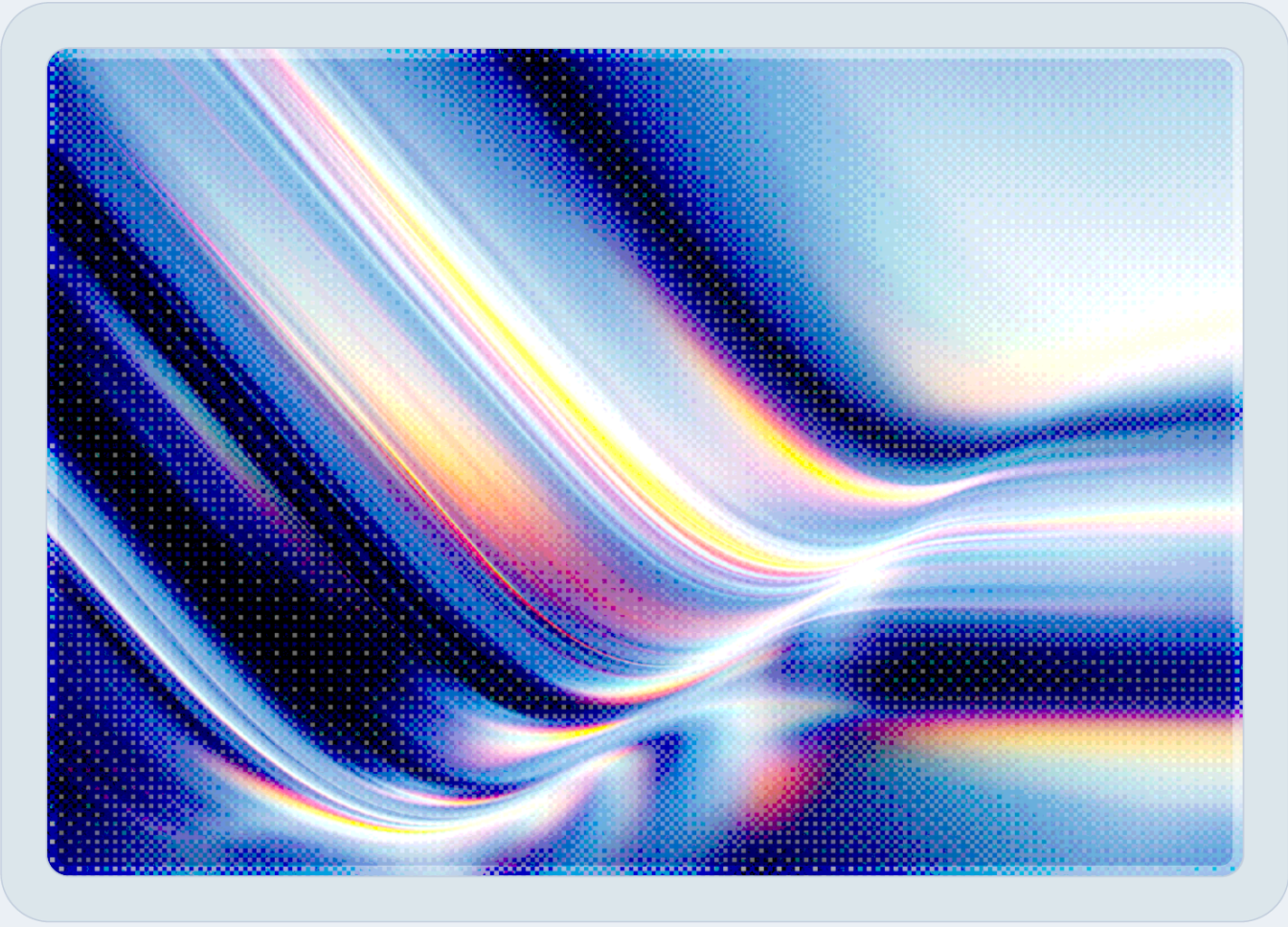
CSAI Foundation | Cloud Security Alliance

