AISC 11 Write-Up

Agentic AI Risks Induced by System-Level Misalignment

Summary

We propose a project that probes AI risks induced by **System-Level Misalignment** in agentic LLM systems i.e., exploiting the agent's model and its interactions with scaffolding/operational environment. Our goal is to study how misalignment and weak security controls reinforce each other, creating bidirectional failures where misaligned agents introduce new attack vectors and agentic system design introduce new attack surfaces. The project progresses along different streams/ideas. Together, these efforts aim to formalize threat models, validate them experimentally, and deliver practical mitigations. This work addresses an under-studied risk vector that complements user- and model-level alignment research and potentially lays out a subfield.

Non-Summary

Motivation

Most alignment work focuses on user-level (bad prompts, jailbreaks) and model-level (objective specification, reward-shaping) failures. However, agentic systems don't operate in silos, they are not just models. They are still software that operate as services with software tools, memory (known as agent scaffolding), network, container, and system privileges and user interaction. This enables **system-level failure modes** (insider-like actions, privilege escalation, exfiltration to attacker owned assets) that are not mitigated by standard RL/robustness research. Our previous work in AI Safety Camp classifies these types of misalignments. The system-level failure modes and addressing AI risks have **pronounced risk implications relevant to the cybersecurity domain.**

Table 1. A Classification of Misalignment Types in Agentic AI Systems.

Misalignment Type	Description	Example in This Paper
User Misalignment	Occurs when the user intentionally or unintentionally requests a harmful or disallowed action.	A user deliberately attempts to "jailbreak" the system or requests instructions for an illicit activity.
Model Misalignment	Arises when the AI model itself errs and disregards a critical safety or preference constraint provided by the user.	The AI recommends a food item containing a severe allergen to an allergic user.
System Misalignment	Refers to flaws in the broader operational environment that permit unsafe behavior, such as with third-party tools.	An agent inadvertently disclosing sensitive financial information to a malicious website or tool.

We aim to **meet the moment by grounding AI safety research in messy, real-world deployments** where agentic systems already operate and fail. While the broader community continues important work on eliciting model capabilities and red-teaming for failure modes, we are motivated by the **lack of attention to the software engineering layer** and the **scarcity of actionable blue-team tools** that help defenders secure agentic systems in practice.

Project Plan

Our main research questions (RQ) aim to study AI risks through the following lens:

1. How lack of capable security controls create new attack surfaces in agentic systems
2. How misalignment with the deployer's intent creates security attack vectors and capabilities
3. Identifying defense mechanisms

Stream 1: Research attack and defense mechanisms for system misalignment threat models that are proven and where risk has materialized

Metric	Finding
Organizations adopting LLMs	98%
LLMs in customer-facing apps	75%
MCPs in customer-facing apps	47%
LLM security fully deployed	54%
MCP security currently onboarding	**24%**
API security currently onboarding	26%
Security cited as top adoption barrier	49%
Organizations refusing AI adoption	2%

This industry survey shows deploy first, secure later behavior. Security controls for AI systems are lagging AI deployments, indicating that the new attack surfaces are ripe for exploitation. A recent Comet prompt injection incident showed how malicious web content could trick an agent into exfiltrating user credentials—illustrating how system components introduce new attack surfaces and cause immediate security risks. At the same time, most defensive security measures still revolve around making models more robust and/or agent monitoring/observability tools rather than research on reducing security risks at the agentic system level.

