

Hello-CTF

github.com/ProbiusOfficial/Hello-CTF

Scanned 24 September 2026 at 09:45 · Report generated 29 October 2026 · HTML · 4,218 stars

OVERALL ASSESSMENT

Medium

Combined score 29 · 29 findings after filtering

Summary

The repository has a medium vulnerability score of 29, which indicates that while the code does not appear to be maliciously crafted, it contains several common insecure coding patterns that an attacker could exploit. These issues include running the application with a writable root filesystem and allowing it to bind to all network interfaces, which means anyone on the internet could potentially access or modify files inside the container. Additionally, there are concerns about using weak hash algorithms and dynamic file paths, but these are standard weaknesses found in many applications rather than signs of hidden malicious intent. The scan also flagged a GitHub Actions workflow that uses mutable tags, which is a supply chain risk but does not necessarily mean the code itself is dangerous. Overall, the findings suggest the project needs careful review to fix these specific security gaps before it can be considered safe for production use.

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 (29)

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

MALICIOUS PATTERN

Safe (0)

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	5	11	16
semgrep	Insecure coding patterns across many languages	2	9	0	11
yara	Signatures of malicious behaviour in source files	0	0	2	2

6 of 29 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	semgrep	By not specifying a USER, a program in the container may run as 'root'. This is a security hazard. If an attacker can control a process running as root, they may have control over the container. Ensure that the last USER in a Dockerfile is a USER other than 'root'.	2	/collector/Dockerfile:12
MEDIUM	semgrep	Service 'hello-ctf' is running with a writable root filesystem. This may allow malicious applications to download and run additional payloads, or modify container files. If an application inside a container has to save something temporarily consider using a tmpfs. Add 'read_only: true' to this servi	1	/docker-compose.yml:3
MEDIUM	semgrep	Running flask app with host 0.0.0.0 could expose the server publicly.	1	/admin/app.py:382
MEDIUM	semgrep	Service 'hello-ctf' allows for privilege escalation via setuid or setgid binaries. Add 'no-new-privileges:true' in 'security_opt' to prevent this.	1	/docker-compose.yml:3
MEDIUM	semgrep	Detected a dynamic value being used with urllib. urllib supports 'file://' schemes, so a dynamic value controlled by a malicious actor may allow them to read arbitrary files. Audit uses of urllib calls to ensure user data cannot control the URLs, or consider using the 'requests' library instead.	1	/collector/app.py:65

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	GitHub Actions step uses a mutable tag or branch reference. Tags and branch names can be silently repointed by the action owner, enabling supply-chain attacks — as seen in the trivy-action and kics-github-action compromises. Pin the reference to a full 40-character commit SHA instead, e.g. <code>`uses: ac</code>	2	<code>/.github/workflows/ghdeploy.yml:26</code>
MEDIUM	bandit	Use of weak MD5 hash for security. Consider <code>usedforsecurity=False</code>	2	<code>/events_update.py:263</code>
MEDIUM	bandit	Audit url open for permitted schemes. Allowing use of <code>file:/</code> or custom schemes is often unexpected.	1	<code>/collector/app.py:65</code>
MEDIUM	bandit	Possible binding to all interfaces.	2	<code>/collector/app.py:184</code>
MEDIUM	semgrep	This tag is missing an 'integrity' subresource integrity attribute. The 'integrity' attribute allows for the browser to verify that externally hosted files (for example from a CDN) are delivered without unexpected manipulation. Without this attribute, if an attacker can modify the externally hosted	3	<code>/overrides/main.html:152</code>
LOW	bandit	Consider possible security implications associated with the subprocess module.	3	<code>/admin/deployer.py:7</code>
LOW	bandit	Starting a process with a partial executable path	3	<code>/admin/repo_sync.py:26</code>
LOW	bandit	subprocess call - check for execution of untrusted input.	5	<code>/admin/repo_sync.py:26</code>
LOW	yara	YARA rule matched: <code>detect_downloader</code> — Downloads a remote resource - common and low-signal alone; see <code>detect_download_and_execute</code> for the composite version	2	<code>/events_update.py</code>

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.