

ACE-Step-1.5

github.com/ace-step/ACE-Step-1.5

Scanned 24 September 2026 at 09:28 · Report generated 29 October 2026 · Python · 12,862 stars

OVERALL ASSESSMENT

Medium

Combined score 80 · 213 findings after filtering

Summary

This repository shows signs of both risky and potentially malicious code, so it should be treated with caution before use. The most serious issue is that several critical secrets like API keys and passwords are hardcoded directly into the source files, which means anyone who sees the code can steal them immediately. While some dependencies lack version locks, which creates supply chain risks, these issues appear to be unintentional rather than intentional attempts to harm others. The overall security score of 31.0 reflects medium-level vulnerabilities that could be exploited if the secrets are leaked or misused.

Written by a language model running locally on the scanning machine, from the findings in this report. The risk levels above and below were decided by the scoring model, not by this summary.

Risk, by type

MalDet scores two different questions separately, because they call for different responses: code can be badly written without being hostile, and hostile code is not always badly written.

VULNERABILITY

Medium (31)

Insecure coding an attacker could exploit. Usually mistakes, and usually fixable in place.

MALICIOUS PATTERN

High (49)

Signs the code may be trying to do harm: hidden payloads, credential access, persistence, exfiltration.

What each detector found

DETECTOR	LOOKS FOR	HIGH	MEDIUM	LOW	TOTAL
bandit	Insecure coding patterns in Python source	0	27	101	128
dep_checker	Vulnerable, unpinned or typosquatted dependencies	0	15	44	59
yara	Signatures of malicious behaviour in source files	13	0	4	17

DETECTOR	LOOKS FOR	HIGH	MEDIUM	LOW	TOTAL
guarddog	Malicious package and install-time behaviour	1	3	5	9

32 of 213 findings were rated more likely genuine than noise by the classifier.

Evidence

Identical findings are grouped, so a single rule firing many times takes one row. Ordered worst first. “Seen” is how many times that finding occurred.

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	guarddog	Detects access to sensitive files (credentials, configs, keys) [threat-filesystem-read]	4	/acestep/ api_server.py :173
HIGH	yara	YARA rule matched: detect_download_and_execute — Downloads a remote resource AND executes it - the actually dangerous combination, not just downloading alone	10	/ start_gradio_ ui_macos.sh
HIGH	yara	YARA rule matched: detect_persistence_mechanism — Modifies a startup/scheduling location - legitimate application code essentially never does this	3	/.claude/ skills/ acestep-docs/ guides/ UPDATE_AND_B ACKUP.md
MEDIUM	dep_checker	Unpinned dependency: 'modelscope' has no version lock — supply chain risk	2	pyproject.to ml
MEDIUM	dep_checker	Unpinned dependency: 'lycoris-lora' has no version lock — supply chain risk	2	pyproject.to ml
MEDIUM	dep_checker	Unpinned dependency: 'xxhash' has no version lock — supply chain risk	1	requirements. txt:62
MEDIUM	dep_checker	Unpinned dependency: 'setuptools' has no version lock — supply chain risk	1	pyproject.to ml
MEDIUM	dep_checker	Unpinned dependency: 'diskcache' has no version lock — supply chain risk	1	pyproject.to ml
MEDIUM	dep_checker	Unpinned dependency: 'diffusers' has no version lock — supply chain risk	1	requirements. txt:23
MEDIUM	dep_checker	Unpinned dependency: 'torchao' has no version lock — supply chain risk	1	pyproject.to ml

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	dep_checker	Unpinned dependency: 'torchvision' has no version lock — supply chain risk	1	pyproject.to ml
MEDIUM	dep_checker	Unpinned dependency: 'nano-vllm' has no version lock — supply chain risk	1	pyproject.to ml
MEDIUM	dep_checker	Unpinned dependency: 'toml' has no version lock — supply chain risk	2	pyproject.to ml
MEDIUM	dep_checker	Unpinned dependency: '--extra-index-url https://download.pytorch.org/whl/cu128' has no version lock — supply chain risk	1	requirements. txt:3
MEDIUM	dep_checker	Unpinned dependency: '--extra-index-url https://download.pytorch.org/whl/cu130' has no version lock — supply chain risk	1	requirements. txt:4
MEDIUM	bandit	Use of unsafe PyTorch load	1	/acestep/ training_v2/ trainer_help ers.py:339
MEDIUM	bandit	Audit url open for permitted schemes. Allowing use of file:/ or custom schemes is often unexpected.	1	/acestep/ text_tasks/ external_lm_ model_discov ery.py:51
MEDIUM	bandit	Call to requests without timeout	4	/scripts/ lora_data_pr epare/ gemini_capti on.py:184
MEDIUM	bandit	Unsafe Hugging Face Hub download without revision pinning in snapshot_download()	2	/acestep/api/ model_downlo ad.py:54
MEDIUM	bandit	Unsafe Hugging Face Hub download without revision pinning in from_pretrained()	19	/acestep/ llm_inference .py:4231
LOW	bandit	Possible hardcoded password: 'ollama'	1	/acestep/ text_tasks/ external_lm_ model_discov ery.py:96
LOW	bandit	Possible hardcoded password: 'ACESTEP_ANTHROPIC_SECRET_PATH'	1	/acestep/ text_tasks/ external_lm_ providers.py: 64

