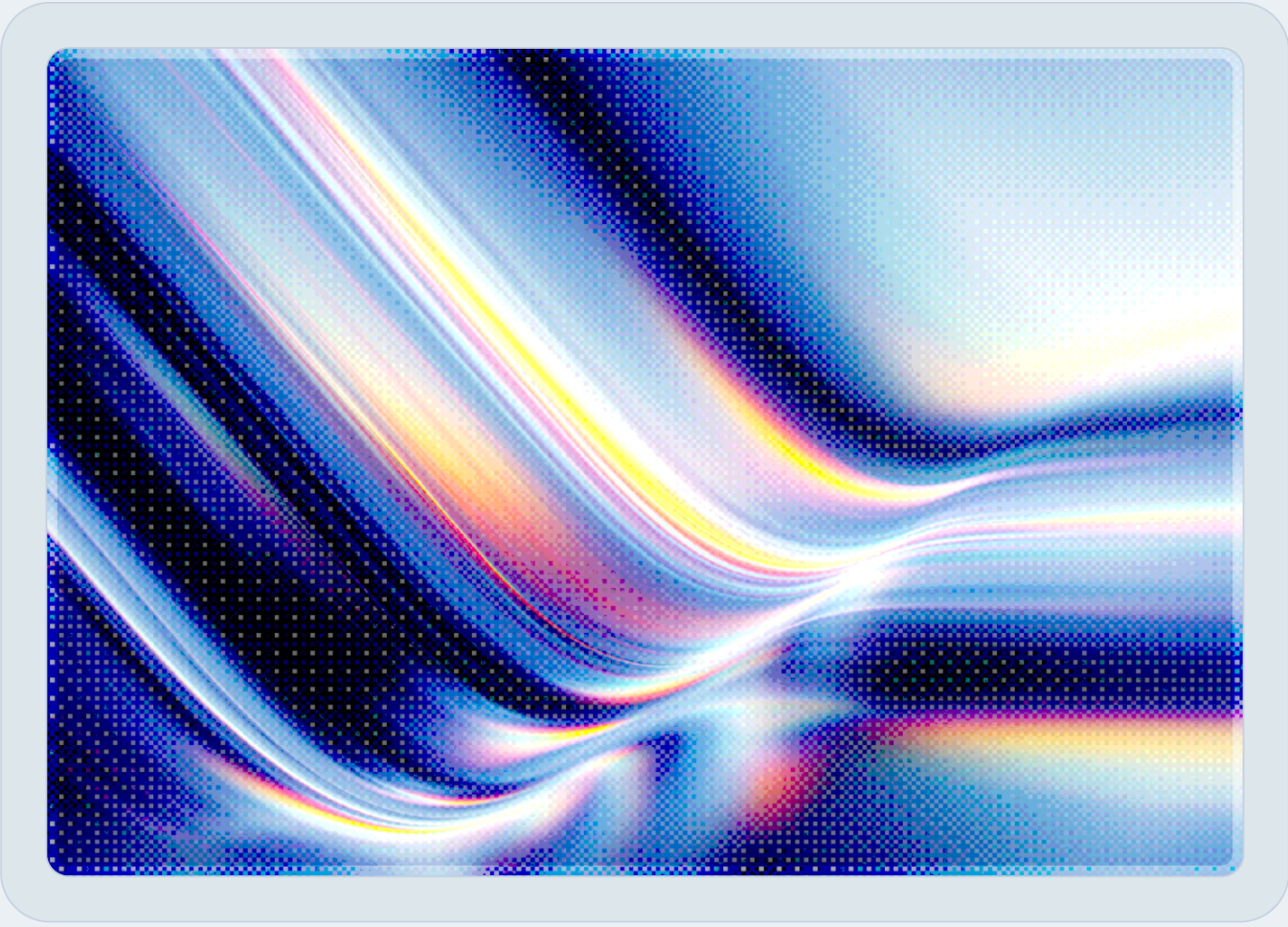
CSAI Foundation | Cloud Security Alliance

Public by Default: Exposed Grok Apps and the AI App-Builder Leak Pattern

Security implications of 35,578 openly reachable Grok-built apps, and guidance for enterprises whose staff use AI app builders

2026-11-11

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

Security firm Claroc tested 37,571 live apps built with Grok's chat-based app builder and found that 35,578 of them (94.7%) opened for anyone without sign-in [1]. Claroc's figures separate two categories of concern. About one in ten open apps, or roughly 3,500, held internal business or personal data, while a narrower set of about 2% held material the researchers rated high or critical, which Claroc extrapolates to 650 to 800 apps [1]. The exposed apps included 24 built by SpaceXSI (formerly xAI) and X staff, among them a confidential advertising proposal to a U.S. federal agency, along with data from 30 other identifiable organizations, including a hospital staff roster of more than 3,500 people and a timesheet app whose script held plaintext passwords [1]. All Grok-specific figures in this note come from Claroc's research and have not been independently verified by CSA.

