

LocalAI

github.com/mudler/LocalAI

Scanned 24 September 2026 at 07:49 · Report generated 29 October 2026 · Go · 49,250 stars

OVERALL ASSESSMENT

Critical

Combined score 1001 · 605 findings after filtering

Summary

This repository has two major red flags that indicate it is unsafe to use without significant changes. The code contains several critical vulnerabilities, such as running unverified user commands and insecure gRPC connections, which could allow attackers to execute arbitrary code or tamper with data in ways that are not just bad practice but potentially dangerous exploits. Additionally, the code includes patterns like downloading remote files and executing them directly, which is a known method for malware delivery. While some of these issues might seem like ordinary mistakes made by developers, the high scores from automated tools suggest they represent intentional or systemic risks rather than simple errors. Because of these findings, the repository should not be used in production environments until these security flaws are fixed and verified.

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

Critical (952)

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
semgrep	Insecure coding patterns across many languages	44	336	8	388

DETECTOR	LOOKS FOR	HIGH	MEDIUM	LOW	TOTAL
bandit	Insecure coding patterns in Python source	1	51	121	173
yara	Signatures of malicious behaviour in source files	21	0	16	37
guarddog	Malicious package and install-time behaviour	4	3	0	7

350 of 605 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	Detected non-static command inside Exec. Audit the input to 'syscall.Exec'. If unverified user data can reach this call site, this is a code injection vulnerability. A malicious actor can inject a malicious script to execute arbitrary code.	4	/core/cli/worker/worker_ds4.go:104
HIGH	semgrep	Found an insecure gRPC connection using 'insecure.NewCredentials()'. This creates a connection without encryption to a gRPC server. A malicious attacker could tamper with the gRPC message, which could compromise the machine. Instead, establish a secure connection with an SSL certificate using the 'g	1	/pkg/grpc/client.go:74
HIGH	semgrep	Detected user input entering a `subprocess` call unsafely. This could result in a command injection vulnerability. An attacker could use this vulnerability to execute arbitrary commands on the host, which allows them to download malware, scan sensitive data, or run any command they wish on the serve	1	/backend/python/mlx-video/backend.py:111
HIGH	semgrep	A `run:` step pipes the output of `curl` or `wget` directly into a shell interpreter. This is the "curl bash" install pattern — if the remote server is compromised or the URL is hijacked, an attacker can execute arbitrary code in your CI runner. Consider downloading the file first, verifying its c	14	/.github/workflows/test-extra.yml:1247

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	semgrep	This GitHub Actions workflow file uses <code>`pull_request_target`</code> and checks out code from the incoming pull request. When using <code>`pull_request_target`</code> , the Action runs in the context of the target repository, which includes access to all repository secrets. Normally, this is safe because the Action only	3	<code>/.github/workflows/disabled/comment-pr.yaml:11</code>
HIGH	guarddog	Detects download-and-execute patterns: fetching a remote file then executing it [threat-process-download-exec]	3	<code>/backend/python/moonshine/test.py:24</code>
HIGH	guarddog	Detects base64 decoding followed by code execution [threat-runtime-obfuscation-base64exec]	1	<code>/core/http/static/assets/pdf.min.js:22</code>
HIGH	semgrep	Using variable interpolation <code>`\${{...}}`</code> with <code>`github`</code> context data in a <code>`run:`</code> step could allow an attacker to inject their own code into the runner. This would allow them to steal secrets and code. <code>`github`</code> context data can have arbitrary user input and should be treated as untrusted. Instead, use	5	<code>/.github/workflows/backend_build_darwin.yaml:343</code>
HIGH	semgrep	Functions reliant on pickle can result in arbitrary code execution. Consider loading from <code>`state_dict`</code> , using fickling, or switching to a safer serialization method like ONNX	1	<code>/backend/python/vibevoice/backend.py:470</code>
HIGH	semgrep	Insecure WebSocket Detected. WebSocket Secure (wss) should be used for all WebSocket connections.	3	<code>/core/http/endpoints/openresponses/websocket.go:68</code>
HIGH	semgrep	Detected non-static command inside Command. Audit the input to 'exec.Command'. If unverified user data can reach this call site, this is a code injection vulnerability. A malicious actor can inject a malicious script to execute arbitrary code.	11	<code>/backend/go/vllm-cpp/video.go:364</code>

