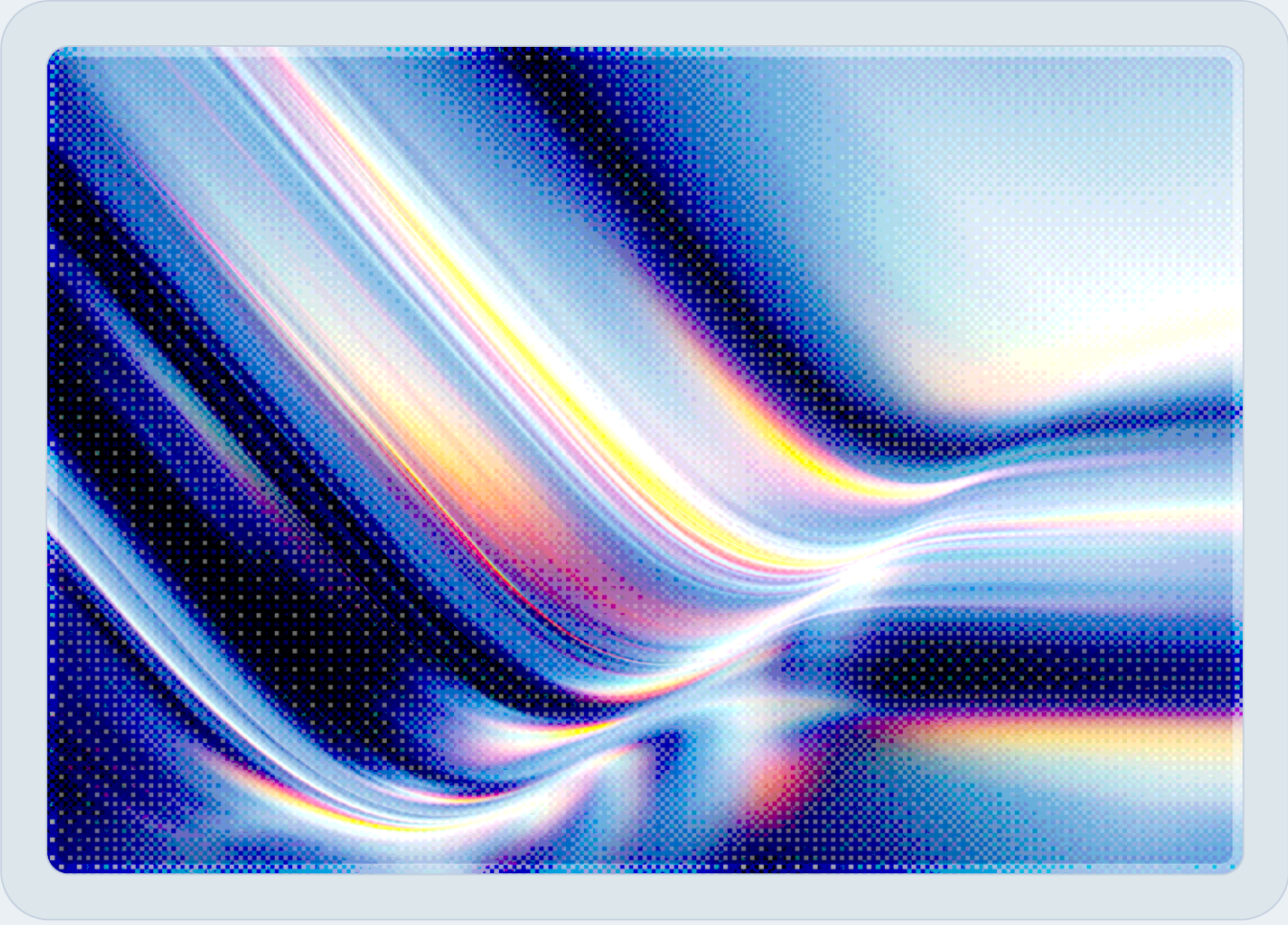
CSAI Foundation | Cloud Security Alliance

Puppet CVE-2026-115731: When the Configuration Server Becomes a Fleet Kill Switch

