INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-05-09 18:46 UTC

Mass Data Exposure via AI 'Vibe Coding' Platforms Affects Thousands of Apps

SECURITY ANALYSIS | HIGH | CVSS 7.5

SCC Item ID	SCC-STY-2026-0119
Type	Security Analysis
Severity	HIGH
CVSS Base Score	7.5
Affected Products	Applications built with AI vibe-coding platforms including Lovable, Base44, Replit, and Netlify; estimated 380,000 publicly accessible apps, ~5,000 confirmed sensitive data exposure
Published	2026-05-07
Discovery Source	Gemini

Executive Summary

RedAccess researchers identified approximately 380,000 publicly accessible applications built on AI-assisted 'vibe coding' platforms, including Lovable, Base44, Replit, and Netlify, with roughly 5,000 confirmed to expose sensitive corporate and personal data. The root cause is insecure default configurations that publish apps openly unless developers explicitly change privacy settings, a design assumption that fails non-technical builders who lack security hygiene awareness. This exposure signals a systemic risk emerging from the democratization of app development: as AI lowers the barrier to building software, it simultaneously widens the attack surface for data harvesting at scale.

Technical Analysis

RedAccess researchers scanned the public internet for applications originating from AI-assisted development platforms and identified approximately 380,000 publicly reachable assets. Of those, roughly 5,000 were confirmed to expose sensitive information, including internal credentials, business logic, and personal data, with no authentication or access controls applied. The root cause maps to three overlapping weaknesses: insecure default visibility settings (CWE-1188), information exposure through accessible resources (CWE-200), and incorrect default file and directory permissions (CWE-276). From an adversarial standpoint, the exposure pattern aligns with MITRE ATT&CK T1083 (File and Directory Discovery), T1530 (Data from Cloud Storage), and T1213 (Data from Information Repositories), representing low-effort, high-yield reconnaissance and collection opportunities requiring no exploit code, just an HTTP request. The structural problem is design intent mismatched with user capability: platforms optimized for frictionless publishing default to public visibility because that serves the primary use case of sharing. Non-technical users building internal tools, customer portals, or

data dashboards with AI assistance have no baseline expectation that 'deploy' means 'publicly visible to anyone.' Security hygiene assumptions embedded in traditional software development workflows, such as private-by-default repositories, environment variable separation, and access control reviews, do not transfer to this user cohort. The finding reflects a broader pattern the security community has observed in low-code and no-code environments: the easier a platform makes building, the harder it becomes to enforce the security controls that professional developers apply through discipline and tooling. CISA's Secure by Design principles directly address this dynamic, calling on vendors to shift default configurations toward safety rather than convenience. Prior to public disclosure by RedAccess (May 2026), none of the named platforms appear to have implemented private-by-default configurations; vendor responses are ongoing and should be tracked per Step 5 action item. The Wired report (Tier 2 source) provides the most substantive coverage of the RedAccess findings and is the recommended primary reference for technical teams.

Action Checklist

1. Step 1: Assess exposure, audit your organization for any use of Lovable, Base44, Replit, Netlify, or comparable AI-assisted development platforms; include shadow IT and departmental tools built outside IT governance
2. Step 2: Review controls, for any identified apps, immediately verify access control settings, confirm private-by-default configurations are enforced, and validate that no credentials or internal data are embedded in app code or assets
3. Step 3: Update threat model, add insecure-default misconfiguration via AI development platforms as a threat vector in your risk register; treat shadow app development as an emerging attack surface category distinct from traditional SaaS sprawl
4. Step 4: Communicate findings, brief leadership on the risk that business units may have deployed internally sensitive tools on public-facing platforms without IT involvement; frame as a governance gap, not individual fault
5. Step 5: Monitor developments, track RedAccess follow-up disclosures, vendor responses from Lovable, Base44, Replit, and Netlify regarding default configuration changes, and any regulatory guidance from CISA on AI-assisted development platforms

IR / Forensic Enrichment

Triage Priority	URGENT
Escalation Criteria	Escalate immediately to legal and privacy counsel if Step 2 secrets scanning or unauthenticated access testing confirms that any publicly exposed app contained PII, PHI, authentication credentials to internal systems, or internal API keys — any of these conditions may trigger mandatory breach notification timelines under GDPR (72 hours), HIPAA, or applicable state privacy statutes, and exceed IR team authority to resolve without legal involvement.

