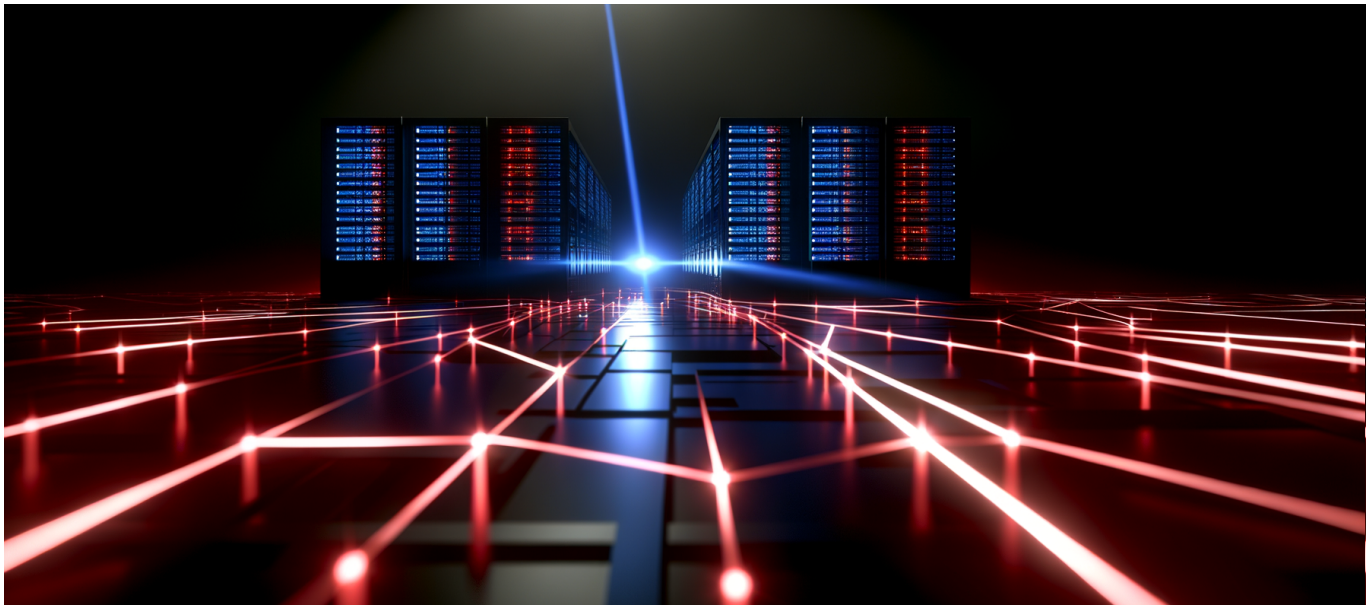

GreyNoise Captures 91,403 Attacks Targeting Every Major LLM

Attackers launched 91,403 sessions against AI infrastructure in 90 days—and they hit every major model from GPT-4o to Claude to Llama. The reconnaissance phase is over; what comes next is exploitation at scale.

The Numbers That Should Keep You Up Tonight

GreyNoise published findings on January 8, 2026, that reveal just how systematically attackers are mapping the AI attack surface. Between October 2025 and January 2026, their honeypots captured two distinct campaigns that paint a sobering picture of what's happening to enterprise AI deployments.

The first campaign exploited Server-Side Request Forgery (SSRF) vulnerabilities in Ollama's model pull functionality and Twilio SMS webhooks. It consisted of 1,688 sessions originating from 62 IP addresses across 27 countries, with a notable spike during the Christmas holiday period—exactly when your security team was understaffed.

The second campaign was pure reconnaissance: 80,469 enumeration sessions over just 11 days, from December 28, 2025 into January 2026. Two IP addresses—134.122.136.119 and 134.122.136.96—systematically probed 73+ LLM endpoints, building what amounts to a target list for future attacks.

