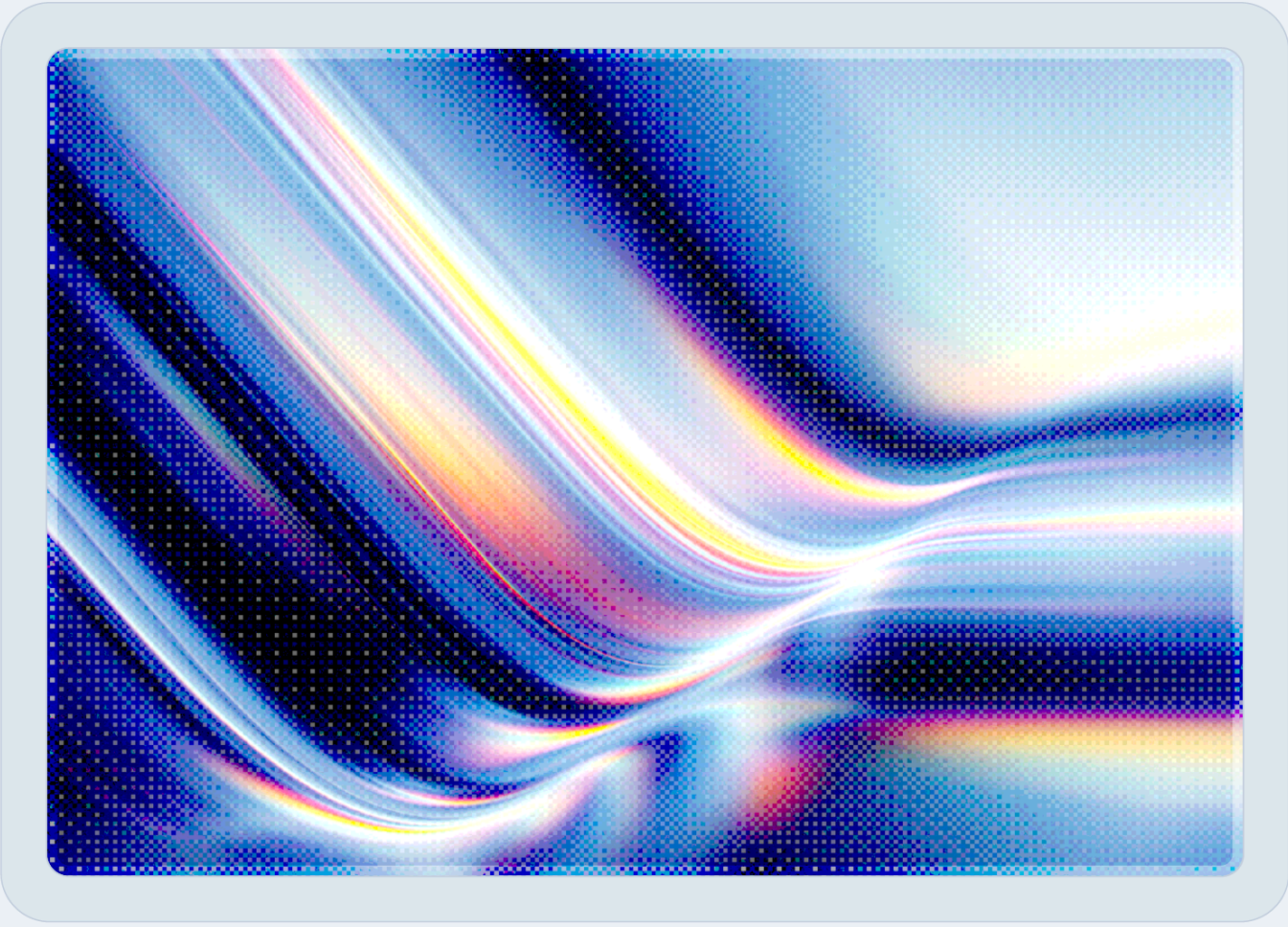
CSAI Foundation | Cloud Security Alliance