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	bandit	By default, jinja2 sets autoescape to False. Consider using autoescape=True or use the select_autoescape function to mitigate XSS vulnerabilities.	1	/backend/python/tinygrad/backend.py:246
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'.	1	/Dockerfile:441
HIGH	yara	YARA rule matched: detect_base64_exec — Encoded payload decoded and then executed	3	/core/http/static/assets/tailwindcss.js
HIGH	yara	YARA rule matched: detect_download_and_execute — Downloads a remote resource AND executes it - the actually dangerous combination, not just downloading alone	12	/core/http/static/assets/highlightjs.js
HIGH	yara	YARA rule matched: detect_obfuscated_string_execution — Reconstructs a string character-by-character or from hex escapes, then immediately executes it - fromCharCode/chr() alone are common in ordinary text-processing code, so this requires both the reconstruction and an execution call together	6	/core/http/static/assets/marked.js
MEDIUM	semgrep	RegExp() called with a `s` function argument, this might allow an attacker to cause a Regular Expression Denial-of-Service (ReDoS) within your application as RegExP blocks the main thread. For this reason, it is recommended to use hardcoded regexes instead. If your regex is run on user-controlled in	1	/core/http/static/assets/tw-elements.js:17

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	Found user-controlled request data passed into <code>'.write(...)'</code> . This could be dangerous if a malicious actor is able to control data into sensitive files. For example, a malicious actor could force rolling of critical log files, or cause a denial-of-service by using up available disk space. Instead, e	12	/backend/python/transformers/backend.py:549
MEDIUM	semgrep	Variable <code>`r`</code> is likely modified and later used on error. In some cases this could result in panics due to a nil dereference	1	/core/http/endpoints/openai/realtime_voicegate.go:254
MEDIUM	semgrep	Iteration over a possibly empty map <code>`neededDefs`</code> . This is likely a bug or redundant code	2	/core/http/endpoints/localai/api_instructions.go:199
MEDIUM	semgrep	Iteration over a possibly empty map <code>`votes`</code> . This is likely a bug or redundant code	1	/core/services/routing/router/knn.go:129
MEDIUM	semgrep	<code>`e.item.replace("[", "").replace`</code> method will only replace the first occurrence when used with a string argument (<code>"["</code>). If this method is used for escaping of dangerous data then there is a possibility for a bypass. Try to use sanitization library instead or use a Regex with a global flag.	1	/core/http/static/assets/alpine.js:5
MEDIUM	semgrep	<code>`e.item.replace`</code> method will only replace the first occurrence when used with a string argument (<code>"{"</code>). If this method is used for escaping of dangerous data then there is a possibility for a bypass. Try to use sanitization library instead or use a Regex with a global flag.	1	/core/http/static/assets/alpine.js:5
MEDIUM	semgrep	<code>`e.item.replace("{", "").replace`</code> method will only replace the first occurrence when used with a string argument (<code>"}"</code>). If this method is used for escaping of dangerous data then there is a possibility for a bypass. Try to use sanitization library instead or use a Regex with a global flag.	1	/core/http/static/assets/alpine.js:5

