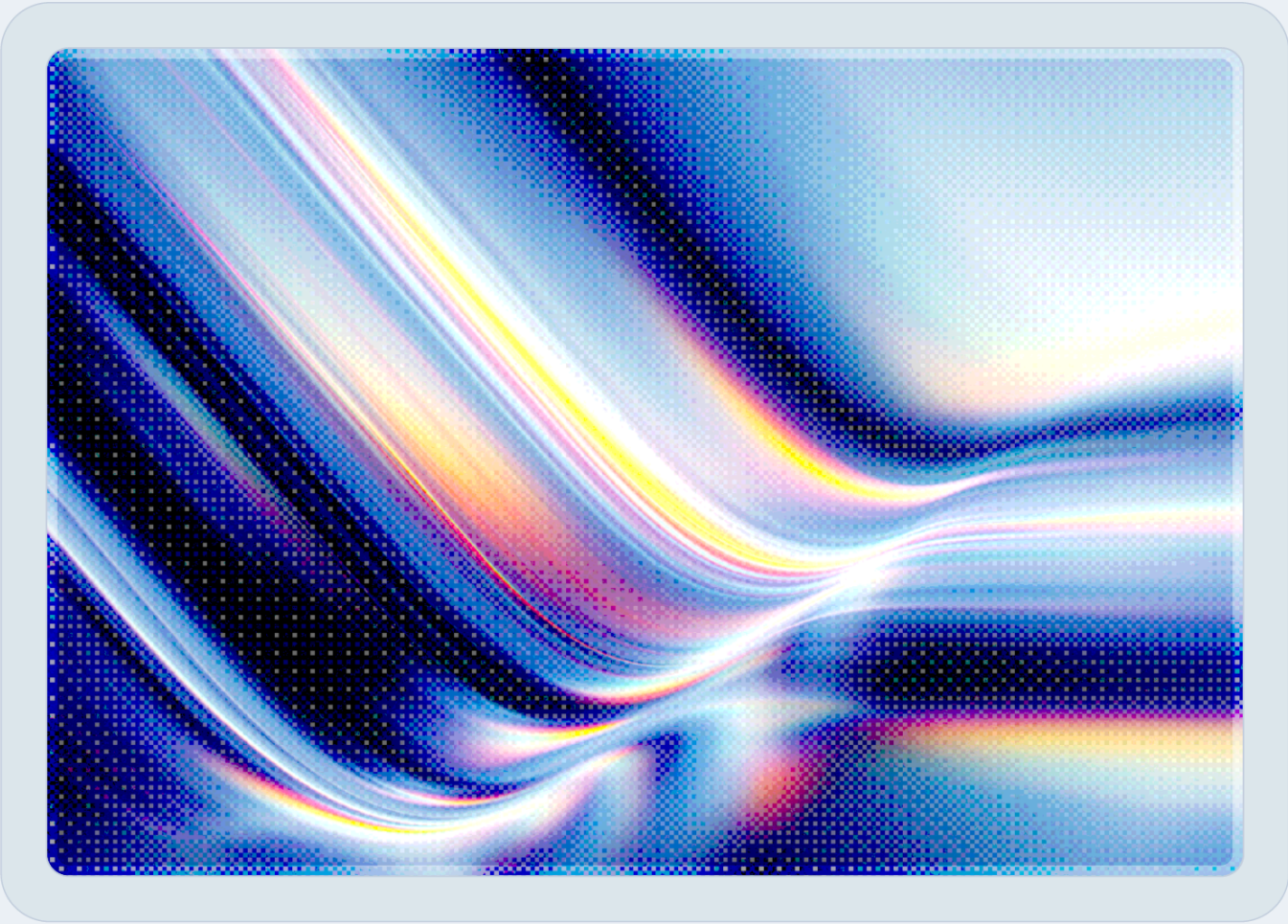
CSAI Foundation | Cloud Security Alliance

ServiceNow Sandbox Escape: Pre-Auth RCE Under Active Attack

2026-07-21

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

Key Takeaways

A critical sandbox-escape vulnerability in the ServiceNow AI Platform, tracked as CVE-2026-6875 and rated CVSS 9.5, is under active exploitation as of the weekend of July 18–19, 2026 [1][2]. The flaw lets an unauthenticated attacker with network access to a vulnerable instance escape ServiceNow's server-side script sandbox and execute arbitrary code, without credentials, phishing, or any prior foothold in the target organization [1]. Threat intelligence firm Defused confirmed that in-the-wild attackers are hitting the same pre-authentication endpoint documented by the vulnerability's discoverer, Searchlight Cyber, but are reaching code execution through a different technical path than the original published proof of concept—a divergence that may indicate independent attacker research rather than simple copy-paste exploitation, though the evidence does not rule out other explanations, such as partial leakage of the original research [1][3].

