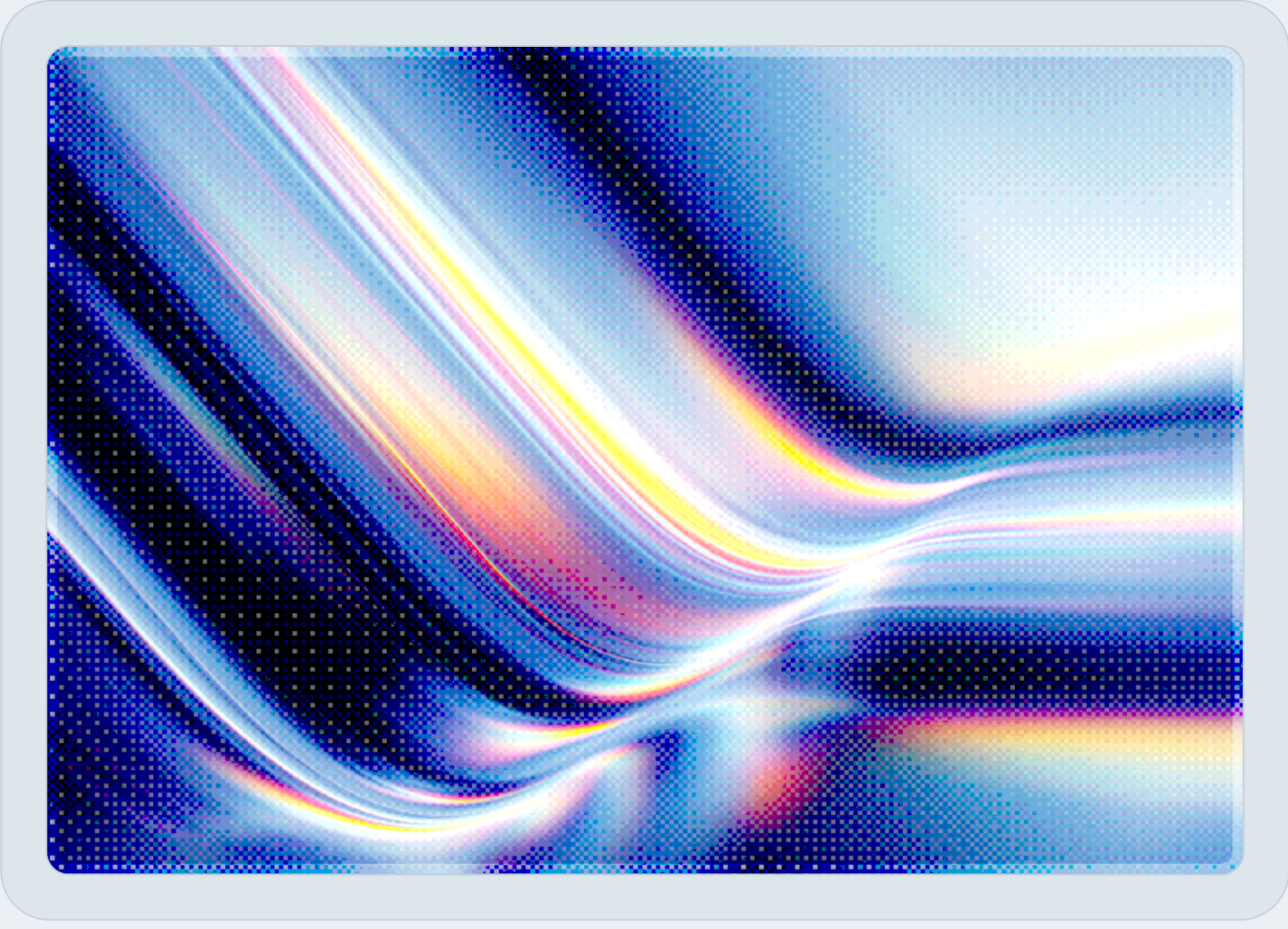
CSAI Foundation | Cloud Security Alliance

AI-Powered Supply Chain Wave: Five Campaigns, One Pattern

Common Tradecraft Across TeamPCP, Context7, prt-scan, GlassWorm, and the Checkmarx Cascade