Stream 1 focuses on identifying and testing new classes of attacks unique to emerging agentic AI systems, along with evaluating defenses against them using real-world tools like Comet, Atlas, or their successors. Our goal is to develop and integrate defense mechanisms into

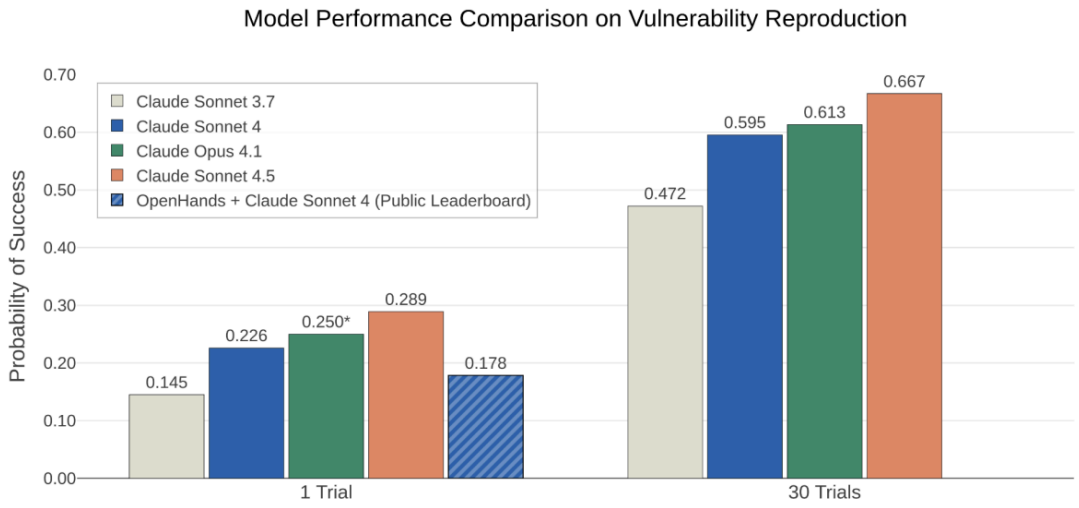
popular frameworks (e.g., LangGraph, CrewAI) to make agentic misalignment defenses both accessible and reliable.

RQ:

1. Are there new classes of attacks in emerging agentic systems architecture that add to the existing taxonomy in this survey paper?
2. How prevalent are established threat models in existing and emerging agentic architectures across different vendor's agentic offerings? The focus will be on agent2environment, agent2memory and action risks.
3. "Attacker Moves Second" says that none of the current state-of-the-art defense mechanisms work against an adaptive attacker, particularly humans. How do existing/newly proposed defense mechanisms hold up against attacks on external interfaces in emerging agentic scaffolding and architectures?
4. What is preventing the wide-scale adoption of these defense mechanisms? Is it possible to democratize defense mechanisms by incorporating them in popular agentic frameworks? What are the challenges in this?

Deliverable: A library/patch that implements techniques like CaMeL to democratize defense mechanisms in popular frameworks like Langgraph, CrewAI or propose new defense mechanisms

Stream 2: Research defense mechanisms for near-term imminent system misalignment threat models where risk is imminent but not materialized yet.



OpenAI's latest system card reports model cyber-capability results without browsing or tool access, likely understating real-world risk, since deployed agents routinely operate with browsing, shell, or API privileges. In contrast, Anthropic's latest system card, which tests agents with tools in realistic environments, demonstrates that models already achieve 30 % success in a single trial and 60 % after 30 trials on capture-the-flag–style exploitation tasks. At the same time, empirical data confirm that LLM-generated code remains insecure at scale where 62 % of

code produced by state-of-the-art models contained at least one security vulnerability. This indicates short-term asymmetry where the ability to exploit code vulnerabilities is increasing, but the ability to write secure code is lagging.

Redwood Research's blog warns that a misaligned model (regardless of internal or externally induced misalignment) may cause subtle regressions in security posture cascading through production systems that traditional alignment research does not yet address. If infrastructure is compromised, it is difficult to guarantee the security of layers built on top of infra

In Stream 2, the focus is AI Control, to build systems that remain secure despite the agent itself being untrustworthy. **Treat all code (AI or human-authored) as untrusted until proven secure,** build mechanisms that continuously detect and block posture regressions at commit time.