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	bandit	Possible hardcoded password: 'ACESTEP_GLM_SECRET_PATH'	1	/acestep/text_tasks/external_lm_providers.py:35
LOW	bandit	Possible hardcoded password: 'ACESTEP_OLLAMA_SECRET_PATH'	1	/acestep/text_tasks/external_lm_providers.py:78
LOW	bandit	Possible hardcoded password: 'ACESTEP_OPENAI_SECRET_PATH'	1	/acestep/text_tasks/external_lm_providers.py:50
LOW	bandit	Possible hardcoded password: 'acestep.external_lm'	1	/acestep/text_tasks/passphrase_store.py:13
LOW	bandit	Possible hardcoded password: 'external_lm_store_passphrase'	1	/acestep/text_tasks/passphrase_store.py:14
LOW	bandit	Possible hardcoded password: 'lyric_token_ids'	1	/acestep/ui/gradio/events/results/session_artifacts.py:22
LOW	bandit	Possible hardcoded password: 'anthropic_messages'	1	/acestep/text_tasks/external_lm_model_discovery.py:92
LOW	guarddog	Detects reading of environment variables (credential access, often contains secrets) [threat-runtime-environment-read]	3	/acestep/api/server_cli.py:38
LOW	bandit	Possible hardcoded password: 'secret-tool'	1	/acestep/text_tasks/passphrase_store.py:18
LOW	guarddog	Detects fully silent process execution (suppressing all output channels) [threat-process-spawn-silent]	2	/acestep/api/runtime_helpers.py:72

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	bandit	Possible hardcoded password: "	6	/acestep/ text_tasks/ secure_secret_store.py:68
LOW	bandit	Try, Except, Continue detected.	4	/acestep/ constrained_logits_processor.py:573
LOW	bandit	Consider possible security implications associated with the subprocess module.	6	/acestep/ gpu_config.py:611
LOW	bandit	Starting a process with a partial executable path	3	/acestep/ gpu_config.py:613
LOW	dep_checker	Loose version constraint: 'python-multipart>=0.0.18' — attacker could push malicious update	2	requirements.txt:33
LOW	bandit	subprocess call - check for execution of untrusted input.	8	/acestep/ text_tasks/ passphrase_store.py:117
LOW	dep_checker	Loose version constraint: 'mlx-lm>=0.20.0' — attacker could push malicious update	1	pyproject.toml
LOW	dep_checker	Loose version constraint: 'mlx>=0.25.2' — attacker could push malicious update	1	pyproject.toml
LOW	dep_checker	Loose version constraint: 'torch>=2.9.1' — attacker could push malicious update	1	pyproject.toml
LOW	dep_checker	Loose version constraint: 'torchao>=0.16.0,<0.17.0' — attacker could push malicious update	1	pyproject.toml
LOW	dep_checker	Loose version constraint: 'torchao>=0.16.0,<0.17.0; platform_machine != 'aarch64' — attacker could push malicious update	1	requirements.txt:41
LOW	dep_checker	Loose version constraint: 'torchaudio>=2.9.1' — attacker could push malicious update	1	pyproject.toml
LOW	dep_checker	Loose version constraint: 'fastapi>=0.110.0' — attacker could push malicious update	2	pyproject.toml

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	dep_checker	Loose version constraint: 'numba>=0.63.1' — attacker could push malicious update	2	pyproject.to ml
LOW	dep_checker	Loose version constraint: 'peft>=0.18.0' — attacker could push malicious update	2	requirements. txt:47
LOW	dep_checker	Loose version constraint: 'scipy>=1.10.1' — attacker could push malicious update	2	pyproject.to ml
LOW	dep_checker	Loose version constraint: 'matplotlib>=3.7.5' — attacker could push malicious update	2	pyproject.to ml
LOW	dep_checker	Loose version constraint: 'loguru>=0.7.3' — attacker could push malicious update	2	pyproject.to ml
LOW	dep_checker	Loose version constraint: 'vector-quantize-pytorch>=1.27.15' — attacker could push malicious update	2	pyproject.to ml
LOW	dep_checker	Loose version constraint: 'lightning>=2.0.0' — attacker could push malicious update	2	requirements. txt:49
LOW	dep_checker	Loose version constraint: 'einops>=0.8.1' — attacker could push malicious update	2	pyproject.to ml
LOW	dep_checker	Loose version constraint: 'transformers>=4.51.0,<4.58.0' — attacker could push malicious update	2	pyproject.to ml
LOW	dep_checker	Loose version constraint: 'accelerate>=1.12.0' — attacker could push malicious update	2	pyproject.to ml
LOW	dep_checker	Loose version constraint: 'tensorboard>=2.0.0' — attacker could push malicious update	2	pyproject.to ml
LOW	dep_checker	Loose version constraint: 'soundfile>=0.13.1' — attacker could push malicious update	2	pyproject.to ml
LOW	dep_checker	Loose version constraint: 'pytorch-wavelets>=1.3.0' — attacker could push malicious update	1	pyproject.to ml
LOW	dep_checker	Loose version constraint: 'diffusers>=0.37.0' — attacker could push malicious update	1	pyproject.to ml
LOW	dep_checker	Loose version constraint: 'pywavelets>=1.9.0' — attacker could push malicious update	1	pyproject.to ml

Showing the 60 most significant of 69 distinct findings. The counts in the previous section cover all of them.

How to read this report

- **This is a static scan.** The code was read, never executed. Behaviour that only appears at runtime is outside what any of these detectors can see.
- **Findings are not all equally real.** A classifier rates each finding on how likely it is to be genuine rather than noise, and that rating weights the scores. Measured against a hand-reviewed sample it reached precision 0.52 and recall 0.62 — useful for ranking, not a verdict on any single finding.
- **A high score is not proof of malice.** Security tools, penetration-testing frameworks and malware research repositories contain the same patterns as real malware by design. The two risk types above exist to keep that distinction visible.
- **Absence of findings is not proof of safety.** A detector that is not installed, or a language none of them parse, produces silence rather than a warning.
- **This scan is a snapshot.** The repository may have changed since 24 September 2026.