2026-04-28

AI-assisted Rapid Research



CSAI | CSA cloud security alliance®

Key Takeaways

Five distinct supply chain campaigns disclosed between February and April 2026 — TeamPCP, the ContextCrush flaw in Context7, the prt-scan GitHub Actions campaign, GlassWorm, and the Checkmarx cascade culminating in the Bitwarden CLI compromise — share enough common tradecraft to be treated as a single phenomenon rather than independent incidents [1][2][3][4][5]. In each case, attackers used large language models or AI-driven automation either to scale the attack itself or to target the credentials, configurations, and documentation feeds that AI coding assistants depend on. The cumulative footprint exceeds 1,000 compromised packages, extensions, and repositories across npm, PyPI, Docker Hub, Open VSX, and GitHub, with confirmed credential theft from at least 16 organizations including the European Commission's AWS environment [1][6].

The unifying pattern is two-sided. On the offensive side, AI lowers the cost of producing convincing cover commits, repository-aware payloads, and language-appropriate injections — the prt-scan campaign's final wave deployed payloads tailored to Python `conftest.py`, JavaScript `package.json`, and Rust `build.rs` conventions, while GlassWorm's commit history mimicked routine documentation tweaks and version bumps at a fidelity that is difficult to produce by hand at scale [3][4]. On the defensive-target side, attackers are increasingly hunting for AI-specific assets: API keys for Anthropic, OpenAI, Cohere, Google, Mistral, Replicate, and others; the configuration files of Claude Code, Cursor, Codex CLI, Kiro, Aider, Gemini CLI, and Continue; and the documentation feeds that flow into Model Context Protocol (MCP) servers [5][7][8]. The result is a supply chain risk model where the development-time AI assistant has become both an attack instrument and a high-value target.

Organizations should treat this wave as a structural shift, not a string of unrelated incidents. The immediate priorities are inventorying every AI coding assistant configured in the environment, enforcing OIDC-based trusted publishing for any package the organization maintains, restricting GitHub Actions `pull_request_target` triggers, and adding LLM provider keys and AI-tool configuration files to the highest-priority secret scanning and rotation lists. The strategic priority is recognizing that "supply chain" now includes the documentation and rules feeds consumed by MCP servers, the marketplaces that distribute IDE extensions, and the CI/CD actions that publish software — each of which has been independently exploited in this wave.

Background

Between February 20 and April 27, 2026, security researchers disclosed five overlapping campaigns whose combined breadth defines the current supply chain threat surface. The earliest in the cluster, GlassWorm, originated in October 2025 as a Visual Studio Marketplace and Open VSX campaign and was reactivated and expanded across npm and GitHub repositories beginning in February 2026, ultimately implicating at least 72 VS Code extensions, 88 npm packages uploaded across three waves, and over 151 GitHub repositories [4][9]. Aikido and Socket independently confirmed that the malicious payloads use invisible Unicode characters — Tags block code points that render as zero-width in editors and terminals — to hide loader code inside otherwise innocuous JavaScript and TypeScript files [4]. The malware uses Solana blockchain transactions as a dead drop resolver to retrieve current command-and-control endpoints, rotates wallet addresses to evade takedowns, and deploys a payload tracked as ZOMBI to harvest cryptocurrency wallets, environment variables, and CI/CD tokens [4].

The ContextCrush flaw in Context7, disclosed by Noma Security on March 5, 2026 after a private fix on February 23, exposed a different point in the AI-development supply chain [2][10]. Context7 is a widely deployed MCP server (roughly 50,000 GitHub stars, over 8 million npm downloads) used by Cursor, Claude Code, and similar assistants to fetch library documentation in real time. Its "Custom Rules" feature served library-maintainer-provided natural-language instructions to every querying AI agent without sanitization, allowing a malicious entry to instruct the assistant to read `.env` files, exfiltrate their contents to an attacker-controlled GitHub repository, and delete local folders — using the assistant's own tool access rather than any traditional code-execution primitive [2]. Upstash patched the flaw with rule filtering, but the disclosure established documentation feeds as a first-class supply chain asset.

The TeamPCP campaign, profiled by Palo Alto Networks Unit 42 on April 6, 2026, executed a cascading multi-victim compromise between February and late March 2026 [1]. The group breached Aqua Security's Trivy on March 19, Checkmarx KICS on March 21, BerriAI LiteLLM on March 23, and Telnyx's Python SDK on March 27, then propagated laterally to 47 additional npm packages within 60 seconds using stolen publishing tokens [1]. The wiper component, CanisterWorm, used Internet Computer Protocol (ICP) canisters as a decentralized C2 — the first documented offensive use of that infrastructure. CVE-2026-33634 (CVSS 9.4) was assigned to the underlying CI/CD vulnerability [11]. Unit 42 attributes more than 300 GB of exfiltrated data and over 500,000 stolen credentials to the campaign, including the breach of the European Commission's AWS environment [1].