NGINX Rift: AI-Discovered 18-Year Flaw Enables Unauthenticated RCE

CVE-2026-42945 and the Emergence of Autonomous Vulnerability Discovery

2026-05-17

AI-assisted Rapid Research



CSAI | CSA cloud security alliance®

Key Takeaways

On May 13, 2026, F5 and security firm DepthFirst disclosed CVE-2026-42945, a critical heap buffer overflow in NGINX's URL rewrite module that had been present in the codebase since 2008 – an 18-year window during which vulnerable deployments were undetected and therefore unmitigated. The flaw carries a CVSS v4 score of 9.2 and affects NGINX Open Source versions 0.6.27 through 1.30.0 and NGINX Plus R32 through R36 [1]. Because NGINX serves approximately 33% of all websites globally [2], the potential exposure surface spans hundreds of millions of internet-facing hosts.

The vulnerability was not found by a human researcher conducting manual code review. It was discovered autonomously by DepthFirst's AI-powered analysis platform in approximately six hours, during which the system also identified three additional memory corruption flaws in the same codebase [3]. A public proof-of-concept exploit is now available on GitHub [4], substantially compressing the timeline from disclosure to potential weaponization.

Security teams should treat this vulnerability as requiring emergency response. Patch NGINX immediately to version 1.30.1 or 1.31.0. Organizations that cannot patch immediately should convert unnamed PCRE captures to named captures in all rewrite directives as a configuration-level mitigation.

Background

NGINX is the world's most widely deployed web server, powering a broad cross-section of the internet's infrastructure – from startup applications to hyperscaler API gateways. As of May 2026, W3Techs data places NGINX's share of known web servers at approximately 33% [2], representing millions of active deployments. Many of those deployments rely on the `ngx_http_rewrite_module`, which provides URL rewriting, conditional logic, and variable manipulation and is among the most commonly configured NGINX components.

The vulnerability at issue, CVE-2026-42945, was introduced into the NGINX source tree with version 0.6.27, released in 2008. The flaw originates from an inconsistency in the rewrite engine's buffer arithmetic: when processing a rewrite rule that uses an unnamed PCRE capture group (for example, `$1` or `$2`) alongside a replacement string containing a question mark, NGINX computes the size of the destination buffer under one set of assumptions but then copies data under a different set. The net

effect is that attacker-controlled bytes drawn from the URI can overwrite memory beyond the allocated heap region in the worker process [5]. The flaw is only triggered when a `rewrite` directive using this pattern is followed by another `rewrite`, `if`, or `set` directive within the same location block — a configuration frequently encountered in applications performing multi-step URL normalization or query string manipulation.

For eighteen years, this logic error was invisible to the security community. No static analysis tool, code audit, or vulnerability scanner surfaced it during NGINX's extensive production lifetime across hundreds of millions of deployments. Its discovery came instead from DepthFirst, an AI-native security research firm that applied its autonomous analysis platform to the NGINX source repository. Starting from a single onboarding action, the platform examined the codebase and returned four memory corruption findings — including CVE-2026-42945 — in approximately six hours [3]. The other three findings from the same session (CVE-2026-42946, a CVSS 8.3 excessive memory allocation issue; CVE-2026-40701, a CVSS 6.3 use-after-free in the SSL module; and CVE-2026-42934, a CVSS 6.3 out-of-bounds read in the charset module) underscore that the AI system was not performing a shallow heuristic scan but rather conducting structured analysis capable of identifying distinct vulnerability classes across multiple NGINX subsystems [6].

DepthFirst reported the findings to the NGINX security team on April 21, 2026. NGINX confirmed all four vulnerabilities on April 24. Coordinated public disclosure and patch release followed on May 13, 2026 [3][11]. A public GitHub repository containing a working proof-of-concept exploit for CVE-2026-42945, demonstrating unauthenticated remote code execution on targets with ASLR disabled, was released alongside the advisory [4].

