The AI Vulnerability Scanning Market: OpenAI Codex Security and the Anthropic/Mozilla Partnership

Unofficial AI-assisted Research

Cloud Security Alliance AI Safety Initiative

2026-03-08

Key Takeaways

On March 6, 2026, two announcements provided the clearest public evidence to date that frontier AI models are operating as primary agents of vulnerability discovery, rather than auxiliary tools that flag known patterns against static signatures. OpenAI launched Codex Security—a context-aware application security agent, meaning one that builds a model of the target application's architecture and trust boundaries before scanning—into research preview, while Anthropic published the results of a two-week collaboration with Mozilla in which Claude Opus 4.6 identified 22 previously unknown vulnerabilities in the Firefox browser. Whether these developments constitute a structural shift or a significant acceleration of trends already underway is a question this note examines; either characterization demands serious attention from enterprise security programs.

The implications for enterprise security teams are significant. AI-powered vulnerability scanning agents can now build repository-specific threat models, validate findings in isolated sandboxes, and propose patches—compressing a workflow that traditionally required multiple specialist roles. At the same time, these tools introduce novel risks: the AI agents themselves represent new attack surfaces (as demonstrated by a command injection vulnerability in the Codex CLI disclosed in August 2025 [4]), and their deployment at scale raises concerns about automation bias, prompt injection via internet-connected contexts, and the dual-use character of tools that, by design, discover exploitable vulnerabilities.

Security organizations should not treat these announcements as a simple acceleration of the status quo. The competitive dynamics, safety architecture decisions, and operational security requirements of AI-native vulnerability scanning differ materially from those of prior-generation static analysis and dynamic testing platforms.

Background

The Trajectory of AI in Application Security

Automated vulnerability discovery is not new. Static application security testing (SAST), dynamic application security testing (DAST), and software composition analysis (SCA) tools have been commercial offerings for over two decades. These tools share a fundamental characteristic: they apply predetermined rule sets, patterns, or signatures against code or running applications. Their limitations—high false positive rates, inability to reason about application-specific context, and difficulty detecting complex logic vulnerabilities—have been well documented [6].

The introduction of large language models into application security initially produced tools that assisted with code review and remediation suggestion rather than autonomous discovery. GitHub Copilot Autofix, which can propose fixes for flagged issues, and Snyk's AI-assisted scanning capabilities represent this first generation: models used as accelerants within established workflows rather than as primary discovery agents [7]. The concern during this period ran in the opposite direction—that AI code generation tools were actively introducing new vulnerabilities by replicating insecure patterns from training data [8].

The announcements of March 6, 2026 represent a meaningful step beyond this first generation, with frontier models now deployed for autonomous discovery rather than assisted review. Both OpenAI and Anthropic are now deploying frontier models—GPT-5.3-Codex and Claude Opus 4.6, respectively—specifically for the task of autonomous vulnerability discovery in production codebases. The distinction matters: these systems are not assisting human reviewers with known vulnerability classes but are independently identifying previously unknown vulnerabilities in widely used open-source projects that have been subject to years of security review, fuzzing, and manual auditing.

The Competitive Landscape

The AI vulnerability scanning market encompasses a heterogeneous set of vendors, ranging from traditional enterprise platforms that have incorporated AI capabilities to AI-native startups building around LLM reasoning. Commercial market research vendors estimate the AI-specific vulnerability scanning segment has grown from approximately $2.4–2.6 billion in 2024 to roughly $3.1 billion in 2025, with projections reaching $9.09 billion by 2034 at a compound annual growth rate of 14–17%; however, these figures reflect the methodological variability common to commercial market sizing reports and have not been independently validated by major analyst firms [13]. Adoption surveys from the same sources suggest that approximately 62% of enterprises plan to incorporate AI-driven scanning

