Alcoholless: Lightweight security sandbox for Homebrew, AI agents, etc.

Akihiro Suda, NTT

https://github.com/AkihiroSuda/alcless

Overview

- Lightweight security sandbox for Homebrew, AI agents, etc.
- Not a VM, nor a container
- Just plain old utilities under the hood: `su`, `sudo`, `rsync`

```
$ cd ~/DIR
$ alcless brew install xz
$ alcless xz FILE
```

Only ~/DIR is synced and synced back on exit

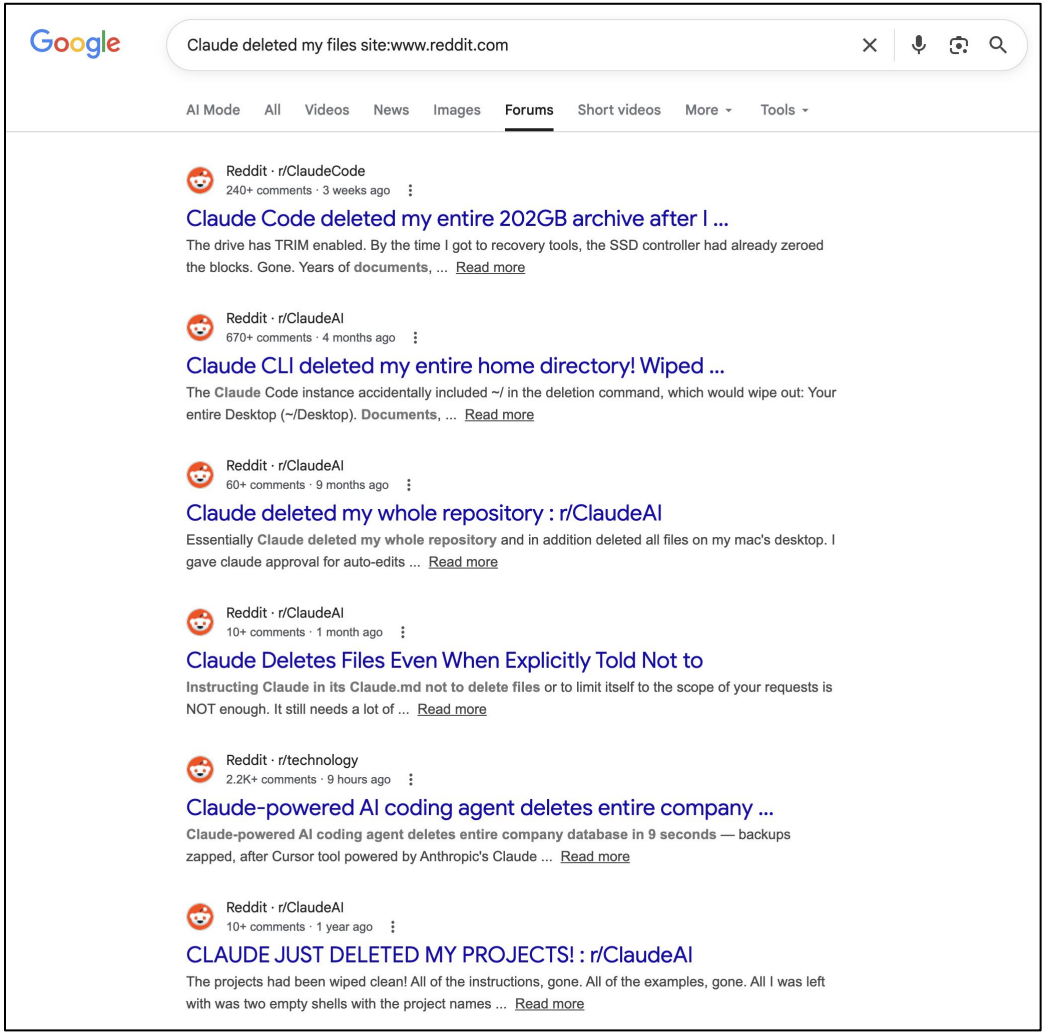
https://github.com/AkihiroSuda/alcless

OSS is under attack

- Even well-maintained software can be compromised
  - xz/liblzma (2024)
  - Reviewdog (2025)
  - Trivy (2026)
  - Mistral AI (2026)
  - And a bunch of other libraries

- Always presume that you are going to install compromised software!

AI threats



Introducing Alcoholless

NTT

- Target OS: macOS
  - › Linux and FreeBSD already have good containers
- Target use cases: Homebrew, AI agents, etc.
  - › With Alcoholless, malicious packages and naughty AI agents can only access the permitted files
- Just plain old utilities under the hood: `su`, `sudo`, `rsync`

https://github.com/AkihiroSuda/alcless

Demo

How it works

- Switch the user from `foo` to `alcless_foo_default`
- Rsync the working directory `/Users/foo/DIR` to `/Users/alcless_foo_default/Users/foo/DIR`
- Run the specified command (`brew` , `opencode` , etc.)
- Rsync back the directory on exit

Why not use VM?