Report deserialization RCE in KEV, CA key exposure, and the unpatched open-source Puppet tail

2026-11-22

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

CVE-2026-115731 is a critical deserialization flaw in the code Puppet Server uses to process agent run reports. Any client that holds a certificate signed by the Puppet certificate authority (CA) can run arbitrary code on the server as the `puppet` service user [2][5]. The flaw is rated CVSS 4.0 9.4 [3][9]. CISA added it to the Known Exploited Vulnerabilities (KEV) catalog on November 17, 2026, citing "evidence of active exploitation," and federal civilian agencies must remediate by December 1, 2026 under Binding Operational Directive (BOD) 26-04 [5][6]. No attacker, campaign, or payload has been named publicly. Perforce has said it is not aware of attacks against its customers [4], so defenders should treat the KEV listing as authoritative while recognizing that public exploitation detail is thin.

The bug matters because of where it sits more than because of how it works. The `puppet` user can read the CA signing key. The researchers who found the flaw showed that a stolen key can be used to mint a certificate that drives Puppet Enterprise's orchestration broker and runs commands as root on managed nodes [3][7]. A single compromised configuration server can therefore become a fleet-wide execution point. Where the server may have been reached by an attacker while vulnerable, patching alone does not remove the risk. In those cases the CA should be treated as compromised and rebuilt [1][3].

Autosigning, which Neodyme reports is common in large deployments [7], turns this authenticated bug into an unauthenticated attack chain, because an attacker who can reach TCP 8140 can enroll a certificate and then exploit the server [1][7][9]. A companion flaw, CVE-2026-115732, rated CVSS 9.0 by Perforce and Corgea (the GitHub advisory scores it 9.5) [3][10], lets a certificate for `evil.target.example` pass authorization rules written for `target.example`. That enables node impersonation and catalog theft [2][3][7].

Fixes for Puppet Enterprise, Puppet Core, and OpenVox shipped on November 3, 2026 [2][8][11]. The community open-source Puppet releases, which Neodyme reproduced as vulnerable back to version 6.24.0 from July 2021, have no official patch and will not receive one [3][7][8]. Organizations still running them need to migrate, not wait.

© 2026 Cloud Security Alliance. AI-assisted Rapid Research.

Background

Puppet is one of the longest-established configuration management platforms. A central Puppet Server (the "primary") compiles a catalog of desired state for each managed node. The agent on each node pulls that catalog, applies it, and sends back a report describing what changed. Trust between agents and the server comes from a private Puppet CA, usually hosted on the primary itself, which issues the client certificates agents use to authenticate over mutual TLS on port 8140 [1][7]. Perforce, which owns Puppet, distributes the commercial Puppet Enterprise and the Puppet Core builds. OpenVox is a community fork maintained by Vox Pupuli after Perforce changed Puppet's licensing in 2025 [3][7].

Alain Rödel of Neodyme AG found both vulnerabilities while preparing for a red team engagement. The company reported them to Perforce and the OpenVox project in mid-October 2026 [1][3][7]. Neodyme states that it "used AI-based analysis to discover the bugs," though "strict steering of the agents and explicit prompting was necessary" [7]. Perforce, Vox Pupuli, and Neodyme coordinated a release on November 3, 2026 at 17:00 UTC. The OpenVox GitHub advisories followed the same evening, and NVD published both CVEs with Perforce as the CNA [3][7][8]. Exactly two weeks later, CISA added CVE-2026-115731 to KEV [5].

The fixed releases are summarized in the table below. One discrepancy deserves attention. Several news reports list the Puppet Core 8.x fix as 8.22.2 [2][4], while Perforce's own product page names it "Puppet Core 9.1.3 & 8.22.1" [11], and Corgea's analysis agrees with 8.22.1 [3]. Operators should confirm the build against Perforce's release notes. Any 8.x build at or above 8.22.1 that Perforce designates as the security release should be treated as fixed.

