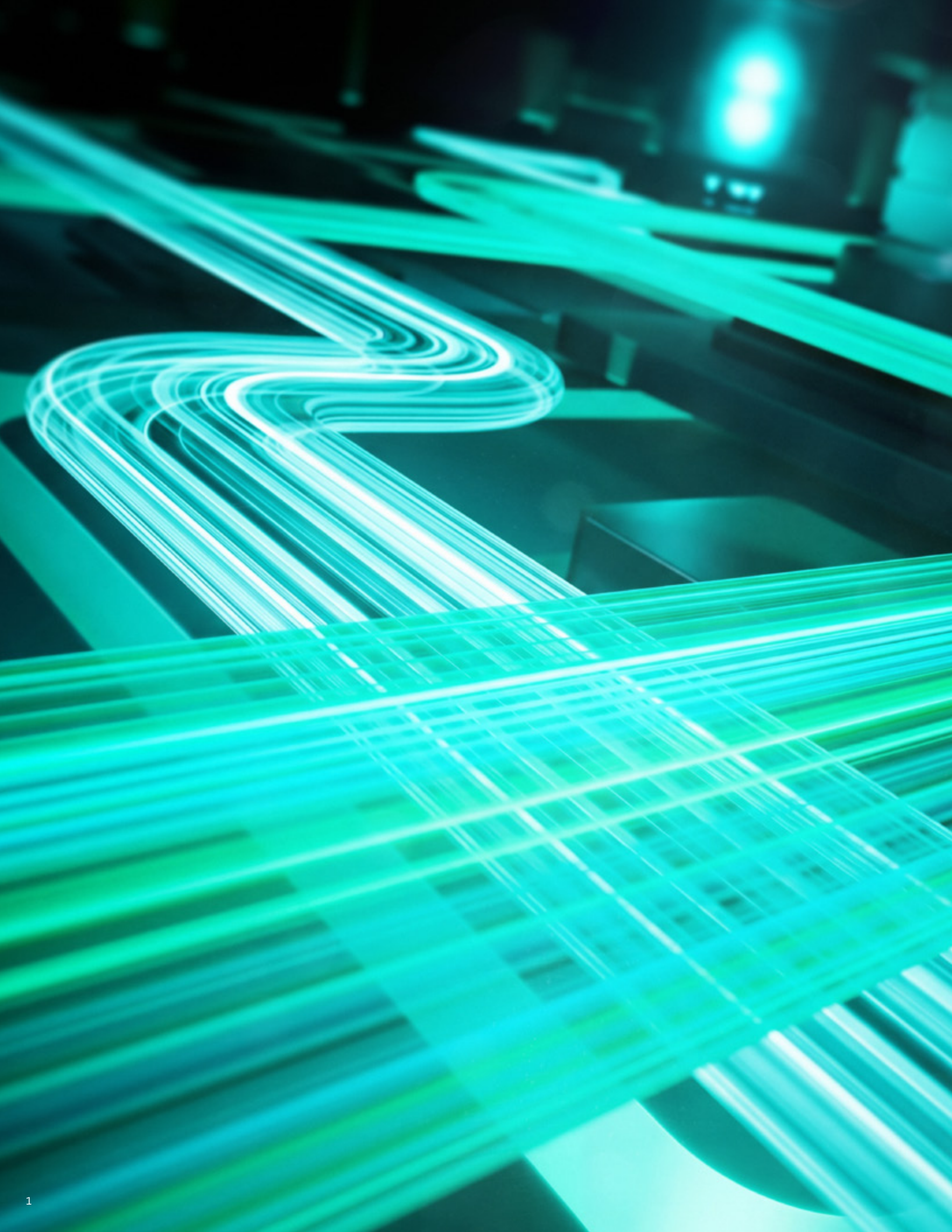
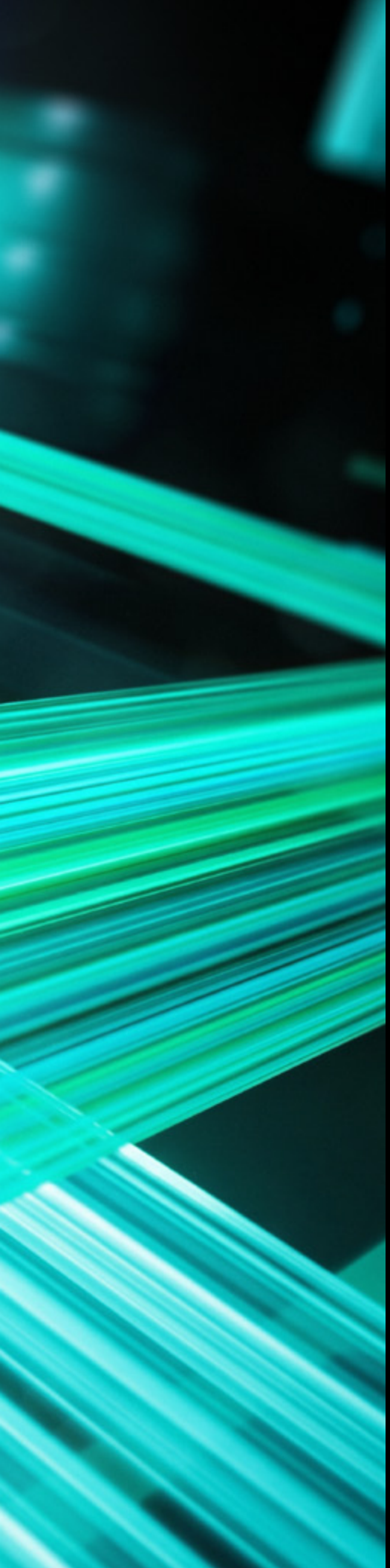
2026

HIDDENLAYER

QUARTERLY UPDATE: APRIL-JUNE

AI THREAT LANDSCAPE REPORT





ABOUT THIS REPORT

This Quarterly AI Threat Landscape Update serves as a companion to HiddenLayer's annual AI Threat Landscape Report. Each quarter, we examine the most significant developments shaping AI security, from emerging attack techniques and vulnerabilities to major incidents, research, and defensive initiatives, to provide organizations with timely insight into the rapidly evolving AI threat landscape.

Access the full 2026 AI Threat Landscape Report here.

