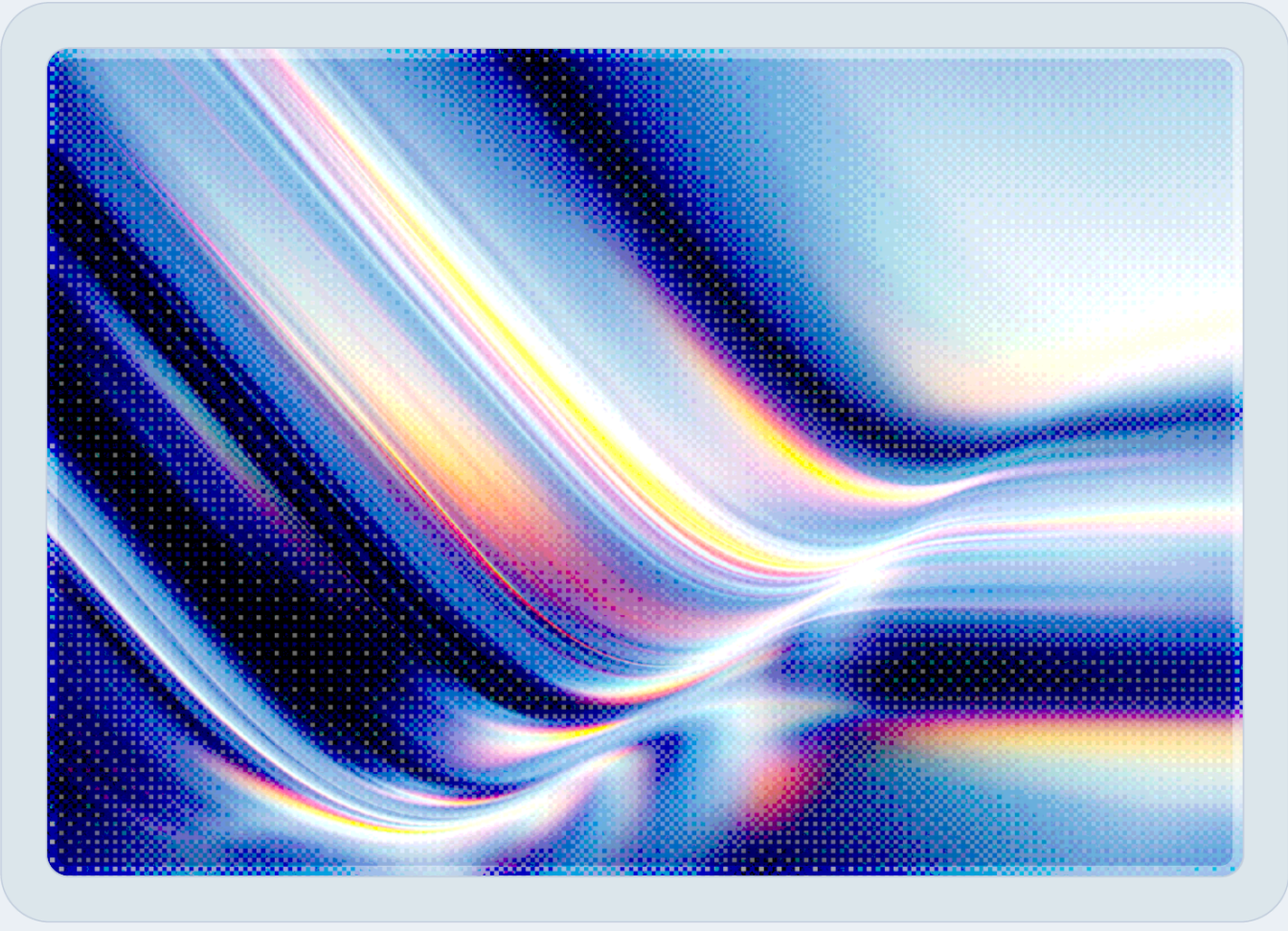
CSAI Foundation | Cloud Security Alliance

Fail2Install and Its Sequel: AI-Found Drupal Installer Takeover

Security Implications and Guidance for CVE-2026-132165 and CVE-2026-136716

2026-11-27

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

© 2026 Cloud Security Alliance. Some rights reserved.

You may download, store, display, view, print, redistribute, and link to this document in its original, unmodified form, provided that attribution to the Cloud Security Alliance is maintained and all trademark and copyright notices remain intact.