RQ:

1. Which static and dynamic signals reliably indicate a commit changes security posture from restrictive → permissive
2. Can semantic/taint-flow diffs detect regressions where sanitized inputs become unsanitized or sensitive sinks are newly reachable
3. How can we quantify and compare threat-model states before and after a commit to measure whether a code change increases or decreases overall system risk?

Deliverable: Drawing inspiration from OpenAI's Aardvark, we will prototype a code delta control component to identify the changes in security posture as an AI control mechanism

Team Composition

Team size: 4-8 (excluding leads)
Required average weekly commitment 10 - 15 hours
Team Member Roles Needed: AI Safety Camp is a great way to get familiar with the tools and technology and get some research experience in a new field in a supportive environment. Even if you don't have the following skills, the willingness to commit to learning and actively contributing to the project is more valuable! We do expect everyone to be able to code hands-on in any one language

	Expertise & Responsibilities	Key Tools / Skills	This is you if…

Software Developer/AI Engineer	Builds and maintains the experimental infrastructure, integrates model APIs, manages version control, and ensures the codebase is functional and operational per developer guidelines.	Python, FastAPI/Flask, Git, Docker, Kubernetes, CI/CD, REST APIs, LangChain, CrewAI, LangGraph, OpenRouter API, LangSmith, MCP Registries	You have hands-on experience slinging code (language-agnostic) and enjoy bringing experimental ideas to life. You're open to learning new tools and frameworks, balance "vibe coding" creativity with responsibility for production-ready systems, and take pride in clean, maintainable code.
AI Alignment Researcher (RL Focus)	Designs experiments to study alignment failures, develops reward modeling and has experience in building verifier agents with variable rewards	PyTorch, JAX, HuggingFace, Gymnasium, Weights & Biases, reward modeling frameworks.	You have a solid grounding in reinforcement learning, are curious about how optimization pressure produces misalignment, and want to design experiments linking behavior to security outcomes.
Data Scientist	Analyzes model outputs, quantifies behavioral drift, builds dashboards for evaluation metrics, and curates datasets for reproducible experiments. Provides statistical evidence for failure patterns and mitigation effectiveness.	Python, Pandas, NumPy, SQL, Jupyter, Seaborn/Matplotlib, MLflow, Streamlit, scikit-learn.	You love finding patterns in messy data, building metrics that make misalignment measurable, and visualizing risk or security-posture trade-offs in intuitive ways.

Security Engineer	Designs and validates threat models, secures experimental pipelines, identifies vulnerabilities in agentic workflows, and hardens environments against subversion or misuse. Advises on Secure Software Development Lifecycle (SSDLC) practices.	OWASP ZAP, Burp Suite, IaC scanning tools.	You've worked with or have exposure to SSDLC and production security. You think adversarially, are curious about how AI systems fail under pressure, and want to explore the intersection of AI security and AI safety

Team Leads	Background	Skills and Role in the Project
Evan Harris	Began white-hat hacking in 2025 with a focus on MCP servers, leading to multiple vulnerability disclosures. Professional software developer since 2018, initially studying genomics and unconscious processing before transitioning into programming through self-study and a 2017 bootcamp. Passionate about using agentic systems for responsible vulnerability detection and disclosure. Connect: Follow on Twitter	Full-stack developer and security researcher with strong hands-on coding experience. Will lead **agentic defense prototyping and automation**, focusing on agents that detect and report vulnerabilities responsibly, file coordinated disclosures, and share defensive software patterns. Committed to contributing **12+ hours weekly** (approx. 2 hrs/day, Mon–Sat) and participating in hackathons.
Preeti Ravindra	9 years of experience at the intersection of **AI and cybersecurity**, specializing in applied research that transforms emerging ideas into production-grade security solutions. Has worked on securing AI data centers and addressing real-world risks similar to those explored	Applied machine learning engineer in security. Will direct **research direction** ensuring the project bridges AI Safety and AI Security. Oversees coordination across technical streams and dissemination of findings to the broader

in this project. Holds multiple publications and patents in AI Security.

research community. **Committed to contributing 10+ hours weekly**

Connect via Website

Team Culture

What our groupmates will bring to the table:

1. Have an active voice in the research direction
2. Scoping and hands-on implementing experiments with high level direction from leads.
3. They will make good tradeoffs on software choices and be creative in utilizing datasets

What our groupmates will get from us

- Mentoring in AI+security/adversarial mindset
- Guidance, Project Management and Conflict Resolution
- Hands-On debugging to some extent

Groupmates will thrive if they enjoy this kind of work environment:

- Ready to be hands-on in coding
- Not afraid to ask questions.
- Willing to share perspectives and making good technical arguments
- Willing to be challenged and explore away from your comfort zone
- High agency and ownership

What this project is not:

- A theoretical project
- Something to coast through while being non-committal on deliverables

By the end of the project, our hope for you is that you walk away with a significant understanding of risks of AI systems and the practical experience of bringing an AI agent for defensive measures online.

Theory of Change

Goals (success criteria)

- Deliver a toolkit to the community to democratize evaluating agentic stacks and mitigate system-level risks.
- Develop and validate AI-control mechanisms that detect and block security-posture regressions in automated code generation and infrastructure updates

- Producing empirical research for the exchange of techniques between AI Safety and AI Security to reduce AI risks.

Non-goals

- We will not study multi-agent security or coordination.
- We will not attempt live attacks on third-party production systems.
- We will not release any exploit scripts that enable real-world compromise without mitigations; red-team artifacts will be carefully sanitized and reviewed by an ethics board.
- We will not attempt to “solve” model-level alignment generally; our focus is on system interactions, operational controls, and applied defenses

This project aims to generate actionable evidence and build deployable defenses for both present and near-future system misalignment risks. In the short term, we empower practitioners to evaluate, patch, and harden existing agentic frameworks through accessible blue-team tooling.In the longer term, we lay the foundation for secure autonomy—developing methods to quantify and prevent posture drift when AI systems themselves author or deploy code.

As a secondary goal, we hope this experiment can provide evidence that is legible and persuasive to researchers across both AI safety and applied security communities. In a spirit of (adversarial) collaboration, we aim to engage alignment researchers, ML practitioners, and systems security engineers in critiquing our threat models and registering predictions about experimental outcomes.

Timeline

The team is split up into groups of researchers and each group tackles one of the streams. This is the “happy path” in execution

Month One: Setup and Design

- Team building, Onboarding, Refine the overall project scope, Rapid experimentation for selecting top 2 experiment designs

Month Two: Execution and Mitigation

- Red Team Phase to execute attack scenarios, Blue Team phase to design/implement proposed mitigations

Month Three: Evaluation, Synthesis and Dissemination

- Synthesize findings from experiments, documentation for tools, feedback from users, AISC presentation prep.

Failure Modes

- **Failure to Reproduce the Threat:** The chosen agentic system may be too unsophisticated (fails to follow instructions) or unexpectedly robust (immune to our injection). We could spend weeks setting up an experiment only to find the core premise doesn't work, yielding no useful data on mitigations.
- **Countermeasure:**
    - **Diversify and Timebox.** In Month One, each pair will prototype their attack against at least two different open-source agentic frameworks. We will timebox this initial "vulnerability discovery" phase to two weeks and proceed with the most promising target.
    - **Eval awareness:** Systems driven by more powerful models will demonstrate awareness that they are being tested. This could lead to intentional sandbagging when monitoring is being performed.
    - **Use Smaller Models for Proof of Concept.** Assuming there is not a reliable counter measure within state of the art models to significantly decrease eval awareness, then smaller models can be used for initial experiments.
- **Lack of publicly available datasets for empirical research:** Lack of diverse, high-quality datasets skews results toward narrow environments or benign behaviors, underrepresenting real-world failure diversity.
- **Countermeasure:**
    - Establish collaborative data contribution pipelines with GitHub/open source communities—to continuously expand the dataset's coverage of agentic behaviors and threat models while enforcing strict ethical review and anonymization
    - Use synthetic datasets
- **Ineffective or Impractical Mitigations:** The mitigations we design might successfully block the attack but render the agent useless (e.g., a network filter that blocks legitimate API calls) or have too many false positives, making them impractical for real-world use.
- **Countermeasure:**
    - **Define Mitigation Metrics Upfront.** For each experiment, we will pre-define not only security success (attack blocked) but also a "usability score" or "performance overhead" metric. This ensures we evaluate the mitigation's practicality, not just its power.
- **Flawed Experimental Environment:** A misconfiguration in our sandbox could lead to two critical failures: 1) The environment is "flaky," producing inconsistent results that make our measurements meaningless. 2) A critical vulnerability allows an experimental payload to "escape" the sandbox, posing an operational security risk.
- **Countermeasure:**
    - **Infrastructure-as-Code and Peer Review.** All experimental environments must be defined using IaC tools (like Docker Compose).
- **Team Desynchronization:** With groups working in parallel, there's a risk of environments drifting apart, inconsistent data collection methods, or duplicated effort in