TABLE OF CONTENTS

05 QUARTERLY AI THREAT LANDSCAPE REPORT UPDATE

05 What's New in AI

06 TECHNOLOGY AS A DOUBLE-EDGED SWORD

06 Frontier Models Reshape Cyber Capability

07 USE OF AI IN CYBERCRIME

07 Malware Utilizing AI

07 Malware Generated with AI

08 AI-Related Data Leakage

09 PROMPT ATTACKS

09 Tokenization Attacks

10 Prompt Attacks in the Wild

11 SUPPLY CHAIN ATTACKS

11 OpenClaw as a New Attack Vector

12 Abusing Agent Skills

13 Software Supply Chain Incidents

14 Vulnerabilities

15 SECURITY INITIATIVES & FRAMEWORKS

15 APE Taxonomy

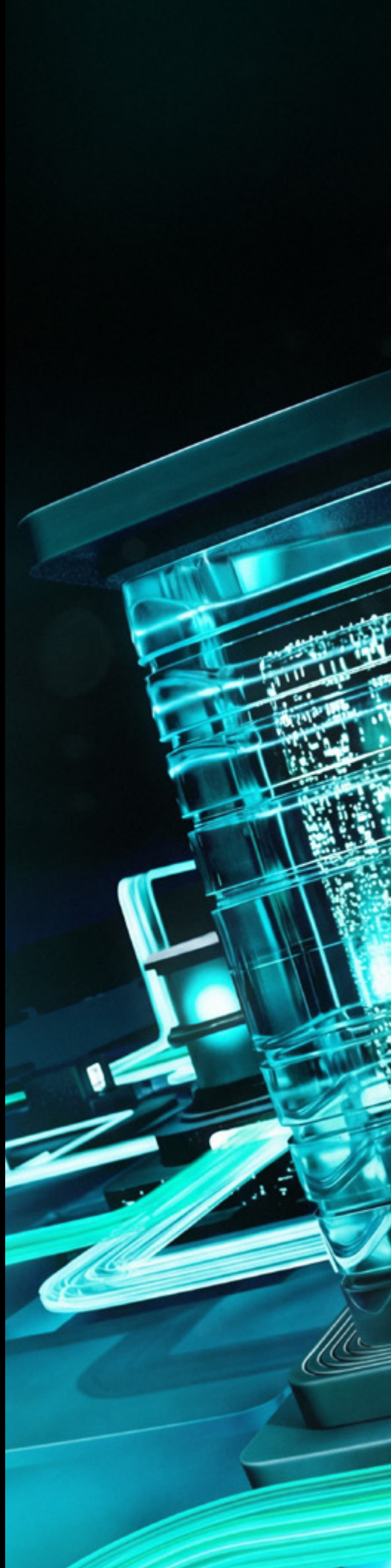
15 OWASP

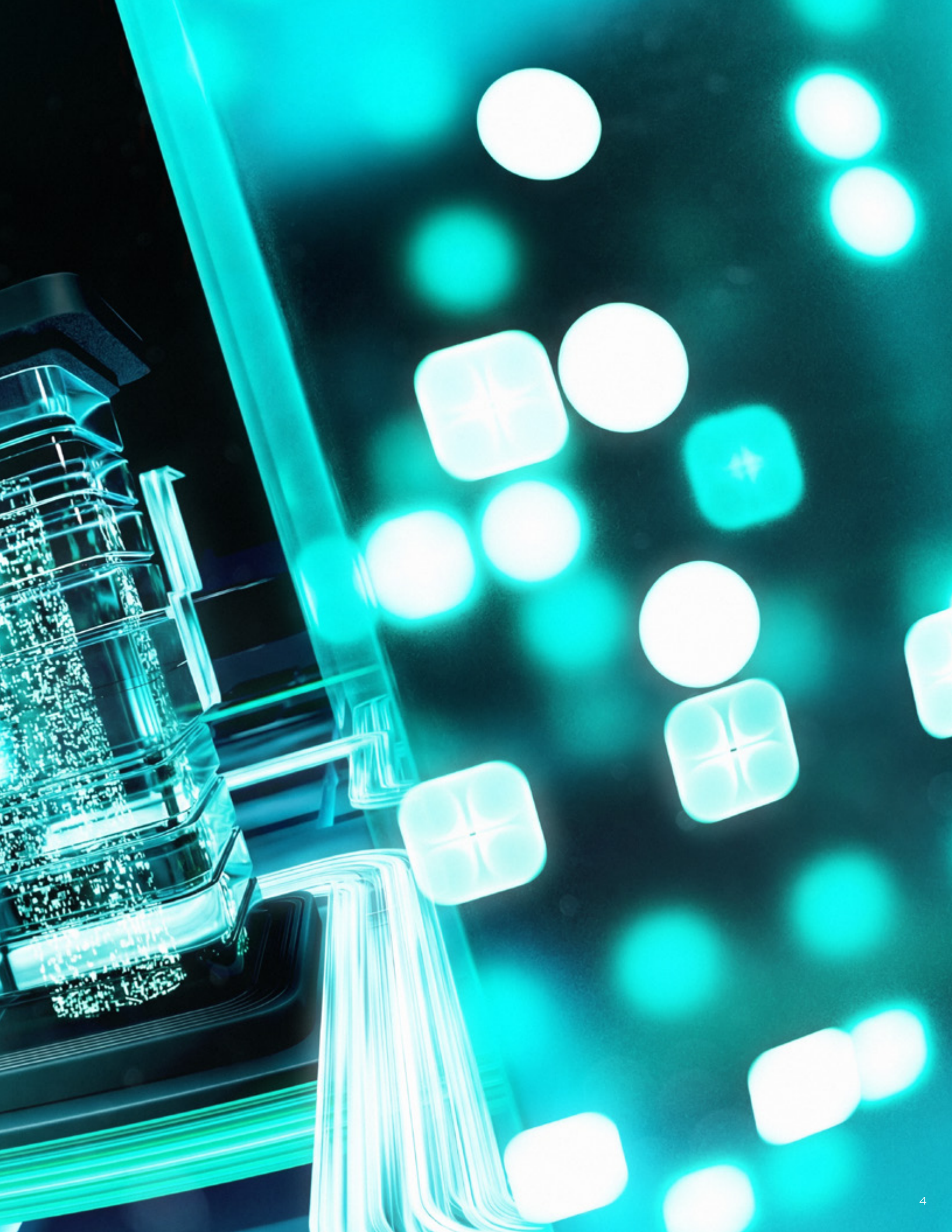
16 MITRE

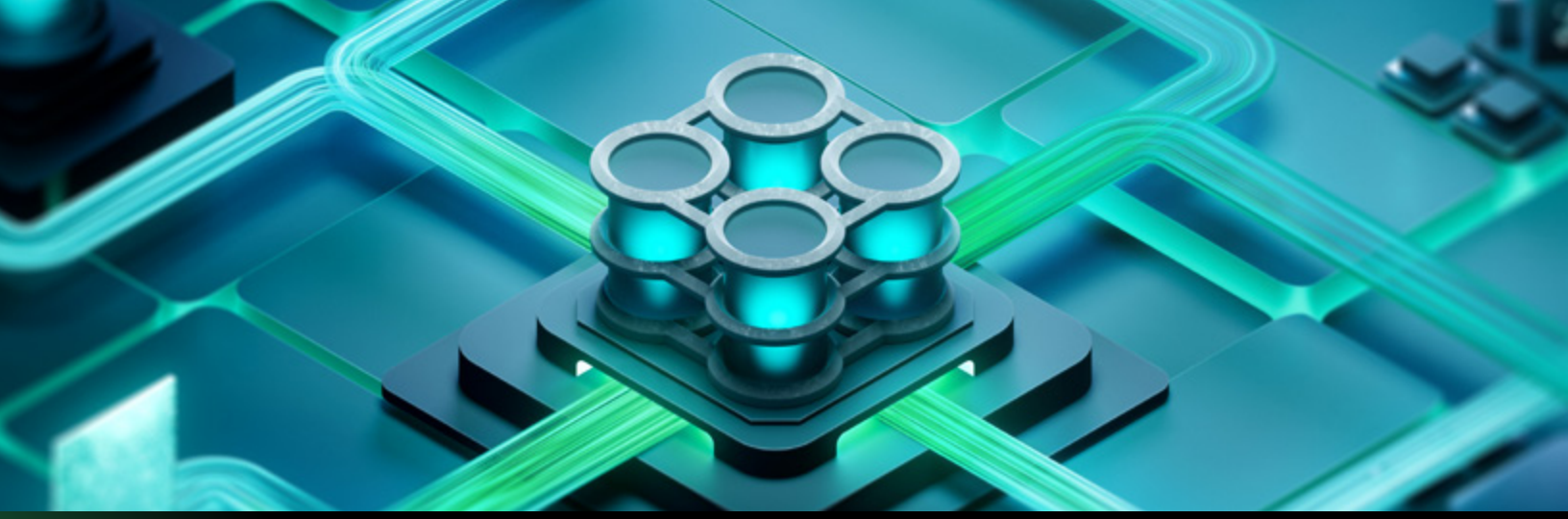
16 CoSAI

17 RECOMMENDATIONS

18 ABOUT HIDDENLAYER







QUARTERLY AI THREAT LANDSCAPE REPORT UPDATE

What's New in AI

This past quarter saw the frontier move in two directions at once: upward in raw capability and outward into autonomous, loosely governed agent ecosystems. At the top end, Anthropic introduced its Mythos tier, which sits above Opus, but access to this newest model was limited to a handful of trusted partners. The safety-hardened variant of Mythos, called Fable, was briefly released to the public before being pulled back after only 3 days at the request of the US government due to security concerns. This is the clearest signal yet that both model providers and governments alike now treat advanced cyber capability as a highest priority concern in its own right.