The exposure did not require attackers to bypass any controls. It came from platform design choices, namely per-app `grok.me` subdomains that passive DNS services record and an access setting preselected to "Anyone with the link" when users published [1][2], and in some cases from flawed generated sign-in logic [1]. In one test app, a Grok-generated sign-in flow added every signed-in user to the team before checking membership, so any Google account could read, edit, and delete records [1]. The lesson for security teams is that generating authentication with AI does not guarantee that the authorization logic is correct.

The incident fits a recurring pattern across AI app builders and AI share features, including Lovable, Base44, Meta Muse, and an earlier Grok issue in which shared chats were indexed by search engines [3][4][5][6]. The pattern is not universal, since at least one major builder already defaults new apps to private [1][2]. Even so, enterprises should treat AI app builders as an unmanaged publishing channel and govern them accordingly.

Background

AI app builders let users describe an application in natural language and receive a working, hosted web app, often with a database, sign-in, and a public address, within minutes. The people using them are often analysts, sales staff, clinicians, and operations managers rather than software engineers. As a result, applications that hold real organizational data can reach the internet without design review, threat

modeling, or any record in the security team's asset inventory. The speed is useful to the business. It also means that the platform's defaults, rather than a security engineer, make the important decisions about who can reach the data.

According to reporting by The Hacker News, the chat-based app builder in Grok, the AI assistant operated by SpaceXSI (the company formerly known as xAI), was opened to all Grok plans on August 19, 2026 [2]. The feature shares a name with Grok Build, SpaceXSI's separate terminal-based coding agent, which was reported in July 2026 to upload users' repositories [2]. This note concerns the hosted apps produced by the chat-based builder, which are commonly described as "Grok Build apps" in coverage of this incident; the coding agent is not the subject here. Each published app is served from its own hostname of the form `<name>.grok.me`, and Grok proposes four random words as the default name [1].

Claroc's researchers discovered the issue on November 8, 2026, and published their findings on November 10 under the name "GhostShares 2" [1]. The name continues their October 2026 work on Meta's Muse agent, in which more than 6,800 share hostnames were found in a commercial passive DNS dataset [4]; Meta paused sharing before reissuing shared items at new addresses [2]. In the Grok study, the researchers listed 43,601 `grok.me` names using common subdomain discovery techniques. They then loaded the first page of each live app once, without signing in, and used an AI model followed by manual review of a random sample to identify pages holding private data [1]. They did not submit forms, call server functions, or use any credentials they observed, and all write and delete testing was limited to two apps the researchers built themselves [1].

Readers should weigh these findings with their provenance in mind. All Grok-specific figures in this note are as reported by Claroc, a commercial security firm publishing a named research series. CSA has not independently verified them, and The Hacker News coverage [2] relies on Claroc's findings rather than corroborating them independently. Because sensitivity was classified by an AI model and extrapolated from a manually reviewed sample, the data-exposure estimates should be read as approximate.

Security Analysis

Scale and Nature of the Exposure

The headline numbers are large, but most exposed apps were harmless. The Hacker News notes that most open apps were games, landing pages, and calculators [2]. The risk sits in the minority that held internal data, and Claroc's figures distinguish between two tiers within that minority. The broader tier, about one in ten open apps (roughly 3,500), held some internal business or personal data. The narrower tier, about 2%

of open apps, held material rated high or critical, and Claroc extrapolates this to 650 to 800 apps by calibrating its AI classifier against a manually reviewed random sample [1]. The table below summarizes the measured figures.

Measure	Value	Source
`grok.me` hostnames discovered	43,601	[1]
Apps still online when tested	37,571	[1]
Apps open without sign-in	35,578 (94.7%)	[1]
Apps returning HTTP 401	1,993 (5.3%)	[1]
Open apps with internal or personal data	About 1 in 10 (roughly 3,500)	[1]
Open apps rated high or critical	About 2%	[1]
Estimated apps with high/critical private data (extrapolated)	650–800	[1]
SpaceXSI and X staff apps exposed	24	[1]
Other identifiable organizations affected	30, incl. 6 Fortune 500	[1]

