LLMScan.online

AI Model Security Scan Report

C

Score 64/100

Report summary

GGUF v3 model | architecture: llama | quantization: Mixed/Unknown | 0 tensors | 2 blocking issues / 2 warnings

Model

Filename	`tokenizer-tamper.gguf`
Format	`gguf`
Architecture	`llama`
Size	`280 B`
SHA-256	`74b288ed732cfc5d63f787a9378176be2e9b77b207362d343d53f5175e1cef8c`
Scan mode	`static-deep`
Scanned at	`2026-08-14T23:56:26+00:00Z`
Report ID	`c849a76a7f3a472fbf47f8c332a7f052`

Findings (4)

SEVERITY	CATEGORY	DETAIL
HIGH	Tokenizer Integrity	**Duplicate vocabulary entries (1 token string(s) claim multiple IDs)** Examples: [('hello', 3, 6)] (token, first id, duplicate id). A correctly produced vocabulary is a 1:1 mapping between string and ID; a repeated string at two IDs can make a detokenizer collapse two distinct tokens into one, or let a crafted ID silently decode to a different string than the one that was encoded. **Recommendation:** Reject this file; re-export the tokenizer from a trusted source
HIGH	Tokenizer Integrity	**Hidden control/bidi-override characters in vocabulary (1+ found)** Examples: [("'ignore\\u202eprevious'", 'bidi-override')]. Unicode bidi-override characters (e.g. U+202E RIGHT-TO-LEFT OVERRIDE) and raw control bytes render invisibly or reorder surrounding text wherever a token is displayed -- chat UIs, terminals, logs -- without changing what the model itself processes. **Recommendation:** Reject this file; sanitize or reject vocabulary entries containing bidi-control or non-printable characters before rendering them anywhere
LOW	Supply Chain	**No license field in GGUF metadata (general.license)** Absence of license information makes it impossible to determine usage rights **Recommendation:** Add general.license to model metadata before distributing; use an SPDX identifier
LOW	Supply Chain	**No embedded provenance metadata** None of general.author, general.source.url, general.base_model.0, or general.source.huggingface.repository are embedded in the GGUF file. Without any origin signal the model cannot be traced to a trusted source or compared against a reference checkpoint for tampering detection. For files uploaded directly, provenance is especially critical because the scanner cannot infer repository origin from the file alone. **Recommendation:** Add at least one of: general.author, general.source.huggingface.repository, or general.source.url

Runtime compatibility

RUNTIME	SUPPORTED	NOTES

llama.cpp	Yes	GGUF v3 container: supported | quant Mixed/Unknown: supported
Ollama	Yes	GGUF via llama.cpp backend: container and quantization supported
LM Studio	Yes	GGUF container and quantization supported; GPU/CPU inference
vLLM	No	GGUF not supported; requires safetensors format
HuggingFace Transformers	No	GGUF not supported; requires safetensors or PyTorch checkpoint

Generated 2026-08-21 06:51 UTC by LLMScan.online (engine v2.0.0, report schema v1.0). Full interactive report: https://llmscan.online/report/c849a76a7f3a472fbf47f8c332a7f052 — Static analysis only; not a guarantee of safety. See llmscan.online/docs/how-it-works for methodology.