Product	Vulnerable	Fixed (November 3, 2026)
Puppet Enterprise	2023.8.0–2023.8.12; 2025.0.0–2025.12.0; 2026.0.0	2023.8.13, 2025.12.1, 2026.0.1 [2][11]
Puppet Core	8.x before the security release; 9.0.0–9.1.2	8.22.1 per Perforce (8.22.2 in some reports), 9.1.3 [3][11]
OpenVox Agent	≤ 8.31.x, ≤ 9.2.x	8.32.0, 9.3.0 [8][9]
OpenVox Server	≤ 8.17.x, ≤ 9.1.x	8.18.0, 9.2.0 [8][9]
OpenVoxDB (CVE-2026-115732)	≤ 8.17.0, ≤ 9.1.0	8.18.0, 9.2.0 [3][8]

Product	Vulnerable	Fixed (November 3, 2026)
Community open-source Puppet	All releases through 8.10.0 (RCE reproduced back to 6.24.0)	No official release; migrate to OpenVox or Puppet Core [3][7][8]

Two further points affect how organizations scope their exposure. Corgea reports that products bundling Puppet Server inherit the exposure, and names Red Hat Satellite among them [3]. Puppet integration is optional in recent Satellite releases, and no downstream vendor advisory has been cited here, so operators of Satellite deployments with Puppet integration enabled should confirm their status with Red Hat. Corgea also notes that GitHub's advisory database holds both CVEs only as unreviewed records with no package attached, so Dependabot and other software composition analysis (SCA) tools that depend on reviewed GitHub advisories may not flag them [3]. Inventories built on SCA alone are likely to undercount affected systems.

Security Analysis

How the report deserialization works

Agents submit run reports with an HTTP PUT to `/puppet/v3/report/<certname>` on the primary [3][7]. When the server rebuilds each event in the report, it calls Puppet's rich-data decoder, `Puppet::Pops::Serialization::FromDataConverter`. That decoder interprets any hash that carries a `__ptype` key as a Puppet type expression [3]. The decoder lets an attacker define an inline `Object` type whose `RubyMethod` annotation holds a method body. Puppet's `RubyGenerator` copies that body unescaped into generated class source and evaluates it with `class_eval` [3][7]. A backtick shell command placed at class-body level therefore runs as soon as the class is defined, without any method ever being called. Corgea traces the root cause to an asymmetry: Puppet wrote events as plain data but read them back as rich data [3]. The OpenVox advisory includes a working proof-of-concept report that executes a shell command on the server [1][9].

The fix converts incoming events with the plain-data `ToStringifiedConverter`, so a `__ptype` key is preserved as inert data rather than interpreted [3]. Corgea notes one side effect: rich-typed values from very old Puppet 5 agents now arrive as hashes rather than objects. Operators with legacy agents should test report processors after upgrading [3]. CISA's catalog entry maps the flaw to CWE-502 (deserialization of untrusted data) and CWE-94 (code injection) [3][6].

From one server to the whole fleet

Code execution as the `puppet` user is serious on its own. The `puppet` user can also read the CA private key, and that changes the scale of the problem [3][9]. Neodyme states that "independent of the configuration, an attacker with RCE on a Puppet server can grab the CA signing keys and impersonate other hosts" [7]. In its Puppet Enterprise proof of concept, Neodyme used the stolen key to mint a certificate with both `serverAuth` and `clientAuth` extended key usages. The researchers then connected to the Puppet Execution Protocol (PXP) websocket broker on port 8142, staged a task, and pushed it to an agent, achieving code execution as root on the managed node [3][7].

How far an attacker can go beyond the server depends on the deployment. PXP is specific to Puppet Enterprise. On OpenVox, follow-on execution depends on whether an orchestration layer such as Choria is configured and whether the `puppet` user can write to code directories or catalogs [3][7]. Neodyme notes that environments hardened to best practice, for example where the catalog and code are owned by a user other than `puppet`, "mitigate direct impact significantly" [7]. Even in hardened environments, however, a stolen CA key undermines the identity of every node the CA has issued a certificate to. Neodyme told The Hacker News that the realistic outcomes include "deployment of ransomware, mass data exfiltration, and potentially lateral movement" [2].