Security Analysis

Vulnerability Mechanics

The exploit path for CVE-2026-42945 requires no credentials and no established session. An unauthenticated attacker sends a single crafted HTTP request whose URI is designed to trigger the buffer arithmetic mismatch during rewrite processing. Under the most common exploitation scenario, the worker process crashes and is automatically restarted by the NGINX master process, creating a denial-of-service condition that is difficult to distinguish from normal traffic spikes without instrumentation at the worker lifecycle level [5].

The more serious scenario — arbitrary code execution — requires ASLR to be disabled or bypassed on the target system. While ASLR is enabled by default on modern Linux distributions, many embedded systems, legacy appliances, containerized workloads with non-standard kernel configurations, and certain cloud-managed NGINX deployments may run with weakened or disabled ASLR. F5's NGINX Plus R32 through R36, used extensively in enterprise API gateway and load balancer deployments, falls within the affected version range [1][9]. The combination of unauthenticated access, a single-request trigger, and a 33% global deployment footprint creates a severe risk profile: no credentials required, a single malformed HTTP request can trigger the flaw, and tens of millions of potentially affected hosts represent a substantial attack surface.

Exposure Scope

The vulnerability is present in any NGINX deployment that meets three simultaneous conditions: an affected version is running; a `rewrite` directive uses unnamed PCRE captures (`$1` , `$2`); and that directive is followed by another `rewrite` , `if` , or `set` directive in the same block [5]. Determining whether a given production configuration meets these conditions requires reviewing all NGINX configuration files and any dynamically loaded configurations, including those embedded in application-level directives written by developers rather than platform administrators. In organizations where NGINX configuration is distributed across development teams, infrastructure-as-code repositories, and Kubernetes ingress controllers, this audit is non-trivial and should not be assumed complete without systematic tooling [5][12].

Beyond the critical flaw, the three companion vulnerabilities discovered in the same session warrant attention. CVE-2026-42946 (CVSS 8.3) affects the SCGI and uWSGI modules, which are common in Python-based application stacks. CVE-2026-40701 (CVSS 6.3) is a use-after-free in the SSL module triggered when mutual TLS verification and OCSP stapling are both enabled — a configuration found in high-assurance services. CVE-2026-42934 (CVSS 6.3) affects the charset conversion module. All four should be remediated together [6].

The AI Discovery Dimension

From a security operations standpoint, the manner in which this vulnerability was found is as significant as the vulnerability itself. Autonomous AI analysis surfaced a critical flaw despite NGINX's 18-year deployment history as one of the world's most widely used web servers — a codebase subject to continuous community code review and automated scanning across hundreds of millions of production deployments, none of which identified this issue. This outcome challenges a longstanding assumption in the security community: that age and deployment scale, combined with community scrutiny, are effective proxies for the absence of hidden critical vulnerabilities.

This case is not without precedent. In October 2024, Google's Big Sleep agent — a collaboration between Google Project Zero and Google DeepMind — identified a previously unknown stack buffer underflow in SQLite before the affected code reached an official release, representing the first documented instance of an AI system autonomously discovering a zero-day in widely deployed open-source software [16]. The pattern emerging from CVE-2026-42945 and earlier cases is that AI systems capable of structured, compositional reasoning over source code can surface vulnerabilities rooted in subtle interactions between program components — precisely the class of flaw that traditional static analysis and fuzzing approaches are least effective at identifying [7][16].

The implication for defenders is dual-edged. Security teams may be able to deploy similar AI-powered analysis against their own proprietary and open-source dependencies to find previously hidden flaws before adversaries do. But adversaries face no equivalent barrier to adopting the same tools. The same autonomous scanning capability that allowed DepthFirst to analyze NGINX in six hours can, in principle, be directed at any target codebase — including closed-source firmware, embedded system software, and enterprise applications where no responsible disclosure process will constrain the exploitation timeline. The public availability of the PoC for CVE-2026-42945 demonstrates how quickly theoretical AI-discovered vulnerabilities can transition to usable offensive capability [4].

