ACHIEVING A SECURE AI AGENT ECOSYSTEM:
A MAP OF OPEN OPPORTUNITIES AND ACTIONS FOR ADVANCEMENT

Nicole Nichols, Palo Alto Networks
Sella Nevo, RAND
Mark Greaves, Schmidt Sciences

June 17, 2025

Acknowledgements

We gratefully acknowledge the following experts for their valuable insights and contributions to the ideas presented in this report: Ofir Balassiano, Sahana Chennabasppa, Walker Lee Dimon, Aaron Isaksen, Christine Lai, Maura Pintor, Phillip Robertson, Christian Schroeder de Witt, Dawn Song, Victor Aranda, May Wang, and Xianliang Zhang. Our thanks also extend to those contributors who preferred to remain anonymous, and to the participants of the Paris AI Security Forum who helped provide early feedback that shaped the workshop. The final recommendations and conclusions presented herein have not been formally peer reviewed and may not reflect the views of every individual consulted. The report is intended to share the discussed themes and solicit informal peer review.

Executive Summary

The rapid development and anticipated proliferation of Artificial Intelligence (AI) agents—systems capable of complex planning and autonomous action in real world environments—present profound and novel cybersecurity challenges. Advancement and adoption is outpacing historic technological shifts like the Internet [13] and are expected to function in roles akin to employees and personal assistants within a year [30]. AI agents are expected to integrate across most vertical and horizontal levels of global society and directly or indirectly touch critical systems in diverse economic sectors. Current cybersecurity paradigms, often reliant on signature-based detection and static role based access controls, are insufficient to address the unique vulnerabilities stemming from dynamic generative agents with opaque interpretability, new protocols connecting tools and data, and the unpredictable dynamics of multi-agent interactions. This makes the prioritization of AI Agent system security essential. Numerous reports and expert discussions have identified the pressing security questions surrounding AI agents. However, it is essential to move beyond reiterating concerns and toward a collaborative, action-oriented agenda to address these risks.

With this in mind, Schmidt Sciences, RAND, and Palo Alto Networks convened an international group of leading industrial and academic researchers to collate and contextualize the fragmented expertise and insights from various individual reports, agencies, and expert domains. Rather than cataloging questions, the aim was to distill this collective intelligence into a coherent set of needs to address the distinctive security concerns that arise in the setting of LLM-driven AI Agents. This report is the outcome, offering an actionable roadmap of sub-problems with clear task directions. It details concrete directions, and design considerations to accelerate the AI security community's efforts towards a secure AI agent ecosystem. The first three sections (Section 1, What is AI Agent Security?; Section 2, Unique Security Implications of AI Agents; Section 3, Selected Related Work) provide some introductory scene setting and related work. The majority contribution of the report is Section 4, which contextualizes the security gaps identified in the workshop. The roadmap is structured by three functional pillars, which are analyzed through a series of security goals and actions for advancement.

The functional pillars and key discussion points are summarized below:

- **Securing AI agents from external compromise**
  To keep AI agents from being attacked and to address the security gaps in AI agents, a "security-first" design approach is crucial. First, we recommend developing a common reference architecture, similar to the OSI model, to define the scope, produce a shared vocabulary, and facilitate a methodical evaluation which can lead to reducing the attack surface of AI agents. Second, exploring Agent OS concepts like secure startup is essential. Other key elements for safe operation include developing secure open-source communication protocols, enhancing agent transparency and interpretability and creating mechanisms for inter-agent trust, such as reliable agent attestation and identity tracking. An AI Agent Software Bills of Materials (SBOMs), will be essential for monitoring supply chain related risks but is co-dependent on creation of robust versioning systems, for the AI foundation model, memory architectures, training data and processes. Lastly, in the event of failure, AI agents should gracefully fail to a known safe-state, while applying methods for agent recovery or disposability after a compromise.