capabilities by 2026, a forward-looking estimate that likewise lacks disclosed methodology or sample size [13]. Established leaders including Qualys VMDR, Tenable One, Checkmarx One, and Veracode retain significant market presence, with the 2025 Gartner Magic Quadrant for Application Security Testing naming Black Duck, HCLSoftware AppScan, and OpenText as Leaders—and recognizing Cycode as a first-time entrant reflecting the rise of unified Application Security Posture Management (ASPM) platforms [14]. Cloud-aligned offerings from Palo Alto Networks (Prisma Cloud/Prisma AIRS) and CrowdStrike (Falcon Exposure Management) have extended vulnerability visibility into cloud workloads and endpoint telemetry. Significant M&A activity in 2025 and early 2026 reflects broader strategic investment in AI-integrated security capabilities: Google's announced $32 billion acquisition of Wiz—primarily a cloud security posture management platform, with regulatory approval still in final stages as of this writing—alongside more directly AI-focused acquisitions including Check Point's acquisition of Lakera Guard and Palo Alto's acquisition of Protect AI, underscore how large security platforms are repositioning around AI-native capabilities [13].

Within this landscape, three distinct market segments are emerging. The first focuses on securing AI-generated code: tools like Snyk's AI-powered scanning and GitHub Copilot Autofix that address the vulnerability debt introduced when developers rely heavily on AI code generation without adequate review. Research has found that AI coding assistants can replicate insecure patterns from their training data, and that developer automation bias—the tendency to accept AI-generated output with reduced scrutiny—compounds this risk [8]. The second segment uses AI agents to discover vulnerabilities in existing codebases with human-like reasoning capabilities—the segment into which OpenAI Codex Security and Anthropic's security research program are now entering. A third and fast-growing segment focuses on securing LLM applications themselves: tools such as Mindgard, Lakera Guard, Garak, and Microsoft's PyRIT address AI-specific attack surfaces—including prompt injection (ranked the top LLM vulnerability in the OWASP 2025 Top 10 for LLM Applications), model inversion, and adversarial input attacks—that conventional SAST and DAST tooling cannot address [15].

Security Analysis

OpenAI Codex Security: Architecture and Findings

Codex Security entered research preview on March 6, 2026, available to Enterprise, Business, and education customers at no cost for the first month [1]. OpenAI began testing a related internal security research agent called "Aardvark" with a limited customer group in October 2025; public reporting links this earlier effort to Codex Security's development, suggesting a development timeline of roughly five months from private beta to public research preview [2].

The system's operational approach departs meaningfully from signature-based scanning. Rather than maintaining a database of known vulnerability patterns, Codex Security builds a repository-specific threat model: it analyzes the structure, trust boundaries, and exposure points of the target codebase before scanning for vulnerabilities, and uses that context to distinguish genuine issues from false positives. This threat model is described as editable by security teams, allowing human oversight of the analytical frame within which automated scanning operates [3]. Findings are then validated in an isolated sandbox environment before surfacing. According to OpenAI's launch announcement, these design choices reduced false positive rates by 50% and decreased over-reported severity findings by 90% during the beta period—figures that have not been independently verified as of this writing [1].

OpenAI has published the following beta performance data: over the 30 days preceding the research preview announcement, Codex Security scanned more than 1.2 million commits across external open-source repositories and identified 792 critical findings and 10,561 high-severity findings [1]. Independent validation of these figures has not been published as of this writing. Among the affected projects are OpenSSH, GnuTLS, GOGS, Thorium, libssh, PHP, and Chromium—codebases that have been subject to extensive security review for years. Fourteen CVEs have been assigned to vulnerabilities uncovered by the agent to date, including three affecting GnuTLS: a heap buffer overflow (CVE-2025-32990), a heap buffer overread in SCT extension parsing (CVE-2025-32989), and a double-free in otherName SAN export (CVE-2025-32988) [2]. The ability to identify new vulnerabilities in GnuTLS—a cryptographic library used across Linux distributions—indicates that the tool operates above the threshold of what conventional fuzzing and static analysis have historically captured.