Current Exploitation Status

As of May 17, 2026 — four days after coordinated disclosure — CVE-2026-42945 has not been confirmed as actively exploited in the wild, and it has not been added to CISA's Known Exploited Vulnerabilities catalog [8]. However, the existence of a working, publicly available proof-of-concept exploit means the operational window before weaponized variants appear in threat actor toolkits is likely measured in days, not weeks. Opportunistic scanning for vulnerable configurations should be assumed to be underway. Organizations relying on KEV catalog additions as their primary patching trigger for this vulnerability should reconsider that posture given the severity score, the scale of exposure, and the public PoC. Given the point-in-time nature of this assessment, readers should verify current KEV status directly at the CISA Known Exploited Vulnerabilities Catalog [8].

Recommendations

Immediate Actions

The priority action is patching. NGINX has released fixed versions 1.30.1 and 1.31.0 for Open Source, and the corresponding NGINX Plus releases. Organizations should treat this as an emergency change requiring expedited patching outside of normal maintenance windows given the CVSS 9.2 score and public PoC availability [1][10][13].

For environments where patching cannot be completed immediately, the recommended workaround is to replace all unnamed PCRE captures (`$1`, `$2`) with named captures in every affected `rewrite` directive. For example, the vulnerable pattern `rewrite ^/(.+)$ /new/$1?; rewrite ...` should be rewritten to use named captures explicitly. This configuration change eliminates the specific code path that triggers the buffer overflow [5]. Organizations should validate that all NGINX configuration files have been reviewed — including those managed by application teams, generated by CI/CD pipelines, or embedded in Kubernetes ingress annotations [13].

Short-Term Mitigations

After patching, security teams should inventory all NGINX-adjacent components within their environment to address the three companion vulnerabilities. CVE-2026-42946 affects deployments using SCGI or uWSGI proxying; CVE-2026-40701 affects mutual TLS configurations with OCSP stapling enabled; CVE-2026-42934 affects deployments using the charset conversion module with disabled proxy buffering. Each of these requires independent assessment and patch verification [6].

Monitoring should be enhanced at the NGINX worker process lifecycle level. Abnormal worker restarts — particularly those correlated with unusual URI patterns or triggered by traffic from unexpected source ranges — may indicate exploitation attempts of CVE-2026-42945 in the DoS-only configuration. Web application firewall rules blocking crafted rewrite-triggering requests may reduce opportunistic exploitation attempts while patching is underway, though WAF rules should not be treated as a primary control — bypass techniques for rule-based mitigations routinely follow the public disclosure of exploit details.

Strategic Considerations

The discovery mechanism for this vulnerability suggests several broader implications for security posture, though the evidence base rests primarily on this single disclosure event and a small number of analogous cases. With that context explicit, the nature of what the DepthFirst platform accomplished [3] points toward a structural shift in how the security community should approach legacy open-source dependencies. Codebases that have been in continuous production use for a decade or more may contain latent vulnerabilities that evaded every prior scanning approach but are now findable with AI-powered compositional analysis — as this disclosure and the 2024 SQLite finding both suggest [3][16]. Critical infrastructure components — web servers, TLS libraries, DNS resolvers, authentication middleware — should be evaluated as candidates for AI-assisted re-analysis as these tools become more accessible.

Organizations should also revisit their software bill of materials (SBOM) practices with this event in mind. Knowing which internal systems, products, and services run which versions of NGINX — including NGINX bundled inside third-party appliances, CDN edge software, and container base images — is a prerequisite for rapid response when vulnerabilities like this emerge. Teams that could not answer "what version of NGINX is running where in our environment?" within hours of the May 13 advisory should treat that gap as a remediation priority in its own right [3].

CSA Resource Alignment

CVE-2026-42945 and the circumstances of its discovery touch multiple areas of CSA's AI safety and cloud security research.