Recovery Notes	After toggling all identified apps to private and rotating any confirmed exposed credentials (API keys, database connection strings, service account tokens found via truffleHog), verify that the platform's access control change took effect by performing a fresh unauthenticated HTTP GET from an external IP to confirm a 401/403 response or authentication redirect. Establish a 90-day enhanced monitoring period during which the crt.sh and Shodan queries from Step 1 are re-run weekly to detect any newly deployed shadow apps by the same business units. Before restoring any app to production use, require the owning team to complete a lightweight security review checklist confirming no hardcoded secrets, access controls enforced, and IT governance notification completed.
Forensic Artifacts	Platform access logs from Lovable, Base44, Replit, or Netlify admin consoles showing app creation dates, visibility setting history (public vs. private toggle events), and collaborator access lists — request via platform support or data export before any remediation that might overwrite audit trails | Exported app source code and asset bundles containing hardcoded secrets, including .env files, JavaScript bundle files, and any configuration JSON files embedded in the app — these are the primary evidence of what data was exposed and must be preserved before the app is modified or deleted | Web server or CDN access logs from the platform showing unauthenticated GET requests to the app's public URL, particularly requests from unfamiliar IP ranges or automated scanner user-agents, which would indicate whether the exposed app was discovered and accessed by external parties prior to containment | IdP/SSO audit logs (Okta System Log, Azure AD Sign-In Logs, Google Workspace Admin Reports) showing OAuth authorization grants to Lovable, Base44, Replit, or Netlify — these reveal which user accounts connected organizational identity to the platform and what scopes were granted, establishing potential lateral access paths | DNS and certificate transparency records from crt.sh showing the full history of subdomains issued for *.netlify.app, *.repl.co, *.lovable.app, and *.base44.app containing your organization's name — provides a timeline of when apps were created and surfaced publicly, which may predate IT awareness by months or years

Per-Action IR Details

Step 1: Assess exposure — audit your organization for any use of Lovable, Base44, Replit, Netlify, or comparable AI-assisted development platforms; include shadow IT and departmental tools built outside IT governance

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection & Analysis: identifying scope of adverse events and correlating information across sources

Controls: NIST IR-4 (Incident Handling), NIST IR-5 (Incident Monitoring), NIST SI-4 (System Monitoring), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 2.1 (Establish and Maintain a Software Inventory)

Compensating: Run DNS and certificate transparency queries using crt.sh and Shodan (free tier) filtering on your organization's domain names and known subdomains to surface publicly reachable apps on Netlify (*.netlify.app), Replit (*.repl.co), Lovable (*.lovable.app), and Base44 (*.base44.app). Supplement with a grep across your SSO/IdP access logs and expense management system for OAuth grants or SaaS billing entries referencing those platform domains. Export results to a spreadsheet — no SIEM required.

Evidence: Before triaging further, capture: (1) full DNS enumeration results for your org's domains against *.netlify.app, *.repl.co, *.lovable.app, *.base44.app subdomains; (2) browser history and bookmarks on shared/developer workstations for sessions to those platforms; (3) OAuth token grants in your IdP (Okta, Azure AD, Google Workspace) showing third-party app authorizations to those platforms; (4) any expense reports or SaaS procurement records referencing those vendors — these establish which business units own the apps before evidence is altered.

Step 2: Review controls — for any identified apps, immediately verify access control settings, confirm private-by-default configurations are enforced, and validate that no credentials or internal data are embedded in app code or assets

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy: stopping additional damage and preserving evidence while limiting exposure

Controls: NIST IR-4 (Incident Handling), NIST AC-3 (Access Enforcement), NIST SI-2 (Flaw Remediation), NIST SI-7 (Software, Firmware, and Information Integrity), CIS 3.3 (Configure Data Access Control Lists), CIS 4.7 (Manage Default Accounts on Enterprise Assets and Software), CIS 6.3 (Require MFA for Externally-Exposed Applications)

Compensating: For each identified app, use the platform's native visibility settings to immediately toggle to private or password-protected access (Netlify: Site Settings > Access Control > Password protection; Replit: Repl settings > Private toggle; Lovable and Base44: Project visibility settings). Simultaneously, clone or export the app source code and run `grep -rEi '(api[_-]?key|secret|password|token|bearer|aws_|db_pass|connectionstring)' ./app-source/` against the exported codebase to detect hardcoded credentials. Use truffleHog (free, open source) for a more thorough secrets scan: `trufflehog filesystem ./app-source/`.

