What is ML data-loader injection?

ML data-loader injection is abusing the configuration or metadata of a machine-learning dataset or model file so that parsing it runs code or reads files. Dataset configs can hold template expressions (Jinja2 SSTI), external file references (HDF5 external storage), or protocol specs (fsspec reference paths) that a loader evaluates on load. Because ML pipelines fetch and parse untrusted artifacts automatically, loading one becomes an execution or file-read primitive. Defend by disabling evaluation and external references, sandboxing parsing, and preferring safe formats.

HOW IT WORKS

01 How it works and example

Several vectors share this shape. Template expressions in a dataset config reach the Python object graph and run code (Jinja2 SSTI). An HDF5 dataset configured for external storage points at a local path and returns that file's contents (HDF5 external storage file read). A reference-protocol spec with manipulated offset fields injects a payload into the loader (fsspec reference abuse). All fire the moment an unsuspecting pipeline loads the artifact.

SOURCES

[1] MITRE ATLAS, Adversarial ML tactics

[2] HuggingFace, Agent intrusion technical timeline

Test whether your AI stack survives real attacks.

securelayer7.net/learn/ai-security/ml-data-loader-injection

Open online