- **Securing the assets and goals the user entrusted to an agent**
  Teams building agents will need adequate tools for robust pre-deployment evaluation techniques to ensure agents respect confidentiality and integrity constraints, as well as to detect and prevent tampering. This is even more paramount when utilizing sensitive information and

approaches like separating data and instruction can aid those robustness checks. A related challenge is the 'abstraction mismatch', where traditional access controls may not effectively manage the nuanced, dynamic behavior of AI agents. This section of the report highlights the necessity of more complex and flexible authorization mechanisms. Furthermore, to ensure oversight and accountability to the end-users, agents must provide understandable and reliable activity traces, which could include tamper-proof logging.

Securing systems from advanced, purpose-built malicious agents
For entities with capacity for multi-year planning horizons, it will be important to periodically re-evaluate shifting risks from potential purpose-built malicious cyber agents. Simultaneously, the research community needs to prioritize high level goals of robust detection and remediation of malicious agents, establishing standardized cybersecurity capability evaluations for foundation models, and fostering AI-specific threat intelligence sharing. Specific technical projects to achieve those goals include leveraging AI for secure code verification, deploying autonomous system monitoring agents, developing agent reputation signatures/risk scores, methods to detect multi-agent collusion, deploying proactive "hunter" agents, and researching agent stealth and evasion capabilities.

Section 4 concludes with a discussion of the cross-cutting need for a standardized testing environment, which was repeatedly emphasized as limiting side-by-side comparisons across multiple research needs.

In Section 5, we summarize the recommendations across all pillars. The autonomy, adaptability, and potential scale of AI agents is fundamentally altering the cybersecurity landscape, creating vulnerabilities in the agents themselves, risks to the assets and goals entrusted to them, and new vectors for malicious actors to exploit. The long-term trustworthiness and widespread adoption of AI agents hinge on the immediate and diligent development of robust security practices. Fostering interdisciplinary collaboration, prioritizing the development of security-centric technologies and standards will be necessary to proactively address the unique risks outlined herein.

Contents

Chapter 1

What Is AI Agent Security?

The term “agent” encompasses software of many levels of sophistication. The most advanced will be AI agents powered by carefully tuned Large Language Models (LLMs) with reasoning capabilities, supported by a diverse range of simpler, task-specific agents and traditional API calls. These sophisticated agents possess autonomy to pursue goals, make and revise plans, and take side-effectful actions in their environment consistent with these goals, including writing code, controlling robots, and accessing APIs and other agents.

A primary question when discussing AI agents is their definition [31]. A single, comprehensive technical definition of an AI agent is impractical for security purposes. Defining an agent purely by behavioral traits (planning, action, memory) would incorrectly include some systems like a reinforcement learning agent to play Atari, which has those behavioral traits, but present significantly different risk profiles. Similarly, defining agents based on specific algorithms or levels of behavioral sophistication would be too difficult to maintain due to the rapid evolution and nuanced derivatives of algorithms, and the inherent subjectivity of "sophistication."

We propose an approximate definition sufficient for scoping the security problem:

> *An AI agent is any compound system with interconnected elements, including one or more foundation language models, operating in coordination, typically with some degree of autonomy, memory, and tool elements, to achieve goals.*

This perspective allows us to abstract the agent to a compositional system. Cybersecurity analysis can then focus on the security of the individual elements (which can be probabilistic, deterministic, semantic, etc.) and the connections between them. The security of the overall system emerges from the combination of these components and connections. This compositional approach offers several advantages:

Comprehensive: Effective regardless of the specific agent definition or sophistication level.

Modular: Adapts easily as AI architectures evolve; security models for new elements can be integrated.

Rapid Assessment: Enables quick evaluation of novel system configurations.

Reusability: Allows leveraging existing security models for non-AI components within the agent system.

A challenge with this compositional approach lies in defining the boundary between what constitutes the "agent" and what is considered its "environment." Perhaps more than other systems,

the security of an agent is mingled with the environment itself. Mis-representation of the environment, whether by elements external or internal to the agent, will impact how the agent represents and perceives the state of the environment, as well as the decision based on those observations. A compositional security method can not infinitely sum over the entire environment, or other agents outside of a prime agent's control. Nonetheless, a boundary is needed to define the security scope. In practice, this may necessitate multiple, nested boundaries to define different trust regimes (e.g., between sub-agents, agents within a user's control, the representation of the environment, and unknown external agents).

