Hive Pro

Threat Level

Red

Hiveforce Labs

THREAT ADVISORY

VULNERABILITY REPORT

Unsafe by Default: KindaRails2Shell Exposes Ruby on Rails to File Read and RCE

Date of Publication	Admiralty Code	TA Number
August 07, 2026	A1	TA2026225

Summary

First Seen: July 29, 2026
Affected Products: Ruby on Rails Active Storage
Impact: Ruby on Rails has patched CVE-2026-66066, dubbed KindaRails2Shell, a critical flaw in Active Storage that leaves unsafe libvips loaders enabled by default, exposing applications that never made a configuration choice. An unauthenticated attacker who can upload an image can read arbitrary files off the server and receive them back rendered as pixels.

CVE

CVE	NAME	AFFECTED PRODUCT	ZERO-DAY	CISA KEV	PATCH
CVE-2026-66066	KindaRails2Shell (Ruby on Rails Active Storage Arbitrary File Read Vulnerability)	Ruby on Rails Active Storage	✗	✗	✓

Vulnerability Details

#1

The Ruby on Rails project has released fixes for CVE-2026-66066 (KindaRails2Shell), a critical arbitrary file read and remote code execution vulnerability in Active Storage, the framework's built-in file attachment and image-processing component. The flaw stems from Active Storage failing to disable libvips operations that the imaging library itself marks as unsafe for untrusted content. Affected applications use libvips for Active Storage image processing and accept image uploads from untrusted users. Rails selects Vips under load_defaults 7.0, and later defaults retain it.

#2

An unauthenticated attacker who can upload an image to an affected application can craft a file that libvips will parse through one of these unsafe loaders, causing the application to read a file of the attacker's choosing off the server filesystem and return its contents rendered as image pixels. The files most valuable to an attacker are the ones the Rails process can always reach: the process environment, which typically holds 'secret_key_base', and the encrypted credentials that the Rails master key unlocks. Once 'secret_key_base' is recovered, the attacker holds the application's master cryptographic key and can forge signed and encrypted material, escalating into full remote code execution.

#3

The root cause sits at the trust boundary between Active Storage and libvips rather than in either component's own logic. libvips reads and writes image formats through operations it calls loaders and savers, many of which are backed by third-party libraries and handle formats unrelated to web imagery. libvips itself marks a subset of these operations as unfuzzed, meaning they have not been hardened against hostile input. Since version 8.13, libvips has offered a mechanism to block those operations outright.

#4

Active Storage never invoked it, so every unfuzzed loader remained reachable by any file the application handed to the processor. The vulnerable setting is 'config.active_storage.variant_processor = :vips', which no subsequent default has changed. No threat actor group or malware family has been linked to activity involving this vulnerability, and the Rails Security Team has stated that it is not aware of exploitation attempts before or after disclosure.

Vulnerability

CVE ID	AFFECTED PRODUCTS	AFFECTED CPE	CWE ID
CVE-2026-66066	Ruby on Rails Active Storage, all versions before 7.2.3.2; version 8.0 and later, before 8.0.5.1; and version 8.1 and later, before 8.1.3.1.	cpe:2.3:a:rubyonrails:rails:-:*:*:*:*:*:*:*	CWE-1188

Recommendations

Upgrade Active Storage Without Waiting for a Patch Window: Move affected applications to Rails 7.2.3.2, 8.0.5.1, or 8.1.3.1, or a later release on those branches. It's advised to upgrade regardless of how a specific deployment is configured. Applications on Rails 7.1 or earlier will not receive a backport because those branches are end of life, so they must be migrated to a supported fixed branch or run one of the workarounds below indefinitely.

Upgrade libvips as a Separate Task: The fix depends on libvips being version 8.13 or later, because earlier builds cannot disable the unsafe operations at all. Upgrading Rails alone leaves an application on older libvips unprotected, and patched Active Storage versions will deliberately raise an exception during boot rather than start in that unsecurable state. Where ruby-vips is installed, ensure it is version 2.2.1 or later. If libvips exists in the dependency tree only for image analysis, removing the 'ruby-vips' gem from the Gemfile is a valid alternative to upgrading it.

Apply the Temporary Workaround Only Where Upgrading Must Wait: On libvips 8.13 or later, the unfuzzed operations can be disabled without upgrading Rails by setting the 'VIPS_BLOCK_UNTRUSTED' environment variable, which libvips reads during initialisation. Applications running ruby-vips 2.2.1 or later can instead call 'Vips.block_untrusted(true)' from an initialiser, which is the same protection the official patch invokes when Active Storage starts. For libvips earlier than 8.13, Rails states there is no workaround other than removing the libvips dependency from the application.

Rotate Every Secret the Application Process Could Read: Upgrading closes the vulnerability but does nothing about a secret that was already exfiltrated, so treat every secret readable by the application process as potentially exposed and replace it. That list covers 'secret_key_base', the master key, whether it lives in 'config/master.key' or arrives as 'RAILS_MASTER_KEY', along with everything in 'config/credentials.yml.enc' that it decrypts, Active Storage service credentials for S3, GCS or Azure, database credentials, and tokens or keys for every third-party service the application calls. Rails is explicit that rotation should only be an intermediate step where unavoidable and that an exposed secret must never be retained as a fallback.

Potential MITRE ATT&CK TTPs

Tactic	Technique	Sub-technique
Initial Access	T1190: Exploit Public-Facing Application	
Defense Evasion	T1036: Masquerading	T1036.008: Masquerade File Type
Execution	T1203: Exploitation for Client Execution	
	T1059: Command and Scripting Interpreter	
Discovery	T1083: File and Directory Discovery	
Collection	T1005: Data from Local System	
Credential Access	T1552: Unsecured Credentials	T1552.001: Credentials In Files
	T1606: Forge Web Credentials	T1606.001: Web Cookies

Patch Link

https://github.com/rails/rails/releases/

References

https://github.com/rails/rails/security/advisories/GHSA-xr9x-r78c-5hrm

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

August 07, 2026 • 08:00 AM

More at www.hivepro.com