httpjail (via) Here's a promising new (experimental) project in the sandboxing space from Ammar Bandukwala at Coder. `httpjail` provides a Rust CLI tool for running an individual process against a custom configured HTTP proxy.

The initial goal is to help run coding agents like Claude Code and Codex CLI with extra rules governing how they interact with outside services. From Ammar's blog post that introduces the new tool, Fine-grained HTTP filtering for Claude Code:

> `httpjail` implements an HTTP(S) interceptor alongside process-level network isolation. Under default configuration, all DNS (udp:53) is permitted and all other non-HTTP(S) traffic is blocked.
>
> `httpjail` rules are either JavaScript expressions or custom programs. This approach makes them far more flexible than traditional rule-oriented firewalls and avoids the learning curve of a DSL.
>
> Block all HTTP requests other than the LLM API traffic itself:
>
> ```
> $ httpjail --js "r.host === 'api.anthropic.com'" -- claude "build something great"
> ```

I tried it out using OpenAI's Codex CLI instead and found this recipe worked:

```
brew upgrade rust
cargo install httpjail # Drops it in `~/.cargo/bin`
httpjail --js "r.host === 'chatgpt.com'" -- codex
```

Within that Codex instance the model ran fine but any attempts to access other URLs (e.g. telling it "`Use curl to fetch simonwillison.net`)" failed at the proxy layer.

This is still at a really early stage but there's a lot I like about this project. Being able to use JavaScript to filter requests via the `--js` option is neat (it's using V8 under the hood), and there's also a `--sh shellscript` option which instead runs a shell program passing environment variables that can be used to determine if the request should be allowed.

At a basic level it works by running a proxy server and setting `HTTP_PROXY` and `HTTPS_PROXY` environment variables so well-behaving software knows how to route requests.

It can also add a bunch of other layers. On Linux it sets up nftables rules to explicitly deny additional network access. There's also a `--docker-run` option which can launch a Docker container with the specified image but first locks that container down to only have network access to the `httpjail` proxy server.

It can intercept, filter and log HTTPS requests too by generating its own certificate and making that available to the underlying process.

I'm always interested in new approaches to sandboxing, and fine-grained network access is a particularly tricky problem to solve. This looks like a very promising step in that direction - I'm looking forward to seeing how this project continues to evolve.

Posted 19th September 2025 at 9:57 pm

Recent articles

- I think "agent" may finally have a widely enough agreed upon definition to be useful jargon now - 18th September 2025
- My review of Claude's new Code Interpreter, released under a very confusing name - 9th September 2025
- Recreating the Apollo AI adoption rate chart with GPT-5, Python and Pyodide - 9th September 2025

http 98 javascript 722 proxies 19 sandboxing 21 security 545 v8 11 rust 83 claude-code 27 codex-cli 5