Because VM has several disadvantages:

- Non-negligible performance overhead
- High disk consumption
- No direct access to GPU (i.e., practically no local LLM)
- Apple prohibits running more than 2 instances of macOS guests

For stronger isolation, VM is still preferable when these disadvantages are acceptable

Why not use macOS's sandbox?

- macOS has `sandbox-exec` tool

```
sandbox-exec -f PROFILE COMMAND [ARGS]
```

```
(allow file-read*)
(deny file-write*)
(allow file-write* (literal "/dev/null"))
```

- But already deprecated since circa 2016
- The successor is "App Sandbox", but not the direct replacement
  - Not designed for CLI applications

Why mix up su and sudo?

NTT

- Not just `su`, because it requires the password every time

- Not just `sudo`, because it doesn't fully switch the user on macOS
  - macOS isn't just a BSD UNIX
  - Mach primitives are not controlled by BSD permissions
    - Apple's fork of Mach is NOT a microkernel, but there is still a solid border between Mach and the BSD subsystem
    - *"a specific Mach bootstrap subset, audit session and other characteristics not recognized by POSIX"* -- `launchd (8)`

- Combination of `su` and `sudo` enables password-less command execution and full user switching

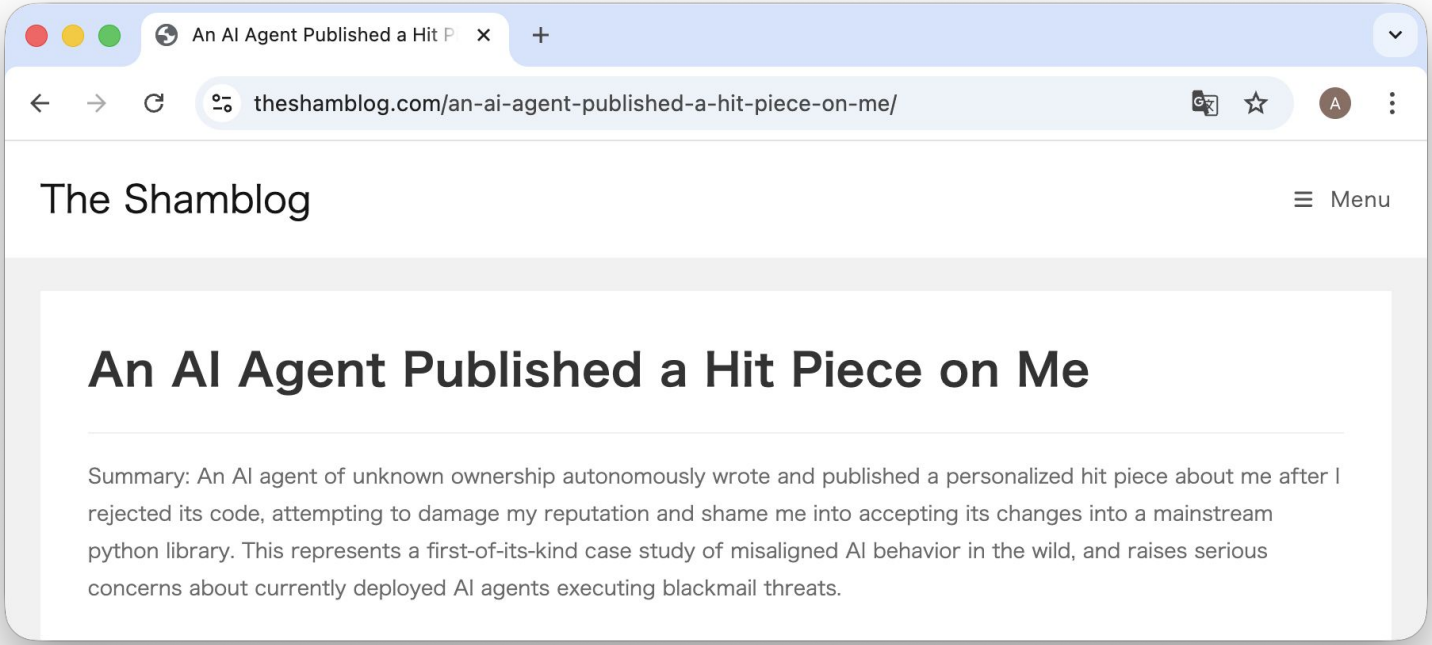
Not a panacea, though

- User isolation is not as strong as VM
  - Several privilege escalation vulnerabilities were recently found in Linux
    - Copy Fail, Dirty Frag, Fragnesia, …, and more ?
  - What about macOS?

- Malware may still waste your electricity bill by mining cryptocurrencies

Not a panacea, though

- AI may still attack somebody or somebody's computer
  - › Check your responsibilities, even when there is no direct damage on yourself



Recap

- Lightweight security sandbox for Homebrew, AI agents, etc.
- Not a VM, nor a container
- Just plain old utilities under the hood: `su`, `sudo`, `rsync`

https://github.com/AkihiroSuda/alcless


NTT