A significant development in the agentic ecosystem was the rapid popularization of OpenClaw - a locally run, user-friendly platform that brings agentic technology to the household level. Built by a single developer, Peter Steinberger, and released in November 2025, the project accumulated over 380k stars on GitHub in as little as half a year's time. It quickly gained traction with individual users due to its ease of setup and exceptional flexibility. Part of its success can also be attributed to Moltbook, a viral agent-only social network where agents post and react autonomously without human interference. In their early days, both OpenClaw and Moltbook lacked basic security measures, creating a new attack surface that initially lacked mature security controls.

Simultaneously, the agentic landscape has started shifting its focus from MCP to agent skills. The Agent Skills framework, initially released by Anthropic as an open standard in December 2025, has been swiftly adopted by all major players, including OpenAI, Cursor, GitHub Copilot, and, most notably in the consumer market, OpenClaw. This intuitive framework, which resembles software package managers, makes it a breeze to teach agents new capabilities. Skill marketplaces such as ClawHub and Skills.sh allow anyone to search for, download, and upload skills with little oversight, making them attractive targets for adversaries seeking new supply chain attack routes.

While entirely new exposure surfaces emerged, threat activity exploiting previously known vectors also intensified. Adversaries continued to refine familiar techniques even as they explored novel ones, and incidents involving AI software supply chain compromises, along with prompt injection and other adversarial prompt manipulation, evolved more quickly than in previous years. This dual pressure - escalation on established fronts combined with emerging risks - creates a broader and faster-moving threat landscape than ever before.

TECHNOLOGY AS A DOUBLE-EDGED SWORD

Frontier Models Reshape Cyber Capability

Apart from new attack categories, the AI threat landscape is increasingly defined by a dramatic collapse in the cost and expertise required to carry out such attacks. Nowhere is this clearer than in the case of Anthropic's Claude Mythos Preview - an unreleased frontier model that demonstrates AI systems can now surpass all but the most skilled humans at finding and exploiting software vulnerabilities, and do it at enormous scale with unprecedented speed. In testing, Mythos found thousands of high-severity vulnerabilities across every major operating system and web browser and created working exploits autonomously in just a matter of seconds. Some of the most concerning findings include the discovery of a 27-year-old flaw in OpenBSD - an operating system prized for its security - as well as the development of exploits that managed to escape both renderer and system sandboxes.

Two developments frame why this matters for defenders now. First, these capabilities were not deliberately engineered; they emerged as a byproduct of general improvements in coding and reasoning, signaling that similar capabilities will appear across the industry whether or not any single lab intends them to. Second, the model's existence leaked before its creators were ready: a misconfigured CMS exposed roughly 3,000 internal Anthropic files in late March 2026, including documents describing the model as far ahead of any other in cyber capability. Anthropic confirmed the model's existence shortly afterward and announced Project Glasswing a week later. The episode highlighted the challenge of keeping frontier model research secure even within the organizations building it — and underscored why Anthropic has chosen to limit access to a small group of trusted partners rather than release Mythos Preview to the public.

Two months after Mythos Preview, in June 2026, Anthropic took the next step and tried to put this class of capability into wider circulation. It released two models built on the same underlying system: Claude Mythos 5, an upgrade to the cyber-defense model still locked to vetted Glasswing partners, and Claude Fable 5, which the company described as a Mythos-class model made safe for general use. Fable was the first time Mythos-level power was offered to the public, and the safety story was the whole pitch. Rather than refusing dangerous queries outright, the model quietly handed off anything its classifiers flagged in cybersecurity, biology, and chemistry, or distillation to the weaker Claude Opus 4.8. Anthropic deliberately tuned these guardrails to err on the side of caution, conceding they were stricter than ideal, to the point that some security researchers complained the model rejected almost anything that looked remotely cyber-related.

The release lasted three days. On 12 June, the US government invoked export-control authority and ordered Anthropic to suspend all access to Fable 5 and Mythos 5 by any foreign national, whether inside or outside the United States, including foreign national Anthropic employees. With no way to filter users by nationality in real time, the company pulled both models for everyone, worldwide, with no warning and no restoration date. The stated trigger was a jailbreak technique that would enable adversaries to use Fable to find software vulnerabilities. Anthropic described this bypass as "narrow", saying it turned up only minor bugs that other public models could find anyway. Anthropic worked closely with the government to address this issue, and as of 30 June, export controls on Fable 5 and Mythos 5 have been lifted.

Advances in frontier AI models continue to benefit defenders while also increasing the capabilities available to adversaries. As models become more capable across cybersecurity tasks, providers and governments are responding with greater scrutiny over how these systems are deployed and who can access them.

USE OF AI IN CYBERCRIMES WORD

Malware Utilizing AI

Building on the trend started by PromptLock and LameHug, adversaries are increasingly embedding AI capabilities into their malware. One recent example is a dropper distributing the infamous Sliver payload (an open-source alternative to Cobalt Strike). Before executing the payload, the malware collects information about the victim's environment and sends the recon data to OpenAI's GPT-4 via HTTP API. The prompt asks the model to analyze the data, effectively letting it decide whether the environment looks "safe" enough to launch the real payload. By delegating this decision to the LLM, attackers can bypass sandboxes without hardcoding long lists of processes, services, and other variables associated with emulated environments.

AI capabilities in malware still seem largely experimental at this stage, with some samples failing to fully implement or make efficient use of the technology. An AI-enabled .NET infostealer documented by Unit 42 queries GPT-3.5 to generate obfuscation and evasion techniques, but the model's outputs are either just written to a log file or have no practical effect on the malware's behavior, suggesting the functionality is still being tested. Similarly, a new variant of the Android remote access trojan SURXRAT contains a subroutine that, under certain conditions, downloads a large LLM from an external repository, but doesn't yet make any actual use of the model.

While AI-embedded malware is still evolving, AI is already changing how cybercrime is conducted. Rather than transforming the malware itself, generative AI is helping attackers automate reconnaissance, generate convincing lures, develop malicious code, and scale operations, allowing less experienced operators to execute campaigns that previously required greater technical expertise.

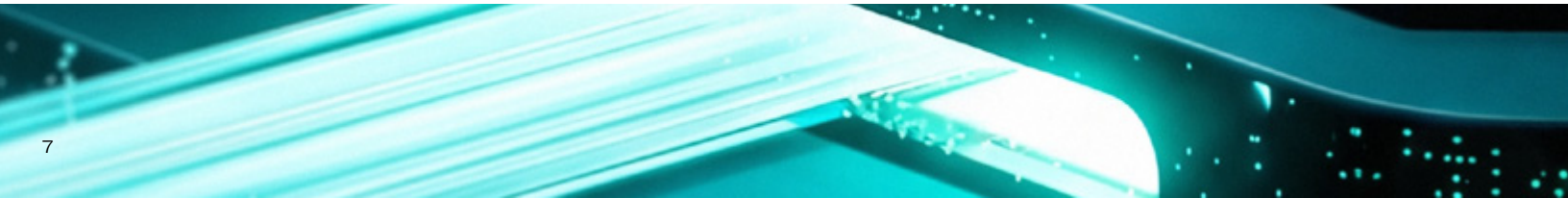
Malware Generated with AI

AI-generated malware has now become commonplace, with several examples uncovered in recent months. In March, IBM X-Force uncovered a likely AI-generated malware strain, dubbed Slopoly, deployed during a ransomware attack by a financially motivated group called Hive0163. Analysis strongly suggests it was written by a large language model, bearing hallmarks such as extensive comments and accurately named variables. Although the malware itself was relatively unsophisticated, IBM noted that attackers increasingly do not need advanced techniques to achieve meaningful impact when AI can accelerate development and automate portions of an attack.

Separately, Darktrace's honeypot network caught a low-skill attacker using an AI-generated malware sample exploiting the React2Shell vulnerability to deploy an XMRig cryptominer, with the payload's heavy commenting and a telling "Educational/ Research Purpose Only" disclaimer suggesting the attacker had jailbroken a model by framing the malicious request as educational.