This document may not be modified or altered. You may quote portions of the document as permitted by the Fair Use provisions of the United States Copyright Act, provided that attribution is given to the Cloud Security Alliance.

This document may be shared on professional and social media platforms in its original form with attribution.

This document was generated with AI assistance and has not undergone official CSA review and approval processes.

Key Takeaways

Two related flaws in the Drupal core installer let an unauthenticated attacker reinstall a live Drupal site, create their own administrator account, and in some configurations run arbitrary code. Both are being exploited, but only the first is credited to an AI tool. That first flaw, CVE-2026-132165 (SA-CORE-2026-017, publicly named "Fail2Install"), was patched on November 18, 2026, and Drupal credits its discovery to Anthropic's OSS Scanner, an AI-driven service that scans open-source projects [1][7]. Drupal received a report of exploitation on live sites starting at 19:00 UTC on November 19, about a day after the fix shipped [1]. CISA added the flaw to its Known Exploited Vulnerabilities (KEV) catalog on November 24 with a three-day federal deadline that falls today, November 27 [11][12]. One week after the first advisory, Drupal published SA-CORE-2026-019 (CVE-2026-136716), which credits Drupal contributors rather than an AI tool. It states that certain sites in "unusual or incomplete install states" could still be taken over after the first patch, including production sites, and that this second flaw is also being exploited in the wild [2][8].

For practitioners, the most pressing point is that the November 18 releases are not enough. Sites must run Drupal 10.6.23, 11.3.23, or 11.4.14 *and* run the database update hook (`drush updatedb` or `update.php`), because the update hook is what moves existing sites out of the vulnerable state [2][10]. Upgrading also does not remove a compromise: Drupal's own testing found that an attacker's backdoor, a changed hash salt, and an attacker's administrator account all survived both upgrading and resetting user 1 [5]. Any reachable installer should therefore be treated as exploitable, and every site whose `/core/install.php` could be reached from the internet at any time since November 18 should be checked for compromise [5][9]. Sites with longer log retention should also review earlier installer traffic, since pre-disclosure exploitation cannot be ruled out. End-of-life branches have no patch at all, so blocking the installer at the web server is the only protection for Drupal 8.9, 9, and unsupported 10.x/11.x minor releases [1][5].

The case is an early data point on AI-driven discovery, not proof of a new timeline. A flaw reported by an automated AI scanner was independently rediscovered by human researchers, and attackers exploited it within roughly a day of the fix, before any public technical write-up appeared [1][6][9]. Public sources do not show how attackers built their exploit.

Background

Drupal is an open-source content management system used by many government, higher-education, and enterprise websites. Like most PHP applications, it ships a web-based installer at `/core/install.php`. On a new deployment, the installer collects database settings, writes them to `settings.php`, and creates the first administrator account (user 1). On an installed site, the installer is supposed to detect the existing database and refuse to run. Drupal's documentation advises administrators to block or remove the installer once a site is live. In practice many deployments leave it reachable, partly because Composer puts the file back every time it installs or updates core [4].

On November 16, 2026, Drupal announced that core security releases would ship on November 18 [3]. The November 18 advisory, SA-CORE-2026-017, rated the issue "Highly critical" (20/25) on Drupal's own scale. It said that "under specific conditions, an attacker can use the installer to attempt to reinstall Drupal," which "can lead to a site takeover or remote code execution" [1]. The advisory credits the discovery to "Anthropic's OSS Scanner, as vulnerability ANT-2026-GMRPK0BU," along with Drupal Security Team members and other contributors [1]. Anthropic launched OSS Scanner on October 8, 2026 as a free, opt-in service modeled on OSS-Fuzz. Enrolled projects receive periodic scans by Anthropic's models, and the resulting reports are "fully model-generated, without human review or triage," including a reproducer and, where available, a candidate patch [14][15]. SecurityWeek noted that this is the third Drupal vulnerability exploited in 2026 [7].

