The AI Odyssey

When Your Model Is the Trojan Horse

Rahul Sharma
Senior Software Engineer | Red Hat

Christopher Nolan's The Odyssey

The Gift at the Gate

Nolan's movie:
Greeks hide soldiers inside a wooden horse

Hugging Face, 2026:
Attackers hide code inside AI model files

*AI generated image

Agenda

THE TROJAN HORSE

1 How a tiny model file gives an attacker full shell access

HOW IT WORKS

2 Pickle Deserialization

3 Real-world Attack Timeline

4 Why scanners fail

THE FIX

5 Model signing with Sigstore

6 SafeTensors

7 Practical Defense Checklist

GET INVOLVED

8 Open source projects accepting contributions today

Hugging Face: The npm of AI

3,000,000+ models hosted
1,000,000+ datasets
1,400,000+ Spaces

- Hugging Face is the npm of AI and just like npm, it is now a **supply chain target**.
- Anyone can upload. Scanners exist, but **attackers keep bypassing them.**

How Big Is the Problem?

244,000

May 2026

Downloads of a fake OpenAI model before removal

100+

JFrog, 2024

Malicious models with reverse shells

63%

ShadowPickle, July 2026

Scanner evasion rate

The Trojan Model

Live Demo

1. See how PyTorch 2.6+ blocks a malicious `.pt` file by default.
2. See how using `weights_only=False` triggers the malicious payload.
3. The moment the reverse shell gives the attacker access to the system.

⚠️ Everything runs locally; no network exploits

What Just Happened?

```
# The model file contained a hidden
python class:

class TrojanModel:
    def __reduce__(self):
        return (os.system,
("reverse_shell_command",))
```

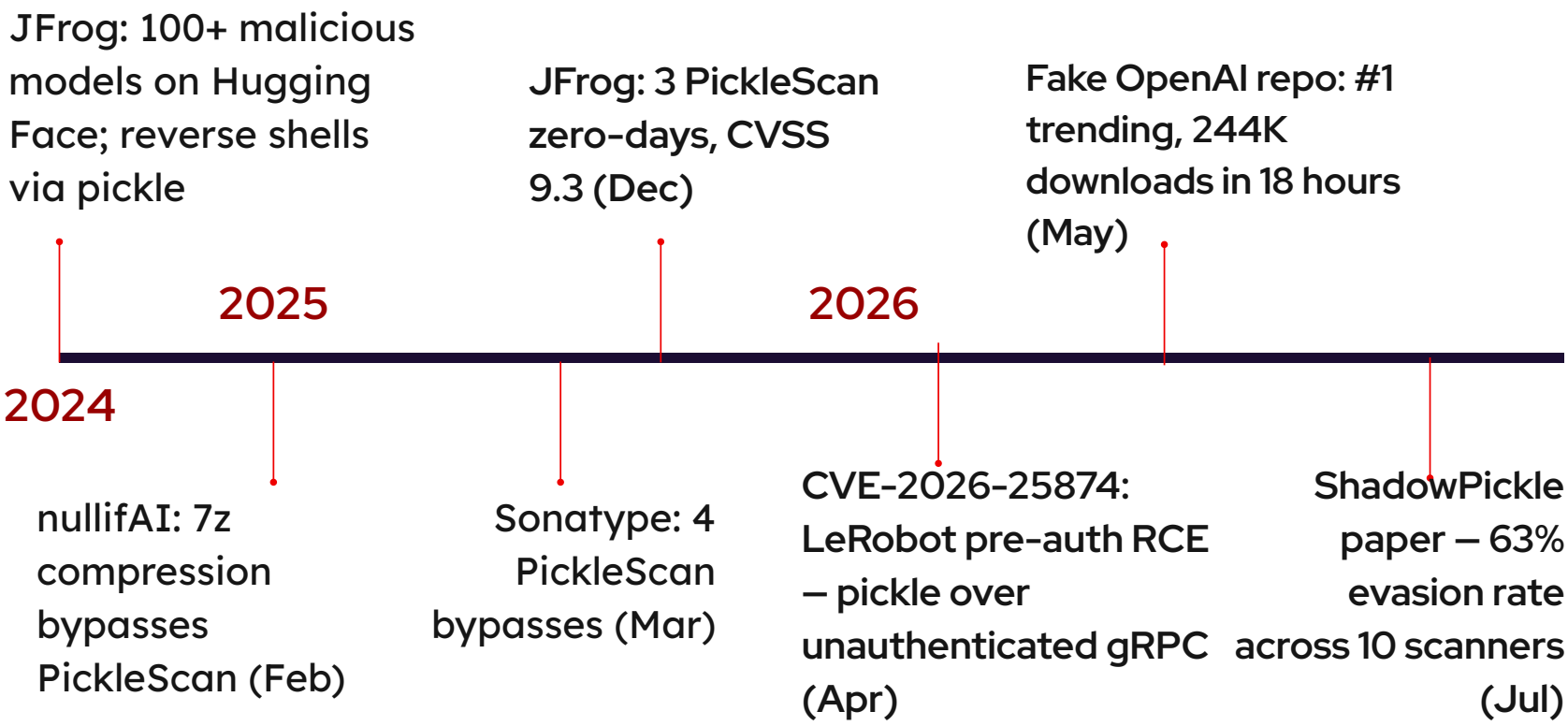
- Small model file
- Few lines of payload
- Full shell access
- No visible errors

Pickle: The Root Cause

⚠️ *This is NOT a bug. Pickle was designed this way.*

- Python docs: "Never unpickle data from an untrusted source."
- But every model download from Hugging Face is data from an untrusted source.

A model file is not just weights. It is a program.