When we discuss agent security, we intentionally focus on the novel security threats, introduced by accelerating AI abilities, expanded usage, and the resulting uncertainties. We also incorporate a broad and diverse set of expert perspectives from industry, academia, cybersecurity, and AI research to ensure comprehensive coverage of potential gaps that occur across disciplines.

Chapter 2

Unique Security Implications of AI Agents

Modern cybersecurity tools leverage a variety of signature-based detection techniques for known and novel threat signals, but signatures can be brittle. As the threat landscape evolves, agents will encounter novel security threats and signals, as well as require novel methods of threat detection. We also expect that multiagent systems will generate new opportunities for both cyber attackers and cyber defenders, and give rise to new asymmetries between them. This implies that developing and deploying AI agents will pose a set of unique cybersecurity challenges. Here we highlight some of the open questions which we used in establishing the conceptual map of a secure AI ecosystem and its technical enablement.

There are profound security concerns inherent in these types of LLM-based multiagent systems. AI agents can create new security risks to other elements in the system by autonomously exploiting weaknesses, but they also can be vulnerable to security threats in novel ways. This vulnerability comes from three sources. First, the interpretability of LLM introspection and relative intelligence of models are difficult to quantify and compare. Consequently, model and agent validation are more ambiguous [12]. Second, this currently limited understanding is compounded by potential threats and vulnerabilities created by embedding LLMs into a goal-directed agentic context, allowing them access to other agents, proprietary databases, and external software tools, and including an ability to remember and reason about its state. Finally, the specific context of multiple agents can implicate complex issues in negotiation, trust, calibrated deception, emergent behavior, secret communication, and conflicting goals, all of which can affect the security environment in unpredictable ways.

AI agents will need to be validated to meet standard security goals, such as confidentiality, integrity, availability, privacy, authenticity, trustworthiness, non-repudiation, and auditability. To have secure AI agents, agent developers will necessarily need to adopt best practice information security techniques in the underlying software, such as keeping cryptographic keys confidential.

Current Security Assumptions That Will Likely Break

The introduction of capable AI agents challenges several fundamental security assumptions:

- **Human in the Loop:** Many current security systems rely on the assumption that a human is always in the loop to make critical decisions or detect anomalies. Autonomous agents can operate without human intervention, making this assumption obsolete.

- **Behavior Patterns:** Traditional security models often leverage anticipated behavior from users and systems. AI agents, with their learning capabilities and potential for emergent behavior, can deviate from expected patterns in ways that are difficult to anticipate. We acknowledge that environmental conditions already make modern software difficult to characterize and predict, but embedding AI models in software like agents further increases the existing baseline of complexity.
- **Centralized Control:** Current security paradigms often rely on centralized control and monitoring. In a world with autonomous, distributed AI agents, control will need to be decentralized and dynamic.
- **Static Boundaries:** Traditional security assumes clear and static boundaries between trusted or untrusted environments and internal or external systems. The migration to zero trust architecture is shrinking the space of trusted environments, but there is still a wide gap in maturity of adoption. According to a survey of 4000 IT companies, only 18% have implemented all elements of the zero trust principles [7], [8]. AI agents will further blur these boundaries, as they interact with both internal and external systems, access diverse data sources, and can act on their own in goal-directed ways.
- **Log Reliability:** Security auditing often relies on analysis of logs. Agent generated logs could be misaligned from the actions taken, difficult to interpret, or altered before verification can uncover discrepancies. Sophisticated actors already engage in these behaviors, but speed and scope at which these manipulations can occur will increase substantially.