ServiceNow disclosed the vulnerability publicly on July 13, 2026, alongside patches for both hosted and self-hosted instances; exploitation began within four days of that disclosure, with a broader wave following over the subsequent weekend [2][4]. Because the ServiceNow AI Platform underpins IT service management, HR case management, security operations, and a growing set of agentic AI workflows across a large share of the Fortune 500, a successful compromise can extend well beyond the platform itself to any connected system that trusts it, including on-premises MID Server proxies used to bridge cloud and internal networks [1][3]. Organizations running self-hosted or partner-hosted ServiceNow instances that have not yet applied the July patches should treat remediation as an emergency, same-day priority.

Background

ServiceNow's AI Platform (the current branding for the company's Now Platform) is a cloud-based workflow and service-management environment; the company reported serving approximately 8,700 customers worldwide, including enterprises and government agencies, as of the end of fiscal year 2025 [5][9]. Those customers run IT service management, HR operations, customer service, and, increasingly, agentic AI automations that act on business data on the platform [5]. Administrators and developers extend the platform through server-side JavaScript that runs inside a controlled execution environment

—ServiceNow's script sandbox—intended to prevent both authenticated low-privilege users and, in pre-authentication contexts, anonymous requesters from running arbitrary code against the underlying Java application server [6].

Security researchers at Searchlight Cyber identified a way to defeat that sandbox and reported it to ServiceNow in early April 2026. ServiceNow pushed emergency restrictions to its own cloud-hosted instances within about 24 hours of the report and made patches available to self-hosted customers and partners through June 2026, culminating in public disclosure of CVE-2026-6875 on July 13, 2026 [1][3][4]. Fixed releases span ServiceNow's Brazil, Australia, Zurich, and Yokohama product families: Brazil Early Availability and General Availability, Australia Patch 2, Zurich Patch 7b and Patch 9, and Yokohama Patch 12 Hot Fix 1b and Patch 13 all contain the fix [4]. Any self-hosted or partner-hosted instance running a release earlier than these should be considered vulnerable.

Defused researchers first observed active exploitation attempts on Friday, July 17—four days after the July 13 disclosure—with a broader wave confirmed over the following weekend of July 18–19 [1][2]. This compressed timeline is consistent with what CSA has observed anecdotally across other pre-authentication disclosures in 2026, where opportunistic scanning and exploit development by unrelated threat actors has tended to follow within days of a patch diff becoming available rather than weeks, though a formal comparison across incidents is outside the scope of this note.

Item	Detail
CVE ID	CVE-2026-6875
CVSS score	9.5 (Critical)
Vulnerability class	Pre-authentication code injection / sandbox escape
Discovered by	Searchlight Cyber
Reported to vendor	Early April 2026
Cloud-hosted fix	Deployed within ~24 hours of report (April 2026)
Self-hosted patches available	Through June 2026
Public disclosure	July 13, 2026

Item	Detail
First observed exploitation	July 17–19, 2026
Fixed releases	Brazil EA/GA, Australia Patch 2, Zurich Patch 7b/9, Yokohama Patch 12 HF1b/Patch 13
Exploited endpoint	`/assessment_thanks.do` (pre-auth)

Table 1: CVE-2026-6875 summary as of July 21, 2026.

Security Analysis

How the Sandbox Escape Works

ServiceNow's platform applies two layers of script restriction. A general sandbox constrains all server-side JavaScript execution, blocking dangerous Java class instantiation and filesystem access while still permitting broad, effectively admin-equivalent, query capability against platform data. A second, stricter layer—applied specifically to untrusted input sources such as query filters—further blocks constructs like `eval`, `new Function()`, and inline function definitions, and restricts the powerful global `gs` (GlideSystem) object's methods [3]. This layered design exists precisely because ServiceNow's GlideRecord query API accepts inline JavaScript expressions inside filter values, a long-standing platform feature used for dynamic query conditions such as `sla_due<javascript:gs.daysAgoStart(0)` [3].