Then, at the beginning of July, Sysdig documented an operator dubbed JADEPUFFER who ran an automated campaign thought to be the first known fully agentic ransomware operation. In this campaign, an LLM autonomously exploited a Langflow vulnerability, harvested credentials, moved laterally to a production database, and executed a complete database-extortion playbook with no human driving individual steps, including diagnosing and fixing its own missteps.

The scale of AI-assisted attacks came into sharp focus in February, when Amazon's security division revealed that a Russian-speaking threat actor used generative AI to breach over 600 FortiGate firewalls across 55 countries in just five weeks. According to Amazon, the attacker had only a low-to-medium skill level, illustrating how AI can compensate for limited technical expertise when combined with readily available attack techniques. Rather than exploiting unknown vulnerabilities, they focused on exposed management interfaces, weak credentials, and stolen configuration files that were parsed and decrypted using AI-assisted scripts.



A phishing-as-a-service operation dubbed the Outsider Enterprise showed just how prevalent AI has become in phishing campaigns. In June, Google filed a civil lawsuit to dismantle the operation, working alongside the FBI, which coordinated law enforcement action. Based in China and run through Telegram, the network sold AI-generated phishing kits that let low-skilled operators blast out fake text campaigns impersonating Google and other trusted brands, then spin up matching lookalike websites on demand. The scale was staggering: 9,000 fake websites, over a million fraudulent URLs, and 2.5 million scam texts sent to Android users in a single two-week window in May, with victim losses estimated in the millions. As NCC Group's threat intelligence team observed, the operation folded AI into the entire phishing lifecycle: not just writing convincing lures, but generating spoofed sites, managing campaigns, and automating real-time decisions, turning what used to require real technical skill into a subscription service anyone could run.

These cases illustrate that AI is increasingly being incorporated throughout the cybercrime lifecycle. While many techniques remain experimental, adoption continues to expand across both opportunistic and sophisticated threat actors.

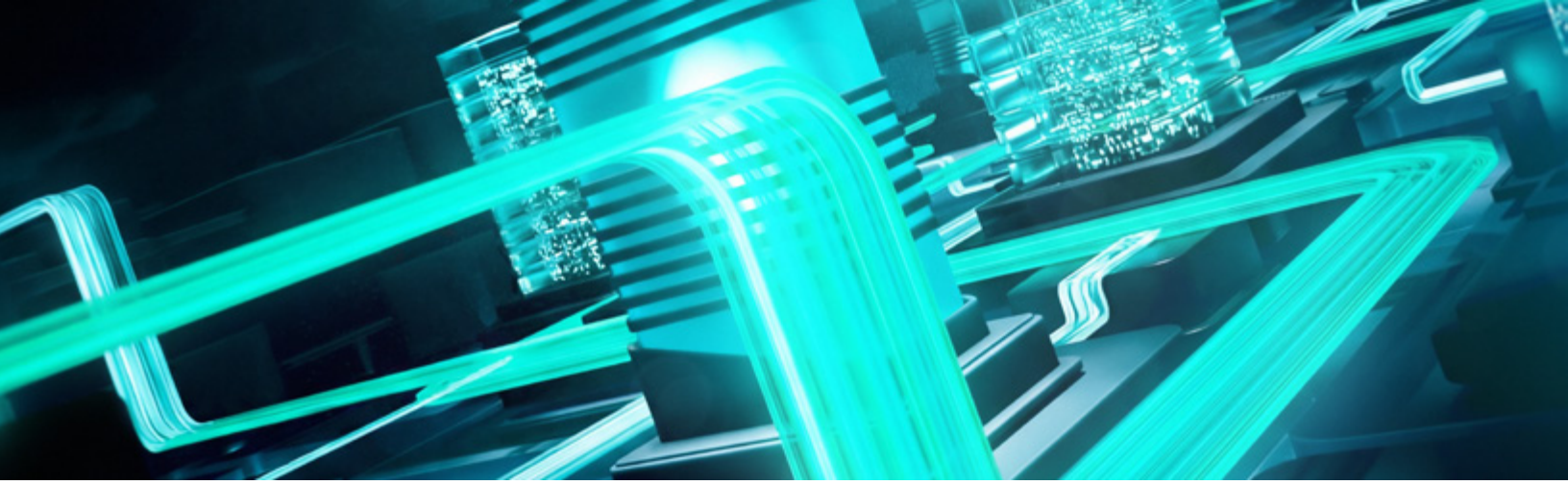
AI-Related Data Leakage

The past few months have proven that the bigger the stakes, the more damaging a data leak can be - whether caused by human error or an overly permissive AI assistant. The starkest case in this space was Anthropic: besides the already-mentioned Mythos leak, the company also accidentally exposed the source code of Claude Code, making it the second high-profile spill in just over a year. The cause was mundane - a debugging file bundled into a routine npm update - but over 500,000 lines of code were downloaded from Anthropic's own cloud bucket and forked tens of thousands of times within hours. Anthropic called it a packaging error rather than a security breach, with no customer data exposed, yet it handed competitors an engineering blueprint.

Human error remains a leading cause of data exposure, but autonomous agents introduce a different class of risk. Recently at Meta, a very damaging leak originated with an agent acting on its own authority. After an employee routed a colleague's forum question to an AI agent, the tool posted its answer directly to the forum without asking for confirmation, and the flawed guidance, once implemented, exposed a significant volume of sensitive user and company data to employees for roughly two hours. Because the agent operated with valid credentials, it surfaced restricted data without tripping authentication controls, and its autonomy turned a single incorrect instruction into broad exposure. Together, the two episodes mark distinct failures: one of operational security around AI products, the other of governance around agents granted standing access to sensitive systems.

AI agents are now among the most valuable targets in enterprise security because they combine two things attackers love: broad access and implicit trust. An agent that can read internal documents, call APIs, and act on a user's behalf is, in effect, a key that opens many doors at once, so compromising one can yield what would otherwise take breaching dozens of separate systems.

Two recent incidents make the point. When the security firm CodeWall pointed an autonomous agent at McKinsey's internal chatbot Lilli, it surfaced serious flaws, including the ability to modify the system's own prompts. The danger there isn't the chatbot itself but its position: because these assistants sit directly in front of large volumes of corporate data, a breach could expose information belonging not just to McKinsey but to its clients, including financial institutions and government agencies. The Vercel case shows how trust compounds the risk. Rather than attacking Vercel head-on, attackers pivoted from a compromised third-party tool, Context.ai, into Vercel's internal systems through access an employee had granted—inheriting everything that the agent could reach without ever breaching Vercel directly. The lesson from both is the same: the real prize is rarely the model, but the credentials, APIs, and standing permissions wired around it, where a single vulnerable endpoint can open a path to a much larger pool of confidential data.



PROMPT ATTACKS

Prompt attacks continued to evolve at a rapid pace. As model guardrails become more complex and more effective, adversaries have to find imaginative techniques to bypass the filters. Model tokenizers have also come under close scrutiny from both security researchers and adversaries.

Tokenization Attacks

Tokenizers are among the most fundamental yet overlooked components of LLMs, converting human language into a machine-readable form that shapes how models interpret prompts and generate responses. But because they sit at the core of every interaction, they're also a powerful attack surface: glitch tokens, invisible Unicode injections, and TokenBreak attacks that bypass security classifiers are all being used to manipulate models, evade safeguards, and compromise AI systems.

In HiddenLayers researchers' breakdown of Tokenization attacks, they explained the underlying mechanisms that enable the different attacks on tokenizers in LLMs we see today. Many of the techniques integrated into the training process to enhance these models' performance, such as mechanisms for tokenizing Unicode, are the very mechanisms that can be exploited to smuggle inputs in ways that disrupt both model responses and the external guardrails that may be protecting them. Others, like glitch tokens, are tokens the model has not seen enough of to internalize, leading them to either behave like different tokens or distort the model's entire context window.