building common tooling. This could invalidate our ability to synthesize a single, coherent taxonomy.

- **Countermeasure:**
    - **Mandatory Weekly Syncs and Centralized Docs.** We will hold a mandatory weekly sync where each pair presents their progress, challenges, and any changes to their environment. All experimental configurations will be managed via version-controlled scripts (e.g., Dockerfiles) in the central GitHub repository.
- **Team Member Withdrawal:** Person 1 of pair A drops out. This could happen for more than one of the pairs. It could happen at any point throughout the course of the project.
- **Countermeasure:**
    - **Flexible Team Composition**. Person 2 of pair A in the above scenario could write up the current position of their experiment, then merge into pair B or C. Two other variations of this are:
    - One or two people from pairs B or C offer partial support to Person 2 in their research path.
    - If Person 2 actually prefers to finish their experiment without bringing in explicit support from another pair (which would be quite sensible if their teammate was to withdraw near the close of their experiment),then this is an additional path.
    - If many people withdraw, then the project scope would be reduced to fit the remaining available time within AISC.
- **Team leader withdrawal:** Either Preeti or Evan has to withdraw for whatever reason.
- **Countermeasure:** Reduce overall scope of project. Perform handoff of any relevant details before complete withdrawal of either team leader.
- **Scope:** The initial exploration phase could unveil that the experimental design is unachievable within the period of AISC.
- **Countermeasure:**
    - **Scoping Down Primary Deliverables.** If completing one experiment and producing a tool as an output for the community seemed like overreaching, then a blog post rather than an open source tool would be made as the end target.

Output

1. The primary goal is creating a code repository of attack and defense mechanisms which can also be a submission to the Call for Tools, Demo Labs, workshops at leading security conferences like BlackHat Arsenal/ DEFCON (GitHub repo)
2. The secondary goal is a paper submission/talk at an AI security conference like IEEE S&P, AAAI AICS, ACSAC, CIKM or equivalent.

Risks and downsides (externalities)

1. **Dual-use publication risk (revealing attack recipes):** publishing precise exploit methods (even sandboxed) can enable malicious actors. *Mitigation:* follow responsible disclosure, withhold or heavily sanitize exploit payloads, coordinate with vendors when applicable, and release mitigations concurrently.

2. **Overconfidence / misapplied hardening:** if mitigations are framed as "silver bullets," product teams might reduce other security hygiene (false sense of safety). *Mitigation:* emphasize layered defense and operational playbooks; publish limitations and non-goals.

3. **Misinterpretation by policymakers or media:** framing could be sensationalized, accelerating poorly informed regulation or bans. *Mitigation:* clear messaging, public FAQ, and collaboration with civil-society and product security teams to contextualize findings.

Acknowledgements

Shout out to Nell Watson for bringing Preeti and Evan together in the 2025 AI Safety Camp, and for demonstrating how to deliver within the container of AISC.