On November 24, Searchlight Cyber researchers published a technical write-up that gave the flaw the name "Fail2Install." They showed a pre-authentication remote code execution chain affecting Drupal 8.9, 9, 10, and 11 across common Apache and PHP-FPM deployments with MySQL or PostgreSQL; Drupal 7 and Drupal 8 versions before 8.9 are not affected [6]. The researchers state that their finding "ended up being a duplicate of ANT-2026-GMRPK0BU, reported by Anthropic's OSS Scanner" [6]. That same day, CISA added CVE-2026-132165 to the KEV catalog, citing Binding Operational Directive (BOD) 26-04 [11]. The catalog lists a due date of November 27 and says forensic triage is required [12]. This matches the BOD 26-04 guidance that internet-exposed, automatable flaws with total technical impact receive a three-day deadline [13].

On November 25, Drupal published SA-CORE-2026-019 (CVE-2026-136716), rated "Highly critical" (21/25) and classified as an access bypass [2]. This note calls it the "sequel"; the name is ours, not Drupal's. The advisory warns that "it is not apparent in the user interface if a site is in the affected vulnerable state." It also states that "any site that left its installer remotely accessible following SA-CORE-2026-017 may still have been compromised" [2]. Unlike the first advisory, SA-CORE-2026-019 does not credit an AI tool [2]. When this note was written, we did not find CVE-2026-136716 in CISA's KEV catalog, although that may change quickly. The timeline below summarizes the sequence of events.

Date (2026)	Event
October 8	Anthropic launches its opt-in OSS Scanner service [14]
November 16	Drupal pre-announces core security releases [3]
November 18	SA-CORE-2026-017 (CVE-2026-132165) fixed in 10.6.22, 11.3.22, 11.4.13 [1]
November 19, 19:00 UTC	Earliest exploitation on live sites reported to Drupal [1]
November 24	Searchlight Cyber publishes the "Fail2Install" write-up; CISA adds CVE-2026-132165 to KEV [6][11]
November 25	SA-CORE-2026-019 (CVE-2026-136716) fixed in 10.6.23, 11.3.23, 11.4.14; exploitation confirmed [2]
November 27	KEV remediation deadline for federal civilian agencies [12]

Security Analysis

How a Resource Limit Becomes a Reinstall

The root cause of Fail2Install is that the installer decides whether Drupal is already installed in a way that does not tolerate errors. In affected versions, the installer's early checks treat a failed database query as evidence that Drupal is not installed yet. One code comment states the reasoning plainly: "Do not trigger an error if the database query fails, since the database might not be set up yet" [9]. The check that does stop on an error runs later. If an attacker can make the database refuse the early connections and then accept a later one within the same request, the installer shows its setup forms on a production site [6][9]. The attacker can then point the site at a database of their choosing and create an administrator account. Searchlight described the result as allowing an attacker "to completely reinstall the website and create our own administrator account" [6].

The technically novel part is how the attacker makes the database fail on demand without credentials. Searchlight and StepSecurity describe three approaches. On Apache with PostgreSQL, the attacker fills the database's default 100 connection slots, which is fewer than Apache's default worker count. On Apache with MySQL, a `Referer` header containing invalid UTF-8 makes Drupal's database logger

open a second connection, so each request uses up connections twice as fast. On PHP-FPM, the attacker exhausts the server's outgoing TCP ports, which takes about 471 new connections per second against Linux's default 28,232 ephemeral ports and 60-second `TIME_WAIT` window [6][9]. StepSecurity's lab tests showed the effect is probabilistic. During one 20-second flood against Drupal 11.4.12, 12% of installer requests returned the database configuration form, while the patched 11.4.13 held [9]. Searchlight benchmarked remote exploitation on servers with four or more vCPUs [6]. Because exploitation needs only sustained request volume against common default configurations, and success is probabilistic rather than impossible, defenders should treat any reachable installer as exploitable rather than relying on server sizing.

Reaching code execution takes one more step. The installer accepts SQLite as a database driver and lets the user choose the database file path. An attacker can choose a path ending in `.php` inside the web root and put PHP code into the site name, which is written into that file. Drupal's stock `.htaccess` rule allows requests to core scripts by checking how the URL ends rather than checking the full path, so a request such as `/modules/x.php/core/x.php` executes the attacker's file [6][9]. StepSecurity reports that this `.htaccess` behavior is unchanged in the fixed releases. The November 18 patch closes the installer entry point, but the same bypass pattern may be useful in future chains [9]. Drupal notes that some, but not all, exploit paths are mitigated when the web server cannot write to or change permissions on `settings.php` [1].