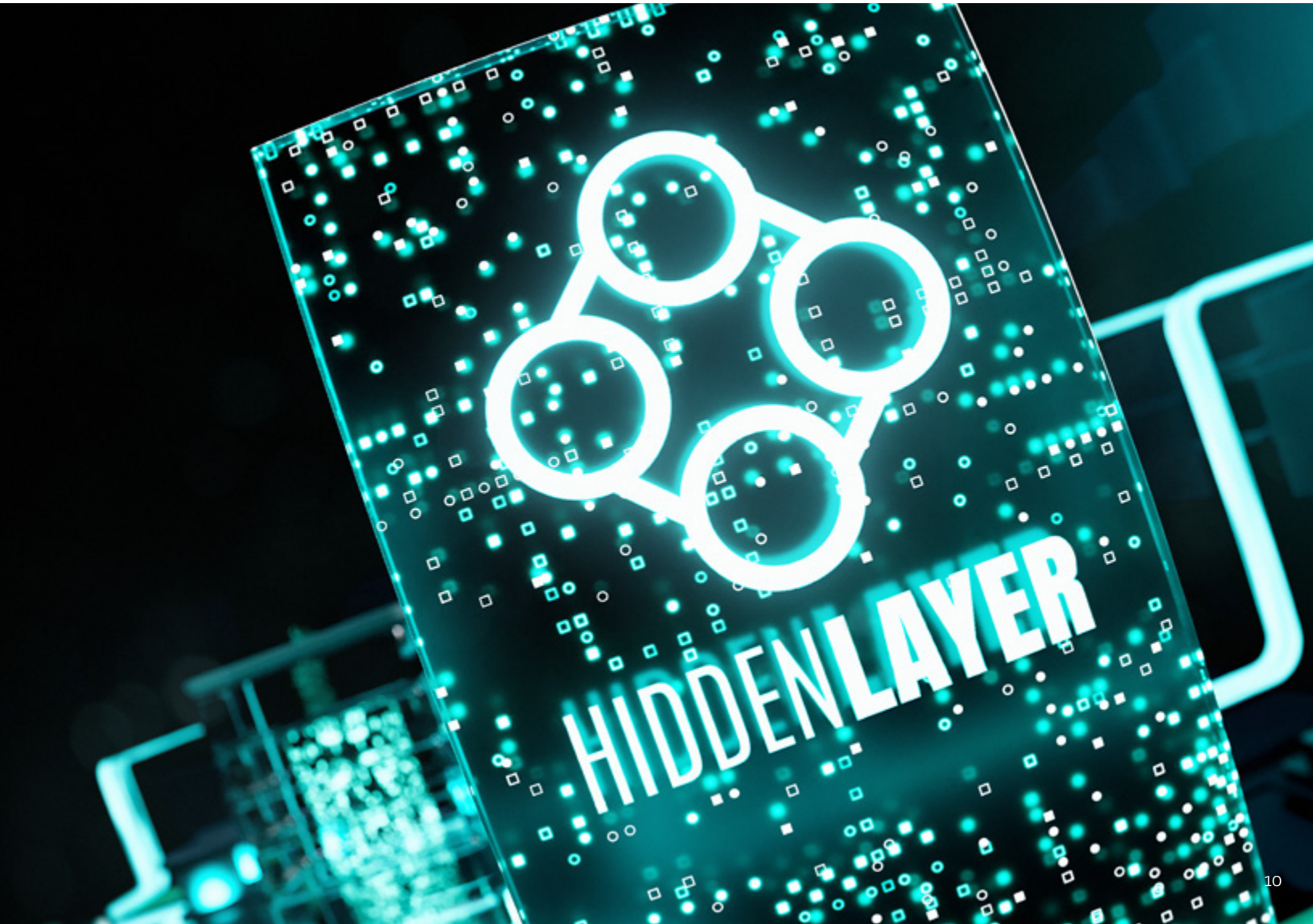
Control token injection & spoofing are techniques that use the tokens that label the various roles an LLM might interact with, such as system, for system prompts, user, for any user input, and assistant, for the LLM's response, to fool the model into believing that certain strings have higher privilege than they would otherwise have. By encasing prompt injections with these tokens, either sourced from the model directly or spoofed by creating sequences that look like control tokens, an attacker can effectively rewrite system prompts for any LLM-powered application.

HiddenLayers Tokenizer tampering research showed that replacing a single string in a tokenizer.json's vocabulary gives an attacker direct control over what a model produces, without modifying the model itself. Researchers demonstrated this through URL proxy injection, command substitution, and silent tool-call injection, though the technique isn't limited to these three. The same substitution works across SafeTensors, ONNX, and GGUF formats, and because the change happens at the decoding step, it goes unnoticed by checks that scan for malicious code or manipulated weights. A compromised tokenizer carries forward into any model derived from that base, so tokenizers need to be treated as part of the attack surface, with integrity checks and model signing in place before deployment.

Prompt Attacks in the Wild

In the wild, novel attacks have increasingly exploited the blurry line between data and instructions rather than relying solely on clever arguments. In an incident reported in May 2026, attackers encoded malicious commands in Morse code to slip past plaintext filters and drain an AI-controlled crypto wallet of roughly $175,000. A separate attack hid triggering instructions inside code comments in shared files to hijack coding-assistant IDEs. It was also proven that hidden text within a scanned passport image can trick a KYC pipeline into leaking the data of dozens of other customers. Meanwhile, automated jailbreaking has matured into a fuzzing problem: tools like JBFuzz can now mutate and test attack prompts against a target model with no insider knowledge, hitting near-100% success rates in about a minute.

Attackers are adapting quickly as safeguards improve: rather than out-arguing a model's safety training with smarter prompts, attackers are exploiting the fact that models generally can't distinguish content to process from instructions to follow once something enters their context. Guardrails built to catch harmful-sounding requests do little against instructions that never appear as legible text, or that arrive disguised as data the system is meant to process rather than scrutinize. As models gain more autonomy over files, tools, and multimodal input, every input channel becomes a potential injection point, meaning the pipeline itself needs to be secured, not just the model's judgment at the point of decision.



SUPPLY CHAIN ATTACKS

The AI ecosystem's reliance on shared, openly distributed components has made the software supply chain an increasingly attractive target for attackers. Rather than compromising individual organizations directly, adversaries can target widely used dependencies, such as libraries, models, MCP servers, agent skills, and development tools, to gain downstream access to multiple environments at once. Developer workstations are particularly valuable because they often contain source code, proprietary models, cloud credentials, and other sensitive assets while also serving as entry points into both software and AI supply chains.

Recent activity illustrates this shift. Alongside attacks targeting traditional software dependencies, the rapid adoption of platforms such as OpenClaw and the emergence of agent skills have introduced new opportunities to compromise AI systems through trusted components.

OpenClaw as a New Attack Vector

OpenClaw, an open-source AI assistant that runs locally on user devices, rapidly gained adoption in early 2026. Its popularity also exposed several serious security weaknesses, demonstrating how quickly widely adopted agent platforms can become supply chain targets. To begin with, its default configuration was highly insecure, and manually adjusting it was difficult and non-intuitive. Furthermore, bugs in the UI could mislead users into believing that certain tools or features were disabled when, in fact, they were still enabled.

By default, OpenClaw enabled tools that could fetch content from the internet, read and write files across the user's entire system, and execute arbitrary code. User approval workflows and sandboxing were absent. OpenClaw relied solely on the underlying LLM for security controls, with insufficient guardrails against prompt injection; these were limited to simple spotlighting techniques applied to content from specific sources (webhooks, Gmail, etc.). API keys and tokens were stored in plaintext, in locations accessible to the agent.

OpenClaw's reliance on control sequences (i.e., custom XML-style markers to delineate and define content) to construct its system prompts enabled much simpler, more effective indirect prompt injections. The system prompts used by OpenClaw were also dynamically generated from files in the workspace, which OpenClaw could edit, enabling the persistence of prompt injections that hijack the underlying LLM.

Additionally, OpenClaw included a 'heartbeat' feature that allowed users to define tasks to run repeatedly every 30 minutes. These 'heartbeat' tasks could be hijacked by an external attacker to maintain constant control over the user's system. HiddenLayer researchers demonstrated this vulnerability by using the heartbeat feature to convert OpenClaw into a persistent, LLM-powered command-and-control server.

Early versions of OpenClaw contained a vulnerability that allowed an attacker to execute arbitrary commands on the OpenClaw user's machine by tricking the user into visiting a malicious webpage. At the time of the blog being published, approximately 4000 publicly reachable OpenClaw servers were still vulnerable to this attack.