Puppet primaries also concentrate secrets. Catalogs routinely carry database credentials, API tokens, and cloud keys. Hiera eyaml private keys, PuppetDB credentials, and r10k or Code Manager deploy keys typically live on or near the server [3]. Public reporting has not confirmed that attackers are harvesting these specific secrets. Corgea's remediation guidance nonetheless treats all of them as exposed if the server was reachable while vulnerable, and CSA concurs with that conservative position [3].

Autosign and CVE-2026-115732 remove the authentication barrier

Requiring a CA-signed certificate looks like a meaningful barrier, but in practice it is often weaker than it appears. Every managed node already holds such a certificate, including CI runners, build agents, and developer workstations, so compromising any one of them is enough [3]. More importantly, many organizations enable autosigning so that new instances can enroll without manual approval. Neodyme observes that "most large companies have autosign enabled" [7]. The OpenVox maintainers state that in such configurations "this is an unauthenticated attack chain" [1][9]. CISA's catalog entry makes the same point: where the CA signs requests without strong verification, an unauthenticated network attacker can obtain a valid certificate [6].

CVE-2026-115732 compounds the problem. The `trapperkeeper-authorization` library enforces Puppet Server's `auth.conf` rules. When comparing hostnames, it stopped at the shorter name, so an exact-host rule also matched any name with additional leading labels, effectively behaving like `*.target.example` [3][7][10]. Neodyme summarized the effect: "an attacker who got a cert for `evil.target.lab.local` can thus authenticate as `target.lab.local`" [7]. With that access an attacker can fetch another node's catalog and submit facts or reports in its name. Neodyme extracted PostgreSQL credentials and administrative API tokens from a catalog this way [3]. CVE-2026-115732 is not in KEV [3][6]. Vox Pupuli warns that custom `auth.conf` rules relying on the old matching behavior may stop matching after the upgrade [8].

Exploitation status and the patch window

CISA's November 17 alert states the KEV addition was made "based on evidence of active exploitation" [5]. Public detail is otherwise sparse. Security Affairs reports that Perforce "is not aware of attacks" [4]. The Hacker News cites an unverified Reddit post, dated November 4, from an administrator who described tracing a misconfigured CA back to a CVE-2026-115731 intrusion [2]. If accurate, that would place exploitation about one day after disclosure, but CSA has not been able to verify the post. Corgea found no public attack reports and noted that the EPSS score before the KEV listing was 0.00366 [3]. That figure illustrates how probability scores can lag for control-plane bugs whose exploitation is not yet visible in public telemetry, even when proof-of-concept code is available.

This case is consistent with a pattern CSA has described across 2026 [12]. Discovery was AI-assisted, although Neodyme stresses that it required close human steering [7], and a working exploit was published alongside the advisory [7][9]. The short path from disclosure to exploitation is most directly explained by that public proof of concept rather than by how the bug was found. Whether AI-assisted discovery will systematically shorten exploitation timelines is a reasonable expectation, but this single case does not demonstrate it. The bug had been present since at least mid-2021 [7], which does show how long serious flaws can persist in mature infrastructure software.

The federal deadline deserves careful reading. Under BOD 26-04, the KEV due date of December 1 falls 14 days after the November 17 listing, and 28 days after the November 3 fix [6][16]. In CSA's view, even this window is too long for a server that holds the root of trust for an entire fleet, because a working exploit has been public since November 3. Organizations should measure their own remediation targets from public disclosure, not from the KEV listing, and set them well inside the federal deadline.

Recommendations

Immediate Actions

Upgrade every Puppet primary, compiler, and CA server to the fixed release for its product line, then restart the server process, as Perforce instructs [1]. Patch servers first, because the server-side fix closes the remote code execution path [3]. Agents should follow: the OpenVox advisory lists the agent package itself among the affected components [9], and Vox Pupuli's upgrade guidance includes fixed agent versions [8]. Organizations running community open-source Puppet have no official release to install. Vox Pupuli links standalone patch scripts that "might or might not apply cleanly" to older Puppet versions [8], but these are not a substitute for a supported build. Those organizations should restrict network access to their servers immediately and schedule a migration to OpenVox 8.32.0 or later or to Puppet Core [3] [8].