Why the First Patch Left a Gap

According to StepSecurity's reading of the patch, the November 18 fix changed the installer in four ways. A database error now stops it with "The configured database could not be verified." It refuses to resume when a recorded install step is paired with an install time. Only the browser session that started an install can continue it, enforced with a token. Finally, it no longer changes permissions on `settings.php` [9]. Drupal's November 25 advisory publishes no exploit details, but Hive Security's analysis of the second fix commit suggests that one combination of stored state passed both new guards: a recorded installer task with no recorded install time [10]. Sites can reach that state through interrupted, scripted, or otherwise unusual installs, and nothing in the administrative interface shows it [2].

According to Hive, the second fix adds a further refusal and a post-update hook, `system_post_update_mark_installation_finished()`, which marks the installation finished and sets `install_time` when it is missing. That hook runs only when administrators run `update.php` or `drush updatedb`, not when new code is deployed [5][10]. Organizations that deploy code through CI/CD pipelines without running database updates may therefore believe they are patched while remaining exposed. The table below compares the two advisories.

Attribute	SA-CORE-2026-017	SA-CORE-2026-019
CVE	CVE-2026-132165 ("Fail2Install")	CVE-2026-136716
Published	November 18, 2026	November 25, 2026
Drupal risk score	Highly critical, 20/25 (complex access)	Highly critical, 21/25 (basic access)
Impact	Site takeover or remote code execution	Access bypass leading to site takeover
Fixed releases	10.6.22, 11.3.22, 11.4.13	10.6.23, 11.3.23, 11.4.14, plus update hook
Credited discovery	Anthropic OSS Scanner (with Drupal contributors)	Drupal contributors; no AI tool credited
Exploited in the wild	Yes, from November 19 [1]	Yes [2]
CISA KEV	Added November 24, due November 27 [12]	Not listed as of this writing

Sources: [1][2][10][12].

Exposure and Detection

The exposed population is significant, though it can be measured only roughly. A Shodan survey by StepSecurity on November 26 identified about 55,800 internet-facing Drupal services. Roughly 31,500 were on branches affected by Fail2Install, and about 15,000 of those were on end-of-life Drupal 8 and 9 releases that will receive no fix [9]. StepSecurity describes these as lower bounds, because Drupal's `X-Generator` header reveals only the major version and the CVE cannot be fingerprinted from version strings alone [9]. Hive Security notes that the figures do not measure exposure to CVE-2026-136716, because that flaw depends on a site's stored install state rather than its version [10]. Drupal has not published a victim count [10].

The two flaws leave different forensic traces. Fail2Install exploitation tends to produce bursts of database connection refusals together with installer requests. These include PostgreSQL's `FATAL: sorry, too many clients already`, MySQL's `SQLSTATE[HY000] [1040] Too many connections`, and PHP's `[2002] Cannot assign requested address`, along with spikes of HTTP 500 responses. StepSecurity saw none of these errors in 600 legitimate high-concurrency lab requests [9], which suggests that connection-refusal bursts coinciding with installer requests are a strong signal; organizations should validate it against their own baseline, since ordinary overload can also produce these errors.

Exploitation of CVE-2026-136716, by contrast, does not need that resource exhaustion and may leave no connection-error burst [10]. Drupal's guidance therefore focuses on installer requests and changes to stored state. Indicators include POST requests to `/core/install.php` returning 302 or 303, responses much larger than the usual "already installed" page, changes to user 1's name, email, or password, and unexpected administrators. In stored state and on disk, investigators should look for a non-empty `install_session_token`, a recent `install_time` on an older site (unless it matches the time the database update was run), extra `hash_salt` lines in `settings.php`, and PHP files in the public files directory outside the Twig cache [5].

What the AI Discovery Angle Shows

This incident is a well-documented case of a critical web-platform flaw, credited to an AI scanning service, being exploited soon after disclosure. The details are informative, though they should not be over-read. The disclosure process followed the coordinated path: Anthropic's scanner reported the flaw, Drupal coordinated a fix, and the advisory credited the source [1]. Two observations deserve attention. First, the flaw was found independently by Searchlight's human researchers [6], which shows that more than one party can converge on the same bug once capable tools and researchers examine widely used code. That convergence is the condition underlying the rediscovery risk CSA has described, although in this case the rediscovering party was a defensive research firm, and no evidence shows that attackers found the flaw independently before the patch. Second, exploitation began about a day after the patch and five days before the public write-up [1][9]. The available reporting does not establish how attackers developed their exploit. Analysis of the public patch is one plausible route, and AI assistance in that analysis cannot be ruled out, but neither is confirmed in public sources. In this case, the gap between the public fix and observed exploitation was roughly a day.