> The attacking IPs weren't amateurs. They carry exploitation histories spanning 200+ CVEs and over 4 million total GreyNoise hits. These are professional operators with established infrastructure.

What makes this particularly concerning is the breadth. According to SecurityWeek's analysis, attackers targeted every major model family: GPT-4o, Claude (Sonnet, Opus, and Haiku variants), Llama 3.x, DeepSeek-R1, Gemini, Mistral, Qwen, and Grok. They used both OpenAI-compatible and Google Gemini API formats, demonstrating fluency with production deployment patterns.

The SSRF attacks showed remarkable consistency. A single JA4H fingerprint—po11nn060000—appeared in 99% of these sessions. This means one toolchain, likely automated, is being weaponized against AI infrastructure at scale.

Why This Matters More Than Typical Scanning Activity

Security teams see scanning activity constantly. Port probes, credential stuffing, vulnerability enumeration—it's background noise. But this is different, and here's why.

These attacks specifically target AI-native infrastructure. The Ollama SSRF vulnerability isn't some legacy web app flaw. It's a weakness in how LLM serving infrastructure handles model downloads. Twilio webhook exploitation targets the voice/SMS integration layer that many AI agents depend on for real-world interaction. Attackers are learning the unique architecture of AI systems.

eSecurity Planet reported that the enumeration campaign used Nuclei-like tooling to test 240+ exploits across probed endpoints. This isn't random spray-and-pray. It's methodical vulnerability assessment designed to find which organizations left doors open.

The Christmas timing wasn’t coincidental. Holiday periods historically see reduced security monitoring and delayed incident response. The attackers knew exactly when to strike.

> The real danger isn’t the 91,403 sessions themselves—it’s the intelligence those sessions gathered. Attackers now know which organizations run which models, which endpoints are exposed, and which have exploitable vulnerabilities.

This intelligence becomes the foundation for targeted attacks. Ransomware operators don’t waste time on hardened targets. They buy or build reconnaissance lists, then hit organizations with known weaknesses. What GreyNoise captured is the list-building phase.

Technical Deep Dive: Attack Methodology and Infrastructure

Understanding how these attacks work requires dissecting both campaigns separately, because they represent different threat actor objectives.

Campaign One: SSRF Exploitation

The SSRF attacks exploited two specific weaknesses. First, Ollama’s model pull functionality allows users to download models from remote sources. When improperly configured, attackers can abuse this to make the server request arbitrary URLs, potentially accessing internal services, cloud metadata endpoints (like AWS’s 169.254.169.254), or exfiltrating data to attacker-controlled servers.

Second, Twilio webhook handlers often run with elevated privileges to process SMS and voice callbacks. If these handlers don’t properly validate incoming requests, attackers can forge webhook payloads that trigger unintended actions—from data exfiltration to command execution.

Two IP addresses dominated the SSRF campaign: 45.88.186.70 (responsible for 49,955 sessions) and 204.76.203.125 (30,514 sessions). Together, they account for over 88% of SSRF activity. This concentration suggests either a single actor using multiple exit points or a shared infrastructure being rented by multiple operators.

The JA4H fingerprint consistency across 99% of SSRF attacks reveals something important about the tooling. JA4H fingerprints identify HTTP client behavior patterns. When nearly all attacks share one fingerprint, they're using the same exploitation framework—likely a custom tool or heavily modified version of existing SSRF exploitation software.

Campaign Two: Systematic Enumeration

The enumeration campaign was technically simpler but strategically more significant. Over 11 days, two IPs sent 80,469 requests probing for the presence of LLM endpoints.

These requests tested for:

- OpenAI-compatible API endpoints (/v1/chat/completions, /v1/models, /v1/embeddings)
- Google Gemini API formats
- Provider-specific endpoints for Anthropic, Mistral, and others
- Self-hosted deployment signatures (Ollama, vLLM, Text Generation Inference)
- Administrative interfaces and health check endpoints

The 73+ distinct endpoint patterns suggest attackers have comprehensively cataloged how organizations deploy LLMs. They're not just looking for "an AI endpoint"—they're fingerprinting exactly which models run where, what API versions are exposed, and what authentication mechanisms (if any) protect them.

