

sglang

github.com/sgl-project/sglang

Scanned 24 September 2026 at 06:41 · Report generated 29 October 2026 · Python · 36,385 stars

OVERALL ASSESSMENT

Critical

Combined score 711 · 8,729 findings after filtering

Summary

This repository has been flagged as extremely dangerous by automated security tools, with a critical vulnerability score of 588.0 and a malicious-pattern score of 122.0, indicating that the code contains severe risks that could be exploited by attackers or signs of intentional harm. The most serious issues involve the use of dynamic code loading and downloading files from the internet to execute them at runtime, which allows an attacker to potentially run arbitrary malware if the code is shared publicly. Additionally, there are multiple instances where shell commands are started without proper validation, creating a high risk for command injection attacks that could allow attackers to execute system commands on your machine. While some findings relate to standard security practices like using deprecated functions or parsing untrusted XML data, these represent common insecure coding patterns rather than evidence of malicious intent. Because the code is hosted on GitHub and appears to be public, it should not be used in any production environment without a thorough review and complete removal of all identified dangerous components.

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

 (588)

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

MALICIOUS PATTERN

Critical

 (122)

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	17	309	8260	8,586
yara	Signatures of malicious behaviour in source files	50	0	71	121
guarddog	Malicious package and install-time behaviour	8	2	12	22

259 of 8,729 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 suspicious autostart persistence mechanisms [threat-filesystem-autostart]	2	/python/ sglang/ multimodal_gen /runtime/ utils/ profiler.py:202
HIGH	guarddog	Detects dynamic code loading: downloading and importing/executing code at runtime [threat-runtime-dynamic-loader]	3	/python/ sglang/srt/ utils/ common.py:4515
HIGH	guarddog	Detects download-and-execute patterns: fetching a remote file then executing it [threat-process-download-exec]	3	/python/ sglang/ kernels/aot/ setup_metal.py:69
HIGH	bandit	tarfile.extractall used without any validation. Please check and discard dangerous members.	1	/scripts/ci/ slurm/ analyze_logs_ with_modal.py: 85
HIGH	bandit	By default, jinja2 sets autoescape to False. Consider using autoescape=True or use the select_autoescape function to mitigate XSS vulnerabilities.	1	/python/ sglang/srt/ parser/ template_dete ction.py:106
HIGH	bandit	Starting a process with a shell, possible injection detected, security issue.	4	/scripts/ playground/ lora/ analyzer.py:19

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	bandit	subprocess call with shell=True identified, security issue.	11	/scripts/ code_sync/ copy_from_oss. py:135
HIGH	yara	YARA rule matched: detect_credential_harvester — References credential/secret material by name AND a deliberate outbound-send API in the same file - narrower than 'reads any env var', which large legitimate codebases do constantly regardless of intent. Still not proof the two are connected (YARA c	6	/benchmark/ hichache/ bench_serving. py
HIGH	yara	YARA rule matched: detect_anti_analysis — Checks for a debugger, VM, or sandbox before continuing - a common malware evasion technique, essentially never present in ordinary application code	2	/python/ sglang/ multimodal_gen /runtime/ utils/ perf_logger.py
HIGH	yara	YARA rule matched: detect_base64_exec — Encoded payload decoded and then executed	5	/python/ sglang/ multimodal_gen /runtime/ pipelines_core /stages/ model_specifi c_stages/ minimax_h3/ material_io.py
HIGH	yara	YARA rule matched: detect_download_and_execute — Downloads a remote resource AND executes it - the actually dangerous combination, not just downloading alone	29	/scripts/ ci_monitor/ ci_auto_bisect .py
HIGH	yara	YARA rule matched: detect_persistence_mechanism — Modifies a startup/scheduling location - legitimate application code essentially never does this	8	/docker/ arm64.Dockerf ile
MEDIUM	bandit	Using xml.etree.ElementTree.XMLParser to parse untrusted XML data is known to be vulnerable to XML attacks. Replace xml.etree.ElementTree.XMLParser with its defusedxml equivalent function or make sure defusedxml.defuse_stdlib() is called	2	/python/ sglang/srt/ function_call/ minicpm5_dete ctor.py:115

