ModelArmor

Cloud Security for AI/ML Workloads

An Open-Source Sandbox for Securing Untrusted PyTorch, TensorFlow, NVIDIA, and JupyterHub Workloads powered by KubeArmor BPF-LSM.

The Challenge: Securing AI/ML in the Cloud

Modern AI/ML development and deployment face critical security vulnerabilities.

Risk Level: **High Risk**

Pickle Module Vulnerability

Python's pickle module poses a significant security risk, potentially allowing arbitrary code execution.

Risk Level: **36%**

Adversarial Attacks

36% of AI systems face compromised outcomes due to adversarial data manipulation.

Risk Level: **Critical**

Exposed GPU/CUDA Resources

Unauthorized GPU toolkit access remains a top concern in high-performance computing environments.

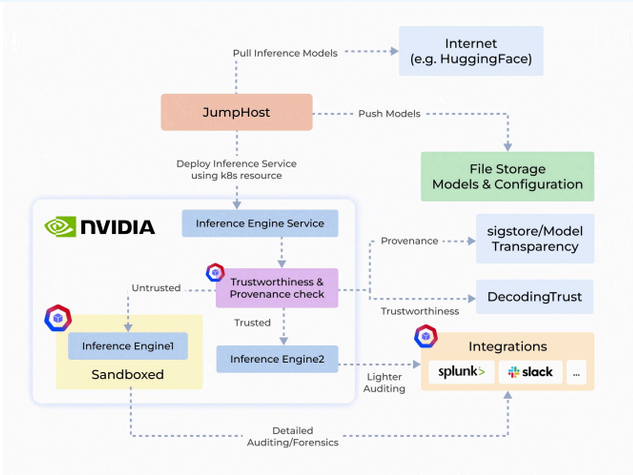
Risk Level: **80%**

Container Breaches

80% of organizations using containers face misconfigurations that lead to vulnerabilities.

The Solution: ModelArmor

Secure isolation for AI/ML workloads with KubeArmor sandboxing



Secure TensorFlow & PyTorch

Isolated execution for TensorFlow and PyTorch models.

Container Hardening

Prevents vulnerabilities in sandboxed container environments.