Zero-Day to Weaponized: AI Inference Under Attack

Rapid Exploitation Targeting AI Inference Platforms

2026-05-07

AI-assisted Rapid Research



CSAI

cloud security alliance®

Key Takeaways

- The median time between public vulnerability disclosure and active exploitation fell from roughly 32 days in 2022 to approximately 5 days by 2023, and 2025 data shows that 32.1% of exploits now appear on or before the CVE's public disclosure date—meaning patches often arrive after adversaries have already begun operating [1][2].
- AI inference platforms—including NVIDIA Triton Inference Server, vLLM, Meta Llama Stack, and Ollama—have emerged as a distinct and high-value attack surface, with critical remote-code-execution (RCE) vulnerabilities documented across all four platforms in 2025 alone [3][4][5][6].
- A shared code reuse pattern ("ShadowMQ") propagated a single unsafe deserialization pattern across Meta, NVIDIA, vLLM, SGLang, and Modular Max Server simultaneously, demonstrating that AI infrastructure inherits and amplifies software supply chain risk [5].
- AI-assisted exploit generation can now produce working proof-of-concept code for published CVEs in 10 to 15 minutes at a cost of approximately one dollar per attempt, compressing the window between disclosure and weaponization below what most enterprise patch cycles can accommodate [1].
- Enterprise remediation cycles have not kept pace: mean time to remediation for complex applications reached five months and ten days in 2026 benchmarks, while approximately 45% of enterprise vulnerabilities remain unpatched after 12 months [1].
- Organizations running AI inference workloads must treat these platforms as critical infrastructure requiring network isolation, runtime monitoring, and supply-chain provenance controls equivalent to those applied to database and authentication systems.

Background

The rapid productionization of large language models has created a new layer of technical infrastructure that did not exist at meaningful scale three years ago. Inference platforms—the software components that receive requests, load model weights, execute forward passes through a neural network, and return

predictions—now sit at the center of AI-powered products across virtually every major industry sector. Platforms such as NVIDIA Triton Inference Server, vLLM, Ollama, SGLang, and Meta Llama Stack are deployed in cloud environments and increasingly on enterprise on-premises hardware to serve real-time AI workloads.

This infrastructure layer inherited the security characteristics of the open-source software ecosystem from which it grew: rapid development cycles, heavy code reuse between projects, minimal historical scrutiny from the security research community, and default configurations that prioritize accessibility over access control. For much of AI's early commercial history, that trade-off was low-stakes because inference endpoints were not widely internet-exposed. That calculus changed as organizations began exposing these systems to broader networks—sometimes inadvertently—and as adversaries recognized that a compromised inference server offers access to proprietary model weights, customer data flowing through the system, and a privileged execution environment within the AI production stack.

The security community began systematically auditing these platforms in earnest in 2024. What followed was a concentrated sequence of disclosures in 2025 revealing that the inference layer is not merely vulnerable in isolated ways, but structurally vulnerable: the same classes of flaws—unsafe deserialization, missing authentication on administrative endpoints, insufficient input validation—appear repeatedly across projects, often because one project's code was adopted by another without the underlying vulnerability being identified or remediated.

The threat is compounded by a broader shift in attacker capability. AI tools themselves now accelerate the vulnerability research and exploit development process, enabling adversaries to convert published CVEs into working exploit code in minutes rather than weeks [1]. The combination of a rich new attack surface in AI inference infrastructure and AI-enabled exploit acceleration creates conditions where the traditional patch-first, protect-later security model is no longer viable.

Security Analysis

The Inference Platform Attack Surface

AI inference platforms expose a combination of attack vectors that, individually, would be recognized as serious in any other production software context. Together they constitute a surface requiring deliberate architectural countermeasures that standard enterprise security controls do not automatically provide.

The most consequential class of vulnerability identified in 2025 involves unsafe deserialization. Python's `pickle` library, used widely in the data science ecosystem to serialize and deserialize tensor objects and model artifacts, will execute arbitrary code embedded in a malicious payload when it deserializes that payload. When inference frameworks expose deserialization paths through network-accessible API endpoints, an attacker who can reach those endpoints can achieve remote code execution without authentication. Researchers at Oligo Security documented exactly this pattern across five frameworks simultaneously in late 2025: vLLM (CVE-2025-30165, CVSS 8.0), NVIDIA TensorRT-LLM (CVE-2025-23254, CVSS 8.8), Meta Llama Stack (CVE-2024-50050), SGLang, and Modular Max Server (CVE-2025-60455) all shared a vulnerable code pattern tracing back to the same ZeroMQ deserialization logic [5]. The researchers described this as the "ShadowMQ" pattern: a vulnerable implementation in one project was copied into others without the security flaw being identified during code adoption, propagating a single root cause across much of the commercial AI inference ecosystem simultaneously.