The SpaceXSI and X staff apps show that the exposure reached the vendor's own workforce. They included a confidential onboarding plan for the enterprise sales team, a territory map covering 594 strategic accounts, eight X advertising proposals (one to a U.S. federal agency and marked confidential), live advertiser account boards, and copies of an X brand safety playbook [1]. Among other organizations, the researchers found a hospital group's staff roster of more than 3,500 employees and an ICU briefing with one patient's labs, imaging, and ventilator status. They also found a fire department's shift board with names and badge numbers, a trucking firm's recruiting list of more than 1,000 people with personal email addresses, and a contractor's timesheet app whose client-side script contained 18 accounts with plaintext passwords, administrator accounts included [1]. Claroc's figures show that 26 of the 30 organizations have more than 1,000 employees, and 12 are based outside the United States [1]. Data of these kinds could plausibly bring regulatory obligations under health privacy, employment, and data protection law, depending on jurisdiction.

Root Cause One: Hostnames Are Not Secrets

The first cause concerns how apps are addressed. Many share features rely on an unguessable link: if only the recipient knows the URL, the content is treated as effectively private. That model holds up better when the secret part sits in the URL path, because DNS resolution never sees the path. Path-based links still leak through logs, referrers, link previews, and crawlers, as the 2025 Grok shared-chat indexing showed [3], so they reduce but do not eliminate discoverability. The model breaks down further when the secret is the hostname. Every visit to `<name>.grok.me` causes a DNS lookup, and passive DNS sensors operated by commercial and community services log those lookups in searchable databases [1]. Claroc notes that the four-random-word naming scheme makes names hard to guess, but guessing is unnecessary once the names appear in passive DNS datasets, many of which offer pattern queries to registered or paying users [1][4].

This is the same mechanism Claroc documented a month earlier in Meta Muse, which suggests it is a class of design flaw rather than a single vendor's mistake [2][4]. Claroc's report notes that ChatGPT's app builder also uses per-app subdomains (in a two-level `<app-name>.<username>.chatgpt.site` format), so those names can also appear in passive DNS. ChatGPT apps, however, default to "Only me," which keeps a discovered hostname from becoming a data exposure [1][2]. In other words, a discoverable address is a problem only when the access control behind it is open. Security teams should therefore assume that any per-app subdomain on a builder platform is public, whatever the vendor says about link secrecy.

Root Cause Two: The Default Access Setting

The second cause is the access default, and the vendor's response centers on it. SpaceXSI told reporters that Grok apps stay private until their creators share them, that publishing involves several steps ending in a legal disclaimer, and that employee accounts default to "Only you" [1][2]. According to Claroc, SpaceXSI also agreed that "Only you" should be the default for users, while saying that this change alone would not have protected the staff apps [1]. Claroc in turn agreed that nothing goes public before the user clicks Publish. However, it found that the access panel shown at that point offered "Anyone with the link," "Invited users," and "Only you," with "Anyone with the link" preselected [1]. When the researchers changed the setting to "Only you," strangers received a 401 error, and unpublishing took an app offline within seconds [1]. The controls existed and worked; they were simply not the default.

The researchers argue that extra publishing steps do not protect data if the final step is preset to everyone [1]. That argument matches long-standing secure-by-default principles. A user building an internal team dashboard has no particular reason to understand passive DNS, and users tend to treat a preselected option as the platform's recommendation, a well-documented default effect. The fact that

24 apps built by SpaceXSI's own staff were exposed, despite the stated employee default, suggests that default settings alone did not prevent exposure even inside the vendor [1]. It is not clear, however, whether the employee default was in place when these apps were published, or whether staff used personal accounts.

Root Cause Three: Generated Authorization Logic

For security teams, the third cause is arguably the most consequential, because it affects apps whose owners believed they had added access control. In Claroc's second test app, a customer list for a team, Grok automatically added "Continue with Google" and "Continue with X" sign-in buttons. The generated code added each signed-in user to the team before checking team membership, so the membership check always passed [1]. An uninvited outside Google account could therefore view, edit, and delete every record [1][2]. Claroc's first test app, a contact list with a database and no sign-in at all, let an anonymous visitor add a row that later visitors could see [1].

This failure mode has appeared before on other platforms. In 2025, CVE-2025-48757 described insufficient Row-Level Security policies in Lovable-generated applications that allowed unauthenticated remote attackers to read or write arbitrary database tables; the CVE record notes that Lovable disputes it, on the grounds that customers are responsible for protecting their apps' data [6]. Researchers reported more than 170 affected applications [7]. Also in 2025, Wiz Research found that Base44 exposed registration and verification endpoints that let anyone join a supposedly private enterprise application using only its publicly visible app identifier, bypassing SSO; Wix fixed the flaw within 24 hours of disclosure [5]. In a broader passive scan of more than 5,600 publicly available vibe-coded applications, Escape reported more than 2,000 vulnerabilities, more than 400 exposed secrets, and 175 instances of personal data including medical records [8]. These studies used different methods and are not directly comparable. Still, the consistent finding is that the visible presence of a sign-in screen in an AI-built app says little about whether authorization is enforced.

