LLMScan.online

AI Model Security Scan Report

B

Score 79/100

Report summary

GGUF v3 model | architecture: llama | quantization: F32 | 9 tensors | 1 blocking issue / 2 warnings

Model

Filename	`zero-dim-tensor.gguf`
Format	`gguf`
Architecture	`llama`
Size	`5.2 KB`
SHA-256	`edb030d10e550cbdaf6ab0cac6e476a403f9ec97f0594452f0726d9d5cc68958`
Scan mode	`static-deep`
Scanned at	`2026-08-10T19:43:43+00:00Z`
Report ID	`d0d741adf7a04b6e93bd5f9ca6668f2f`

Findings (3)

SEVERITY	CATEGORY	DETAIL
HIGH	Tensor Integrity	**Tensors with zero dimensions: ['blk.0.ffn_gate.weight']** Zero-dimension tensors are invalid and may cause runtime crashes or exploits **Recommendation:** Re-quantize the model from source; this may indicate intentional corruption
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 F32: 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