CVE-2026-6875 abuses that same GlideRecord filter mechanism from a pre-authentication entry point. The `/assessment_thanks.do` page accepts a `sysparm_assessable_type` parameter that, in the vulnerable versions, is evaluated as a filter expression inside the stricter sandbox context without requiring a login [3]. Searchlight Cyber's published research describes a gadget chain that pivots through ServiceNow's script-include loading mechanism: by redefining object properties via `Object.defineProperty` and then triggering `gs.include()` to load a script library, the attacker forces the platform to evaluate an attacker-controlled payload inside an "unrestricted context" that the include process treats as trusted, rather than the constrained context normally applied to filter

input [3]. The practical effect is that a single crafted, unauthenticated HTTP request can escape both sandbox layers and reach a code-execution primitive equivalent to running arbitrary server-side Java-backed script.

The detail that most concerns defenders is not the original proof of concept itself but what Defused observed in the wild: attackers hitting the identical `/assessment_thanks.do` sink documented by Searchlight Cyber, yet reaching the code-execution primitive via a different gadget chain [1][2]. That divergence suggests at least one threat actor may have independently re-derived a working exploit path rather than reusing published code verbatim—though alternative explanations, such as partial leakage of the original research or convergent-but-distinct discovery, cannot be excluded. If confirmed, this pattern would imply that signature-based detection tied to the original proof of concept's exact payload may miss related attacks.

Impact of Successful Exploitation

Because the sandbox escape grants code execution equivalent to the platform's own trusted script context, a successful attacker can read and modify any data table on the instance, including customer records, HR case data, incident tickets, and any credentials or configuration data stored in ServiceNow itself [1][3]. Searchlight Cyber's own proof-of-concept work demonstrated that this access extends to creating new administrator accounts at will, which would establish durable, credentialed persistence that survives the underlying vulnerability being patched [3]. Where the instance is integrated with an on-premises MID Server—a common architecture used to let cloud ServiceNow reach internal systems such as Active Directory, on-prem databases, or internal APIs—Searchlight Cyber documented a path to run shell commands on those proxy servers, effectively using a SaaS compromise as a pivot into the customer's internal network [1][3].

The unauthenticated nature of the flaw is also a reminder that SaaS platforms do not inherently reduce exposed attack surface relative to self-managed infrastructure. No phishing, no stolen credentials, and no insider access are required; any instance reachable over the network with a vulnerable version is exposed by default. This risk may be compounded by ServiceNow's growing integration of generative AI features—the platform's "AI Platform" branding reflects this shift toward large language model-driven workflows and agents operating over the same data tables this vulnerability exposes—meaning a compromise could also expose whatever data and tool-access those AI-driven automations are configured to use.

The Vendor's Structural Fix

Beyond patching the specific gadget chain, ServiceNow is rolling out a feature called Guarded Script that changes the underlying trust model rather than only closing this one hole. Guarded Script restricts sandboxed code accepted from untrusted contexts, such as filter parameters, to a single simple expression—eliminating variable declarations, control-flow statements, function declarations, assignment operators, and multi-statement payloads entirely [3][4]. This is a meaningfully different mitigation strategy than a targeted patch: it narrows what is expressible in the sandbox's threat model rather than attempting to enumerate and block every possible gadget chain, which should reduce the viability of related-but-distinct escape techniques against the same underlying GlideRecord filter mechanism.

Recommendations

Immediate Actions

Self-hosted and partner-hosted ServiceNow customers should confirm their instance is running Brazil EA/GA, Australia Patch 2, Zurich Patch 7b or later, or Yokohama Patch 12 Hot Fix 1b or later, and apply the appropriate patch immediately if not already current [4]. Given confirmed active exploitation, this should be treated as an emergency change outside normal patch cadence, not scheduled into a routine maintenance window. Instance owners should also confirm whether Guarded Script is enabled per ServiceNow's KB2944435 guidance, since it provides defense-in-depth against exploitation techniques beyond the originally reported gadget chain [3][4].

Security teams should review web server and application logs for requests to `/assessment_thanks.do` with unusual `sysparm_assessable_type` parameter values, particularly any containing `javascript:` prefixes, `Object.defineProperty`, `gs.include`, or `Object.extendsObject` patterns, and treat any match as a likely compromise requiring incident response rather than a blocked-and-forgotten event [3].

