Mass Exfiltration of Cloud Source-Code Repositories

CAPA is tracking a sustained, global campaign in which credential-stealing malware harvests the credentials and session tokens of high-privilege developer and DevOps accounts, then uses that access to automatically clone and exfiltrate their cloud source-code repositories (hosted on services such as GitHub and GitLab). Thousands of repositories, including documentation, PII, secrets, production scripts and operational data, have been exfiltrated. Activity has been observed in Australia and internationally, with clear relevance to energy and critical-infrastructure operators whose IT repositories can provide a pathway to OT.

THREAT TYPE	PRIMARY OBJECTIVE	ATTRIBUTION
Source-code & secret exfiltration Automated bulk repository theft	**Data exfiltration** Secondary: onward credential theft	**Affiliate network** Shared creds & tooling, not one actor
INITIAL ACCESS	**AFFECTED**	**GEOGRAPHY**
Credential stealer on developer endpoint Privileged / universal-read accounts	**Cloud source repositories** GitHub, GitLab and similar	**Global, incl. Australia** Automated, indiscriminate targeting

OVERVIEW

What CAPA is observing

The campaign is opportunistic and highly automated. Rather than targeting one organisation, affiliates share a marketplace of stolen credentials and reusable tooling. A credential stealer, delivered via phishing, malicious packages, cracked software and browser-resident theft, captures the credentials, personal-access tokens and **live session cookies** of developers, preferentially those of **DevOps and platform engineers who hold broad or universal read access**. Because a valid session cookie is often included, multi-factor authentication is bypassed and the attacker resumes an already-authenticated session. Automated tooling then enumerates and clones every visible repository in bulk within minutes. Exfiltration is the primary goal; a secondary, self-reinforcing objective follows, as the cloned code is machine-scanned for further secrets that are recirculated to seed the next intrusion, the same “steal-then-scan-then-exfil” pattern behind the most damaging supply-chain events of the past year.

TECHNIQUES & ATTACK PATTERN

How the attack unfolds

1. **Credential theft.** A credential stealer on a developer/DevOps endpoint harvests stored credentials, PATs, OAuth grants and active session cookies. Delivery via phishing, trojanised packages, or unmanaged devices.
2. **Authenticated access, no MFA.** Stolen PATs, OAuth grants and SSH keys authenticate git and API reads directly, with no MFA challenge; a stolen cookie likewise resumes a web session. MFA does not gate token-based read or clone.
3. **Bulk enumeration & clone.** Automated tooling lists all visible repositories and clones/archives them en masse in a compressed time window. Volumetric and machine-speed; the key detection signal.
4. **Exfiltration.** Repositories (code, docs, PII, production scripts and operational data) are exfiltrated to attacker-controlled storage or public dump repositories. Primary objective achieved.
5. **Secondary credential harvesting.** Exfiltrated content is scanned for embedded secrets, which are recirculated to affiliates to seed further intrusions. Self-reinforcing loop.

T1078 Valid Accounts · T1539 Steal Web Session Cookie · T1555 Credentials from Password Stores · T1213 Data from Information Repositories · T1119 Automated Collection · T1552 Unsecured Credentials · T1567 Exfiltration over Web Service

WHY IT MATTERS FOR ELECTRICITY UTILITIES

The IT-to-OT pathway

For an electricity utility the danger is rarely the exfiltrated data alone; it is what that data enables. Repositories routinely hold exactly the information an adversary needs to map a route from IT into operational technology: network and single-line diagrams, VPN, jump-host and vendor remote-access configurations, engineering-workstation and control-system integration code, and hard-coded or embedded credentials. With this, an attacker can reuse legitimate remote-access paths, abuse trusted IT-to-OT connections, and impersonate engineering or vendor access to pivot from a corporate breach toward generation, substation and control-centre systems. The strategic risk is that a routine IT theft becomes the reconnaissance package for an OT intrusion.

RECOMMENDED MITIGATIVE ACTIONS

What to do now

PRIORITY 1

Enforce least privilege to contain blast radius

- Eliminate organisation-wide read; scope repository access to teams on a need-to-know basis, and retire standing high-privilege “god” DevOps accounts in favour of just-in-time elevation.
- Replace broad personal-access tokens and OAuth grants with **fine-grained, repo-scoped, short-lived** tokens (cap lifetime ≤90 days).
- Enforce SSO with **phishing-resistant MFA (FIDO2 / passkeys)** to blunt the credential phishing that seeds these intrusions; MFA does not protect stolen tokens or cookies, so pair it with the short-lived, revocable tokens above.

PRIORITY 2

Monitor for mass-clone and bulk-read events

- Stream your cloud source-repository audit logs (clone, repo-download and contents-API events) into your SIEM.
- Alert on **volumetric anomalies**: many repositories cloned by one identity in a short period, clone rates far above baseline, or access from a new device, ASN or geography.
- Treat API-based bulk reads and archive downloads as clone-equivalent; do not monitor `git clone` alone.

PRIORITY 3

Gate package consumption through sanitised, delayed sources

- Route all npm / PyPI / Maven / NuGet / container pulls through an **internal proxy** (e.g. Artifactory / Nexus), with no direct developer access to public registries.
- Impose a **cooldown of ~1 week** before newly published versions become installable, so malicious releases are withdrawn before they reach you.
- Scan proxied artefacts for malware and secrets, maintain an allowlist, and disable install-time scripts (`--ignore-scripts`; block pre/post-install hooks).

PRIORITY 4

Restrict platform access to managed network paths

- Require access to cloud source-repository platforms via **corporate VPN or ZTNA**, and enable platform **IP allow-listing** where available.
- Enforce managed-device / device-posture checks; block access from unmanaged endpoints where credential stealers reside.
- Review and restrict third-party OAuth apps and integrations; revoke unsanctioned or dormant grants.

IF YOU SUSPECT COMPROMISE

Revoke all tokens and sessions for the affected identity, force org-wide re-authentication, and rotate every secret in the cloned repositories (assume they are burned). Review the audit log for the full clone scope and notify under your SOCI / privacy obligations where thresholds are met.

C\

ABOUT THIS ADVISORY

CAPA Intelligence is an Australian cybersecurity company serving the global electricity and renewables sector and its critical-infrastructure supply chains. This advisory is not a substitute for tailored professional advice.

Questions or incident support: info@capaintelligence.com

CAPA Intelligence · Threat Advisory

capaintelligence.com