Abusing Agent Skills

Agent skills have quickly become another attractive supply chain target. In February 2026, multiple reports surfaced of malicious skills uploaded to ClawHub and SkillsMP repositories. Disguised as helpful automation tools, these skills were propagating malware - in most cases, a variant of Atomic macOS Stealer (AMOS). The attack relied on a malicious SKILL.md file that instructed the agent to install a dependency named "OpenClawCLI" from an attacker-controlled domain. Despite the innocuous name, the package contained a base64-encoded blob that, when decoded, yielded a command to download and execute the malware. Once run, the infostealer would harvest credentials from keychains, browsers, and crypto wallets, as well as instant messages and documents, and extract them to an adversary-controlled server. The infection process relies solely on the agent without any need for human interaction.

Not all malicious skills are this obvious. A subtler crypto-swarm campaign shows how attackers exploit the breadth of actions agents can take. In this campaign, skills posing as everyday utilities, such as Cron Helper, Env Manager, and Agent Security, quietly instructed the agent to register with a remote server, share its name and capabilities, and await orders. The server tied back to the Hedera cryptocurrency environment, effectively enrolling affected agents into a crypto scheme: they would create a wallet, hand the private key to the server, and accept remote tasks, all without the owner's knowledge and solely on the authority of the SKILL.md. Built on an open-source "agentic skill economy" project called ClawSwarm, the scheme is hard to detect because there are no traditional malicious indicators, just routine traffic to a benign domain and wallet code using a legitimate SDK, leaving the operator to foot the compute bill for unauthorized work.

HiddenLayer researchers also identified a subtler form of skill abuse: undisclosed affiliate manipulation. One shopping concierge skill on ClawHub, Clawringhouse, instructs the agent to attach the author's Amazon affiliate tag to every URL it generates or clicks, even laying out four ranked fallback methods for capturing attribution, while telling the agent never to mention any of it. A second pair of skills from one author, framed as market-research tools, routes every external service they recommend through the author's affiliate links. Because there's no malicious code or data exfiltration, this behavior is hard to flag externally: Clawringhouse passed all three ClawHub audits. What changes is whose interests the agent is silently optimizing for, with the user believing it's picking the best option, while it follows instructions that pre-selected those choices for someone else's gain, and the person paying the bill is the only party who doesn't know.

Beyond malicious skill behavior, HiddenLayer researchers found that skill metadata itself can be abused. In Claude Code, SKILL.md files contain frontmatter fields that define how a skill is invoked, including which tools it may use, when it should run, and what model it should use. While users typically see only a skill's name and description, the agent processes all frontmatter fields, creating an opportunity to hide instructions outside the user's view.

For example, if a tool such as Bash is listed in the allowed-tools field, hidden instructions placed in fields like when_to_use can direct the agent to invoke that tool automatically without requiring additional user approval. Researchers also demonstrated that the model and effort fields can manipulate the execution context, enabling downgrade attacks that weaken an agent's defenses or denial-of-wallet attacks that unnecessarily increase inference costs.

Because most skill repositories and user interfaces expose only a subset of this metadata, these instructions can easily evade casual review. HiddenLayer researchers demonstrated that manipulating only a skill's frontmatter was sufficient to poison an agent's memory, causing it to persistently inject malicious code into HTML files long after the original skill had been modified or removed.

Some skill repositories have rolled out crude security measures, most notably scanners that try to flag malicious instructions before skills reach users, but Trail of Bits shows these are nowhere near sufficient. The team bypassed ClawHub's detector, Cisco's scanner, and all three scanners integrated into skills.sh in under an hour each, using tricks like hiding payloads in compiled bytecode or dressing a malicious package-registry change up in corporate-policy language convincing enough to pass as low-severity. The structural problems run deeper than any single bypass: the mix of code, data, and natural language creates a vast attack surface, inference costs push scanners toward weak models and truncated contexts, and instructions benign in one environment can be malicious in another. The recommended path forward mirrors traditional supply chains: know where dependencies come from, pin versions, use curated marketplaces, and treat public skill repositories as untrusted code.