The Recurring Pattern

The Grok incident is also the second major exposure tied to Grok's sharing features. In August 2025, Fortune reported that more than 370,000 shared Grok conversations had been indexed by search engines after the share feature published unique URLs that were open to crawlers, apparently without users knowing [3]. The table below places the recent incidents side by side.

Incident	Exposure mechanism	Reported scale	Vendor response
Grok shared chats (Aug 2025)	Share URLs indexed by search engines	370,000+ conversations [3]	Not stated in cited source
Lovable (2025)	Missing or weak database RLS in generated apps	170+ apps [7]	Lovable disputed the CVE [6]
Base44 (Jul 2025)	Unrestricted registration endpoints bypassing SSO	Platform-wide [5]	Fixed within 24 hours [5]
Meta Muse (Oct 2026)	Share hostnames visible in passive DNS	6,800+ hostnames [4]	Sharing paused; new addresses issued [2]
Grok apps (Nov 2026)	Per-app subdomains in passive DNS plus open default access	35,578 open apps [1]	24 staff apps removed within ~2 days; third-party apps not yet addressed [1][2]

In most of these cases, the platform made the publishing decision easy and fast, while the access decision was either preset to open or delegated to generated code that the app's creator was unlikely to review. Base44 differs in that the flaw sat in the platform's own endpoints rather than in a default or in generated code. Not every vendor follows this pattern either: ChatGPT's builder defaults to "Only me" [1][2]. Where the pattern does hold, the affected data sits outside the organization's identity provider, data loss prevention tooling, and asset inventory.

The vendor response to the Grok findings should be read against a short disclosure window. Claroc reported the issue to SpaceXSI on November 8 and the additional staff-app findings on November 9, then published on November 10 [1]. Within that window, SpaceXSI removed the 24 staff apps but not other users' apps, and its public statement did not address notifying those owners [1][2]. Removing customers' content unilaterally is a non-trivial decision for any platform, and the vendor's response may change after this note's publication. Claroc had emailed three affected organizations directly and withheld all hostnames and app titles from its publication [1]. Until the vendor's position becomes clearer, organizations whose employees use Grok's app builder should not assume that anyone else will tell them about their own exposed apps.

Recommendations

Immediate Actions

Security teams should first find out whether their organization has apps on `grok.me` and comparable builder domains. Claroc recommends searching passive DNS for builder domains and testing results from an incognito browser that is not signed in [1]. Because a pattern search on a builder domain returns every customer's apps, teams should test only those results that can be attributed to their own organization, for example by app title, embedded corporate email domains, or internal survey responses, and should limit testing to an unauthenticated page load. Where a passive DNS service is unavailable, a short internal survey and a review of web proxy and DNS logs for builder domains can identify which staff are publishing.

Any app found to hold internal data should be switched to "Invited users" or "Only you," or unpublished, and teams should confirm the change by checking that an outside visitor receives an error page [1][2]. Security teams usually cannot unpublish an app they do not own. Apps with no identifiable owner should therefore be reported to the platform vendor for takedown, and, where an employee account can be identified, unpublished through that account. Teams should also inspect the scripts loaded by any exposed app for embedded names, email addresses, API keys, and passwords, since Claroc's timesheet example shows that generated client-side code can contain credentials directly [1]. Any credential found this way should be treated as compromised and rotated, because it was readable by anyone while the app was public and may have been captured by crawlers or scanners.

Short-Term Mitigations

Within the next few weeks, organizations should identify which AI app builders employees use. Claroc's checklist includes Grok, Lovable, Replit, v0, Bolt, Base44, and Google AI Studio [1][2]. With that inventory in hand, organizations should decide which builders are sanctioned for which data classes and configure sanctioned platforms, where tenant controls permit, so that the default share setting is restricted. Users should receive brief, specific guidance that per-app web addresses are discoverable and that "Anyone with the link" means anyone.

Access testing deserves particular attention. For any app that implements sign-in, owners should test access with an outside account that is not on the invite list before relying on it, because generated membership checks may not do what they appear to do [1]. On the network side, proxy or secure web gateway rules can flag uploads of sensitive file types to builder domains, so that citizen-built apps holding regulated data surface for review.

