CSAI Foundation

Cloud Security Alliance AI Safety Initiative

Adobe Reader Zero-Day: Four Months Targeting Energy Infrastructure

Unpatched PDF Exploit Conducts Fingerprinting Reconnaissance Against Oil and Gas Sector

Unofficial AI-assisted Research

2026-04-10

Key Takeaways

- An unpatched zero-day vulnerability in Adobe Acrobat Reader has been actively exploited since at least late 2025 — with the earliest known artifact dated November 28, 2025, though multiple sources cite December 2025 as the first confirmed in-the-wild exploitation date — with no CVE assigned and no patch released as of April 10, 2026, leaving users of current Adobe Reader versions at risk pending Adobe's official confirmation of the affected version scope.
- The exploit requires no user interaction beyond opening a malicious PDF file, abusing built-in Acrobat JavaScript APIs (`util.readFileIntoStream` and `RSS.addFeed`) to perform system fingerprinting and exfiltrate local data to attacker-controlled infrastructure.
- Lure documents use Russian-language content depicting gas supply disruption and emergency response scenarios, indicating deliberate targeting of government, energy sector, and critical infrastructure organizations in the Russian-speaking world.
- Two attacker-controlled command-and-control servers (169.40.2.68:45191 and 188.214.34.20:34123) and the domain `ado-read-parser[.]com` are confirmed indicators of compromise; network defenders can additionally detect malicious traffic by filtering HTTP/HTTPS requests bearing the `Adobe Synchronizer` User-Agent string.
- Until Adobe releases a patch, organizations should disable JavaScript execution in Adobe Reader through policy, block identified indicators at the network perimeter, and subject all inbound PDF attachments to gateway-level inspection.

Background

Adobe Acrobat Reader is among the most widely deployed document-viewing applications globally, with a user base spanning virtually every sector of the global economy. Its prevalence in regulated industries — energy, finance, government, and healthcare — makes it a high-value target for adversaries seeking broad initial access or targeted espionage. The PDF format itself has a long history as an attack delivery mechanism: malicious PDFs have been weaponized in state-sponsored campaigns, criminal ransomware operations, and targeted phishing for well over a decade, making the continued exploitation of Reader vulnerabilities an enduring concern for enterprise defenders.

This particular campaign came to light on March 26, 2026, when security researcher Haifei Li — founder of the EXPMON exploit detection sandbox — encountered a suspicious PDF submitted to his platform. EXPMON flagged the file using its "detection in depth" methodology, which identifies behavioral anomalies associated with file-based exploits rather than relying solely on signature-matching. Li subsequently identified a related artifact, titled `Invoice540.pdf`, that had been uploaded to the VirusTotal malware aggregation platform on November 28, 2025, establishing the earliest known artifact associated with this campaign at more than four months before public disclosure — though the upload date alone does not confirm that victim organizations were actively targeted at that time [1][2]. Several sources, including The Hacker News and BleepingComputer, characterize the campaign as beginning in December 2025, likely based on in-the-wild first-seen metadata distinct from the VirusTotal submission timestamp, while others cite November [1][5][6]. The continued presence of campaign-related files in circulation through late March suggests the operation was active across a multi-month window leading up to public disclosure [3].

Adobe was notified of the vulnerability on approximately April 7, 2026. As of April 9, 2026, the company had neither released a patch nor made a public statement acknowledging the vulnerability [4][5]. No CVE identifier has been assigned. This combination of prolonged in-the-wild exploitation and vendor silence represents a significant systemic risk for organizations that depend on Adobe Reader as part of their standard document workflows.

Security Analysis

Technical Mechanism: API Abuse Within the PDF Sandbox

The exploit's technical design is distinctive in that it avoids the memory corruption techniques that have historically characterized high-profile PDF exploits. Rather than triggering a buffer overflow or use-after-free condition, the attack abuses legitimate, privileged JavaScript APIs that are exposed within the Adobe Reader runtime environment. When a victim opens the malicious PDF, heavily obfuscated JavaScript begins executing automatically — no additional user interaction, macro approval, or security prompt is required [3].

The attack proceeds in two stages. In the first stage, the JavaScript executes reconnaissance against the victim's system. It calls `util.readFileIntoStream()`, an Acrobat API that allows the sandboxed Reader process to read arbitrary files accessible on the local filesystem. It then uses `RSS.addFeed()`, a web-update API, to transmit the harvested data to attacker-controlled servers and to receive additional JavaScript payloads in return [6][7]. The information collected during this

fingerprinting phase includes operating system version, language and locale settings, Adobe Reader version number, and local file paths. This reconnaissance capability allows the threat actor to profile each victim system before deciding whether to proceed with more destructive second-stage payloads.

Researcher Haifei Li described the initial exploit as having "the capability to collect and leak various types of information, potentially followed by remote code execution (RCE) and sandbox escape (SBX) exploits" [1]. The conditional nature of the second-stage delivery is operationally significant: by filtering victims based on system fingerprints, the campaign maintains a lower profile and concentrates destructive capabilities only against high-value targets that meet attacker-defined criteria. This selectivity is characteristic of targeted espionage operations rather than commodity malware distribution, though profiling-based delivery has also appeared in high-end ransomware campaigns.