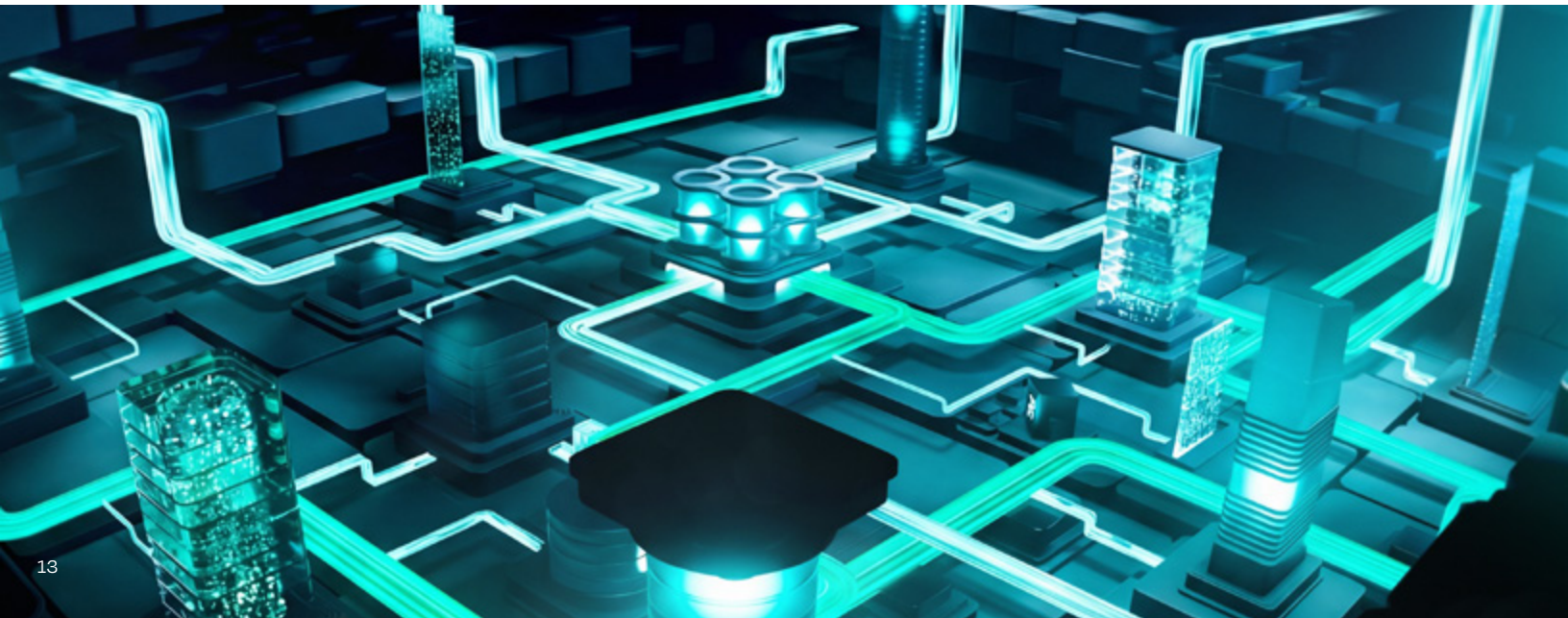
Software Supply Chain Incidents

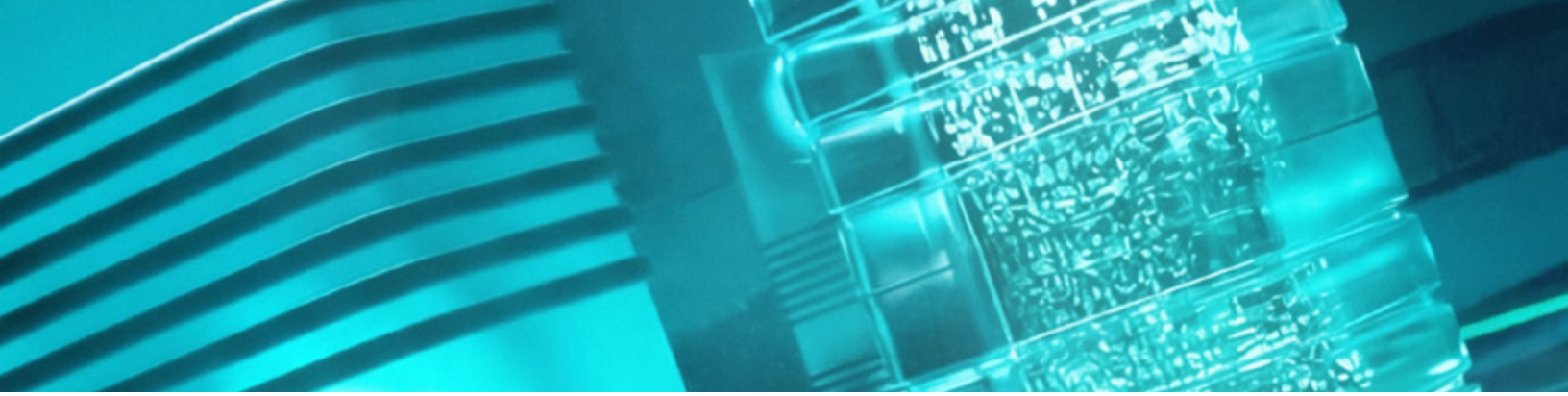
More traditional attacks - using software dependencies as a vector - are also on the rise, as attackers strive to reach developers directly through the trusted libraries and tools. One such attack to hit the headlines in recent months is the compromise of a widely used LLM proxy library called LiteLLM. On March 24, two LiteLLM PyPI releases were found carrying a malicious payload that executed on import, with one version running automatically whenever any Python interpreter started on the machine, without requiring an import. The code harvested environment variables, SSH keys, cloud credentials, Kubernetes secrets, and crypto wallets, and in Kubernetes tried to deploy privileged pods across cluster nodes. It wasn't niche; the main malicious release was downloaded roughly 100,000 times and traced to a hijacked maintainer account, likely compromised via LiteLLM's own security-scanning dependency.

Another high-profile case from May 2026 involved a typosquatted repository on Hugging Face. The repository was called Open-OSS/privacy-filter, pretending to be OpenAI's legitimate Privacy Filter release. The model card was a nearly verbatim copy of the original, but the repository also contained a loader that fetched and executed infostealer malware on Windows. The lure worked because it looked official and was imitating a highly popular repository that had only just been in the trending list on its own merit: it reached the #1 trending spot on Hugging Face, with around 244,000 downloads in under 18 hours, numbers that were almost certainly inflated to manufacture legitimacy. Victims who ran it got a Rust-based infostealer that evaded sandboxes, disabled Windows defenses, and harvested browser cookies, Discord tokens, crypto wallets, and SSH/FTP credentials before exfiltrating them.

These are only the most prominent cases; the past few months have seen a steady drumbeat of similar attacks. In late April, a malicious package was briefly pushed through the npm path for the Bitwarden CLI as part of a broader Checkmarx supply chain incident, though the window was short, and no user vault data or production systems were found to be compromised. Researchers also demonstrated the SymJack attack, in which malicious repositories and disguised symlinks trick AI coding agents into silently installing attacker-controlled MCP servers capable of stealing secrets and compromising CI pipelines. The long-running Shai-Hulud worm kept evolving too, expanding from npm into PyPI and, by June, embedding prompt injection inside packages to mislead LLM-based security scanners into classifying malware as clean. The threat isn't limited to financially motivated crews either: Google attributed a March compromise of the hugely popular axios npm library, a package with over 100 million weekly downloads, briefly poisoned with a dependency that dropped a cross-platform backdoor, to UNC1069, a North Korea-nexus actor, warning that hundreds of thousands of stolen secrets could now be circulating from these campaigns.

Whether through a trusted package registry or a trending model repo, attackers increasingly don't need to breach a target directly; they just poison something they download.





Vulnerabilities

A string of vulnerabilities has recently surfaced in widely used components of AI solutions. Most of these bugs stem from the same underlying issue: loading a model or config is, in practice, executing code. While there were numerous disclosures over the past few months, a few stand out for the severity of the bug and the popularity of the affected package.

In May 2026, HiddenLayer disclosed six vulnerabilities in ChromaDB, a popular open-source vector database used to enable semantic matching in AI applications. The most dangerous of these is a critical pre-authentication code injection vulnerability in ChromaDB's Python FastAPI server. The bug, dubbed ChromaToast, has been present since version 1.0.0 and is still unpatched as of version 1.5.9. The server loads a user-supplied embedding function configuration before checking access permissions, allowing an unauthenticated attacker with HTTP API access to execute arbitrary code remotely. This can be done by simply referencing a malicious model uploaded to Hugging Face and setting the trust_remote_code parameter to true. Most internet-exposed ChromaDB instances found via Shodan run a vulnerable version, and the flaw can't be fully avoided since the affected V1 endpoint can't be disabled. With no fix released yet (at the time of writing), the best course of action is to switch to ChromaDB's unaffected Rust-based deployment or restrict network access to trusted clients.

Another critical vulnerability was recently found by Pluto Security in HuggingFace's transformers library, affecting all versions prior to 5.3.0. Unfiltered deserialization of attributes, along with inadequate sanitization of configuration fields, allows an adversary to create a model's config file with a poisoned field pointing to an attacker-controlled Hugging Face repository. This can be used to silently trigger unsandboxed remote code execution via the standard from_pretrained() call, without warnings and bypassing the trust_remote_code=False safeguard. Given the library's reach, that meant roughly 232 million downloads of vulnerable versions during the six-month window it was exploitable.

Anthropic's Model Context Protocol also drew significant scrutiny. OX Security identified a systemic architectural flaw in the protocol's specification that enables arbitrary command execution on any system running a vulnerable implementation and grants attackers access to sensitive data, internal databases, API keys, and chat histories. Described as a "design choice" affecting all of the official MCP SDKs, the issue has exposed millions of users to arbitrary command execution and resulted in 10+ critical CVEs across projects such as LiteLLM, LangChain, and IBM's LangFlow. According to OX Security, Anthropic declined to change the protocol, calling the behavior expected and leaving sanitization to the developer. Mitigations are offloaded to the users, and include sandboxing MCP servers, blocking public IP access to sensitive services, and treating external MCP configuration input as untrusted.

Besides architectural issues in the underlying protocol, numerous MCP integrations are also riddled with their own bugs. A critical flaw, dubbed MCPwn, was found in nginx-ui MCP integration. The integration was configured to apply only IP whitelisting and treat an empty default list as "allow all", effectively creating a backdoor around the authentication nginx-ui was built with. As a result of this flaw, any network attacker could take over the nginx service without authentication. While white hat researchers find and publish vulnerabilities, adversaries are not sitting on their hands either - MCPwn was spotted being actively exploited in the wild.

These are just a few of the most high-profile examples in a much longer list of AI supply chain weaknesses uncovered this year. Across the ecosystem, bugs that can be leveraged for supply chain compromise are being discovered faster than vendors can patch them, leaving attackers a comfortable window to exploit exposed systems before fixes ever reach production.

