Credential & Supply-Chain Exposure Self-Assessment Checklist

A practical audit for reducing the risks highlighted by recent supply-chain breach claims (source code, cloud credentials, and CI/CD secrets) — regardless of whether the claim is confirmed.

Large-scale breach claims involving stolen source code and cloud credentials have become one of the most common patterns in corporate intrusions — whether the claim is fully verified or not. This checklist walks through the exposure points attackers most often target: leaked secrets, unmonitored repository access, over-privileged cloud credentials, and slow incident response. Use it as a recurring internal audit, not a one-time exercise.

1 Secrets & Credential Hygiene

- [ ] **No plaintext secrets in code or configs** — API keys, PATs, and service account credentials live in a secrets manager (Azure Key Vault, AWS Secrets Manager, HashiCorp Vault, etc.), not committed to source or pasted into CI/CD variables.
- [ ] **Every secret has a named owner** and a documented rotation/expiration date — no indefinite-lifetime tokens.
- [ ] **Git history has been scanned for historical secret exposure**, not just the current branch (tools: gitleaks, trufflehog, GitHub secret scanning).
- [ ] **SSH keys are per-user and per-purpose** — no shared or service-wide keys reused across systems.
- [ ] **An emergency rotation runbook exists** and has been tested in the last 12 months, covering PATs, storage keys, SSH keys, and RSA certificates.

2 Source Code & Repository Security

- [ ] **Repository access is reviewed on a fixed schedule** (quarterly minimum) and former employees/contractors are removed promptly.
- [ ] **Branch protection and required code review** are enabled on all production and deployment-related repositories.
- [ ] **Automated secret scanning runs on every commit and pull request**, not just on a periodic schedule.
- [ ] **Repositories containing build, deploy, or infrastructure configs are never publicly accessible**, even temporarily during setup or testing.
- [ ] **Audit logging is enabled for repository clone and download events**, with alerting on bulk or off-hours activity.

3 Cloud & Identity Access Management

- [ ] **Personal Access Tokens (PATs) use least-privilege scopes** — never org-wide admin access for routine automation.

- [ ] **Cloud storage access keys are short-lived or use managed identities** instead of long-lived static keys wherever the platform supports it.
- [ ] **Multi-factor authentication is enforced** on all developer, admin, and DevOps portal accounts — no exceptions for service or shared accounts.
- [ ] **Conditional access or IP allowlisting is configured** on cloud DevOps and source control portals.
- [ ] **Standing privileged access is reviewed quarterly** — a documented answer to “who can currently touch production” exists and is current.

4 Detection & Monitoring

- [ ] **Alerts are configured for anomalous repository clone volume** or access outside normal working hours/locations.
- [ ] **Logs are retained long enough to support forensic investigation** — 90 days minimum for access and authentication logs.
- [ ] **Dark web / breach-forum monitoring is in place** for your company name, domains, and key executive names.
- [ ] **A defined triage process exists for unverified breach claims** — see the response framework below — so the first 24 hours aren't improvised.

5 Incident Response Readiness

- [ ] **The written IR plan explicitly covers data theft and extortion scenarios**, not only ransomware/encryption events.
- [ ] **Legal, communications, and technical response roles are assigned in advance**, with current contact information on file.
- [ ] **Customer and client notification thresholds and templates are pre-drafted** so disclosure decisions aren't made from scratch under pressure.
- [ ] **Key vendors and supply-chain partners have a designated security contact** for receiving and validating third-party disclosure.

Responding to an Unverified Breach Claim: 4 Steps

1. Separate confirmed facts from attacker claims. Note publicly what your organization can verify (e.g., “we identified and remediated an isolated incident”) without validating or denying the attacker's specific figures until forensics confirm them.

2. Treat any referenced credential types as compromised until proven otherwise. If a claim references PATs, SSH keys, or storage access keys, begin rotation for the systems in scope immediately — rotation cost is low; delay cost is not.

3. Check the actor's track record before reacting to the headline number. Data brokers have repeatedly inflated claimed record counts and dataset sizes to increase sale credibility. A documented history of exaggeration should lower — not eliminate — the urgency of your response, and should shape how confidently you state the scope publicly.

4. Watch for a data sample leak. A published sample is the clearest signal that a claim has moved from unverified to substantiated, and should trigger escalation to full incident response.

Using this checklist: Score one point per checked item (24 total). 20–24: strong posture. 14–19: workable, but close the gaps above within one quarter. Below 14: treat credential and repository hardening as an immediate priority, not a roadmap item.