The Nuclei-like tooling used in these probes deserves attention. Nuclei is a legitimate vulnerability scanner that uses YAML templates to define detection and exploitation logic. Attackers frequently fork or emulate Nuclei's approach for reconnaissance at scale. Testing 240+ exploits across probed endpoints means this wasn't passive fingerprinting—it was active vulnerability assessment.

What Most Coverage Gets Wrong

The security press coverage of this report falls into a predictable pattern: breathless alarm followed by generic recommendations. "Patch your systems! Monitor your logs! Use authentication!" This advice isn't wrong, but it misses the structural issues that make AI infrastructure uniquely vulnerable.

First, the authentication problem is harder than it looks. Many AI deployments use API keys for authentication—tokens that don't rotate, don't expire, and often get embedded in client applications. The number of GitHub repositories containing exposed OpenAI keys is staggering. AI API security inherited the worst practices of the SaaS era.

Second, nobody knows their AI attack surface. Ask a typical engineering team how many LLM endpoints their organization exposes. They'll give you a number. Then ask about the experimental deployments, the hackathon projects, the "temporary" inference servers someone spun up six months ago. The real number is higher—sometimes by multiples.

Third, AI infrastructure monitoring is immature. Traditional security tools know what a SQL injection looks like. They know how to detect port scans and credential stuffing. But what does malicious LLM probing look like? The enumeration requests GreyNoise captured look nearly identical to legitimate API exploration. Distinguishing reconnaissance from normal traffic requires AI-specific detection logic that most organizations haven't built.

> The most dangerous assumption is that your AI deployment is small enough to escape notice. These attacks probed 73+ endpoint patterns systematically. If you run any LLM infrastructure, you were likely probed.

Here's what's underreported: the SSRF attacks against Ollama specifically exploit self-hosted deployments. Organizations running Ollama typically do so because they want local model inference—for privacy, cost, or latency reasons. These are often sophisticated teams making deliberate infrastructure choices. Yet they're still vulnerable, because self-hosting creates a larger attack surface than API consumption.

The irony is thick. Teams avoiding cloud AI providers for security reasons may be creating more vulnerability by running their own inference infrastructure without equivalent security investment.

What You Should Actually Do

Generic advice won't help. Here's a concrete action plan organized by effort level.

This Week: Visibility

Run a discovery scan for AI infrastructure. Look for:

- Processes listening on ports 11434 (Ollama default), 8080, 8000 (common inference servers)
- DNS entries containing “llm”, “ai”, “inference”, “ollama”, “vllm”
- Cloud resources tagged with AI-related labels
- API gateway routes containing /v1/chat, /v1/completions, /generate, /api/generate

You’ll find deployments you forgot existed. Document them. Assign owners.

Query your logs for the specific attacking IPs: 45.88.186.70, 204.76.203.125, 134.122.136.119, 134.122.136.96. If these IPs hit your infrastructure between October 2025 and January 2026, you were probed. Determine what they found.

This Month: Hardening

Implement authentication on every LLM endpoint. This sounds obvious, but SCWorld’s coverage of the attacks specifically noted that exposed, unauthenticated endpoints were primary targets.

For Ollama specifically: disable the model pull functionality if you don’t need it. If you do need it, restrict it to internal networks and implement allow-listing for model sources. The SSRF vulnerability exploits the assumption that pull requests are benign.

For Twilio integrations: implement webhook signature validation. Every Twilio request includes a signature you can verify against your auth token. If you’re not checking signatures, attackers can forge webhooks.

Segment AI infrastructure from your broader network. LLM inference servers don’t need access to your production databases. They don’t need to reach internal APIs. Implement network policies that restrict what these servers can contact.