The prt-scan campaign, documented by Wiz Research on April 14, 2026, ran for six waves between March 11 and April 3, with a single actor operating six disposable accounts to open more than 500 malicious GitHub pull requests — 475 in a single 26-hour window from the account `ezmtebo` [3]. The

campaign abused the `pull_request_target` workflow trigger, which executes in the base repository's context with access to its secrets, and successfully compromised at least two npm packages across 106 versions. The technical evolution across the six waves is consequential: early payloads were unobfuscated bash scripts, while the final wave deployed AI-generated, repository-aware wrappers with language-appropriate injection points and base64-encoded multi-stage loaders [3].

The Checkmarx cascade, disclosed in stages between March 23 and April 27, 2026, ties the cluster together. Checkmarx confirmed that its KICS Docker images, GitHub Actions, and VS Code extensions were compromised on March 23, with attackers later posting exfiltrated GitHub repository data on the dark web [5][12]. On April 22, attackers used the same `checkmarx/ast-github-action` artifact to inject a malicious workflow into Bitwarden's CI/CD pipeline, publishing `@bitwarden/cli@2026.4.0` for a 90-minute window during which 334 developer machines installed the trojanized package [7]. The Bitwarden payload specifically harvested the configuration files of Claude Code, Cursor, Codex CLI, Aider, Kiro, and Gemini CLI alongside SSH keys, npm tokens, and CI/CD secrets, and exfiltrated to `audit.checkmarx[.]cx` — a domain impersonating a Checkmarx audit endpoint [7]. Researchers attribute orchestration to TeamPCP based on the embedded string "Shai-Hulud: The Third Coming" [7].

Security Analysis

AI as the Offensive Force Multiplier

A consistent thread across four of the five campaigns is the use of large language models to lower the cost of plausibility. Aikido's analysis of GlassWorm's commit history found that malicious injections arrived alongside ordinary-looking documentation tweaks, version bumps, small refactors, and bug fixes, calibrated to each repository's existing voice — a pattern Aikido attributes to LLM-generated cover commits [4][9]. The prt-scan campaign's final wave moved from generic bash to wrappers that inspected the target repository's tech stack and produced injection points matched to its conventions, with Wiz noting that the same operator account scaled from manual-feeling early waves to over 475 pull requests in 26 hours once tooling matured [3]. The SANDWORM_MODE typosquat campaign, disclosed by Socket on February 20 with 19 malicious npm packages mimicking AI-development utilities such as `claud-code` and `cloude-code`, embedded prompt-injection payloads inside its rogue MCP server that read SSH keys, AWS credentials, `.npmrc` files, and `.env` files, and harvested API keys for nine LLM providers including Anthropic, OpenAI, Google, Mistral, Cohere, and Together [8][13].

The economic effect is to compress the work that previously gated mass campaigns. Generating credible cover commits, target-aware payloads, and convincing typosquatted package descriptions used to require either deep familiarity with each victim project or substantial manual effort. Each campaign in this wave shows attackers paying that cost once, in tooling, and amortizing it across hundreds of targets. The same shift that has made AI coding assistants productive for defenders has made automated, repository-aware injection accessible to attackers with modest sophistication.

AI Assistants and Their Inputs as the Defensive Target

Where AI is not the weapon, it is the prize. The Bitwarden CLI payload's enumeration of `~/.claude`, `~/.cursor`, `~/.codex`, `~/.aider`, `~/.kiro`, and Gemini CLI configuration paths is the most explicit example, with researchers describing it as the first documented supply chain attack that explicitly targets AI coding assistant credentials [7]. The SANDWORM_MODE worm extends the same logic into propagation: by injecting a malicious MCP server into the assistant's configuration, the malware turns the trusted-tool channel into a persistent prompt-injection vector that survives repository changes [8]. The Context7 ContextCrush flaw shows that the documentation channel feeding these assistants is itself a viable initial-access surface: a single malicious library entry serves poisoned instructions to every developer who queries it, with no need to compromise either the package registry or the developer's machine [2].