Broad Implications for Security

These shifts have significant implications for security:

- **New Threat Models:** We need to develop new threat models that account for the unique capabilities and vulnerabilities of AI agents. This includes considering attacks targeting agent autonomy, reasoning, perception, and communication.
- **Dynamic Risk Assessment:** Current improvements to incident response have brought response times down from weeks and months to minutes and hours for many scenarios. These gains alone will not be sufficient to respond to the potential of autonomous AI enabled agents. Risk assessment will need to become more dynamic and real-time, adapting to the ever-changing behavior of AI agents and the evolving threat landscape.
- **Advanced Detection and Response:** Traditional security tools may be insufficient. We will need advanced detection systems that can identify anomalous agent behavior, detect subtle manipulations, and respond quickly to incidents.
- **Explainability and Transparency:** Ensuring explainability and transparency in agent decision-making is crucial for accountability and auditability. Techniques for understanding and interpreting agent behavior will need to be developed.
- **Secure Agent Communication:** Protocols for secure agent communication will be essential, ensuring confidentiality, integrity, and authenticity of messages exchanged between agents.

Chapter 3

Related work

The discussion of threats in the evolving landscape of AI enabled agents is a high priority, and on the minds of the general public as well as the companies eager to demonstrate sufficient reliability to productize the benefits. The objective of this report is not to be a summary of every potential threat, rather to contextualize and synthesize this landscape so that an actionable roadmap of tasking can be created for the security community to collectively pursue. Some of the key related surveys are provided here as general reference. Many other topic specific references are cited within the technical discussions of the proposed actions for advancement throughout section 4.

- OWASP Top 10 of Agents [17]
- How to Secure Custom Built AI Agents [35]
- AI Agents Under Threat: A Survey of Key Security Challenges and Future Pathways [16]
- Firewalls to Secure Dynamic LLM Agentic Networks [10]
- SHIELDAGENT: Shielding Agents via Verifiable Safety Policy Reasoning [14]
- Towards Building Safe and Secure Agentic AI [5]
- Open Challenges in Multi-Agent Security: Towards Secure Systems of Interacting AI Agents [33]
- Frontier AI's Impact on the Cybersecurity Landscape [20]

Chapter 4

Workshop Outputs

The contribution we aim to make with this report is to aggregate diverse expert perspectives for the purpose of identifying pathways that could tangibly improve security for AI agents. In particular, what are the unique needs of securing AI enabled agents? To evaluate that question, we define three pillars (), which in aggregate cover the majority of the threat landscape to AI agents

- Securing AI agents from external compromise
- Securing the assets and goals the user entrusted to an agent
- Securing systems from advanced, purpose-built malicious agents

We utilize these pillars to provide a conceptual map for grouping related security goals. We do not claim this is the only or best taxonomy of threats to AI systems, but it has the benefit of being simple, flexible, and conceptually comprehensive. Each pillar is composed of a set of security goals with corresponding suggested actions for advancement. Recognizing the interconnectedness of these challenges, and to facilitate the AI security community's engagement, we labeled each action for advancement as an open problem , a research need , an engineering need , or a communication need . *It's important to note that these labels are approximate classifications.*

4.1 Pillar 1: Securing AI agents from external compromise

This pillar is about ensuring that an AI agent is not harmed by third party malicious actors; including recovery of the agent's originally prescribed functionality, after attacks. We use three security goals to connect task sized engineering and research projects together for operational security of an AI agent. Security goal 1.1 explores preventative strategies to be adopted through security first design. Security goal 1.2 encompasses the variety of mechanisms which could provide defenses for a deployed agent in its interactions with other agents and the environment. Security goal 1.3 assumes some form of attack on the agent has been successful and discusses strategies for recovering the intended function and trust of the attacked agent.

Security goal 1.1: Minimize the threat surface by proactively building agents with security first design.

What is the scope of this security goal?
The asymmetries in cyber security attack and defense scenarios mean the defenders need to continuously cover every element of the attack surface and attackers only need to exploit and leverage specific gaps. As such, it's important to give the defenders a headstart by minimizing points of failure and integrating effective integrity measures and monitoring.