The Sophos Threat Intelligence team confirmed two SHA-256 hashes associated with the campaign: `54077a5b15638e354fa02318623775b7a1cc0e8c21e59bcbab333035369e377f` (Invoice540.pdf) and `65dca34b04416f9a113f09718cbe51e11fd58e7287b7863e37f393ed4d25dde7` (yummy_adobe_exploit_uwu.pdf). Sophos products detect these artifacts under the signatures Troj/PDF-BG and Malware/Callhome [8].

Targeting Profile: Critical Infrastructure in the Energy Sector

The social engineering layer of the campaign provides the clearest signal about intended targets. Both observed malicious PDFs display Russian-language documents as visual decoys. The content depicted — gas supply disruption scenarios and emergency response procedures — is designed to appear credible to recipients working in or adjacent to Russia's energy sector [7][9]. Malware analyst Giuseppe Massaro observed that the thematic content strongly suggests targeting of Russian-speaking organizations in government, energy, and critical infrastructure roles [7]. Independent researcher Gi7w0rm reached similar conclusions based on the lure content and language choice [3].

The targeting of oil and gas infrastructure fits a well-documented pattern in which state-aligned or state-sponsored actors use document-borne exploits to gain initial access to energy organizations, establish persistent reconnaissance footholds, and position themselves for potential disruptive operations. The fingerprinting-first architecture reinforces this assessment: before deploying any destructive capability, the operator confirms the precise nature of the compromised environment, minimizing operational mistakes and maximizing the value of each intrusion.

Attribution to a specific threat group has not been established. The sophistication of the exploit, the deliberate sector and geographic targeting, and the use of a two-stage, profiling-based delivery architecture are all consistent with a capable, patient threat actor — likely state-affiliated — but public

reporting had not produced confirmatory technical attribution as of April 10, 2026. Organizations operating in or adjacent to the Russian energy sector should nonetheless treat this threat as a credible, ongoing espionage campaign.

Command-and-Control Infrastructure and Detection Opportunities

Two distinct IP-and-port combinations have been confirmed as command-and-control endpoints: 169.40.2.68:45191 and 188.214.34.20:34123 [7][8]. The attacker-controlled domain `ado-read-parser[.]com` has also been identified as part of the C2 infrastructure. Network defenders have an additional detection opportunity in the HTTP/HTTPS User-Agent string: malicious traffic from the exploited Reader process uses the string "Adobe Synchronizer," a string that security researchers have not associated with any known legitimate Adobe software update behavior and which can therefore serve as a reliable detection signal [6]. Additionally, analysts monitoring endpoint telemetry should look for instances of `AdobeCollabSync.exe` establishing outbound connections to external IP addresses, which researchers have flagged as abnormal behavior that may indicate active exploitation [7].

The Disclosure Gap as a Systemic Risk Factor

The multi-month window between the earliest known exploitation and public disclosure illustrates a structural problem in the vulnerability disclosure ecosystem. The CVE/NVD system depends on vendor cooperation to assign identifiers, and enterprise patch management processes depend on CVE numbers to prioritize and track remediation. When exploitation precedes disclosure — as it does here — those downstream risk management processes receive no signal and offer no protection against the unpatched vulnerability. Organizations that rely exclusively on CVE assignments and vendor patch advisories — without commercial threat intelligence feeds or researcher-community monitoring — would have had no signal that a zero-day affecting their standard PDF viewer was being actively exploited for reconnaissance against critical infrastructure.

This gap is compounded by Adobe's continued public silence as of the date of this publication. The absence of any vendor acknowledgment or interim guidance leaves security teams without authoritative information about scope, affected versions, or planned remediation timelines. Organizations should not interpret the absence of an official advisory as an indication that risk is low.

Recommendations

Immediate Actions

The most impactful near-term mitigation available to organizations is disabling JavaScript execution within Adobe Reader and Acrobat. Administrators can implement this at the individual workstation level through the application preferences (Edit > Preferences > JavaScript > uncheck "Enable Acrobat JavaScript"). For enterprise-wide enforcement, the following registry key can be deployed via Group Policy or endpoint management platforms on Windows systems [6][7]:

```
HKLM\SOFTWARE\Policies\Adobe\Acrobat Reader\DC\FeatureLockDown
Value: bDisableJavaScript = 1 (DWORD)
```

This configuration blocks the JavaScript execution path that the exploit relies upon. Organizations should be aware that disabling Acrobat JavaScript may affect workflows that depend on JavaScript-enabled PDF forms; a brief assessment of business-critical form dependencies is advisable before broad deployment.

In parallel, organizations should immediately block the confirmed C2 indicators at the network perimeter: 169.40.2.68, 188.214.34.20, and `ado-read-parser[.]com`. Firewall and proxy rules should additionally filter any HTTP/HTTPS traffic whose User-Agent header contains the string "Adobe Synchronizer," as this is a reliable network-level indicator of active exploitation.

