Chainguard

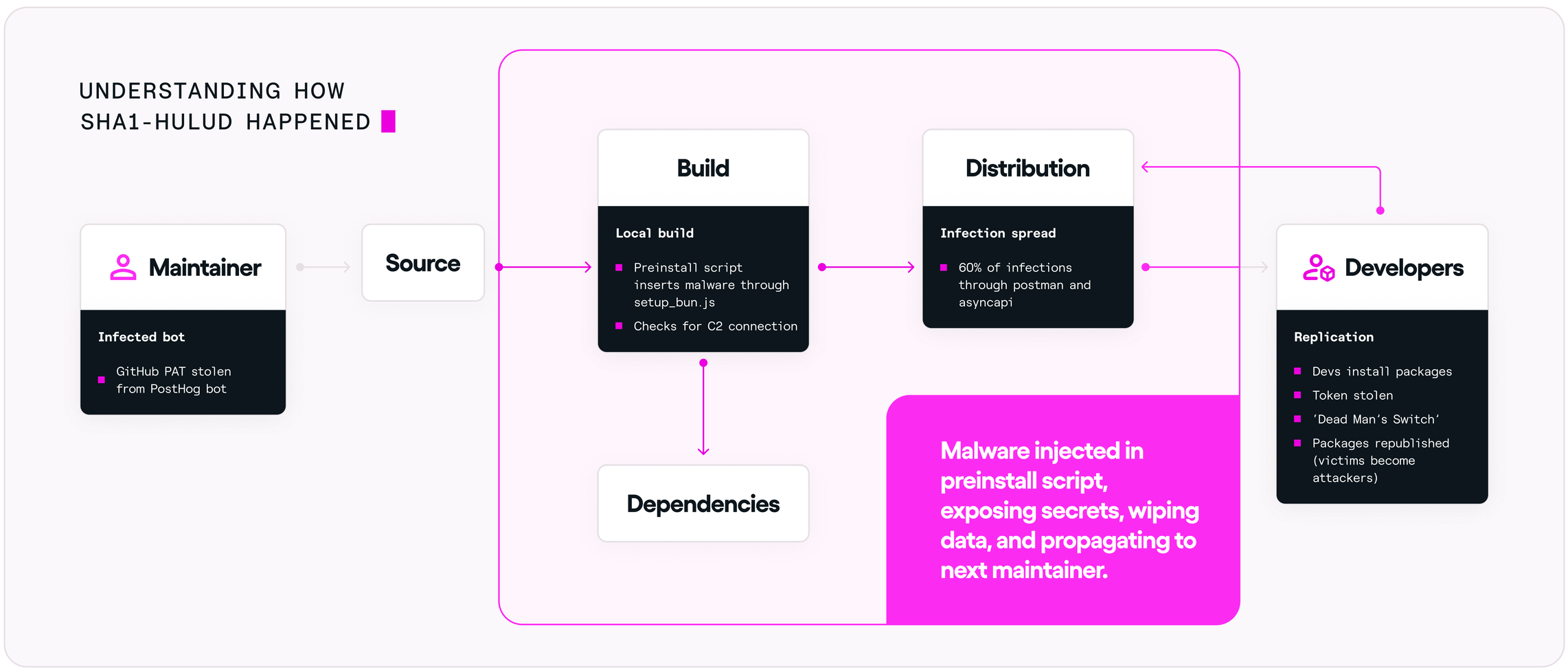
MALWARE UNPACKED

Sha1-Hulud: The Preinstall Worm That Hijacked 26,000 Repos

What Happened?

"Sha1-Hulud" is a sophisticated worm that weaponized the npm install command itself. Instead of waiting for a package to be fully installed, the malware triggers immediately via a preinstall hook (`setup_bun.js`). This script downloads the Bun runtime to execute a heavily obfuscated payload (`bun_environment.js`) in memory.

Standard security tools scan files only after the installation finishes. Sha1-Hulud exploits this blind spot by executing during the download process—stealing secrets before a scan occurs. Once active, the worm operates autonomously: it harvests the victim's npm tokens, downloads every package they maintain, injects itself, bumps the version number, and republishes the poisoned versions—all in minutes. If the worm is unable to find credentials, it wipes the victim's home directory.



Impact

Blast Radius

~500 packages poisoned (132M+ downloads) and 30,000+ impacted repositories in 72 hours.

Data Exfiltration

700+ AWS, GCP, and Azure credentials and 700+ GitHub access tokens made public.

Brand Damage

Trojanized packages from industry giants like Zapier, Postman, and PostHog spread the worm.

Destructive Risk

Features a "dead man's switch" capable of wiping the victim's home directory if C2 access is lost.

The Chainguard Difference

Chainguard Libraries prevents this entire class of attack because every artifact is built from verifiable upstream source, not registry artifacts. While the rest industry downloads the poisoned tarball from npm (where the preinstall script lives), Chainguard builds the software directly from the verified source code. We bypass the registry entirely—never ingesting the malicious code. Plus, our build pipeline strictly disables install-time scripts by default, as these scripts are known to contain malware. In turn, Chainguard neutralizes the vector before it can execute.

Prevent entire classes of supply chain attacks

Access 100K+ libraries built in an isolated, tamper-proof environment that neutralizes build-time and distribution-based malware injections by default.

Eliminate "are we impacted?" fire drills

When the next headline-grabbing library attack hits, don't stall development to prove a negative. Insulate your team against the panic and disruption of upstream compromises.

Streamline compliance evidence

Prove that your libraries are protected from third-party manipulation by providing auditors with automated provenance and signed SBOMs that verify component integrity.

LEARN MORE AT CHAINGUARD.DEV/LIBRARIES.