Strategic Considerations

Over the longer term, AI app builders should be governed as a publishing channel and a SaaS dependency, not as a personal productivity tool. Vendor assessments for these platforms should ask about default access settings, organization-scoped sharing (limited to the company's email domain), audit logs of published apps, and administrative takedown capability. Claroc's four proposed fixes for Grok offer a useful baseline for buyers: serve shared apps on unguessable paths under a single host so links never become DNS names; add an organization-only sharing option; make "Only you" the default and show who can open an app before it goes live; and warn at publish time when an app has a database but no sign-in [1].

Organizations should also bring citizen-built apps into existing asset management and identity governance, so that each published app has an accountable owner, a data classification, and a review date. AI providers that ship sharing features should carry secure-by-default responsibilities comparable to those that cloud storage providers eventually accepted for public buckets. The repetition of this pattern across vendors since 2025 suggests that market pressure alone has not yet produced consistent secure defaults across vendors, although some, such as ChatGPT's builder, already default to private [1][3][4][5].

CSA Resource Alignment

CSA's **AI Controls Matrix (AICM) v1.1** provides the control vocabulary for this incident [9]. Its Data Security and Privacy Lifecycle Management (DSP) domain covers data classification and the handling of sensitive data in AI-produced artifacts. Its Identity and Access Management (IAM) domain addresses the access-default and generated-authorization failures described above, while the Application and Interface Security (AIS) domain applies to the generated code, including the membership-check flaw. Organizations can use AICM to frame vendor questionnaires for AI app builders and to set internal requirements for when such tools may hold regulated data.

CSA's **SaaS Security Capability Framework (SSCF)** defines customer-facing security capabilities that SaaS providers should make configurable, across domains including Identity and Access Management, Data Security and Privacy Lifecycle Management, and Logging and Monitoring [10]. AI app builders are SaaS platforms whose users publish content. The SSCF's expectations about admin-configurable sharing controls, audit logging, and tenant-level enforcement correspond closely to the gaps this incident exposed, especially the absence of an organization-scoped sharing option and of central administrative visibility into published apps.

CSA's **Data Security within AI Environments** maps AICM data security controls across the AI lifecycle and proposes additional controls, including one for shadow AI detection [11]. That proposal applies directly here, because the Grok apps holding hospital rosters and ICU briefings are shadow AI deployments that likely hold regulated health data. CSA's **Shadow Access and AI** examines how unmanaged access paths emerge outside identity governance and how AI can help detect them [12]. A published app at a discoverable hostname with open or broken access control is an example of shadow access, and the detection and remediation practices in that paper apply to inventorying and closing these apps. Finally, the CSA Labs research note **The Vibe Coding Governance Gap** addresses citizen developers deploying apps on vibe-coding platforms that default to public access, along with the visibility and inventory measures needed to close that gap [13].

References

[1] Aprozper, Asaf (Claroc Research). "GhostShares 2: Grok Apps Exposed xAI's Business Secrets and People's Private Data." Claroc Blog, November 10, 2026.

[2] Khandelwal, Swati. "Over 35,000 Grok-Built Apps Exposed Online, Some Leaking Confidential Data and Passwords." The Hacker News, November 11, 2026.

[3] Nolan, Beatrice. "Thousands of private user conversations with Elon Musk's Grok AI chatbot have been exposed on Google Search." Fortune, August 22, 2025.

[4] Ofonagoro, Joseph Chisom. "One DNS Query Revealed 6,800 Meta Muse Chat Links." eSecurity Planet, October 21, 2026.

[5] Nagli, Gal (Wiz Research). "Critical Vulnerability in AI Vibe Coding Platform Base44." Wiz Blog, July 29, 2025.

[6] National Institute of Standards and Technology. "CVE-2025-48757 Detail." National Vulnerability Database, May 30, 2025.

[7] Superblocks. "Lovable Vulnerability Explained: How 170+ Apps Were Exposed." Superblocks Blog, March 2026.

[8] Escape. "Methodology: How We Discovered Over 2k High-Impact Vulnerabilities in Apps Built with Vibe Coding Platforms." Escape Blog, October 2025.

[9] Cloud Security Alliance. "AI Controls Matrix (AICM) v1.1." Cloud Security Alliance, 2025.

[10] Cloud Security Alliance. "SaaS Security Capability Framework (SSCF) v1.0.1." Cloud Security Alliance.

[11] Cloud Security Alliance. "Data Security within AI Environments." Cloud Security Alliance, December 2025.

[12] Cloud Security Alliance. "Shadow Access and AI." Cloud Security Alliance, March 2025.

[13] Cloud Security Alliance. "The Vibe Coding Governance Gap." CSA Labs, June 2026.

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