Hive Pro

Threat Level

Red

Hiveforce Labs

THREAT ADVISORY

VULNERABILITY REPORT

Squidbleed: Decades-Old Parser Flaw Exposes Sensitive Proxy Data

Date of Publication	Admiralty Code	TA Number
June 24, 2026	A1	TA2026175

Summary

First Seen: January 18, 1997 (introduced via Squid commit bb97dd37a); publicly disclosed June 10, 2026
Affected Products: Squid Web Proxy (all versions in default configuration)
Impact: CVE-2026-47729, dubbed Squidbleed, is a newly disclosed information disclosure flaw in the Squid web proxy that can expose sensitive data from the proxy's memory to remote attackers. Caused by a decades-old parsing bug in Squid's FTP gateway, the vulnerability allows a malicious FTP server to trigger an out-of-bounds read and retrieve fragments of previously processed data, including HTTP requests and authentication headers. While exploitation requires access to a trusted proxy and an attacker-controlled FTP server, the flaw affects all Squid versions in their default configuration. Although no active attacks have been reported, the vulnerability highlights how legacy code can quietly persist for years before exposing critical security risks.

CVE

CVE	NAME	AFFECTED PRODUCT	ZERO-DAY	CISA KEV	PATCH
CVE-2026-47729	Squidbleed (Squid Proxy Memory Leak Vulnerability)	Squid Software Foundation Squid (Web Proxy)	✗	✗	✓

Vulnerability Details

#1 The flaw tracked as CVE-2026-47729, dubbed Squidbleed, is an out-of-bounds read vulnerability (CWE-125) affecting the FTP gateway component of the Squid web proxy. The issue stems from improper input validation within the FTP directory-listing parser, allowing a remote attacker to read memory beyond the intended buffer. Researchers have drawn comparisons to Heartbleed because the bug can expose fragments of process memory to a remote party through a similar read-past-the-buffer condition.

#2 The vulnerable code dates to January 1997 and was originally introduced to accommodate the formatting quirks of NetWare FTP servers, which inserted additional spaces between timestamps and filenames in directory listings. A logic error in the parser causes it to continue scanning whitespace even after reaching the end of the string. Because the C standard treats the terminating null byte as part of the string searched by strchr(), the loop never exits as intended when it encounters the end of the input.

#3 An attacker can exploit the flaw by persuading the proxy to retrieve a directory listing from an FTP server under their control and supplying a specially crafted entry that contains only a timestamp and no filename. As the parser advances past the string boundary, it eventually reaches adjacent heap memory and interprets the contents as a filename. Squid then copies this unintended data and returns it to the attacker as part of the directory listing response.

#4 The impact is amplified by Squid's memory management design, which relies on recycled buffer pools without clearing previously used data. As a result, memory regions that once stored HTTP requests may still contain sensitive information when reused. Researchers demonstrated that under certain conditions, an attacker could recover remnants of a victim's traffic, including HTTP Authorization headers, from a shared proxy environment.

#5 Every Squid release is considered vulnerable in its default configuration because FTP support is enabled by default, and port 21 is included in the standard Safe_ports access list. Exploitation, however, requires two conditions: the attacker must already have permission to use the proxy, and the proxy must be able to connect to an attacker-controlled FTP server. Exposure is largely limited to plaintext HTTP traffic or deployments where Squid terminates TLS, since standard HTTPS traffic passing through CONNECT tunnels remains opaque to the proxy.

#6 Although proof-of-concept code has been made public, there is currently no evidence of active exploitation. The vulnerability was responsibly disclosed after being reported in April 2026, making it a patched and publicly known issue.

Vulnerability

CVE ID	AFFECTED PRODUCTS	AFFECTED CPE	CWE ID
CVE-2026-47729	Squid Web Proxy (all versions in default configuration prior to the upstream fix)	cpe:2.3:a:squid-cache:squid:*:*:*:*:*:*:*:*	CWE-125

Recommendations

Disable FTP Support Immediately: The most effective and immediate mitigation is to turn off FTP handling in Squid, which removes this attack surface entirely. Most organizations carry near-zero legitimate FTP traffic since Chromium-based browsers dropped FTP support years ago, so denying FTP via an ACL, removing port 21 from the Safe_ports ACL, and blocking the proxy's outbound port 21 at the firewall costs little and protects every build regardless of patch status.

Apply the Fix and Verify It: Update to a Squid build that contains the null-terminator check in src/clients/FtpGateway.cc, but verify the fix is actually present rather than trusting a version number. Public reporting has been inconsistent: the Squid maintainer initially indicated the fix shipped in 7.6, then corrected that the released fix lands in 7.7 (Squid 7.6 addressed a separate vulnerability, CVE-2026-50012), so confirm the guard "while (*copyFrom && strchr(w_space, *copyFrom))" is in your build or your distribution's backport before considering the issue closed.

Audit Exposed Credentials and Rotate Secrets: Because the flaw silently leaks other users' HTTP request data, assume that credentials, session tokens, and API keys transmitted in cleartext HTTP through an affected proxy may have been exposed. Rotate potentially affected secrets, invalidate active sessions, and prioritize any service that was accessed over plain HTTP through a shared Squid instance.

Enforce Encryption in Transit: Reduce the value of any future memory-disclosure flaw by ensuring sensitive traffic uses end-to-end TLS rather than cleartext HTTP, since HTTPS relayed as a CONNECT tunnel is not exposed by this class of bug. Eliminate plain-HTTP authentication flows and migrate legacy internal applications to HTTPS.

Potential MITRE ATT&CK TTPs

Tactic	Technique	Sub-technique
Credential Access	T1212: Exploitation for Credential Access	
	T1528: Steal Application Access Token	
Resource Development	T1588: Obtain Capabilities	T1588.006: Vulnerabilities

Patch Link

https://github.com/squid-cache/squid/releases

References

https://blog.calif.io/p/squidbleed-cve-2026-47729

https://github.com/0xBlackash/CVE-2026-47729

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 Demo of **HivePro**.

REPORT GENERATED ON

June 24, 2026 • 07:50 AM

More at www.hivepro.com