavioral monitoring that is directly applicable to LOTA defense [5].

References

[1] Ben Nassi et al. "The Promptware Kill Chain: How Prompt Injections Gradually Evolved Into a Multi-Step Malware Delivery Mechanism." arXiv:2601.09625, January 2026.

[2] Aaron Costello, AppOmni. "ServiceNow AI Agents Can Be Tricked Into Acting Against Each Other via Second-Order Prompts." The Hacker News, November 2025.

[3] Researchers. "Architecture Matters for Multi-Agent Security." arXiv:2604.23459, April 2026.

[4] Andrew Kew. "Your AI Agent Is the New Attack Vector. It Just Wants to Help." DEV Community, 2026. References Straiker agentic AI red team findings; see also Straiker AI Security Research.

[5] Cloud Security Alliance AI Safety Initiative. "Agent Commander: Promptware Turns AI Agents into C2 Infrastructure." CSA Lab Space, March 2026.

[6] Researchers. "Memory Poisoning and Secure Multi-Agent Systems." arXiv:2603.20357, March 2026.

[7] Various researchers; Invariant Labs (original tool poisoning disclosure, April 2025). "Model Context Protocol Threat Modeling and Analysis of Vulnerabilities to Prompt Injection with Tool Poisoning." arXiv:2603.22489, 2026.

[8] Checkmarx Zero. "11 Emerging AI Security Risks with MCP (Model Context Protocol)." Checkmarx, 2026.

[9] Christian Schneider. "AI Agents as Attack Pivots: The New Lateral Movement." Christian Schneider Blog, 2026.

[10] OWASP. "OWASP Top 10 for LLM Applications 2025." OWASP, 2024.

[11] Cloud Security Alliance. "Agentic AI Threat Modeling Framework: MAESTRO." CSA Blog, February 2025.

[12] Cloud Security Alliance. "Applying MAESTRO to Real-World Agentic AI Threat Models: From Framework to CI/CD Pipeline." CSA Blog, February 2026.

[13] Cloud Security Alliance. "AI Controls Matrix (AICM): Framework for Trustworthy AI." CSA, 2025.

[14] Cloud Security Alliance. "CSA STAR for AI: AI Security Assurance and Trust Framework." CSA, 2025.