The model underlying Codex Security, GPT-5.3-Codex, occupies a significant position within OpenAI's own safety framework. OpenAI has designated it the first model treated as having "high cybersecurity capability" under its Preparedness Framework, a classification that triggers additional safeguards including training the model to refuse clearly malicious requests such as credential theft [5]. This classification acknowledges an inherent tension: a model capable of autonomously discovering vulnerabilities in production software is also, by definition, a tool with material offensive potential.

The Anthropic/Mozilla Partnership: AI-Assisted Security Research at Scale

The Anthropic/Mozilla collaboration, announced simultaneously with the Codex Security launch, demonstrates a different deployment model for AI-powered vulnerability discovery. Rather than a productized scanning platform, Anthropic's Frontier Red Team engaged directly with Mozilla engineers over a two-week period in February 2026, deploying Claude Opus 4.6 against Firefox's C++ codebase. The collaboration scanned approximately 6,000 C++ files and submitted 112 unique vulnerability reports, of which 22 represented distinct CVEs [10]. Mozilla classified 14 of these as high-severity. It is worth

noting that the 22 accepted CVEs represent a subset of the 112 submissions; the remaining reports likely reflect deduplication, findings below the CVE threshold, or issues classified as informational rather than exploitable vulnerabilities.

The scale of these findings relative to historical remediation rates is notable. According to joint communications from Anthropic and Mozilla, the high-severity findings from the two-week collaboration represented nearly one-fifth of all high-severity Firefox vulnerabilities remediated throughout all of 2025 [10]. Firefox 148.0 incorporated fixes for the majority of the identified issues. This ratio—two weeks of AI-assisted review producing one-fifth of a year's high-severity remediation—reflects both the efficiency of AI-scale code review and the degree to which complex C++ memory safety issues remain undiscovered by conventional methods.

The collaboration also surfaced evidence about discovery speed that warrants attention. Claude identified a use-after-free vulnerability in Firefox's JavaScript engine within twenty minutes of beginning its exploration of that component, subsequently finding 50 unique crashing inputs during the validation phase [10]. This rapid initial discovery, in a code area subject to years of fuzzing and manual review, suggests that AI reasoning about code semantics can surface vulnerability classes that exhaustive automated input generation misses—specifically, logic vulnerabilities where the path to exploitation requires understanding the intended behavior of a system, not merely finding inputs that produce crashes.

Anthropic also published information about the tool's offensive capability boundary. Across hundreds of attempts using approximately $4,000 in API credits, Claude was able to construct crude browser exploits in only two cases [10]. This asymmetry—vulnerability discovery capability substantially exceeding exploitation capability—may reflect both a deliberate design philosophy and the current state of the technology's exploitation-chaining capabilities. Anthropic has indicated it expects this gap to narrow over time [10], which informs the safety architecture decisions surrounding such tools. Future plans include expanding AI-assisted security research to the Linux kernel, releasing Claude Code Security in limited research preview, and developing patching agents and task verifiers for automated remediation.

Risks Introduced by AI-Native Security Tools

The capabilities demonstrated by both platforms introduce security risks that security teams must evaluate alongside the defensive benefits. The most direct risk is that the AI agents themselves constitute new attack surfaces within development and security workflows.

This risk materialized concretely in 2025, when Check Point Research disclosed CVE-2025-61260, a command injection vulnerability in the Codex CLI tool [4]. The flaw allowed attackers to execute arbitrary commands on developers' machines by placing malicious configuration files within a project

repository. The attack worked through two repository-resident files: a `.env` file redirecting the configuration directory and a `.codex/config.toml` file declaring MCP server entries with attacker-controlled command payloads; when a developer ran Codex CLI against the repository, the tool parsed and executed these entries silently at startup without user consent or interactive approval. Check Point disclosed the issue to OpenAI on August 7, 2025; OpenAI patched it in Codex CLI version 0.23.0 on August 20; the finding was publicly disclosed on December 1, 2025 [4]. The incident is structurally representative of a broader class of risk: AI tools that read and process repository contents inherit the trust context of those contents, and supply-chain compromise of a repository can become a vector for compromising the security teams reviewing it.