The pattern points to a category of asset that traditional supply chain defenses do not consistently cover: the configuration of an AI assistant, the documentation feed that conditions its responses, and the API keys it uses to call upstream model providers. These assets cross the boundary between a developer's local environment, the cloud SaaS that hosts the model, and the open registries that distribute extensions and MCP servers. A typical SBOM does not enumerate them, and a typical secret scanner does not always flag them.

Self-Propagation, Decentralized C2, and Stealth

Three of the five campaigns deployed self-propagating worms or near-equivalents. TeamPCP's npm worm used credentials stolen from the first compromise to publish poisoned versions of any package the victim maintained, infecting 47 additional packages across multiple namespaces in under 60 seconds [1]. SANDWORM_MODE used the same primitive to spread between developer environments without operator intervention, deliberately turning each compromised maintainer into a fan-out node [8]. GlassWorm's iterative reappearances — March, with the Open VSX expansion, then April with a Python-repository fork researchers dubbed ForceMemo — show a campaign whose tooling outlives any individual takedown [4][14].

Decentralized command and control compounds the durability problem. GlassWorm's use of Solana transactions as a dead drop resolver lets the operator rotate C2 endpoints by signing a new transaction rather than registering new domains, sidestepping the takedown channels security vendors use against conventional infrastructure [4]. TeamPCP's CanisterWorm wiper does the equivalent on the Internet Computer Protocol [1]. The two innovations together make blocklist-based defenses materially less effective than they were against earlier supply chain campaigns.

Stealth completes the picture. GlassWorm's invisible-Unicode payloads do not appear in code review or syntax-highlighted diffs without specialized tooling [4]. The prt-scan campaign's malicious workflow files were committed under PR titles like "ci: update build configuration" and matched the surrounding repository style closely enough that a substantial fraction of repository owners reviewed the PR without raising alarms [3]. The cumulative effect is that the human review processes most organizations rely on to catch supply chain attacks are calibrated against an attacker who is neither AI-augmented nor patient enough to mimic project conventions — a calibration this wave invalidates.

Trust Inversion in Security Tooling

The most uncomfortable property of this wave is that the compromised software is disproportionately the software organizations buy to defend the supply chain. Trivy and KICS are vulnerability scanners. Checkmarx publishes static application security testing tooling. LiteLLM is an AI gateway whose central purpose is brokering and securing access to model providers. Bitwarden distributes a password manager. Each of these vendors holds privileged positions inside customer CI/CD pipelines, with the implicit trust that comes with a security label. Compromising any one of them yields lateral access disproportionate to the effort, and the attackers in this wave appear to have selected targets with that property in mind.

The implication is that "use a security tool" is no longer a sufficient mitigation by itself. The question shifts to whether the security tool's own software supply chain — its publishing keys, its CI/CD workflow, its container images, its IDE extensions — is independently verifiable and segmented. The Checkmarx cascade is the proof point: a single compromised GitHub Action distributed under the Checkmarx namespace was sufficient to trojanize Bitwarden's CLI three weeks later, despite Bitwarden's own infrastructure being uninvolved in the original intrusion [7].

Recommendations

This wave requires a response that spans immediate triage, near-term hardening, and longer-term program adjustments. The immediate triage addresses known indicators of compromise and the most actively exploited misconfigurations. The hardening actions close the structural gaps — in publishing

credential scope, provenance attestation, and MCP server governance — that made lateral spread possible once an initial foothold was established. The strategic layer recasts the scope of supply chain risk in terms of the AI-specific assets and channels this wave demonstrated are now in scope for adversaries.

Immediate Actions

The first priority is establishing an accurate inventory of AI coding assistant deployments. Every AI coding assistant configured in development environments should be identified, and its configuration directories — `~/.claude`, `~/.cursor`, `~/.codex`, `~/.aider`, `~/.kiro`, the Gemini CLI paths, and equivalents — should be treated as high-priority secret stores [7]. These paths warrant addition to EDR monitoring, host-level file integrity baselines, and exfiltration prevention rules. In parallel, every LLM provider API key stored in any developer environment exposed to npm, PyPI, or VS Code extension installations during the campaign window should be rotated — this includes keys for Anthropic, OpenAI, Cohere, Google, Mistral, Replicate, Together, Fireworks AI, and others catalogued in the SANDWORM_MODE disclosure [8].