NVIDIA Triton Inference Server presented a different but equally serious vulnerability architecture. Wiz Research disclosed a three-CVE chain in August 2025: CVE-2025-23319 enabled an unauthenticated remote attacker to leak the unique name of the server's internal shared memory region by triggering a crafted exception; CVE-2025-23320 allowed that leaked identifier to be used to register the internal memory region and gain unauthorized read/write access; and CVE-2025-23334 chained those capabilities into remote code execution by manipulating IPC message queues [3]. The chain required no credentials and only network access to the Triton Python backend. The responsible disclosure window—from report submission to public patch—was 81 days, during which most defenders had no public signal of the risk and could not act without independent discovery [3].

Ollama, a widely deployed local and enterprise inference runtime, presents a different risk profile rooted in architectural defaults. Because Ollama does not require authentication by default on its API endpoints (port 11434), any user or process with network access to a running instance can perform model management operations, access model artifacts, or abuse compute resources. CVE-2025-63389 formalized this gap, documenting that authentication bypass through unauthenticated model management APIs affects versions through v0.12.3 [6][15]. A separate flaw, CVE-2025-51471, enables cross-domain authentication token theft by exploiting improper domain validation in the model pull authentication flow, allowing attackers who control a malicious model registry to harvest tokens granting access to the victim's private Ollama registries [6]. These vulnerabilities affect deployments where Ollama has been configured with any network exposure beyond localhost—a common configuration in enterprise and cloud environments where the instance is intended to serve multiple users or applications.

The Compressed Exploitation Lifecycle

The exploitation of AI inference platform vulnerabilities does not follow the patterns that established vulnerability management programs were designed to handle. Classic enterprise patch management assumes a window measured in weeks between a CVE's publication and meaningful exploitation activity. That window no longer reliably exists.

CSA research documents that the median time to exploit fell from roughly 32 days in 2022 to approximately 5 days by 2023 [1][2]. The 2025 figures reveal a further and more troubling shift: 32.1% of newly tracked exploits emerged on or before the CVE's public disclosure date—an 8.5 percentage-point increase from 2024 [1]. Mandiant's threat intelligence reporting for 2025 documented a more extreme outcome: the average time to exploit across that year was negative one day, meaning exploitation was already underway before patches became publicly available. The heaviest concentration of weaponization activity occurs within the first month of a CVE's publication, with a growing share appearing on or before the disclosure date itself [1].

A significant driver of this acceleration is AI-assisted exploit development, which has reduced the time and cost of converting published CVEs into working proof-of-concept code. LLM-based tools and multi-agent frameworks can now analyze published CVE descriptions and associated source code diffs, generate proof-of-concept exploit code, and verify exploitability—all in under 15 minutes at costs as low as one dollar per attempt [1]. The CVE-Genie framework, a multi-agent system designed for this purpose, reproduced 51% of the CVEs in the study's evaluated sample with verifiable working exploits at an average cost of $2.77 per CVE [1]. Separately, an AI agent swarm identified more than 100 exploitable kernel vulnerabilities across major hardware vendors in 30 days at a total cost of $600 [1]. These capabilities are not confined to nation-state actors or elite research teams; they represent the commoditization of exploit development.

Between October 2025 and January 2026, GreyNoise's honeypot infrastructure captured 91,403 attack sessions targeting exposed LLM endpoints, representing at least two systematic campaigns mapping the attack surface of misconfigured AI inference deployments [7]. The volume and coordination of these scans indicate that adversaries are actively inventorying AI infrastructure at scale and are positioned to convert that reconnaissance into exploitation once a viable CVE is published—or even before, if the vulnerability can be discovered independently.

Structural Risk Factors

Several factors distinguish AI inference platforms from other categories of production infrastructure in ways that require explicit attention from security architects.

First, inference platforms are inherently supply-chain-dense. A single inference deployment typically involves a base container from a vendor like NVIDIA, a framework like vLLM or SGLang, Python packages from PyPI, model weights from Hugging Face or a proprietary registry, and configuration from a private repository. Each layer represents a potential injection point. The ShadowMQ vulnerability pattern demonstrates that a single flaw introduced into one upstream project can propagate to multiple downstream consumers without the security flaw being identified during code adoption. The CSA LLM Threats Taxonomy identifies this supply chain inheritance dynamic as a structural risk requiring dedicated controls at each layer boundary [8].

Second, AI inference platforms often run with elevated privileges relative to the sensitivity of the operations they support. GPU workloads require significant host resource access. Shared-memory IPC patterns between inference workers create intra-host communication channels. Model weights are high-value intellectual property. A compromised inference server therefore offers not just code execution, but access to proprietary model artifacts, customer inference data, and lateral movement paths within the AI production environment.