A second risk category involves the agentic behavior of these tools when operating in internet-connected contexts. OpenAI's own documentation notes that enabling internet access for Codex Security can introduce prompt injection vulnerabilities, credential leakage, and inadvertent use of code with license restrictions, and recommends limiting access to trusted domains and safe HTTP methods [5]. The risk of prompt injection into security scanning agents is qualitatively different from prompt injection into general-purpose assistants: a security agent that can access credentials, file systems, and CI/CD pipelines represents a high-value target for attackers who can influence its context.

A third concern, identified by security researchers studying the broader class of AI software engineering agents, is automation bias among the security practitioners who review AI-generated findings [9]. When an agent scans millions of commits and surfaces 10,000 high-severity issues, the operational challenge shifts from discovery to triage. If reviewers systematically over-trust AI confidence assessments and under-scrutinize flagged issues, the practical false positive rate at the decision layer may not reflect the tool's technical precision. While the specific behavioral dynamics in security review contexts have not been empirically studied, the concern is well-grounded: automation bias in AI-generated code development has been documented [8], and the structural conditions that produced it—high output volume, confident AI presentation, and reviewer overload—are present in AI-native scanning deployments as well.

Competitive and Strategic Implications

The simultaneous emergence of AI-native vulnerability scanning from two frontier model providers creates a new competitive dynamic within the application security market. Traditional vendors have incorporated AI capabilities incrementally, primarily using machine learning for prioritization and risk scoring rather than for primary discovery. OpenAI and Anthropic are entering with tools built around the reasoning capabilities of their most powerful models—capabilities that most traditional security vendors currently access only through third-party API integrations rather than owning and task-specifically tuning.

This creates differentiation around vulnerability classes. Signature-based tools and conventional fuzz testing are well-calibrated for known vulnerability patterns and input-triggered crashes. AI-native tools appear to offer comparative advantage in identifying logic vulnerabilities, complex memory safety issues in large codebases, and vulnerabilities that require understanding application semantics rather than simply finding anomalous inputs. The GnuTLS CVEs and the Firefox use-after-free reflect this profile.

At the same time, AI-native tools face integration challenges that established vendors have solved. Codex Security currently operates exclusively with GitHub repositories connected through Codex Web [3], limiting its reach compared to platforms that support heterogeneous SCM environments, on-premises deployments, and integration with enterprise ticketing and patch management workflows. The research preview model adopted by both OpenAI and Anthropic also reflects the early maturity of these capabilities: unlike Generally Available commercial products, research preview tools typically lack the contractual SLAs, audit logging, data residency controls, and enterprise support structures that compliance-oriented organizations require.

Recommendations

Immediate Actions

Security teams operating in organizations with access to Codex Security's research preview should evaluate the tool's GitHub integration with specific attention to the permissions model. Before connecting production repositories, teams should review what access the tool requires, how findings are stored and retained by OpenAI, and whether internet access should be enabled given the prompt injection risks documented in the tool's own security guidance.

Organizations that use Codex CLI in local developer environments should verify that Codex CLI version 0.23.0 or later is deployed across all affected machines. The August 2025 command injection vulnerability demonstrated that AI developer tools processing repository contents can be exploited through supply-chain compromise of those repositories.

Short-Term Mitigations

Security programs should develop triage workflows that account for the volume characteristics of AI-native scanning output before broad deployment. Codex Security's beta data showed 792 critical and 10,561 high-severity findings across a set of open-source repositories; an enterprise deployment

scanning internally developed software could produce similar or larger volumes. Without dedicated triage capacity and clear escalation criteria, the practical benefit of discovering true positives may be offset by the operational overhead of processing false positives.

Teams should also establish explainability requirements for AI vulnerability findings before integrating them into remediation SLAs. AI-generated findings may lack the explicit rule references and CWE classifications that traditional SAST tools provide, complicating both developer remediation and compliance reporting. Requiring that AI-generated findings include evidence artifacts—crash reproducers, proof-of-concept test cases, or the threat model reasoning behind a finding—reduces automation bias and supports human review.