Evidence: Before modifying any access settings, capture screenshots with timestamps of the current public/private toggle state for each identified app — this documents the insecure-default condition as found. Export or archive the full app source code, including any environment variable configurations or `.env` files visible in the platform IDE. Record the public URL and perform a single unauthenticated HTTP GET to confirm what data a public visitor could access — log the full response headers and body. This pre-containment snapshot is your evidence of exposure scope.

Step 3: Update threat model — add insecure-default misconfiguration via AI development platforms as a threat vector in your risk register; treat shadow app development as an emerging attack surface category distinct from traditional SaaS sprawl

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: using lessons learned to update policies, procedures, and threat awareness

Controls: NIST IR-8 (Incident Response Plan), NIST RA-3 (Risk Assessment), NIST SI-5 (Security Alerts, Advisories, and Directives), NIST CA-7 (Continuous Monitoring), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: Add a standing agenda item to your monthly risk register review to query crt.sh for new certificates issued to *.netlify.app, *.repl.co, *.lovable.app, and *.base44.app subdomains containing your organization's name or known product names. Create a free Google Alert for '[your org name] site:netlify.app OR site:repl.co OR site:lovable.app' to surface newly indexed public apps. Document this as a named threat scenario — 'AI Vibe Coding Shadow App Exposure' — in your risk register with likelihood and impact ratings grounded in the RedAccess finding of ~1.3% confirmed exposure rate (5,000 of 380,000 apps).

Evidence: Preserve the RedAccess research report (publication date, researcher names, methodology summary) as a named threat intelligence source in your risk register entry. Document which specific platforms (Lovable, Base44, Replit, Netlify) were confirmed in your environment during Step 1 — this scopes the residual risk. Retain the Step 2 evidence package (screenshots, secrets scan output) as supporting evidence for the risk register entry's impact rating.

Step 4: Communicate findings — brief leadership on the risk that business units may have deployed internally sensitive tools on public-facing platforms without IT involvement; frame as a governance gap, not individual fault

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: reporting findings, communicating lessons learned, and updating organizational awareness

Controls: NIST IR-6 (Incident Reporting), NIST IR-8 (Incident Response Plan), NIST AU-6 (Audit Record Review, Analysis, and Reporting), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Prepare a one-page executive brief that leads with the RedAccess statistic (380,000 publicly accessible AI-built apps, ~5,000 with confirmed sensitive data exposure) to establish external context before disclosing internal findings. Quantify your organization's specific exposure by listing the number of apps found, the platforms involved (Lovable, Base44, Replit, Netlify), and the data types potentially exposed — not hypothetical categories, but the actual credential types or data fields identified in Step 2's secrets scan. Frame remediation as a configuration governance action (enforcing private-by-default), not a personnel disciplinary matter, citing the insecure-default design assumption by the platforms as the root cause.

Evidence: Compile the complete evidence package from Steps 1 and 2 before the leadership brief: the asset discovery results showing which apps exist, the pre-containment screenshots showing public exposure state, and the secrets scan output showing what data was at risk. If any app exposed PII, PHI, or credentials to internal systems, flag this explicitly in the brief — these conditions may trigger breach notification obligations under GDPR, HIPAA, or state privacy laws and require legal review before the brief is delivered. Worth noting this touches regulatory/legal interpretation — verify with your legal counsel whether discovered PII or PHI exposure triggers mandatory notification requirements before finalizing the brief.