Short-Term Mitigations

Email gateway and web proxy controls represent a second line of defense. Organizations should configure gateway-level inspection to scan all inbound PDF attachments in a sandboxed environment before delivery. Solutions that perform behavioral analysis — rather than solely signature matching — are better positioned to detect this exploit given the obfuscated JavaScript and API-abuse technique. Security awareness communications should remind staff that this vulnerability is triggered by simply opening a PDF, including those received as email attachments or accessed via a browser, with no further interaction required.

Endpoint detection and response (EDR) platforms should be evaluated for coverage of the Sophos detection signatures (Troj/PDF-BG, Malware/Callhome) and for process behavior monitoring that would flag `AdobeCollabSync.exe` initiating outbound network connections. Organizations without EDR

visibility at this level of granularity should prioritize the network-based indicators as a compensating control.

As an interim measure for the highest-risk environments, organizations may consider routing PDF viewing through an alternative application. Foxit Reader and other PDF viewers do not share Adobe Reader's attack surface and would not be susceptible to this specific exploit chain, though they carry their own vulnerability histories that require ongoing management.

Strategic Considerations

This incident reinforces the need for organizations to maintain zero-day response plans that do not assume CVE availability or vendor acknowledgment as prerequisites for action. Vendor silence and the absence of a CVE identifier should not gate defensive response when credible public reporting — from researchers and multiple reputable security publications — describes active exploitation. Security teams operating in critical infrastructure sectors should adopt a standing posture of monitoring exploit intelligence sources such as EXPMON, VirusTotal, and vendor threat intelligence feeds as early-warning channels that operate independently of formal disclosure processes.

Longer term, the pervasive deployment of Adobe Reader across enterprise environments warrants inclusion in application rationalization reviews. Where PDF viewing is the sole use case, organizations may find that managed, browser-native PDF rendering offers a reduced attack surface relative to full-featured desktop applications. Any migration of this kind should be assessed against document workflow requirements, including digital signatures, dynamic forms, and accessibility features.

CSA Resource Alignment

This incident connects directly to guidance and frameworks published by the Cloud Security Alliance for organizations managing endpoint security and critical infrastructure risk.

CSA's **Zero Trust Advancement Center** addresses precisely the threat model illustrated here: an initial-access technique that exploits a trusted, ubiquitous client application to fingerprint and profile a target environment before delivering destructive second-stage payloads [10]. The Zero Trust principle of "never trust, always verify" applies not only to network traffic but to the assumption that a fully-patched, current application is free of exploitable vulnerabilities. Zero Trust implementations that segment document processing workflows, restrict lateral movement from endpoint compromise, and verify the integrity of outbound data streams would limit the reconnaissance and exfiltration phases of this attack chain.

The **MAESTRO** framework for agentic AI threat modeling identifies document ingestion pipelines as a high-risk attack surface, particularly when automated processing systems handle PDFs without human review. Organizations deploying AI-assisted document analysis tools that rely on Adobe Reader as a rendering component should review their pipeline architectures for exposure to this exploit; an agent that automatically opens untrusted PDFs for analysis would be a high-value exploitation path.

Within the **AI Controls Matrix (AICM)**, which supersedes the Cloud Controls Matrix as a comprehensive security control reference, applicable control domains include Endpoint Security, Vulnerability and Patch Management, and Threat and Vulnerability Intelligence. The AICM's controls around patch lifecycle governance — specifying maximum acceptable remediation timeframes and requiring compensating controls when vendors are non-responsive — are directly relevant to managing exposure during the current unpatched window.

CSA's **Security Trust Assurance and Risk (STAR)** program provides a framework for organizations to assess and communicate the security posture of their endpoint and application stacks. Organizations using STAR for third-party risk assessments should consider adding Adobe Reader JavaScript policy status as a reportable control indicator while this vulnerability remains unpatched.

References

[1] Haifei Li / EXPMON. "Adobe Reader Zero-Day Exploited via Malicious PDFs Since December 2025." The Hacker News, April 9, 2026.

[2] SecurityWeek. "Adobe Reader Zero-Day Exploited for Months: Researcher." SecurityWeek, April 2026.

[3] The Register. "Months-Old Adobe Reader Zero-Day Uses PDFs to Size Up Targets." The Register, April 9, 2026.

[4] CSO Online. "Hackers Have Been Exploiting an Unpatched Adobe Reader Vulnerability for Months." CSO Online, April 9, 2026.

[5] HackRead. "Adobe Reader Zero-Day Exploited to Steal Data via Malicious PDFs." HackRead, April 9, 2026.

[6] BleepingComputer. "Hackers Exploiting Acrobat Reader Zero-Day Flaw Since December." BleepingComputer, April 9, 2026.

[7] Help Net Security. "Acrobat Reader Zero-Day Exploited in the Wild for Many Months." Help Net Security, April 9, 2026.

[8] Sophos. "Adobe Reader Zero-Day Vulnerability in Active Exploitation." Sophos Threat Intelligence Blog, April 9, 2026.

[9] Security Affairs. "Malicious PDF Reveals Active Adobe Reader Zero-Day in the Wild." Security Affairs, April 9, 2026.

[10] Cloud Security Alliance. "Zero Trust Advancement Center." Cloud Security Alliance, 2023.

