Hive Pro

Threat Level

Red

Hiveforce Labs

THREAT ADVISORY

VULNERABILITY REPORT

MCPwn: Critical Nginx UI Bug Opens the Door to Remote Control

Date of Publication	Admiralty Code	TA Number
April 16, 2026	A1	TA2026103

Summary

First Seen: March 04, 2026
Affected Products: Nginx UI
Impact: CVE-2026-33032, dubbed MCPwn, exposes a critical flaw in Nginx UI that effectively removes the need for authentication to access powerful backend functions. Due to a misconfiguration in how security controls are applied, attackers can directly interact with the /mcp_message endpoint, left unprotected by authentication and further weakened by a fail-open IP whitelist that allows all requests by default. This oversight enables anyone to invoke sensitive MCP tools, ranging from configuration changes to full-service restarts, without credentials. With exploitation already observed in the wild and thousands of exposed instances across cloud environments, this vulnerability turns misconfigured deployments into easy targets, making immediate patching and proper access control enforcement essential.

CVE

CVE	NAME	AFFECTED PRODUCT	ZERO-DAY	CISA KEV	PATCH
CVE-2026-33032	MCPwn (Nginx Authentication Bypass Vulnerability)	Nginx UI	✗	✗	✓

Vulnerability Details

#1

CVE-2026-33032, codenamed MCPwn by Pluto Security, exposes a critical security gap rooted in improper access control design. The flaw stems from an architectural inconsistency in how authentication is enforced across the MCP Server-Sent Events (SSE) transport layer in Nginx UI. The implementation splits communication into two endpoints, a persistent listening channel (GET /mcp) and an action channel (POST /mcp_message), both of which ultimately rely on the same backend handler, mcp.ServeHTTP(), to process tool operations. However, authentication middleware is selectively applied only to the /mcp endpoint, leaving the /mcp_message route completely exposed.

#2

Compounding the issue is a flawed fallback security mechanism. The /mcp_message endpoint relies solely on an IP whitelist for access control, but this safeguard is ineffective due to a fail-open design. In default configurations where the IP whitelist is unset, the middleware permits all incoming requests without restriction. This behavior, explicitly coded to allow execution when no whitelist entries exist, effectively renders fresh installations fully accessible to unauthenticated users. The combination of missing authentication and permissive default settings creates a dangerous condition where any external actor can directly interact with sensitive backend functionality.

#3

Exploitation is both simple and highly impactful. An attacker can initiate a session by issuing a GET /mcp request to establish an SSE connection and retrieve a session identifier. This is followed by crafted POST requests to /mcp_message, embedding JSON-RPC payloads that invoke available MCP tools. Of the 12 exposed tools, several enable high-impact actions such as modifying configurations, reloading services, or restarting Nginx, effectively granting attackers the ability to tamper with server behavior or disrupt operations. Others provide visibility into system configurations, enabling reconnaissance without any authentication barrier. Notably, these actions require no credentials, no API keys, tokens, or session validation, making exploitation trivial.

#4

The vulnerability affects all Nginx UI versions up to and including 2.3.5, placing a significant number of deployments at risk. Evidence suggests that exploitation is already underway in the wild. Security researchers have flagged CVE-2026-33032 as one of the most actively targeted vulnerabilities, assigning it a near-critical risk score. Internet-wide scans have identified thousands of exposed instances across major cloud providers, many of which are publicly accessible and running with default configurations.

#5 Given the ease of exploitation, the absence of authentication controls, and confirmed in-the-wild activity, this vulnerability represents a serious operational risk. Administrators should treat it as a priority, ensuring that affected systems are updated immediately and that access controls are properly enforced to prevent unauthorized interaction with MCP endpoints.

Vulnerability

CVE ID	AFFECTED PRODUCTS	AFFECTED CPE	CWE ID
CVE-2026-33032	Nginx UI (all versions through v2.3.5)	cpe:2.3:a:nginxui:nginx_ui:*:*:*:*:*:*:*:*	CWE-306

Recommendations

Update Nginx UI to the Latest Version Immediately: The fix released on March 15, 2026, adds the missing `AuthRequired()` middleware to the `/mcp_message` endpoint. Given confirmed active exploitation in the wild, upgrading is the most critical and urgent remediation step. Organizations should prioritize this update ahead of standard patch cycles.

Disable MCP or Restrict IP Whitelisting as an Emergency Stopgap: If an immediate upgrade is not feasible, disable the MCP integration entirely or configure the IP whitelist to explicitly allow only trusted management hosts. Do not leave the default fail-open whitelist configuration in place, as it permits unrestricted access from any network-reachable host.