While patching proceeds, disable naive autosigning, meaning `autosign = true` or broad globs in `autosign.conf`. Replace it with policy-based autosigning that validates a challenge secret embedded in the certificate signing request, or with manual signing [3][8]. Restrict TCP 8140, and on Puppet Enterprise TCP 8142, to the network segments that host managed nodes, and never expose them to the internet [3]. Review the list of signed certificates for nodes you do not recognize [1].

Hunt for exploitation before declaring a server clean. Corgea suggests looking for report bodies containing `__ptype` together with `Object[`, `RubyMethod`, or `annotations`, and for report PUTs from recently signed or unknown certnames [3]. Other indicators are child processes of the puppetserver JVM or the `puppet` user such as `sh`, `bash`, `curl`, `wget`, `python`, or `nc`, reads of `ca_key.pem`, and certnames that add an extra leading label to a known node name, which would indicate CVE-2026-115732 [3]. Not every report processor stores raw events, so finding no matches does not prove a server is clean [3].

Short-Term Mitigations

A server should be treated as having a compromised CA if, while vulnerable, it was reachable by untrusted clients or ran with autosign [1][3]. Because any node holding a CA-signed certificate can exploit the flaw [3], the same conclusion applies where port 8140 was broadly reachable, where any managed node is known or suspected to have been compromised, or where hunting results are inconclusive. Start by preserving forensic evidence. Then rebuild the CA, reissue every agent certificate, and revoke the old ones. Rotate Hiera eyaml keys, secrets embedded in catalogs, PuppetDB credentials, Code Manager and

r10k deploy keys, and any cloud credentials the server could reach. Finally, review the control repository and the PXP or orchestrator job history for changes made since November 3 [3]. This work is disruptive, but a stolen CA key otherwise lets an attacker impersonate any node or server indefinitely.

After upgrading, retest custom `auth.conf` rules, because the CVE-2026-115732 fix changes how hostnames match [3][8]. Confirm that catalogs and Puppet code are owned by an account other than `puppet`. Neodyme reports that this hardening significantly limits direct follow-on impact through catalog or code tampering [7]. It does not protect the CA key, however, and on Puppet Enterprise a stolen key still allows execution through the PXP broker, so it complements the CA rebuild above rather than replacing it. Because SCA tools are unlikely to flag these CVEs, inventory Puppet deployments directly, including copies that may be bundled inside other products such as Red Hat Satellite [3].

Strategic Considerations

CSA recommends classifying configuration management servers, orchestration brokers, and their certificate authorities as Tier 0 assets, on par with identity providers and hypervisor management. That classification should bring dedicated administrative paths, network isolation, and patch service levels measured in days. Where the platform allows, move the CA signing key off the primary, either to a separate CA host or to hardware-backed key storage, so that compromising the report-processing service no longer exposes the root of trust. Organizations should also set an internal policy on forks and end-of-life community builds. In this incident, community open-source Puppet installations are the one population with no patch available, and in many organizations such installations sit outside formal asset ownership and vendor support processes.

Finally, the AI-assisted discovery of a bug that had existed since at least 2021 should inform planning. Mature infrastructure codebases are increasingly being re-examined with AI-assisted tooling, and other configuration and orchestration platforms that deserialize agent-supplied data are reasonable candidates for similar findings. Programs that rely on EPSS or vendor exploitation statements to prioritize work should add a rule that control-plane software with a public proof of concept is patched immediately, whatever its probability score.

CSA Resource Alignment

CSA's research note *The Collapsing Patch Window: Project Zero's Full X and AI-Speed Remediation* [12] most directly anticipates this event. It argues that AI-assisted discovery is compressing the time between disclosure and exploitation, and that remediation programs built around 30-day cycles no longer match attacker tempo. CVE-2026-115731 is consistent with that analysis: AI-assisted (though closely human-

steered) discovery of a bug at least five years old, a working exploit published with the advisory, and a KEV listing within two weeks. As discussed above, the speed of exploitation in this case owes more to the public proof of concept than to the discovery method. The recommendation to patch control-plane software with public proof-of-concept code immediately, regardless of score, applies that note's guidance to configuration management. CSA's analysis of *CISA BOD 26-04: AI Threat Forces 3-Day Critical Patch Mandate* [16] explains the directive that governs the federal deadline for this CVE and how its timelines are set.