Third, the operational patterns of AI teams create detection gaps. GPU instances are frequently treated as ephemeral workloads, rebuilt from container images rather than patched in place, creating environments where vulnerability scanning tools may not have dwell time to generate findings before the instance is replaced. At the same time, those environments may be rebuilt from images that contain the same unpatched vulnerability repeatedly. Monitoring tooling for AI inference workloads has not yet reached the maturity of equivalent tooling for traditional application servers, creating visibility gaps that adversaries can exploit before defenders detect anomalous activity.

Recommendations

Immediate Actions

Organizations operating AI inference platforms should treat any of the platforms discussed in this note as potentially affected until a confirmed patch status is established for their specific deployment. For NVIDIA Triton deployments, update to version 25.07 or later, which addresses CVE-2025-23319, CVE-2025-23320, and CVE-2025-23334 [3][13]. For vLLM, update to version 0.8.0 or later for the ShadowMQ/CVE-2025-30165 fix [4][14], and verify separately whether the tensor deserialization path in CVE-2025-62164 has been addressed for deployments using the Completions API with embedding inputs. For Ollama, restrict API exposure to localhost or an authenticated reverse proxy; verify current patch availability for CVE-2025-63389 against Ollama's published advisories [6][15], as no patched

version had been confirmed as of the referenced advisory's publication date, making network isolation the primary compensating control. For NVIDIA TensorRT-LLM, update to 0.18.2 or later. For Meta Llama Stack, update to v0.0.41 or later.

Beyond immediate patching, organizations should audit current network exposure of inference endpoints. Inference servers that are directly reachable from broader corporate networks, internet-facing subnets, or through permissive security group rules should be isolated behind a network boundary with explicit ingress controls. The default-no-authentication posture of platforms like Ollama is not an advisory recommendation—it is a precondition that GreyNoise's recorded scan campaigns have already demonstrated adversaries are actively probing.

Short-Term Mitigations

Organizations should implement or verify the following controls within the next 30 days. Network segmentation should place AI inference workloads in a dedicated segment with tightly scoped egress rules; inference servers should be able to reach external model registries only over explicitly permitted, authenticated paths, and should have no general internet egress. Web application firewall or API gateway rules should be applied to any externally or broadly exposed inference endpoints, with payload size limits and content type enforcement to reduce the deserialization attack surface. Supply chain provenance controls should be applied to container images and model weights: cryptographic signing (e.g., Sigstore, NVIDIA's model signing capabilities) should be required for all artifacts entering the inference pipeline.

Runtime monitoring for AI inference workloads should be established if not already in place. Anomalous API call patterns—unexpectedly large payloads, calls to administrative or model management endpoints from non-privileged clients, high-frequency requests from single sources—warrant alerting and investigation. The 91,403 attack sessions recorded by GreyNoise's honeypots demonstrate that organized scanning activity produces detectable patterns; those patterns should be searchable in whatever logging infrastructure an organization has deployed for its inference environment.

Organizations should also establish a dedicated watch process for CVEs affecting their specific inference platform dependencies. Given that exploit timelines frequently run to hours or days for high-severity vulnerabilities, a passive approach of waiting for scheduled patch cycles is not appropriate. CISA's Known Exploited Vulnerabilities catalog and VulnCheck's NVD feed provide exploitability signals beyond raw CVSS scores that should inform triage prioritization.

Strategic Considerations

The structural vulnerabilities described in this note will not be resolved by a single patch cycle. AI inference infrastructure will continue to evolve rapidly, and the security research community's interest in this attack surface has only intensified since 2024. Organizations should plan for an ongoing cadence of significant disclosures affecting AI inference components and build the operational capability to respond rapidly.

At a platform selection and architecture level, prefer inference deployments that support authentication natively and are being maintained by vendors with active vulnerability management programs and a demonstrated history of responsible disclosure coordination. Both NVIDIA and the vLLM project demonstrated reasonable disclosure practices for the 2025 vulnerabilities discussed here; that track record should be a factor in platform decisions. Projects with minimal security disclosure history or no published security contact mechanism represent elevated risk.

For organizations building AI inference into regulated or high-sensitivity environments, model the inference platform itself as a critical component requiring its own threat model. The CSA MAESTRO framework provides a seven-layer agentic AI architecture that is directly applicable to inference deployments: foundation model risks (Layer 1), data operations risks around training data and weights (Layer 2), and deployment infrastructure risks (Layer 4) all have specific mappings to the vulnerability classes described in this note [9]. Threat models should be updated when new CVEs are disclosed, not only when major architectural changes occur.