Strategic Considerations

The Anthropic/Mozilla collaboration provides a template worth examining for organizations that maintain critical open-source dependencies. A time-bounded, focused AI-assisted security review of a dependency—conducted either through a vendor partnership or through internal deployment of AI scanning tools—can surface vulnerability debt that conventional SAST and fuzzing programs miss. The reported API credit cost of approximately $4,000 for the two-week engagement suggests the model-compute component of such a program is tractable for organizations of varying sizes, though personnel and operational costs for a comparable focused review would be additional.

Security leaders should also monitor how OpenAI and Anthropic address dual-use governance for these platforms as they mature. GPT-5.3-Codex's classification under OpenAI's Preparedness Framework as a "high cybersecurity capability" model represents the first instance of a vendor applying a formal AI safety designation to a security product. How this designation translates into operational access controls, use case restrictions, and customer eligibility requirements will shape the risk profile of the tool in enterprise environments. CSA's AI Organizational Responsibilities guidance recommends that organizations deploying AI tools with high cybersecurity capability assess the potential for misuse by insider threats and establish controls commensurate with the tool's offensive potential [11].

Taken together, the events of March 6, 2026 mark a convergence point: two of the most capable AI models now available are being applied directly to one of security's most persistent challenges. The organizations best positioned to benefit are those that treat these tools not as drop-in replacements for existing SAST or DAST platforms, but as a qualitatively different class of capability requiring new governance structures, new triage workflows, and new attention to the security of the tools themselves.

CSA Resource Alignment

The developments described in this note engage multiple CSA frameworks and working group outputs.

The CSA AI Organizational Responsibilities series—specifically the "AI Tools and Applications" volume—is directly applicable to organizations evaluating Codex Security and similar tools [11]. That guidance addresses procurement due diligence for AI tools, third-party data exposure risks, and the governance structures required when AI tools operate with access to sensitive codebases. The command injection vulnerability in Codex CLI also implicates the supply chain security guidance in CSA's work on AI developer tooling.

The CSA MAESTRO framework for agentic AI threat modeling provides the analytical structure most relevant to Codex Security's deployment model. MAESTRO's seven-layer threat taxonomy—which encompasses model behavior, agent runtime, tool integrations, and infrastructure—maps directly to the risk categories described above: prompt injection through internet access, credential leakage, command injection through repository contents, and automation bias at the human review layer. Organizations deploying AI security scanning agents should conduct a MAESTRO-aligned threat model of the agent's own attack surface before using it to assess the attack surface of production systems.

The CSA Cloud Controls Matrix (CCM) Domain AIS (Application and Interface Security) provides the control baseline against which AI-native scanning tools should be evaluated during procurement. Organizations subject to CSA STAR certification should assess whether findings from AI-native tools satisfy the evidence requirements for CCM controls related to vulnerability management, given that AI-generated findings may differ structurally from the outputs of traditional SAST/DAST tools.

The CSA Top Concerns with Vulnerability Data report, published through the Vulnerability Data Working Group, documents the known limitations of CVSS scoring and CVE quality that AI-native tools inherit [6]. The concerns described in that report—context-insensitive severity scoring, data quality variability, and the challenge of prioritization at scale—apply with equal force to AI-generated vulnerability findings. Organizations should not assume that AI-native tools resolve these structural problems; they may in fact amplify them by producing findings at a volume that strains existing prioritization workflows.

Finally, the CSA Agentic AI Red Teaming Guide provides practical guidance for evaluating AI agents, including security agents, for adversarial robustness [12]. Before deploying AI vulnerability scanning tools in environments where they have access to sensitive codebases and credentials, organizations should conduct adversarial testing of the agents themselves, specifically to evaluate their behavior when exposed to malicious repository contents and adversarially crafted inputs.