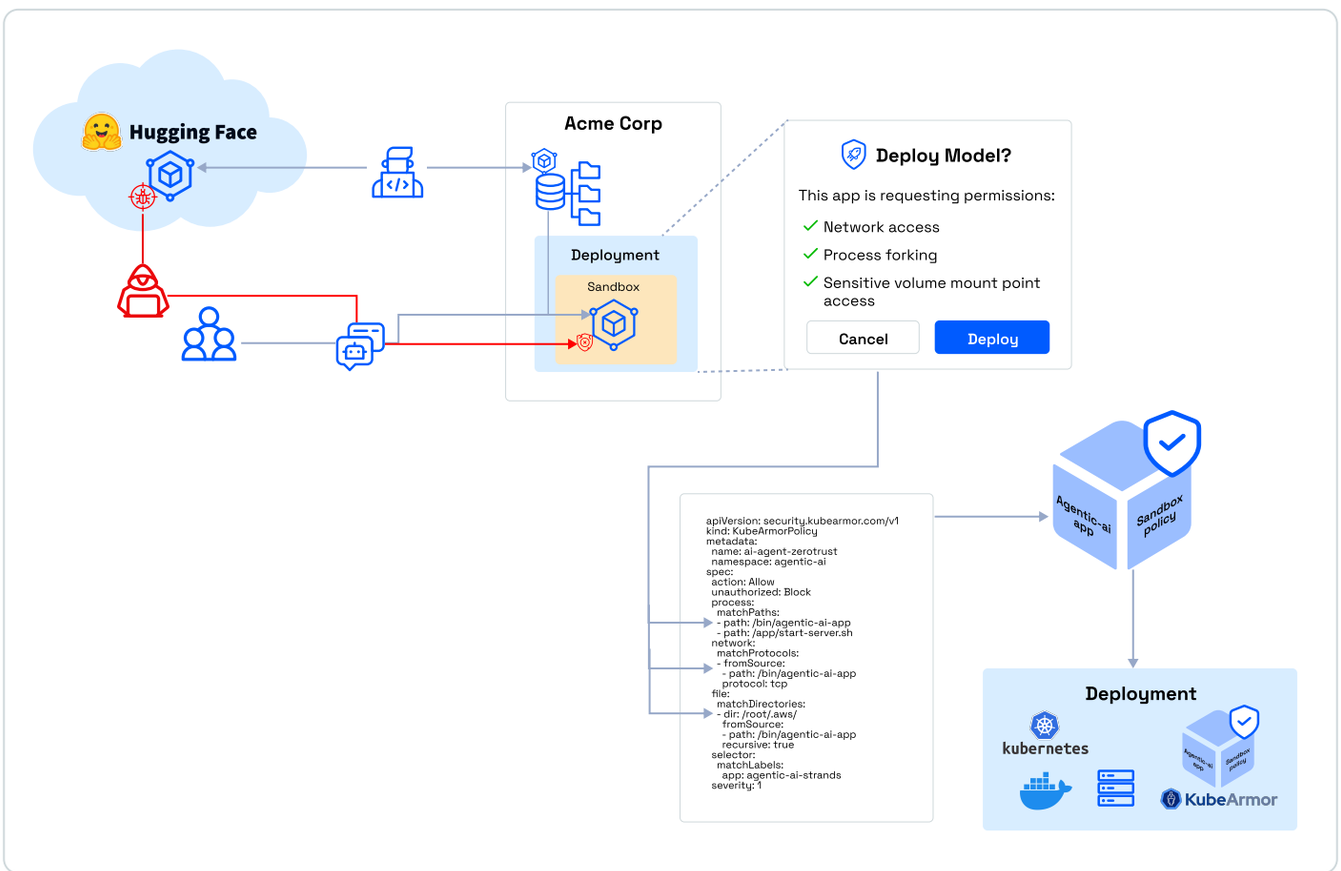
Sandboxed Testing

Ensures untrusted applications execute securely.

GPU/CUDA Security

Secures NVIDIA GPU toolkits from unauthorized access.

Deployment Architecture



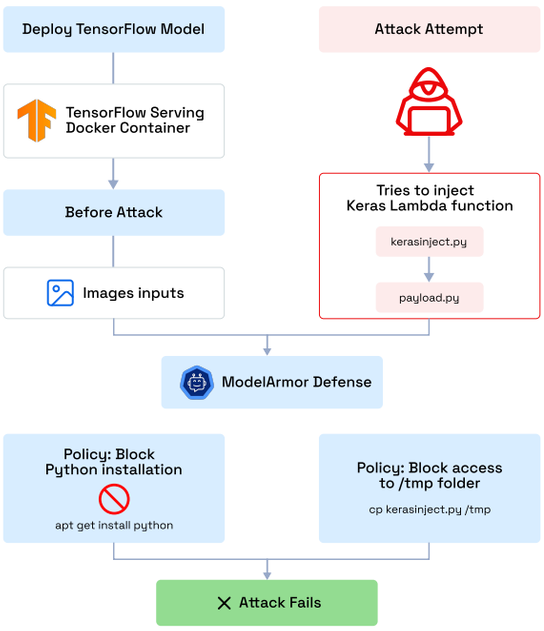
ModelArmor

Cloud Security for AI/ML Workloads

An Open-Source Sandbox for Securing Untrusted PyTorch, TensorFlow, NVIDIA, and JupyterHub Workloads powered by KubeArmor BPF-LSM.

ModelArmor in Action

Defending Against Model Corruption Attack



Without ModelArmor

⚠️ Attacker installs Python inside the container to execute malicious scripts.

⚠️ Copies kerasinject.py into the /tmp folder without any restriction.

⚠️ Injects a Keras Lambda function into the model, enabling arbitrary code execution.

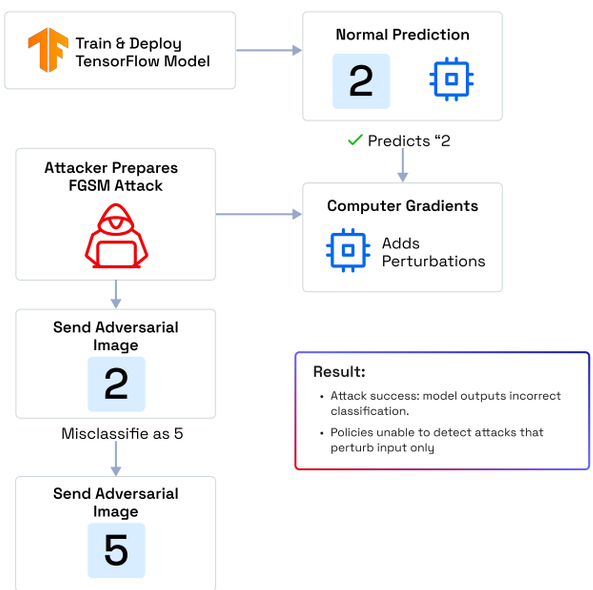
With ModelArmor

✅ Policy blocks installation of Python binaries inside the container.

✅ Prevents access to sensitive directories like /tmp, stopping the upload of malicious scripts.

✅ Model corruption is mitigated as the attack chain is broken at multiple points.

FGSM Adversarial Attack on a TensorFlow Model



Without ModelArmor

⚠️ Attack success: model outputs incorrect classification.

⚠️ Policies unable to detect attacks that perturb input only.

With ModelArmor

✅ Proactively simulates adversarial attacks using *Automated Red Teaming*.

✅ Secures model behavior with input validation and anomaly detection, akin to an *LLM Prompt Firewall* for ML workloads.

✅ Protects against sophisticated input-level manipulations.

Sandboxing Agentic AI

Isolates untrusted models, blocking unauthorized actions

Zero Trust Policies

Granular process, network, and volume controls

Lightweight

No MicroVM overhead, supports any framework

NVIDIA GPU Protection

Prevents unauthorized access to GPU toolkits.

Blocks Package Installs & Network Call

Block /tmp access. breaks attack chains & blocks package installations like nmap & python

Sandboxed untrusted model execution

Untrusted code execution and file access restricted