GitHub Actions workflows require targeted review. Audit for `pull_request_target` triggers and remove or restrict them, as the trigger executes with access to repository secrets when a fork-based PR runs the workflow; prt-scan demonstrated active scanning for this misconfiguration at a rate of 475 PRs in 26 hours from a single account [3]. Where the trigger cannot be removed, gate it behind explicit approval and avoid checking out the PR head commit during the workflow. Any version of `@bitwarden/cli@2026.4.0` installed between 5:57 PM and 7:30 PM ET on April 22, 2026 should be treated as an active compromise indicator [7], and any consumed `checkmarx/ast-github-action` artifact should be audited against Checkmarx's published remediation guidance [5].

Suspect packages, extensions, and images should be pulled from inventories and replaced. This set includes the 19 SANDWORM_MODE typosquats disclosed by Socket, the GlassWorm npm packages disclosed by Endor Labs, the TeamPCP-poisoned LiteLLM 1.82.7/1.82.8 and Telnyx Python SDK versions, the trojanized Bitwarden CLI, the GlassWorm-affected VS Code and Open VSX extensions cataloged by Socket and Aikido, and any KICS Docker image pulled between March 21 and Checkmarx's remediation [4][7][8].

Short-Term Mitigations

The structural gap that enabled TeamPCP's lateral spread — long-lived, broadly scoped publishing tokens — can be closed through OIDC-based trusted publishing. Adopting trusted publishing for any package the organization maintains on npm, PyPI, or container registries eliminates the token class that

TeamPCP used to fan out from a single compromise to 47 additional packages in under a minute [1]. Where trusted publishing is not yet supported by a registry, publishing tokens should be restricted to single-package scopes with the shortest practical expiration.

Provenance verification on package consumption provides a complementary control on the consumer side. Implementing `npm audit signatures` and SLSA-level attestations published through GitHub Actions creates a verification path that the Bitwarden incident illustrates clearly: the trojanized CLI was published from Bitwarden's legitimate identity but with a CI/CD payload that diverged from the public source repository, an inconsistency that source-matched provenance attestations would have flagged before installation.

MCP server registrations deserve dedicated governance. The SANDWORM_MODE worm's persistence mechanism is a malicious MCP entry that survives the removal of the original npm package, while ContextCrush demonstrates that even legitimate MCP servers can serve poisoned instructions through registry feeds [2][8]. An allowlist of approved MCP servers, with alerting on any new registration and mandatory human review for any MCP entry that requests file system or network tool access, closes both the persistence and the documentation-poisoning vectors this wave exploited. Detection should also cover indicators specific to this campaign cluster: Tags-block code points and zero-width joiners in source files (GlassWorm), the Solana wallet addresses and ICP canister identifiers published by Aikido and Unit 42 [1][4], and outbound traffic to `audit.checkmarx[.]cx`, which should be blocked at egress regardless of whether Checkmarx tooling is in use [7].

Strategic Considerations

The wave's defining message is that the boundary of "supply chain" has expanded to include the documentation channels feeding AI assistants (Context7), the marketplaces distributing IDE extensions (Open VSX, VS Code Marketplace), the GitHub Actions namespaces from which CI/CD steps are pulled (Checkmarx), the package registries (npm, PyPI), and the MCP server registries that AI assistants consult at runtime. A supply chain risk program scoped only to direct dependencies declared in package manifests will miss several of these channels entirely.

The Checkmarx cascade also reframes vendor procurement. Compromise of a security vendor's publishing pipeline can propagate to a customer organization through a path the customer does not control. Procurement should explicitly inquire about each vendor's CI/CD security posture, publishing-credential scope, artifact signing, and incident-disclosure timelines, with contractual commitments for prompt notification of pipeline-integrity incidents. Organizations should plan for additional waves rather than treat this cluster as a closing episode: the economics that produced these five campaigns — AI

making cover commits and repository-aware payloads cheap, AI assistants creating high-value credential assets in developer environments, security vendors offering disproportionate lateral reach when compromised — remain in place.