The sequel adds a separate lesson. The first fix addressed the conditions it targeted, but it did not cover every stored state a site's installer could be in [2][10]; whether that gap was visible in the original report is not public. Faster discovery does not shorten the work of fully characterizing a flaw's variants, and organizations should plan for follow-on advisories after high-severity fixes.

Recommendations

Immediate Actions

Organizations running Drupal should act today rather than wait for normal patch cycles, because both flaws are exploited and the federal KEV deadline expires today. The priority is to close the entry point and then establish whether it was used before it was closed. The sequence below matters: blocking comes first because it is fast and reversible, and evidence capture must precede the upgrade because the update changes the very values investigators need.

The first step is to block the installer at the web server. On Apache, add the following to the server configuration; because `<Location>` performs a prefix match, it also covers paths such as `/core/install.php/foo`. Environments limited to `.htaccess` files need an equivalent `<Files>`-based rule instead.

```
<Location "/core/install.php">
    Require all denied
</Location>
```

On nginx, use the prefix match `location ^~ /core/install.php { return 404; }`, not an exact match. In either case, confirm that `/core/install.php`, `/install.php`, and `/core/install.php/foo` all return 403 or 404 [4].

Before updating, teams should capture evidence by saving logs, the database, and the file tree, and by recording `drush state:get install_task`, `install_session_token`, `install_time`, and user 1's details, because updating changes these values [5]. They should then upgrade to 10.6.23, 11.3.23, or 11.4.14 and run `drush updatedb`. Afterwards, they should confirm that `install_task` reads `done`. On sites that were in the vulnerable state, the update hook records the update time as `install_time`; any other recent `install_time` on an older site warrants investigation [5][10]. If indicators of compromise are present, the site should be rebuilt rather than repaired: rebuild the code from a trusted source, restore the database from a backup made before the first suspicious request, rotate every secret including database credentials and the hash salt, and end all sessions [5].

Short-Term Mitigations

Over the following weeks, organizations should make the installer's absence a permanent part of their builds rather than a manual change. Drupal recommends both deleting `install.php` through Composer `post-install-cmd` and `post-update-cmd` scripts and blocking it at the web server. It also recommends adding `.htaccess` rules through the `drupal-scaffold` append mechanism so that scaffolding does not overwrite them [4]. Teams should make `settings.php` and the modules and themes directories read-only to the web server user; the official Docker image does not do this by default [9].

Detection and compensating controls complement these build changes. Per-source connection caps reduced exploitation in StepSecurity's tests, though the firm notes that this "raises the bar rather than closing the bug" [9]. Security operations teams should add correlation rules that alert on connection-refusal bursts combined with installer requests, and on any request matching `\.php/core/[^]*\.php` [9]. Sites on end-of-life Drupal 8.9, 9, 10.5.x, or 11.2.x should keep the installer blocked permanently and treat migration to a supported release as urgent [1][5]. According to Drupal, Drupal Steward subscribers are protected against the known attack vectors for both advisories [1][2].

Strategic Considerations

This incident supports several program-level changes. Vulnerability management programs should treat "patched" as a verified state, not a deployment event. Where a vendor fix requires database migrations or state changes, deployment pipelines should run them and check the result. Risk teams should also expect follow-on advisories after high-severity flaws, whether found by humans or AI. A first patch can be correct for the reported path but incomplete for its variants, and faster discovery can increase how often this happens without changing the underlying pattern. Remediation tickets for such flaws should therefore remain open, or be scheduled for re-verification, for a defined period after the fix is deployed.

Organizations that adopt the risk-based model in CISA BOD 26-04 will find this case a clear example of its three-day tier [13]. Internet exposure, full impact, and automatable exploitation were all present, and the timeline left little room for slower processes. Finally, organizations that maintain or depend on open-source projects should consider enrolling in AI-driven scanning services. They should also plan for the triage load and the shorter disclosure-to-exploit windows that such scanning may bring [14][16].