CSA's analysis of *SCKit and Rogue ScreenConnect Clients: MSP Remote Support as Attack Path* [13] examined the same structural risk in remote-support tooling: one management platform with privileged reach into many endpoints becomes the most valuable target in the estate. Puppet extends that pattern from remote monitoring and management into infrastructure-as-code. The control-plane hardening that note recommends, including network isolation, strict enrollment, and auditing of administrative actions, applies directly to Puppet primaries and orchestration brokers. Similarly, *GitLab CVSS 10 Commits-API Flaw Under CISA Mandate* [14] treated a DevOps platform as a KEV-listed control plane. Its guidance on assuming secrets were exposed and rotating credentials held by build and deployment infrastructure parallels the CA rebuild and secret rotation steps recommended here.

For control mapping, the CSA AI Controls Matrix (AICM) v1.1 [15] provides the framework. Its Threat and Vulnerability Management (TVM) domain covers remediation timelines and prioritization of exploited vulnerabilities. Its Cryptography, Encryption and Key Management (CEK) domain addresses protection and rotation of signing keys such as the Puppet CA key. Its Identity and Access Management (IAM) domain covers the certificate enrollment and authorization weaknesses behind autosign abuse and CVE-2026-115732. Organizations that provision AI training clusters, inference hosts, or agent runtimes through Puppet should note that a compromised configuration server inherits control of those workloads, which brings the incident within the scope of AICM's infrastructure and supply-chain controls.

References

[1] Zorz, Zeljka. "Exploited Puppet flaw puts the server that configures your endpoint fleet at risk (CVE-2026-115731)." Help Net Security, November 19, 2026.

[2] Lakshmanan, Ravie. "Hackers Exploiting Critical Puppet RCE Flaw Affecting Enterprise Management Servers." The Hacker News, November 18, 2026.

[3] Corgea Research. "Puppet CVE-2026-115731 is now in CISA KEV: any agent certificate can run code on the Puppet Server through a crafted report." Corgea, November 17, 2026.

[4] Paganini, Pierluigi. "U.S. CISA adds Perforce Puppet Enterprise flaw to its Known Exploited Vulnerabilities catalog." Security Affairs, November 18, 2026.

[5] CISA. "CISA Adds One Known Exploited Vulnerability to Catalog." Cybersecurity and Infrastructure Security Agency, November 17, 2026.

[6] CISA. "Known Exploited Vulnerabilities Catalog." Cybersecurity and Infrastructure Security Agency, accessed November 22, 2026.

[7] Neodyme AG (Alain Rödel). "Pulling the right strings for RCE on a Puppet." Neodyme, November 3, 2026.

[8] Vox Pupuli. "Critical CVE Updates in all known versions of Puppet and OpenVox." Vox Pupuli, November 3, 2026 (updated November 4, 2026).

[9] OpenVox Project. "Authenticated RCE on OpenVox server via Event Deserialization (GHSA-ffr6-rwpm-p5wq)." GitHub Security Advisory, November 3, 2026.

[10] OpenVox Project. "Flawed Hostname Check allows Puppet Client Impersonation (GHSA-q4hm-33rc-572g)." GitHub Security Advisory, November 3, 2026.

[11] Perforce Puppet. "What's New in Puppet Enterprise and Puppet Core." Puppet by Perforce, November 2026.

[12] Cloud Security Alliance. "The Collapsing Patch Window: Project Zero's Full X and AI-Speed Remediation." CSA Lab Space, November 17, 2026.

[13] Cloud Security Alliance. "SCKit and Rogue ScreenConnect Clients: MSP Remote Support as Attack Path." CSA Lab Space, November 2026.

[14] Cloud Security Alliance. "GitLab CVSS 10 Commits-API Flaw Under CISA Mandate." CSA Lab Space, September 2026.

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

[16] Cloud Security Alliance. "CISA BOD 26-04: AI Threat Forces 3-Day Critical Patch Mandate." Cloud Security Alliance, 2026.