The Attacks Are Escalating

THE CORE PROBLEM

Why Scanners Fail

DEFENSE	WHAT BROKE IT	RESULT
PickleScan blocklist	pkgutil.resolve_name("os:system") indirection (CVE-2026-3490)	11+ RCE chains, all report CLEAN
weights_only=True	SETITEM opcode on Tensor → heap corruption (CVE-2026-24747)	RCE even with the "safe" flag
ShadowPickle (July 2026)	Overwrites whitelisted modules during deserialization	63% evasion across 10 scanners

- Pickle is a virtual machine. Every patch → Attackers find a new bypass.
- Both the framework defense and the scanner were broken in 2026.

BETTER SCANNING IS NOT THE ANSWER

The Real Fix: Two Paths Forward

ELIMINATE THE FORMAT

- SafeTensors or ONNX
- Removes the entire attack surface.
- No pickle = No exploit

VERIFY THE AUTHOR

- Use OpenSSF Model Signing (OMS) + Sigstore.
- Works with any format.
- Sign before you share · verify before you load
  Signed = Trusted

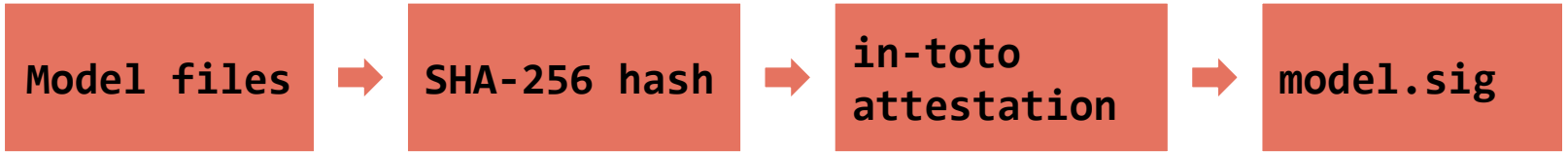
What is Model Signing?

The same trust chain you use for containers - applied to AI

```
Container images →  sign  →  verify  →  deploy
trusted
AI models        →  sign  →  verify  →  load trusted
```

- OpenSSF Model Signing (OMS = Open Model Signing)
- Formalized June 2025
- Backed by Google · NVIDIA · HiddenLayer · Red Hat

How OMS Works



Note: Recorded in Rekor transparency log

Key Insights:

- Sigstore handles keys automatically (OIDC = login with Google/GitHub)
- Signature is detached; lives alongside model as model.sig
- Verification needs no network access to the original signer

Three Commands. That's It.

Sigstore handles everything.

```
$ pip install model-signing

$ model_signing sign ./my-model

$ model_signing verify
./my-model --signature
model.sig
```

- No Key Management
- No GPG
- No Certificate Rotation

Sign >> Tamper >> Catch

Live Demo

1. Sign and verify a SafeTensors model before deployment.
2. Confirm model integrity with successful signature verification.
3. Tamper with the model and trigger a signature mismatch.
4. Verify again; tampering caught at the gate

No pickle = No exploit

SafeTensors: Kill the Attack Surface

	Pickle	SafeTensors
Code execution on load	YES ⚠️	NO ✅
Arbitrary Python objects	YES	NO
Zero-copy loading	NO	YES
Governance	Python stdlib	PyTorch Foundation

SafeTensors joined the PyTorch Foundation in April 2026.
It is now the officially recommended format for model distribution.

LEARNINGS

What We Should Do Today

1. Use `SafeTensors` for all new models
2. Use `torch.load(..., weights_only=True)` at minimum
3. Sign models before sharing: `pip install model-signing`
4. Verify signatures before loading any model
5. Pin models by commit hash, not branch name

LEARNINGS

What We Should Never Do

1. Never `torch.load()` on untrusted files without `weights_only=True`

2. Never `set trust_remote_code=True` without reading the code.

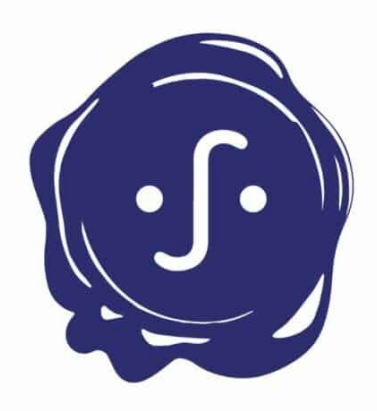
   This literally means 'download and execute arbitrary Python code from this repository.'

3. Never load pickle models outside the sandbox.

   Use containers or VMs for legacy pickle files.

ALL OPEN SOURCE

Get Involved



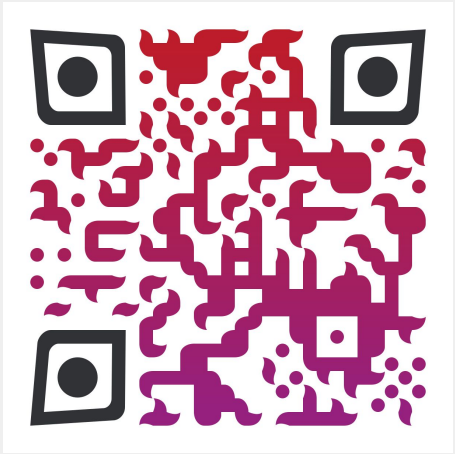
MODEL SIGNING
The signing spec and CLI tools

SAFETENSORS
The safe model format; now a PyTorch Foundation project

FICKLING
pickle decompiler by Trail of Bits

Thank You.

Questions? Drop them in the chat!



https://bit.ly/oosc-ai-odyssey

Scan to access resources & references from this talk.