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	<code>`e.item.replace`</code> method will only replace the first occurrence when used with a string argument (<code>"["</code>). If this method is used for escaping of dangerous data then there is a possibility for a bypass. Try to use sanitization library instead or use a Regex with a global flag.	1	<code>/core/http/static/assets/alpine.js:5</code>
MEDIUM	semgrep	This Dependabot configuration does not set a cooldown period. Newly published packages can be malicious or unstable. Add a <code>`cooldown`</code> block with <code>`default-days: 7`</code> to each <code>`package-ecosystem`</code> entry under <code>`updates`</code> to wait 7 days before proposing updates to newly published package versions. Reference:	11	<code>/.github/dependabot.yml:4</code>
MEDIUM	guarddog	Detects access to sensitive files (credentials, configs, keys) [threat-filesystem-read]	3	<code>/cmd/local-ai/main.go:25</code>
MEDIUM	semgrep	Semgrep could not determine that the argument to <code>'template.HTML()'</code> is a constant. <code>'template.HTML()'</code> and similar does not escape contents. Be absolutely sure there is no user-controlled data in this template. If user data can reach this template, you may have a XSS vulnerability. Instead, do not use	1	<code>/core/http/render.go:88</code>
MEDIUM	semgrep	Downcasting or changing sign of an integer with <code>`int32`</code> method	4	<code>/backend/go/localvqe/golocalvqe.go:123</code>
MEDIUM	semgrep	The Origin header in the HTTP WebSocket handshake is used to guarantee that the connection accepted by the WebSocket is from a trusted origin domain. Failure to enforce can lead to Cross Site Request Forgery (CSRF). As per "gorilla/websocket" documentation: "A CheckOrigin function should carefully v	1	<code>/core/services/nodes/file_transfer_server.go:1197</code>
MEDIUM	semgrep	A session cookie was detected without setting the 'Secure' flag. The 'secure' flag for cookies prevents the client from transmitting the cookie over insecure channels such as HTTP. Set the 'Secure' flag by setting 'Secure' to 'true' in the Options struct.	7	<code>/core/http/auth/oauth.go:242</code>

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	Detected SHA1 hash algorithm which is considered insecure. SHA1 is not collision resistant and is therefore not suitable as a cryptographic signature. Use SHA256 or SHA3 instead.	1	/pkg/modelartifacts/materializer.go:598
MEDIUM	semgrep	Detected MD5 hash algorithm which is considered insecure. MD5 is not collision resistant and is therefore not suitable as a cryptographic signature. Use SHA256 or SHA3 instead.	1	/pkg/utils/hash.go:9
MEDIUM	semgrep	`MinVersion` is missing from this TLS configuration. By default, as of Go 1.22, TLS 1.2 is currently used as the minimum. General purpose web applications should default to TLS 1.3 with all other protocols disabled. Only where it is known that a web server must support legacy clients with unsupport	4	/core/services/cloudproxy/mitm/proxy.go:242
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.	2	/backend/python/insightface/test.py:74
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. `uses: ac	246	/.github/workflows/disabled/notify-models.yaml:98
MEDIUM	semgrep	When working with web applications that involve rendering user-generated content, it's important to properly escape any HTML content to prevent Cross-Site Scripting (XSS) attacks. In Go, the `text/template` package does not automatically escape HTML content, which can leave your application vuln	6	/core/templates/multimodal.go:5

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	Deserializing into <code>`interface{}`</code> allows arbitrary data structures and types, which can lead to security vulnerabilities (CWE-502). Use a concrete struct type instead.	4	/core/http/endpoints/localai/systemone.go:206
MEDIUM	semgrep	Detected the use of <code>exec()</code> . <code>exec()</code> can be dangerous if used to evaluate dynamic content. If this content can be input from outside the program, this may be a code injection vulnerability. Ensure evaluated content is not definable by external sources.	1	/backend/python/trl/reward_functions.py:173
MEDIUM	bandit	Use of unsafe PyTorch load	1	/backend/python/vibevoice/backend.py:470
MEDIUM	semgrep	Checks for disabling of TLS/SSL certificate verification. This should only be used for debugging purposes because it leads to vulnerability to MTM attacks.	1	/backend/go/valkey-store/store.go:293
MEDIUM	bandit	Audit url open for permitted schemes. Allowing use of <code>file:/</code> or custom schemes is often unexpected.	2	/backend/python/insightface/smoke.py:151
MEDIUM	semgrep	Avoid using <code>`pickle`</code> , which is known to lead to code execution vulnerabilities. When unpickling, the serialized data could be manipulated to run arbitrary code. Instead, consider serializing the relevant data as JSON or a similar text-based serialization format.	2	/backend/python/qwen-tts/backend.py:617
MEDIUM	semgrep	Detected direct use of <code>jinja2</code> . If not done properly, this may bypass HTML escaping which opens up the application to cross-site scripting (XSS) vulnerabilities. Prefer using the Flask method <code>'render_template()'</code> and templates with a <code>'.html'</code> extension in order to prevent XSS.	1	/backend/python/tinygrad/backend.py:246
MEDIUM	semgrep	Untrusted user input in <code>`importlib.import_module()`</code> function allows an attacker to load arbitrary code. Avoid dynamic values in <code>`importlib.import_module()`</code> or use a whitelist to prevent running untrusted code.	1	/backend/python/diffusers/diffusers_dynamic_loader.py:172