CSA Resource Alignment

The CSA AI Controls Matrix (AICM) provides the most directly applicable framework for the issues this wave raises [15]. The Threat and Vulnerability Management domain establishes control objectives for inventory, monitoring, and timely remediation of vulnerabilities in AI infrastructure — a category that this wave demonstrates includes MCP servers, AI gateways, IDE extensions, and assistant configuration files. The Identity and Access Management domain addresses the credential-rotation and least-privilege practices that limit the blast radius of stolen publishing tokens, which TeamPCP's worm-based propagation made decisive. The Supply Chain Management, Transparency, and Accountability domain provides the controls that govern AI-specific software components and their build provenance, which the OIDC-based trusted publishing recommendation implements in practice. AICM's role as a superset of the Cloud Controls Matrix is particularly load-bearing here because many of the affected components — IDE extensions, MCP servers, documentation feeds — are AI-specific and not addressed in CCM alone.

The CSA MAESTRO framework for agentic AI threat modeling speaks directly to the prompt-injection-via-documentation pattern that ContextCrush exemplifies and the malicious-MCP pattern that SANDWORM_MODE deploys [16]. MAESTRO Layer 4 (deployment and infrastructure) and Layer 6 (model-serving and gateway) cover the runtime channels through which these attacks operate, while Layer 1 (foundation models and training) is implicated by attacks that target stored model-provider credentials. Threat modeling against MAESTRO would have anticipated several of the attack vectors documented in this wave before their first observed use.

CSA's Securing LLM-Backed Systems guidance and the LLM Threats Taxonomy describe the layered controls — authentication, authorization, secret scoping, audit logging, and anomaly detection — that prevent single-point compromises from cascading into AI-system-wide failures [17][18]. CSA's Zero Trust guidance applies to the developer endpoint that hosts AI assistant configurations: those hosts hold credentials with cross-cloud and cross-provider impact and should be governed under the same Zero Trust posture as production infrastructure [19]. The CSA STAR program provides an assurance mechanism for vendors whose publishing pipelines are now part of customers' supply chain blast radius; security tool procurement should prefer vendors with STAR-level documentation that specifically covers CI/CD integrity and publishing-credential scoping.

References

[1] Unit 42, Palo Alto Networks. "Weaponizing the Protectors: TeamPCP's Multi-Stage Supply Chain Attack on Security Infrastructure." Palo Alto Networks, April 2026.

[2] Noma Security. "ContextCrush: The Context7 MCP Server Vulnerability Hiding in Plain Sight." Noma Security, March 2026.

[3] Wiz Research. "Six Accounts, One Actor: Inside the prt-scan Supply Chain Campaign." Wiz, April 2026.

[4] The Hacker News. "GlassWorm Supply-Chain Attack Abuses 72 Open VSX Extensions to Target Developers." The Hacker News, March 2026.

[5] Checkmarx. "Supply Chain Security Incident Update." Checkmarx Blog, April 2026.

[6] GitGuardian. "No Off Season: Three Supply Chain Campaigns Hit npm, PyPI, and Docker Hub in 48 Hours." GitGuardian Blog, April 2026.

[7] The Hacker News. "Bitwarden CLI Compromised in Ongoing Checkmarx Supply Chain Campaign." The Hacker News, April 2026.

[8] Socket. "SANDWORM_MODE: Shai-Hulud-Style npm Worm Hijacks CI Workflow and AI Toolchain." Socket, February 2026.

[9] Aikido. "Glassworm Returns: Invisible Unicode Malware Found in 150+ GitHub Repositories." Aikido, March 2026.

[10] SC Media. "Context7 MCP Server Flaw Could Allow Malicious Instructions for AI Assistants." SC Media, March 2026.

[11] Burns & McDonnell 1898 Advisories. "March 2026 Developer Supply Chain Attack Wave: TeamPCP CI/CD Infrastructure Campaign (CVE-2026-33634)." 1898 & Co., April 2026.

[12] The Hacker News. "Checkmarx Confirms GitHub Repository Data Posted on Dark Web After March 23 Attack." The Hacker News, April 2026.

[13] Help Net Security. "Self-Spreading npm Malware Targets Developers in New Supply Chain Attack." Help Net Security, February 2026.

[14] SecurityWeek. "ForceMemo: Python Repositories Compromised in GlassWorm Aftermath." SecurityWeek, April 2026.

[15] Cloud Security Alliance. "AI Controls Matrix." CSA, 2025.

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

[17] Cloud Security Alliance. "Securing LLM Backed Systems: Essential Authorization Practices." CSA, 2024.

[18] Cloud Security Alliance. "CSA Large Language Model (LLM) Threats Taxonomy." CSA, 2024.

[19] Cloud Security Alliance. "Zero Trust Advancement Center." CSA, 2024.