References

1. OpenAI, "Codex Security: now in research preview," openai.com, March 6, 2026. https://openai.com/index/codex-security-now-in-research-preview/

2. The Hacker News, "OpenAI Codex Security Scanned 1.2 Million Commits and Found 10,561 High-Severity Issues," thehackernews.com, March 2026. https://thehackernews.com/2026/03/openai-codex-security-scanned-12.html

3. OpenAI, "Codex Security," developers.openai.com, March 2026. https://developers.openai.com/codex/security/

4. Check Point Research, "OpenAI Codex CLI Command Injection Vulnerability (CVE-2025-61260)," research.checkpoint.com, December 1, 2025. https://research.checkpoint.com/2025/openai-codex-cli-command-injection-vulnerability/; SecurityWeek, "Vulnerability in OpenAI Coding Agent Could Facilitate Attacks on Developers," securityweek.com, 2025. https://www.securityweek.com/vulnerability-in-openai-coding-agent-could-facilitate-attacks-on-developers/

5. OpenAI, "GPT-5.3-Codex System Card," cdn.openai.com, February 5, 2026. https://cdn.openai.com/pdf/23eca107-a9b1-4d2c-b156-7deb4fbc697c/GPT-5-3-Codex-System-Card-02.pdf

6. CSA Vulnerability Data Working Group, "Top Concerns with Vulnerability Data," cloudsecurityalliance.org, 2024.

7. TechTarget, "GitHub Copilot Autofix tackles vulnerabilities with AI," searchsecurity.techtarget.com, 2025. https://www.techtarget.com/searchsecurity/news/366603045/GitHub-Copilot-Autofix-tackles-vulnerabilities-with-AI

8. Snyk Labs, "Copilot amplifies insecure codebases by replicating vulnerabilities in your projects," labs.snyk.io, 2025. https://labs.snyk.io/resources/copilot-amplifies-insecure-codebases-by-replicating-vulnerabilities/

9. Pillar Security, "The Hidden Security Risks of SWE Agents like OpenAI Codex and Devin AI," pillar.security, 2025. https://www.pillar.security/blog/the-hidden-security-risks-of-swe-agents-like-openai-codex-and-devin-ai

10. Anthropic, "Partnering with Mozilla to improve Firefox's security," anthropic.com, March 6, 2026. https://www.anthropic.com/news/mozilla-firefox-security; Mozilla, "Hardening Firefox with Anthropic's Red Team," blog.mozilla.org, March 6, 2026. https://blog.mozilla.org/en/firefox/hardening-firefox-anthropic-red-team/; TechCrunch, "Anthropic's Claude found 22 vulnerabilities in Firefox over two weeks," techcrunch.com, March 6, 2026. https://techcrunch.com/2026/03/06/anthropics-claude-found-22-vulnerabilities-in-firefox-over-two-weeks/

11. CSA AI Organizational Responsibility Working Group, "AI Organizational Responsibilities: AI Tools and Applications," cloudsecurityalliance.org, 2025.

12. CSA AI Organizational Responsibilities Working Group, "Agentic AI Red Teaming Guide," cloudsecurityalliance.org, 2025.

13. market.us / The Business Research Company, "AI Vulnerability Scanning Market Size and Forecast 2025–2034," market.us, 2025. https://market.us/report/ai-vulnerability-scanning-market/

14. Gartner, "Magic Quadrant for Application Security Testing," gartner.com, October 6, 2025 (coverage via Black Duck analyst report). https://www.blackduck.com/resources/analyst-reports/gartner-magic-quadrant-appsec.html

15. Mindgard, "Best AI Security Tools for LLM Protection (2026)," mindgard.ai, 2026. https://mindgard.ai/blog/best-ai-security-tools-for-llm-and-genai; Lakera, "Top LLM Security Tools," lakera.ai, 2025 (note: Lakera was acquired by Check Point in late 2025; this URL may redirect post-acquisition). https://www.lakera.ai/blog/llm-security-tools