This Quarter: Detection

Build detection rules for LLM enumeration patterns. Monitor for:

- Sequential requests to /v1/models from single IPs
- 404 responses followed by immediate requests to alternate endpoint patterns
- Requests to endpoints you don't actually expose (attackers testing whether Mistral endpoints exist on your Claude deployment)
- High-volume requests to health check or metadata endpoints

The JA4H fingerprint po11nn060000 should go directly into your block list and detection rules. It's now a known-bad signature for AI infrastructure attacks.

Consider deploying honeypots. GreyNoise captured this intelligence because they run infrastructure designed to attract attackers. A honeypot LLM endpoint—one that logs requests but serves no real model—can give you visibility into who's probing your network.

Architecture Considerations

If you're building new AI infrastructure, design for hostility. Assume enumeration and exploitation attempts are constant.

Run inference behind API gateways that handle authentication, rate limiting, and request validation before traffic reaches your models. The gateway becomes your security control point.

Implement model access as a capability, not a URL. Instead of exposing /v1/chat/completions publicly, require callers to obtain time-limited, scoped tokens that grant access to specific models for specific purposes. This limits the blast radius when tokens leak.

Log everything. Model requests, response metadata (not content—that creates privacy issues), authentication events, error conditions. When the next GreyNoise report names your infrastructure, you want forensic capability.

Where This Goes Next

The reconnaissance phase has concluded. Over the next 6-12 months, expect three developments.

Targeted exploitation campaigns will begin. Attackers now have lists of organizations running specific models with specific vulnerabilities. Those lists have

value. They'll either be used directly or sold to ransomware operators looking for initial access. Q2 2026 will likely see the first major AI infrastructure breach directly attributable to this reconnaissance activity.

AI-specific vulnerabilities will get more attention. The security research community follows attacker activity. As AI infrastructure becomes a more attractive target, researchers will invest more effort finding vulnerabilities. Expect a wave of CVEs affecting inference servers, model serving frameworks, and AI orchestration tools. The Ollama SSRF is just the beginning.

Cloud providers will tighten defaults. AWS, Azure, and GCP all offer managed AI infrastructure. As self-hosted deployments prove vulnerable, cloud providers will market security as a differentiator. Expect new security-focused features: managed API authentication, built-in anomaly detection, network isolation by default.

> The organizations that get ahead of this are the ones treating AI infrastructure with the same security rigor as their primary production systems. The ones that don't will learn hard lessons.

One underrated possibility: AI systems as attack infrastructure, not just targets. An compromised LLM endpoint with network access becomes a powerful tool for lateral movement, social engineering, and data exfiltration. The 91,403 attack sessions GreyNoise captured might be preliminary reconnaissance for turning victim AI systems into attack platforms.

The Bigger Picture

GreyNoise's data reveals something fundamental about where we are in the AI infrastructure lifecycle. The technology matured faster than the security practices needed to protect it.

Two years ago, most organizations had no LLM infrastructure. Today, it's everywhere—customer service bots, internal knowledge systems, code assistants, data analysis tools. This infrastructure went from experimental to critical faster than security teams could adapt.

The attackers noticed. The 91,403 sessions represent a systematic effort to map

this new attack surface. The targeting of every major model family shows they’re not betting on any single technology winning. They’re preparing to exploit whatever organizations deploy.

For CTOs and senior engineers, this report should change how you think about AI projects. The question isn’t whether to use AI—that ship has sailed. The question is whether you’re treating AI infrastructure as production-critical systems requiring production-grade security.

Most organizations aren’t. The experimental mindset that enabled rapid AI adoption now creates risk. Proof-of-concept deployments become permanent. Internal tools get external exposure. Authentication remains “something we’ll add later.”

Later has arrived. The attackers aren’t waiting.

The 91,403 attack sessions GreyNoise captured aren’t the threat—they’re the warning: your AI infrastructure is now on someone’s target list, and the only question is whether you harden it before they exploit it.