Patch cadence governance should be updated to establish inference platforms as a distinct asset class with a separate, accelerated remediation SLA. The five-month-and-ten-day mean time to remediation documented in 2026 enterprise benchmarks [1] is incompatible with an exploitation environment where a significant and growing share of exploits are launched within 24 hours of disclosure. Realistic targets for AI inference critical-severity vulnerabilities should be measured in days, not months, with compensating network controls applied within hours while patching proceeds.

CSA Resource Alignment

This research note connects directly to several active CSA frameworks and publications that provide implementation guidance beyond what is possible in this format.

The **MAESTRO Framework** (Multi-Agent Environment, Security, Threat, Risk, & Outcome) offers a structured threat modeling methodology applicable to AI inference deployments. Its seven-layer reference architecture maps inference platform components to specific threat categories and enables security architects to reason systematically about which controls apply at each layer [9].

The **CSA AI Controls Matrix (AICM)** and associated auditing guidelines provide control specifications applicable across model provider, cloud service provider, and application provider roles. Organizations operating inference platforms will find relevant controls in the infrastructure security and supply chain domains [10].

The **CSA LLM Threats Taxonomy** formalizes the categories of threats to LLM-backed systems, including infrastructure-level attacks on inference components, and provides a common vocabulary for communicating risk findings across technical and governance audiences [8].

Securing LLM-Backed Systems: Essential Authorization Practices provides direct implementation guidance on authorization architecture for LLM deployments, addressing the authentication and access control gaps exemplified by the Ollama vulnerability pattern [11].

The **Agentic AI Red Teaming Guide** includes supply chain and dependency attack methodologies that are directly applicable to testing AI inference platform security posture, including dependency injection and runtime security validation procedures [12].

The **CSA Cloud Controls Matrix (CCM)** domains covering Application and Interface Security (AIS), Infrastructure and Virtualization Security (IVS), and Supply Chain Management, Transparency and Accountability (STA) provide control frameworks applicable to inference infrastructure within broader cloud governance programs.

Organizations seeking to assess their AI inference security posture against CSA STAR criteria should review the AICM implementation guidelines for cloud service providers and model providers, which address the infrastructure and operational controls relevant to inference deployments.

References

[1] Cloud Security Alliance Labs. "The Collapsing Exploit Window: AI-Speed Vulnerability Weaponization." CSA Labs, 2026.

[2] CyberMindr. "Average Time-to-Exploit in 2025." CyberMindr Blog, 2025.

[3] Wiz Research. "Breaking NVIDIA Triton: CVE-2025-23319 – A Vulnerability Chain Leading to AI Server Takeover." Wiz Blog, August 2025.

[4] ZeroPath. "vLLM CVE-2025-62164: Memory Corruption via Unsafe Tensor Deserialization." ZeroPath Blog, 2025.

[5] CSO Online. "Copy-Paste Vulnerability Hits AI Inference Frameworks at Meta, Nvidia, and Microsoft." CSO Online, November 2025.

[6] Security Boulevard. "Ollama Unauthorized Access Vulnerability Due to Improper Configuration (CNVD-2025-04094)." Security Boulevard, March 2025.

[7] VentureBeat. "11 Runtime Attacks Driving CISOs to Deploy Inference Security Platforms in 2026." VentureBeat, 2026.

[8] Cloud Security Alliance. "Large Language Model (LLM) Threats Taxonomy." CSA AI Controls Framework Working Group, 2024.

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

[10] Cloud Security Alliance. "AI Controls Matrix (AICM)." CSA AI Controls Framework Working Group, 2024.

[11] Cloud Security Alliance. "Securing LLM-Backed Systems: Essential Authorization Practices." CSA AI Technology and Risk Working Group, 2024.

[12] Cloud Security Alliance. "Agentic AI Red Teaming Guide." CSA AI Organizational Responsibilities Working Group, 2025.

[13] NVIDIA. "Security Bulletin: NVIDIA Triton Inference Server – August 2025." NVIDIA Product Security, August 2025.

[14] GitHub Advisory Database. "vLLM Deserialization Vulnerability – CVE-2025-62164." GitHub, 2025.

[15] GitHub Advisory Database. "Ollama Missing Authentication – CVE-2025-63389." GitHub, 2025.