Step 5: Monitor developments — track RedAccess follow-up disclosures, vendor responses from Lovable, Base44, Replit, and Netlify regarding default configuration changes, and any regulatory guidance from CISA on AI-assisted development platforms

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: integrating threat intelligence and external disclosures into ongoing improvement

Controls: NIST SI-5 (Security Alerts, Advisories, and Directives), NIST IR-5 (Incident Monitoring), NIST CA-7 (Continuous Monitoring), NIST DE.AE-07 (Cyber threat intelligence integrated into adverse event analysis), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Subscribe to CISA's Known Exploited Vulnerabilities (KEV) feed via RSS (https://www.cisa.gov/sites/default/files/feeds/known_exploited_vulnerabilities.json) and set a filtered alert for any entry referencing Netlify, Replit, Lovable, or Base44. Monitor the RedAccess blog and their GitHub (redaccess-research) directly for follow-up disclosure posts. Set platform-specific changelog monitoring: Netlify (https://www.netlify.com/changelog/), Replit (https://blog.replit.com), and track their GitHub release pages for commits touching access control or default privacy settings. Create a recurring 30-day calendar reminder to re-run the crt.sh and Shodan queries from Step 1 to detect newly deployed shadow apps.

Evidence: Maintain a dated intelligence log documenting each vendor's response to the RedAccess disclosure — specifically whether Lovable, Base44, Replit, or Netlify change their default app visibility from public to private, and when. If a vendor announces a default configuration change, re-audit all apps built on that platform after the change date to verify the new default was applied retroactively or only to new deployments — this distinction determines whether previously exposed apps remain at risk. Archive the original RedAccess report URL and publication date as the threat intelligence anchor for this monitoring program.

Detection Guidance

Security teams should prioritize three detection angles. First, inventory discovery: query DNS, CASB telemetry, and browser proxy logs for traffic to known AI development platform domains (lovable.dev, base44.com, replit.com, netlify.app) originating from corporate devices or identities. This surfaces shadow apps before external researchers find them. Second, credential and secrets scanning: if any apps are identified, scan their public-facing assets for embedded API keys, database connection strings, internal hostnames, or service account tokens using tools aligned with OWASP secret detection guidance. Third, access log anomalies: for apps already deployed, review platform-provided access logs for unexpected external IP ranges, high-volume directory traversal patterns consistent with T1083, or bulk data requests consistent with T1530 and T1213. From a policy audit perspective, review your software development lifecycle and procurement policies for coverage of

AI-assisted development tools; most current policies do not address this category. CISA's Secure by Design guidance provides a framework for evaluating vendor default configurations as part of vendor risk assessments.

Indicators of Compromise

Type	Value	Context	Confidence
URL	`Pending — refer to RedAccess research disclosure and the Wired report for any published asset indicators`	RedAccess identified approximately 5,000 specific exposed application URLs and assets; individual IOC values were not reproduced in the available source material	LOW

Framework Mappings

MITRE-ATTACK

- **T1083** — File and Directory Discovery
- **T1530** — Data from Cloud Storage
- **T1213** — Data from Information Repositories

OWASP-TOP10-2021

- **A01:2021** — Broken Access Control

NIST-800-53R5

- **AC-3** — Access Enforcement
- **SC-28** — Protection of Information at Rest

HIPAA-SECURITY

- **164.312(a)(1)** — Access Control
- **164.312(d)** — Person or Entity Authentication

CIS-V8

- **6.3** — Require MFA for Externally-Exposed Applications

SOC2-TSC

- **CC6.1** — Logical access security software, infrastructure, and architectures

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1083	File and Directory Discovery	Discovery
T1530	Data from Cloud Storage	Collection

Technique ID	Technique Name	Tactic
T1213	Data from Information Repositories	Collection

Sources

Source	URL	Tier
gemini	https://securityboulevard.com/2026/05/thousands-of-vibe-coded-apps-...	T3
Thousands of Vibe-Coded Apps Expose Corporate and ... - WIRED	https://www.wired.com/story/thousands-of-vibe-coded-apps-expose-cor...	T2
Companies like Lovable, Base44, Replit, and Netlify use AI to let ...	https://www.facebook.com/wired/posts/companies-like-lovable-base44-...	T3
Researchers say more than 5,000 apps built with AI coding tools like ...	https://www.reddit.com/r/vibecoding/comments/1t6dzfa/researchers_sa...	T3
AI vibe-coding apps leak sensitive data - Axios	https://www.axios.com/2026/05/07/loveable-replit-vibe-coding-privacy	T3

Generated 2026-05-09 18:46 UTC by TJS Security Command Center