Audit Nginx Configuration Files for Unauthorized Changes: Review the `conf.d/` and `sites-enabled/` directories for unfamiliar or suspicious configuration files that may have been injected through unauthenticated MCP tool calls. Examine Nginx access logs for unexpected requests to `/mcp_message` originating from untrusted IP addresses, and investigate any configuration reload events that do not correlate with authorized administrative activity.

Restrict Network Exposure of the Nginx UI Backend Port: Ensure that the Nginx UI backend port (default 9000) is not exposed to the public internet or untrusted network segments. Deploy firewall rules or network segmentation to limit access to authorized management workstations and administrative VPN subnets only.

Rotate Credentials and Review Administrative Accounts: If exploitation is suspected, rotate all administrative credentials for Nginx UI, including JWT secrets. Review active user accounts for unauthorized additions and invalidate all existing sessions. Additionally, inspect Nginx log format configurations for injected directives designed to capture Authorization headers or other sensitive data.

Potential MITRE ATT&CK TTPs

Tactic	Technique	Sub-technique
Initial Access	T1190: Exploit Public-Facing Application	
Execution	T1059: Command and Scripting Interpreter	
Persistence	T1505: Server Software Component	
Privilege Escalation	T1068: Exploitation for Privilege Escalation	
Discovery	T1083: File and Directory Discovery	
Collection	T1557: Adversary-in-the-Middle	
Impact	T1489: Service Stop	
	T1565: Data Manipulation	T1565.002: Transmitted Data Manipulation

Patch Link

https://github.com/0xJacky/nginx-ui/commit/413dc63

https://github.com/0xJacky/nginx-ui/releases

References

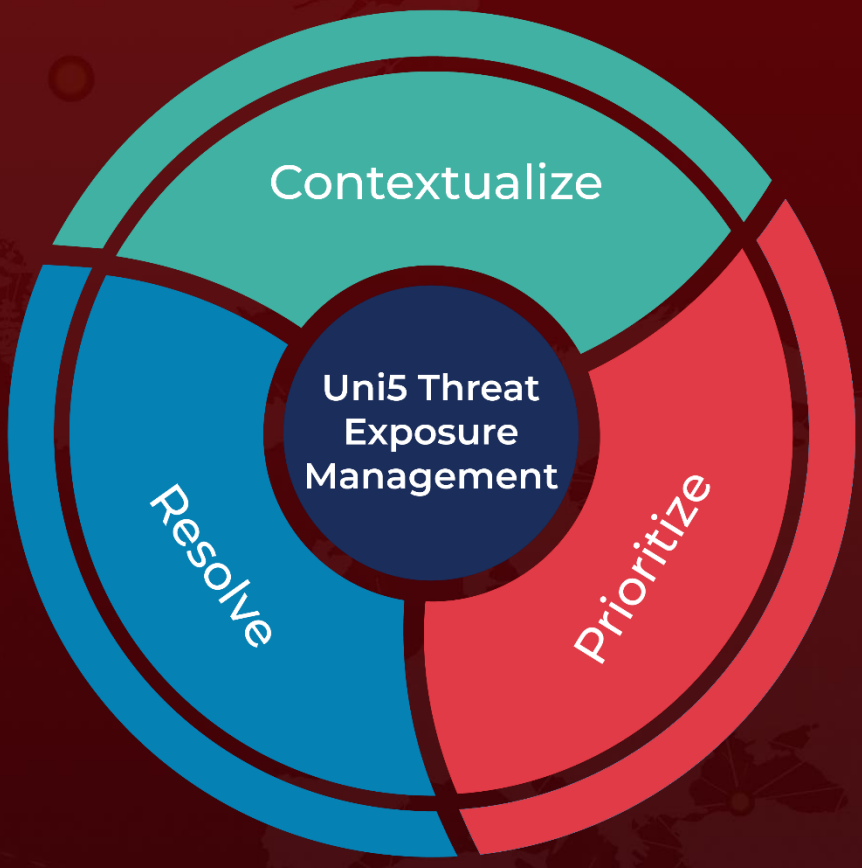
https://github.com/0xJacky/nginx-ui/security/advisories/GHSA-h6c2-x2m2-mwhf

https://pluto.security/blog/mcp-bug-nginx-security-vulnerability-cvss-9-8/

What Next?

At **Hive Pro**, it is our mission to detect the most likely threats to your organization and to help you prevent them from happening.

Book a free demo with **HivePro Uni5**: Threat Exposure Management Platform.



REPORT GENERATED ON

April 16, 2026 • 9:00 AM

More at www.hivepro.com