AI coding agent sandbox: how the Codex sandbox limits files and network access

26 November 2026 · 5 min read · blogs.iictp.org/codex-sandbox-ai-coding-agent-permissions-explained

Short answer: the Codex sandbox is the operating-system boundary that limits which files and network hosts commands run by OpenAI's Codex coding agent can reach, and OpenAI described its design on 18 November 2026. It uses permission profiles enforced by Seatbelt on macOS and by bubblewrap with seccomp on Linux, with Windows moving to Microsoft's MXC library. It matters because an agent that runs shell commands has your access unless something outside the model restricts it.

TL;DR The short version

- The Codex sandbox is the operating-system boundary that limits which files and network hosts commands from OpenAI's Codex coding agent can reach, described by OpenAI on 18 November 2026.
- It uses permission profiles enforced by Seatbelt on macOS and bubblewrap with seccomp on Linux, with Windows moving to Microsoft's MXC library.
- An agent that runs shell commands has your access unless something outside the model restricts it, so set sandbox profiles rather than relying on approval prompts alone.

Read online, listen to it, or change the text size: https://blogs.iictp.org/codex-sandbox-ai-coding-agent-permissions-explained

What is the Codex sandbox?

A sandbox limits what a program can access while it runs. In the post by Michael Bolin, the Codex sandbox lets the agent work inside your existing development environment, instead of a separate virtual machine or container, within filesystem and network limits that you control. OpenAI gives the example of an iOS project: a Docker-only setup could not run the native Mac build tools or the iPhone simulator without a second set of host restrictions.

How it works: permission profiles and enforcement

- **Permission profiles.** Defined under [permissions] in config.toml. A profile can extend a built-in one and mark paths as read, write or deny.
- **Default workspace profile.** With a Git repository as the root, :workspace allows writes inside the workspace but keeps .git and .codex read-only, allows broad reads, and disables network access.
- **Per-command escalation.** The model can ask for extra permissions or to run outside the sandbox, and the harness decides, by prompting you or through auto-review.
- **Network proxy.** For domain allow and deny rules, a local HTTP and SOCKS5 proxy checks each request, and the OS blocks direct connections.

- **macOS and Linux.** Seatbelt through sandbox-exec on macOS; bubblewrap plus seccomp-bpf filters on Linux, inherited by the whole process tree.

What is new compared with approval prompts alone

Asking a person to approve every command does not scale, and an approval cannot show what a script will do after it starts. A kernel-enforced boundary applies to the command and to every child process it launches, including ones that outlive it. The post also shows codex sandbox --permission-profile <name> -- <command>, which wraps any command, so you can restrict a third-party MCP server that would otherwise run with your full permissions.

Results and behaviour reported by the source

- The post publishes no benchmark. It documents behaviour and trade-offs.
- Domain rules accept exact names and wildcards, and an explicit deny wins over an allow.
- Domain rules are enforced only when the network proxy feature is enabled. Without it, a profile with network enabled allows direct access.
- The proxy resolves a hostname only after the request is allowed, because even a DNS lookup can leak data. OpenAI warns that broad wildcards or DNS access can still permit DNS-based exfiltration.
- Managed requirements in requirements.toml let administrators restrict which profiles users may select, including blocking full access.

Limitations and open questions

- OpenAI states that harmful actions are still possible inside the limits. Its example: an agent edits a package in node_modules, Git ignores the change, and your editor later runs it outside the sandbox.
- Broad reads mean files elsewhere on the machine, such as configuration files with API keys, may remain readable.
- On Linux, deny globs are expanded to concrete paths before each command, so matching files created afterwards are not added to that command’s restrictions.
- Apple’s manual page marks sandbox-exec as deprecated, although OpenAI plans to keep using Seatbelt for now. Sources were read on 26 November 2026. Products, rules and standards change, so check the linked originals before you rely on any detail.

How to apply it to your team

1. List what the agent really needs to write and which hosts it needs to reach.
2. Create a named profile that extends :workspace, and mark .env files and secret folders as deny.
3. Turn on the network proxy before relying on domain rules, then test that a blocked host really fails.
4. Wrap each MCP server in its own profile, pin its package version and remove network access where it needs none.
5. Use managed requirements to stop full-access mode on work machines.

What it means for developers and platform teams

Treat agent permissions like service-account permissions: least privilege, written down and tested. Even if you use a different coding agent, the checklist transfers: what can it read, what can it write, where can it connect, who can raise those limits, and what runs later outside the boundary. See least privilege and prompt injection for why the boundary matters.

Frequently asked questions

- **What is an AI coding agent sandbox?** An enforced limit on the files, network hosts and system resources that commands run by the agent can access.
- **Does the Codex sandbox allow network access by default?** No. The post says the default workspace profile disables it.
- **Can the Codex sandbox protect other tools?** The codex sandbox command can run any command, including an MCP server, under a profile.
- **Is a sandbox enough on its own?** No. OpenAI describes it as one layer alongside model training, alignment and approval controls.

Related reading

- What is least privilege?
- What is prompt injection?
- What is Model Context Protocol?
- AI Practitioner certification

Related credential: AI Practitioner. Explore the exam facts and syllabus.

Sources and further reading

1. Michael Bolin, OpenAI. Inside the Codex sandbox. 18 November 2026. openai.com/index/inside-the-codex-sandbox

Published by IICTP, an independent technical credential body, on 26 November 2026. The current version of this post is at https://blogs.iictp.org/codex-sandbox-ai-coding-agent-permissions-explained. More guides: blogs.iictp.org.