Short-Term Mitigations

Instances that were internet-reachable and unpatched at any point between July 13 and the date of patching should be treated as potentially compromised. Incident response teams should audit for newly created administrator accounts, unexpected scheduled jobs or script includes, and anomalous outbound connections from the instance or from any connected MID Server, since attacker persistence

mechanisms of this kind often outlive the underlying vulnerability, based on patterns observed in comparable incidents [1][3]. Organizations with MID Server integrations should specifically review MID Server command execution logs for the affected window, given the documented path from instance compromise to MID Server code execution.

Where immediate patching is not feasible, organizations should work with ServiceNow or their hosting partner to restrict network reachability to the affected pre-authentication endpoints as an interim compensating control, recognizing that this is a stopgap rather than a substitute for patching.

Strategic Considerations

This incident is a reminder that SaaS and PaaS platforms, despite offloading infrastructure management to the vendor, do not eliminate the need for customer-side vulnerability and patch management discipline; self-hosted and partner-hosted instances in particular carry patch-application responsibility that customers must actively track rather than assume is handled automatically. Organizations should build ServiceNow and comparable SaaS platform advisories into the same emergency-patching workflow used for internet-facing infrastructure, rather than treating vendor SaaS security bulletins as lower-priority than on-premises CVEs.

The specific attack path here—exploiting a legitimate, long-standing platform feature (inline JavaScript in query filters) rather than a simple coding bug—also illustrates a broader lesson for platforms that expose scripting or agentic capability to less-trusted contexts. As enterprise platforms add generative AI and agentic automation atop existing scripting engines, the attack surface created by legacy "power user" features designed for authenticated, trusted contexts can become a liability when those same code paths are reachable, even indirectly, from unauthenticated or lower-trust input. Security architecture reviews for AI-enabled SaaS platforms should specifically ask whether any pre-authentication code path can reach an interpreter or scripting engine originally designed only for trusted, authenticated use.

CSA Resource Alignment

This incident connects to several areas of active Cloud Security Alliance research and guidance.

CSA's AI Controls Matrix (AICM v1.1), a 247-control-objective framework spanning 18 security domains, addresses vulnerability and patch management obligations that apply directly here: timely remediation of critical vendor advisories, restriction of pre-authentication attack surface, and vulnerability management responsibilities that persist even when the underlying infrastructure is vendor-hosted [7].

Because CVE-2026-6875 affects an "AI Platform" that increasingly hosts agentic automations operating over sensitive business data, AICM's coverage of AI application and interface security is directly relevant to assessing what a sandbox-escape compromise could expose beyond the traditional ITSM data set.

CSA's Zero Trust Principles and Guidance for Identity and Access Management is also germane, even though this vulnerability bypasses authentication entirely rather than exploiting a weak credential. The guidance's core argument—that access decisions should be based on continuous, risk-based verification rather than a binary trust boundary at the perimeter—applies to how organizations should treat SaaS platforms like ServiceNow going forward: a compromised, "trusted" internal SaaS integration point (in this case, the MID Server bridge to on-premises systems) should not be implicitly trusted by downstream systems simply because it sits inside a previously-authenticated integration channel [8]. Applying that principle would limit the blast radius of exactly the kind of SaaS-to-internal-network pivot this vulnerability enables.

References

[1] Ionut Ilascu. "Critical ServiceNow code execution flaw now exploited in attacks." BleepingComputer, July 20, 2026.

[2] Help Net Security. "ServiceNow pre-auth RCE exploited in the wild (CVE-2026-6875)." Help Net Security, July 20, 2026.

[3] Searchlight Cyber. "Smashing the ServiceNow Sandbox – Pre Authentication RCE." Searchlight Cyber Research Center, 2026.

[4] ServiceNow. "CVE-2026-6875 - Sandbox Escape in ServiceNow AI Platform." ServiceNow Now Support Portal, 2026.

[5] ServiceNow. "ServiceNow AI Platform." ServiceNow Product Page, 2026.

[6] Pierluigi Paganini. "Attackers exploit critical ServiceNow RCE flaw CVE-2026-6875." Security Affairs, July 20, 2026.

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

[8] Cloud Security Alliance. "Zero Trust Principles and Guidance for Identity and Access Management (IAM)." Cloud Security Alliance, 2023.

[9] ServiceNow, Inc. "Form 10-K for Fiscal Year Ended December 31, 2025." U.S. Securities and Exchange Commission, 2026.

