Zero Trust Cryptographic Confinement for AI Agents

Project Proposer: Charles Munson, Charles.Munson@ll.mit.edu
Open Source: Yes
Mentors:

- Charles Munson, Charles.Munson@ll.mit.edu

Preferred Experience:

- Required (at least one team member): Proficiency in Python, Go or Rust (languages used heavily in Keylime and systems engineering)
- Required (at least one team member): Basic understanding of Public Key Infrastructure (PKI) and digital signatures
- Valuable: Experience provisioning and managing Virtual Machines (VMs) or bare-metal instances on cloud platforms
- Nice to have: Experience working with the Mass Open Cloud (MOC); familiarity with remote attestation frameworks (Keylime)

Project Background:

As Artificial Intelligence agents are increasingly integrated into production environments, granting them the ability to execute real-world workloads, traditional security boundaries are proving insufficient. AI models can use "semantic manipulation" (e.g., prompt injection or context hijacking) to bypass simple API gateways and static access control policies (ABAC/RBAC).
To safely deploy AI, we need a cryptographically enforced "Actuator Space" – an isolated execution environment where an AI cannot independently execute workloads. Instead, actions must be authorized by a distributed human-in-the-loop quorum and executed in a highly attested environment. This project aims to build a prototype implementation of this concept using existing open-source and local academic cloud infrastructure.

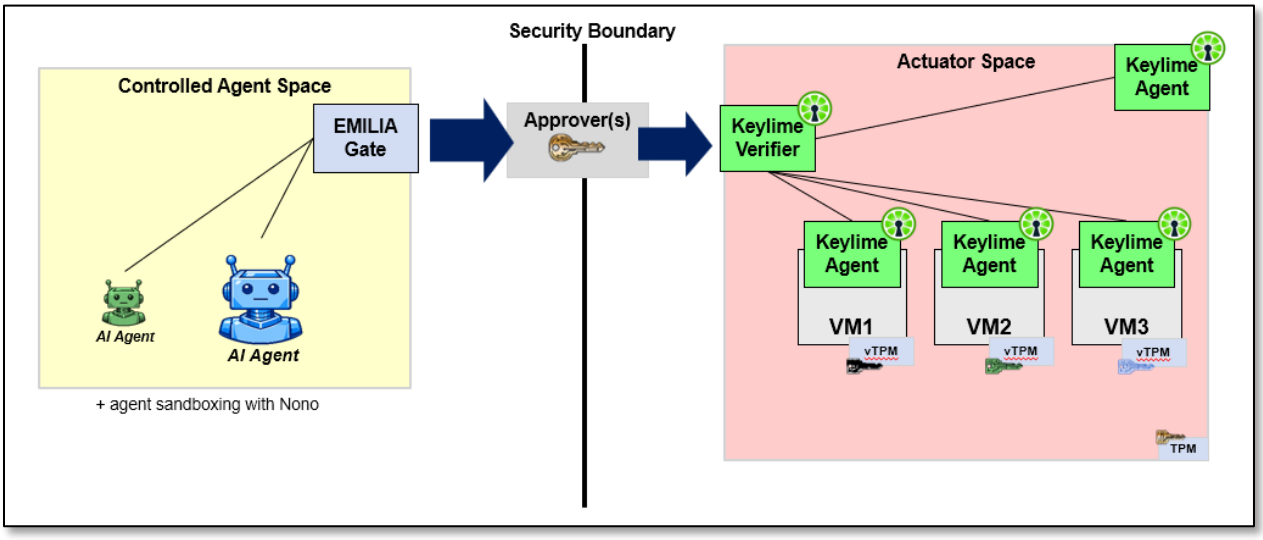
Project Description:

The student team will architect and build a working prototype of an Agentic AI Zero Trust pipeline. The core objective is to integrate several existing technologies to prove out a secure handoff from an AI agent to an isolated, attested execution environment.
Because Keylime relies on hardware/hypervisor-level TPMs (Trusted Platform Modules) and cannot currently attest inside standard containers, workloads will be deployed onto dedicated Virtual Machines (VMs) or bare-metal instances leveraging the Mass Open Cloud (MOC).
The pipeline will consist of:

1. Agent Confinement: Utilizing tools like Nono to sandbox an AI agent attempting to request a workload execution.
2. Distributed Authorization: Leveraging the EMILIA protocol (EMILIA Gate). When the AI requests an action, EMILIA Gate will require a multi-signature (multisig) from a set of authorized administrators. The action proceeds only if a designated threshold of approvers (a quorum) authorizes the action by signing.

3. Attested Execution (Keylime): The approved quorum acts as the "Tenant" for Keylime. Keylime will perform remote hardware/OS-level attestation on a target VM/bare-metal node (e.g., spun up within the Mass Open Cloud). Once attestation is successful, Keylime will securely bootstrap and deploy the monitored workload on behalf of the AI agent.

The final deliverable should be a functional, end-to-end demonstration where an AI agent's request is paused, cryptographically signed by a multisig quorum, and securely deployed onto an attested MOC (or similar) instance.



Learning Outcomes:

Students will gain highly relevant, hands-on experience in cutting-edge cloud security paradigms. Specific skills include:

- Designing and implementing Zero Trust Architectures (ZTA)
- Applying cryptographic concepts (multisig, PKI) to distributed systems
- Working with Remote Attestation and Trusted Execution Environments (using Keylime and TPMs)
- Hands-on cloud engineering utilizing the local Mass Open Cloud (MOC) Alliance infrastructure

Resources:

- EMILIA Protocol
  - https://www.emiliaprotocol.ai
  - https://github.com/emiliaprotocol/emilia-protocol/
- Keylime: https://keylime.dev
- Nono: https://nono.sh
- MOC: https://massopen.cloud

This material is based upon work supported by the Department of the Air Force under Air Force Contract No. FA8702-15-D-0001 or FA8702-25-D-B002. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Department of the Air Force.