The MAESTRO framework for agentic AI threat modeling [14] provides context for understanding autonomous vulnerability scanners as a new class of security-relevant AI agent operating at Layer 3 (Agentic AI Tools). The same architectural reasoning that MAESTRO applies to offensive agentic threats — autonomous goal-directed behavior, tool use without continuous human oversight, output that directly affects security posture — is directly relevant to defensive autonomous analysis platforms like the one DepthFirst deployed. Organizations evaluating such tools should apply MAESTRO's guidance on agent trust boundaries, output validation, and operational scope constraints before deploying AI vulnerability scanners at production scale.

CSA's AI Controls Matrix (AICM) [15] addresses AI system risk through its vulnerability and lifecycle management control domains. The NGINX Rift case clearly illustrates the scenario those controls anticipate: an AI-mediated change to the vulnerability discovery pipeline that produces security findings

at a speed and scale that exceeds human analyst capacity to validate and prioritize. AICM's controls for human-in-the-loop oversight and AI output verification are directly applicable to any organization integrating autonomous scanning into its vulnerability management program.

CSA's STAR for AI registry and AI-CAIQ questionnaire provide a mechanism for AI security service providers — including autonomous vulnerability discovery firms — to demonstrate transparency about their methodologies, data handling practices, and scope boundaries. As more security vendors incorporate AI-driven analysis into their products, procurement and vendor risk teams should include STAR for AI assessments as part of their evaluation criteria.

Finally, CSA's Zero Trust guidance applies directly to the recommended response posture: assume that any unpatched, externally reachable NGINX instance is a viable exploitation target regardless of perceived traffic profile, WAF coverage, or historical incident history. The existence of a public PoC substantially narrows the window between "theoretical vulnerability" and "operational threat actor capability," and organizations that delay patching in favor of detective controls are accepting a risk posture that conflicts with the Zero Trust principle of minimizing attack surface — detection supplements but does not substitute for reduction of exploitable exposure.

References

[1] NIST NVD. "CVE-2026-42945 Detail." National Vulnerability Database, May 2026.

[2] W3Techs. "Usage Statistics and Market Share of Nginx." W3Techs Web Technology Surveys, May 2026.

[3] DepthFirst. "NGINX Rift: Achieving NGINX Remote Code Execution via an 18-Year-Old Vulnerability." DepthFirst Security Research, May 2026.

[4] DepthFirstDisclosures. "Nginx-Rift: exploit for CVE-2026-42945." GitHub, May 2026.

[5] Picus Security. "NGINX Rift: CVE-2026-42945 Critical Heap Buffer Overflow Vulnerability Explained." Picus Security Blog, May 2026.

[6] BleepingComputer. "18-year-old NGINX vulnerability allows DoS, potential RCE." BleepingComputer, May 2026.

[7] CSO Online. "AI agent finds 18-year-old remote code execution flaw in Nginx." CSO Online, May 2026.

[8] CISA. "Known Exploited Vulnerabilities Catalog." Cybersecurity and Infrastructure Security Agency, 2026.

[9] The Hacker News. "18-Year-Old NGINX Rewrite Module Flaw Enables Unauthenticated RCE." The Hacker News, May 2026.

[10] AlmaLinux. "NGINX Rift (CVE-2026-42945) Patches Released." AlmaLinux Blog, May 13, 2026.

[11] Security Affairs. "NGINX Rift: an 18-year-old flaw in the world's most deployed web server just came to light." Security Affairs, May 2026.

[12] Orca Security. "NGINX Rewrite Module Flaw (CVE-2026-42945)." Orca Security Blog, May 2026.

[13] nginx.org. "NGINX Security Advisories." F5/NGINX, May 2026.

[14] Cloud Security Alliance. "Agentic AI Threat Modeling Framework: MAESTRO." CSA AI Safety Initiative, 2025.

[15] Cloud Security Alliance. "AI Controls Matrix (AICM)." CSA AI Controls Working Group, 2025.

[16] Google Project Zero. "From Naptime to Big Sleep: Using Large Language Models To Catch Vulnerabilities In Real-World Code." Google Project Zero, October 2024.