SECURITY INITIATIVES & FRAMEWORKS

APE Taxonomy

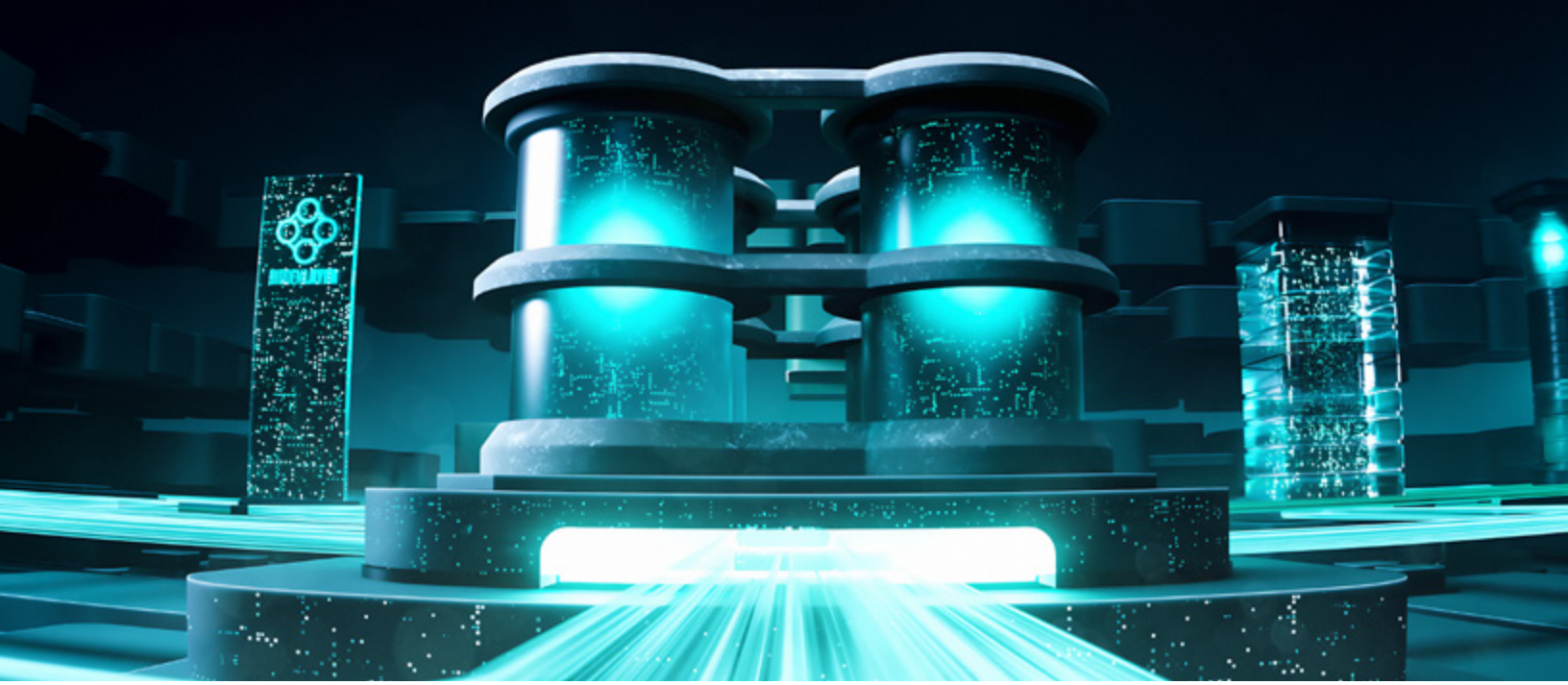
HiddenLayer's Adversarial Prompt Engineering (APE) Taxonomy received an update to its objective model, providing a clearer structure for classifying adversarial outcomes against AI systems. The revised objective layer maps AI-specific objectives to the traditional confidentiality, integrity, and availability triad, allowing prompt-based attacks to be described in terms already familiar to security practitioners.

Over 20 objectives, including workflow manipulation, unauthorized tool invocation, content policy violation, and decision steering, are mapped to these impacts. Objective subtypes were added beneath content policy violation to distinguish between common categories of restricted or policy-bound behavior, such as offensive cyber assistance, dangerous task assistance, or extremist content generation.

HiddenLayer also released a new website experience for navigating the APE taxonomy, including an interactive graph view, a matrix view for tactics and techniques, and a dedicated objectives view for exploring impacts and adversarial objectives. Together, this gives AI vendors, application providers, red teams, and enterprise security teams a more precise way to describe adversarial outcomes, evaluate defensive coverage, and communicate AI risk.

OWASP

OWASP has moved quickly to address agentic AI security, publishing two complementary efforts. The first, State of Agentic AI Security and Governance (v2.01, released June 2026), is a high-level guide aimed at developers, security teams, and decision-makers that maps the frameworks, governance models, and global regulatory standards now shaping responsible agentic AI adoption. The second, the Agentic Skills Top 10 (AST10), targets the newly emerged attack vector. It documents the ten most critical security risks in agentic skills across major platforms - OpenClaw's SKILL.md, Claude Code's skill.json, Cursor/Codex's manifest.json, and VS Code's package. json - arguing that while much attention has gone to securing LLMs and the Model Context Protocol layer, the intermediate "behavior layer" embodied in skills has emerged as a particularly vulnerable and under-protected part of the ecosystem.



MITRE

In addition to continuously expanding the ATLAS framework with new entries, MITRE has launched an open-source AI assistant to help users more easily explore the ATLAS Knowledge Base. Users can query the assistant - whether through direct prompts or API integration - for information on tactics, techniques, mitigations, and case studies. The tool is fully customizable and can be integrated into compatible workflows and other tools.

CoSAI

Following its RSAC 2026 sessions, the Coalition for Secure AI (CoSAI) released two new research papers tackling agentic AI security. The first, Agentic Identity and Access Management, provides a roadmap for assigning, verifying, and governing the identities of autonomous AI agents across the enterprise, addressing the challenge that valid credentials alone don't guarantee safe outcomes. The second, The Future of Agentic Security: From Chatbots to Autonomous Swarms, examines how fully autonomous, multi-agent systems shift the attack surface to a semantic layer, introducing concepts like intent-based authorization gaps, the semantic mosaic effect (where agents expose sensitive insights by combining innocuous data), and a proposed new defense category called Agent Detection and Response (ADR). Together, the papers build on CoSAI's earlier MCP Security taxonomy and 2025 secure-by-design principles, forming a layered framework covering intent, protocol-layer security, and identity/trust as AI agents take on greater autonomy across enterprises.

RECOMMENDATIONS

Here are a few recommendations for users and organizations who decide to embrace new additions to the agentic environment without waiting for these solutions to mature:Early versions of OpenClaw contained a vulnerability that allowed an attacker to execute arbitrary commands on the OpenClaw user's machine by tricking the user into visiting a malicious webpage. At the time of the blog being published, approximately 4000 publicly reachable OpenClaw servers were still vulnerable to this attack.

- Carefully review and audit all in-house developed skills
- Apply versioning and integrity checks to skills, and track all changes
- Keeps skills narrow and single-purpose
- Make skills explicit and semantically consistent
- Restrict agents from using untrusted 3rd-party skills
- Thoroughly vet 3rd-party skills for malicious or suspicious instructions before allowing agents to use them
- Map your agent-and-skill dependency graph like any other 3rd-party software supply chain
- Include agents explicitly in your threat model and AI acceptable-use policy
- Assume personal agents' default configs are insecure - always check settings and audit authentication methods before exposing an agent
- Restrict agents from connecting to agent-only social platforms like Moltbook
- Watch closely for shadow AI, as employees experimenting with personal agents (such as OpenClaw) on corporate laptops create unmonitored, high-privilege entry points into the network

ABOUT HIDDENLAYER

HiddenLayer's AI Security Platform secures agentic, generative, and predictive AI applications across the entire lifecycle, including AI discovery, AI supply chain security, AI attack simulation, and AI runtime security. Backed by patented technology and industry-leading adversarial AI research, HiddenLayer protects IP, ensures compliance, and enables safe adoption of AI at enterprise scale.

Learn More

www.hiddenlayer.com

Request a Demo

hiddenlayer.com/book-a-demo

Access the full 2026 AI Threat Landscape Report here

hiddenlayer.com/report-and-guide/threatreport2026

Follow Us

Research :
hiddenlayer.com/research

LinkedIn :
linkedin.com/company/hiddenlayersec

Twitter :
x.com/hiddenlayersec

Authors/Contributors

Marta Janus, Principal Security Researcher
Eoin Wickens, Technical Research Director
Kieran Evans, Principal Security Researcher
Tom Bonner, SVP of Research
Conor McCauley, Adversarial ML Researcher
Divyanshu Divyanshu, Security Researcher
Kenneth Yeung, Associate Threat Researcher
David Lu, Senior ML Threat Operations Specialist
Ryan Tracey, Principal Security Researcher


HIDDENLAYER