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	guarddog	Detects unicode homoglyphs and uncommon characters used for obfuscation [threat-runtime-obfuscation-unicode]	1	/benchmark/bench_linear_attention/bench_cutedsl_kda_decode.py:434
MEDIUM	bandit	Use of insecure and deprecated function (mktemp).	1	/benchmark/hicache/bench_buffer_mode.py:261
MEDIUM	bandit	Chmod setting a permissive mask 0o755 on file (script_path).	1	/python/sglang/multimodal_gen/benchmarks/datasets.py:175
MEDIUM	guarddog	Detects access to sensitive files (credentials, configs, keys) [threat-filesystem-read]	1	/sgl-model-gateway/scripts/run_benchmarks.py:181
MEDIUM	bandit	Using xml.etree.ElementTree.fromstring to parse untrusted XML data is known to be vulnerable to XML attacks. Replace xml.etree.ElementTree.fromstring with its defusedxml equivalent function or make sure defusedxml.defuse_stdlib() is called	2	/python/sglang/srt/function_call/minicpm5_detector.py:118
MEDIUM	bandit	Use of unsafe PyTorch load	14	/python/sglang/multimodal_gen/tools/build_modelopt_fp8_transformer.py:630
MEDIUM	bandit	Use of weak SHA1 hash for security. Consider usedforsecurity=False	14	/python/sglang/srt/model_loader/loader.py:2088
MEDIUM	bandit	Use of weak MD5 hash for security. Consider usedforsecurity=False	1	/scripts/ci/cuda/warmup_deep_gemm.py:387

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	bandit	Audit url open for permitted schemes. Allowing use of file:/ or custom schemes is often unexpected.	16	/.claude/ skills/sglang- prod-incident- triage/ scripts/ incident_arti fact_tool.py:5 7
MEDIUM	bandit	Possible SQL injection vector through string-based query construction.	11	/python/ sglang/srt/ utils/ rpd_utils.py:2 15
MEDIUM	bandit	Use of possibly insecure function - consider using safer ast.literal_eval.	6	/python/ sglang/srt/ utils/ rank_consensu s_checker.py:2 46
MEDIUM	bandit	Use of exec detected.	2	/benchmark/io/ bench_mm_toke n_padding.py:5 8
MEDIUM	bandit	Pickle and modules that wrap it can be unsafe when used to deserialize untrusted data, possible security issue.	33	/benchmark/ hicache/ data_processi ng.py:459
MEDIUM	bandit	Possible binding to all interfaces.	30	/python/ sglang/srt/ entrypoints/ sidecar.py:35
MEDIUM	bandit	Probable insecure usage of temp file/directory.	39	/python/ sglang/srt/ utils/ stale_shm_cle anup.py:20
MEDIUM	bandit	Call to requests without timeout	39	/python/ sglang/srt/ entrypoints/ http_server.py :2335
MEDIUM	bandit	Unsafe Hugging Face Hub download without revision pinning in snapshot_download()	11	/python/ sglang/srt/ debug_utils/ model_truncat or.py:57

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	bandit	Unsafe Hugging Face Hub download without revision pinning in hf_hub_download()	6	/python/ sglang/srt/ model_loader/ loader.py:2763
MEDIUM	bandit	Unsafe Hugging Face Hub download without revision pinning in load_dataset()	5	/benchmark/ asr/ bench_sglang. py:268
MEDIUM	bandit	Unsafe Hugging Face Hub download without revision pinning in from_pretrained()	76	/scripts/ playground/ lora/ lora_hf_play. py:21
LOW	bandit	Possible hardcoded password: '...'	1	/scripts/ playground/ lora/ lora_hf_play. py:8
LOW	bandit	Possible hardcoded password: '<image>'	12	/python/ sglang/srt/ multimodal/ processors/ internvl.py:46
LOW	bandit	Possible hardcoded password: '<tool_call>'	9	/python/ sglang/srt/ function_call/ glm47_moe_det ector.py:311
LOW	bandit	Possible hardcoded password: '</tool_call>'	5	/python/ sglang/srt/ function_call/ mimo_detector. py:155
LOW	bandit	Possible hardcoded password: 'image'	4	/python/ sglang/ multimodal_gen /configs/ pipeline_conf igs/ longcat_image. py:102
LOW	bandit	Possible hardcoded password: '[IMG]'	4	/python/ sglang/srt/ multimodal/ processors/ pixtral.py:30