CSA Resource Alignment

CSA's November 2026 white paper *The Disclosure Bottleneck: When AI Discovery Outpaces Patch Consumption* argues that the backlog of known but unpatched flaws is a systemic risk. As capable models are pointed at the same widely used code, the chance that an adversary independently rediscovers a queued flaw rises over time [16]. Fail2Install partly illustrates this concern: an AI-reported flaw was independently rediscovered by human researchers, which shows that more than one party can converge on the same bug. Exploitation in this case began after the public fix, however, and no evidence shows that attackers found the flaw independently beforehand. The paper's emphasis on the stages between a fix and a deployed system matches the sequel's lesson that deploying code without running the update hook leaves sites exposed.

CSA's *AI-Accelerated Vulnerability Discovery and the Patch Debt Crisis* describes how autonomous discovery tools are increasing CVE volume and shortening exploitation windows, creating a structural "patch debt" that conventional patch cadences cannot absorb [17]. The roughly 15,000 internet-facing Drupal 8 and 9 services that will never receive a fix for this flaw [9] are a clear example of that debt. For these sites, compensating controls such as blocking the installer, followed by migration to a supported release, are the only remaining protection.

CSA's research note *CISA's BOD 26-04: A Risk-Based Reset of Patch Rules* explains the directive's risk-based prioritization model and recommends that enterprises adopt it beyond the federal sector [18]. CVE-2026-132165 is a direct example of the directive's three-day tier, and the note provides a framework for building that tier into enterprise service-level agreements. Mapped to the *AI Controls Matrix (AICM) v1.1*, the recommendations in this note fall primarily under the Threat and Vulnerability Management (TVM) domain, including remediation verification and compensating controls for unsupported software. They also fall under the Application and Interface Security (AIS) domain, which covers securing deployment-time components such as installers [19].

References

[1] Drupal Security Team. "Drupal core - Highly critical - Remote code execution - SA-CORE-2026-017." Drupal.org, November 18, 2026 (updated with exploitation notice).

[2] Drupal Security Team. "Drupal core - Highly critical - Access bypass - SA-CORE-2026-019." Drupal.org, November 25, 2026.

[3] Drupal Security Team. "Upcoming critical Drupal core releases on November 18, 2026 - PSA-2026-11-16." Drupal.org, November 16, 2026.

[4] Drupal.org. "Securing the installer (install.php)." Drupal Documentation, updated November 27, 2026.

[5] Drupal.org. "Checking whether your site was compromised through the installer." Drupal Documentation, updated November 26, 2026.

[6] Searchlight Cyber. "Fail2Install: Turning System Limits into Pre-Auth RCE in Drupal." Searchlight Cyber Research, November 24, 2026.

[7] Kovacs, Eduard. "Drupal Flaw Found by Anthropic Scanner Exploited in Attacks." SecurityWeek, November 25, 2026.

[8] Kovacs, Eduard. "Drupal Installer Patch Left a Gap Attackers Are Already Exploiting." SecurityWeek, November 26, 2026.

[9] Shaikh, Owais. "Fail2Install (CVE-2026-132165): Drupal Installer Flaw Lets Attackers Take Over Sites and Run Code." StepSecurity, November 26, 2026.

[10] Hive Security. "Drupal CVE-2026-136716: The Installer Still Answers on Sites That Look Finished." Hive Security Blog, November 26, 2026.

[11] CISA. "CISA Adds Three Known Exploited Vulnerabilities to Catalog." CISA, November 24, 2026.

[12] CISA. "Known Exploited Vulnerabilities Catalog." CISA, accessed November 27, 2026.

[13] CISA. "BOD 26-04: Implementation Guidance for Prioritizing Security Updates Based on Risk." CISA, June 10, 2026 (updated August 25, 2026).

[14] Anthropic Frontier Red Team. "Launching an opt-in vulnerability-finding service for open-source software." Anthropic, October 8, 2026.

[15] Anthropic. "Introducing the Anthropic Cyber Mission." Anthropic, October 8, 2026.

[16] Cloud Security Alliance. "The Disclosure Bottleneck: When AI Discovery Outpaces Patch Consumption." CSA Lab Space, November 6, 2026.

[17] Cloud Security Alliance. "AI-Accelerated Vulnerability Discovery and the Patch Debt Crisis." CSA Lab Space, June 17, 2026.

[18] Cloud Security Alliance. "CISA's BOD 26-04: A Risk-Based Reset of Patch Rules." CSA Lab Space, August 20, 2026.

[19] Cloud Security Alliance. "AI Controls Matrix (AICM) v1.1." Cloud Security Alliance, June 22, 2026.