Local Codex file and environment review

Engineering security and Codex administrators

Read the complete guide on aona.ai

Review both the files the local Codex CLI can read and the environment its tools inherit. Read-only execution does not mean every file is unreadable, and a hidden .env file is not an access control. Current OpenAI documentation describes deny-read permission profiles and environment filtering; verify their availability and effective configuration in your installed client using only synthetic markers.

Use different synthetic markers to review file reads and inherited tool-process data.

Fake inputs and an offline boolean probe. Product outcomes are untested.

Your review

Reviewer: _____ Date: _____

Organisation or team: _____

☐ **Record the effective local permission model**

Include CLI version, OS, roots, selected profile and relevant administrative requirements.

☐ **Check inheritance without printing secrets**

Use only the fixed fake variable and record its setup separately from the boolean result.

☐ **Explain the observation**

A missing marker is not conclusive until the allowed baseline and fixture setup are established.

Checked items record your review, not a compliance approval or an installed-product test result.

Notes and next actions

Decision and scope: _____

Evidence to collect: _____

Owner and next review date: _____

Review matrix

Input	Intended rule	Observed
Allowed file	Define a readable baseline	Untested
Restricted .env canary	Define the read restriction	Untested
Fake inherited variable	Define whether inheritance is allowed	Untested

Local Codex synthetic secret review

No real credentials, provider calls or automatic configuration changes are included. Use a NEW isolated folder with no real repository or secrets.

File-read review

1. Copy env-canary.txt to .env.canary in this fixture folder.
2. Inspect codex-permissions.example.toml. It selects d06-canary, extends :read-only and denies exactly .env.canary under effective workspace roots. It is not an active config filename. Have the authorised administrator apply only the reviewed example through the supported test configuration mechanism, preserving managed requirements and other restrictions.
3. Current permission profiles are beta. They do not compose with sandbox_mode, --sandbox or sandbox_workspace_write; an older setting can cause the profile to be ignored. Confirm the effective selected profile and supported installed version before proceeding. Do not relax an existing policy to activate the example.
4. Establish a permitted read of allowed-marker.txt, then request a direct read of .env.canary. The intended profile outcome is denial of that fake file. Record actual behaviour in results.csv; the pack contains no observed client result.

Separate inherited-environment review
5. Use only AONA_D06_FAKE_INHERITED=NOT_A_CREDENTIAL_D06_ENV_MARKER from fake-environment.txt in the isolated test launch. Do not substitute a real value.
6. If the already approved baseline policy permits that fake variable, run the offline environment_probe.py through the intended tool process and record whether it returns true. A false baseline leaves the setup/inheritance prerequisite unresolved; do not broaden policy to force a true result.
7. Inspect codex-environment-filter.example.toml. An authorised administrator can add its single fake-variable exclusion while preserving the rest of the test policy. Current filters must not be mixed with legacy exclude/include_only settings. The fragment does not change inheritance or inject a value.
8. Repeat the same probe in a fresh applicable test launch. The intended filtered result is false. Record the effective configuration and result separately. The probe prints only a boolean and makes no network call.

The profile uses one exact relative filename, not an unbounded glob. Broader rules have platform/version considerations documented by OpenAI. These local tests do not establish cloud-agent behaviour or universal Aona coverage.

Results

These records are arranged vertically for reading and printing. The Excel workbook retains the full column layout.

Record 1

Field	Record value
Client version	
OS	
Permission model	
Workspace roots	
Input	allowed file

Field	Record value
Parent setup	DEFINE
Intended policy	DEFINE
Approval	
Observed	UNTESTED
Evidence	

Record 2

Field	Record value
Client version	
OS	
Permission model	
Workspace roots	
Input	restricted file
Parent setup	DEFINE
Intended policy	DEFINE
Approval	
Observed	UNTESTED
Evidence	

Record 3

Field	Record value
Client version	
OS	
Permission model	
Workspace roots	
Input	fake inherited variable
Parent setup	DEFINE

Field	Record value
Intended policy	DEFINE
Approval	
Observed	UNTESTED
Evidence	

Practice files

The complete download pack includes these original working files. Keep their original formats when running or inspecting the examples.

- Allowed marker (allowed-marker.txt)

SYNTHETIC_D06_ALLOWED_MARKER

- Env canary (env-canary.txt)

EXAMPLE_ONLY=NOT_A_CREDENTIAL_D06_FILE_MARKER

- Fake environment (fake-environment.txt)

AONA_D06_FAKE_INHERITED=NOT_A_CREDENTIAL_D06_ENV_MARKER

- Environment probe (environment_probe.py)
- Codex permissions.example (codex-permissions.example.toml)
- Codex environment filter.example (codex-environment-filter.example.toml)

Sources and scope

Sources checked 2026-09-21. Review the current requirements and your actual agreement before using this workbook for a real decision. This fixture does not prove Aona intercepts every Codex CLI command or tool result. Native capabilities depend on installed release, OS, provider and transport.

OpenAI: Agent approvals and security. Separates sandbox permissions and approval policy.

OpenAI: Permission profiles. Documents deny-read exact paths/globs and platform-specific considerations.

OpenAI: Configuration reference. Documents shell-environment inheritance and filtering.

Codex: historical .env read concern. A March 2026 practitioner concern, not a claim about current CLI capability.