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	Detection of dangerouslySetInnerHTML from non-constant definition. This can inadvertently expose users to cross-site scripting (XSS) attacks if this comes from user-provided input. If you have to use dangerouslySetInnerHTML, consider using a sanitization library such as DOMPurify to sanitize your HT	5	/core/http/react-ui/src/components/CanvasPanel.jsx:141
MEDIUM	semgrep	Detected directly writing or similar in 'http.ResponseWriter.write()'. This bypasses HTML escaping that prevents cross-site scripting vulnerabilities. Instead, use the 'html/template' package and render data using 'template.Execute()'.	2	/core/services/nodes/file_transfer_server.go:181
MEDIUM	semgrep	Possibility of prototype polluting function detected. By adding or modifying attributes of an object prototype, it is possible to create attributes that exist on every object, or replace critical attributes with malicious ones. This can be problematic if the software depends on existence or non-exis	1	/core/http/react-ui/src/pages/ModelEditor.jsx:65
MEDIUM	semgrep	RegExp() called with a `e` function argument, this might allow an attacker to cause a Regular Expression Denial-of-Service (ReDoS) within your application as RegExP blocks the main thread. For this reason, it is recommended to use hardcoded regexes instead. If your regex is run on user-controlled in	1	/core/http/static/assets/tw-elements.js:17
MEDIUM	bandit	Use of exec detected.	1	/backend/python/trl/reward_functions.py:173
MEDIUM	semgrep	Do not use `math/rand`. Use `crypto/rand` instead.	11	/core/http/endpoints/localai/systemone.go:7
MEDIUM	bandit	Pickle and modules that wrap it can be unsafe when used to deserialize untrusted data, possible security issue.	1	/backend/python/qwen-tts/backend.py:568

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	In Kubernetes, each pod runs in its own isolated environment with its own set of security policies. However, certain container images may contain `setuid` or `setgid` binaries that could allow an attacker to perform privilege escalation and gain access to sensitive resources. To mitigate this risk,	1	/website/static/install/kubernetes.yaml:80
MEDIUM	bandit	Probable insecure usage of temp file/directory.	12	/backend/python/speaker-recognition/test.py:55
MEDIUM	bandit	Call to requests without timeout	3	/.github/check_and_update.py:30
MEDIUM	bandit	Unsafe Hugging Face Hub download without revision pinning in load_dataset()	9	/backend/python/trl/backend.py:319
MEDIUM	bandit	Unsafe Hugging Face Hub download without revision pinning in snapshot_download()	3	/backend/python/tinygrad/backend.py:89
MEDIUM	bandit	Unsafe Hugging Face Hub download without revision pinning in hf_hub_download()	3	/backend/python/tinygrad/backend.py:396
MEDIUM	bandit	Unsafe Hugging Face Hub download without revision pinning in from_pretrained()	16	/backend/python/trl/backend.py:770
LOW	bandit	Possible hardcoded password: '</think>'	1	/backend/python/vllm/test.py:566

Showing the 60 most significant of 68 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.