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	bandit	Possible hardcoded password: '< begin_of_image >'	2	/python/ sglang/srt/ multimodal/ processors/ glm4v.py:457
LOW	bandit	Possible hardcoded password: '< tool_calls_end >'	2	/python/ sglang/srt/ function_call/ deepseekv31_d etector.py:48
LOW	bandit	Possible hardcoded password: '<im_patch>'	2	/python/ sglang/srt/ multimodal/ processors/ step3_vl.py:34 2
LOW	bandit	Possible hardcoded password: '< video_pad >'	2	/python/ sglang/srt/ multimodal/ processors/ minicpmv4_6.py :334
LOW	bandit	Possible hardcoded password: '< end_of_image >'	2	/python/ sglang/srt/ multimodal/ processors/ glm4v.py:458
LOW	bandit	Possible hardcoded password: 'concat_steps'	2	/python/ sglang/srt/ debug_utils/ comparator/ aligner/ token_aligner/ entrypoint.py: 57
LOW	bandit	Possible hardcoded password: 'attn_cp_size'	1	/python/ sglang/srt/ arg_groups/ mega_moe_hook. py:135
LOW	bandit	Possible hardcoded password: '< tool_call_end >'	1	/python/ sglang/srt/ function_call/ lfm2_detector. py:455

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	bandit	Possible hardcoded password: '< video_start >'	1	/python/ sglang/srt/ multimodal/ processors/ bailing_mm.py: 39
LOW	bandit	Possible hardcoded password: ' </tool_call>'	1	/python/ sglang/srt/ function_call/ qwen25_detect or.py:39
LOW	bandit	Possible hardcoded password: '< action_start > < plugin >'	1	/python/ sglang/srt/ function_call/ internlm_dete ctor.py:50
LOW	bandit	Possible hardcoded password: '</dots_function_call>'	1	/python/ sglang/srt/ function_call/ dots_detector. py:46
LOW	bandit	Possible hardcoded password: '< start >assistant< channel >commentary'	1	/python/ sglang/srt/ function_call/ gpt_oss_detec tor.py:30
LOW	bandit	Possible hardcoded password: '</think>'	1	/python/ sglang/srt/ function_call/ deepseekv41_d etector.py:20
LOW	bandit	Possible hardcoded password: 'token_topk'	1	/python/ sglang/srt/ layers/moe/ token_dispatc her/ base.py:295
LOW	bandit	Possible hardcoded password: '<dots_function_call>'	1	/python/ sglang/srt/ function_call/ dots_detector. py:45

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	bandit	Possible hardcoded password: '< video >'	1	/python/ sglang/srt/ multimodal/ processors/ glm4v.py:456
LOW	bandit	Possible hardcoded password: '< END_TEXT >'	1	/python/ sglang/srt/ parser/ reasoning_parser.py:1773
LOW	bandit	Possible hardcoded password: 'avg_pool_2d'	1	/python/ sglang/srt/ models/ phi4mm.py:103
LOW	bandit	Possible hardcoded password: '<pad>'	1	/python/ sglang/srt/ multimodal/ processors/ pixtral.py:29
LOW	bandit	Possible hardcoded password: ']<]image[>['	1	/python/ sglang/srt/ multimodal/ processors/ minimax_m3_vl.py:183

Showing the 60 most significant of 187 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.