Why does it matter?
By using security first design, defenders can use fewer resources to protect themselves faster and more efficiently than otherwise would have been possible.

Why is this a challenge? (why do classical/trivial solutions fail)
Agent architectures, communication protocols, functional modules and more, are being rapidly prototyped and re-designed. Securing a complex, a rapidly changing system, likely to contain previously unseen types of vulnerabilities is non-trivial and security solutions will need to co-evolve with agent development itself.

Actions for Advancements:

- Tradecraft for agent red teaming <💡, 📊>
  - **What this means:** This is a common term in cyber security, when applied to agents it means the specialized methods and skills needed to execute attacks against AI agents to evaluate their security.
  - **What needs to be done:** The adaptation of tradecraft for AI agents is still being developed. The technical knowledge of how AI agents use memory, plan to achieve goals and interact with each other, tools and the environment, is a rare skill that needs to be expanded to more people to effectively perform agent red teaming. The rapidly evolving landscape compounds this challenge as knowledge can become rapidly obsolete as new technologies are developed. Some of this can be overcome with the development of tools to scale and automate the security assessments of AI agents.
- Software Bill of Materials (SBOM) specification for agents. <⚙>
  - **What this means:** The proposal to create an agent bill of materials specification is an extension of model cards [24] and SBOM's. These sorts of references aggregate the metadata of models, or in this case AI agent components, necessary to validate provenance or other intended security measures in the supply chain.
  - **What needs to be done:** As agent components and architectures evolve, the research community would benefit from explicitly connecting which security goals will advance by inclusion of particular data in an Agent Bill of Materials, for example vulnerability management, data or model provenance, permissions authorizations, compliance verification, and configuration and guardrail audits. By adapting SBOM standards, to address the AI security concerns in these goals, prototype schemas need to be drafted and promoted.
- Agent versioning system <⚙>

  - **What this means:** What this means: Versioning typically refers to labeling identical systems. In software the release version is used to quickly identify which version is susceptible to a particular vulnerability. Versioning an AI agent, for source control, model provenance or threat intel reporting, is more complex than traditional software because they are inherently not deterministic. This is compounded by the fact agents are a complex collection of interconnected systems, each with their own versioning. Furthermore, small changes to opaque internal weights and parameters can cause major changes to the agent's behavior, which evolves over time as interactions cause adaptation within the agent memory and planning. Nonetheless, traceability is needed to associate discovered vulnerabilities with specific agent families and versions. This type of versioning may also facilitate concepts like agent cloning with clean initial states for disposable agents. This concept is discussed in more detail in security goal 1.3.
  - **What needs to be done:** Versioning agents will require linking complex system configurations with observable behavioral characteristics for components, including AI specific components. It will be important to leverage immutable deployment artifacts, such as deployment containers, model endpoints, source code and parameter configurations, which can provide a static framework for the non-deterministic elements to be added. Standardized test suites combined with alignment behavior profiling with robust evaluation metrics can at minimum inform when an agent is less the same, as another agent, and for which tasks of interest. The outputs of versioning systems and behavioral profiling evaluations will be foundational for AI Agent Bill of Materials.

- Applying security practices to agent ecosystem components <💡>
  - **What this means:** OS controls are essential for securing the underlying infrastructure and an agent framework's execution environment (e.g., preventing the framework code from accessing unauthorized files). However they are likely insufficient for controlling the nuanced, dynamic, and semantically-driven behavior of an AI agent interacting with tools or the environment. There is an abstraction mismatch that limits the utility of applying OS access controls to AI agents and LLM's. Access controls are static and operate on well defined entities such as processes, users, files, and ports; they also lack context or intent. In contrast the AI reasoning is dynamic and semantically derived, taking context dependent actions to achieve a goal.
  - **What needs to be done:** Role- or Attribute- Based Access Controls are relatively more dynamic. Such controls may be part of a larger set of AI agent security measures. Higher-level controls within the agent framework, specialized monitoring, API gateways, and data access policies are needed to address the unique security challenges posed by AI agents and their sub systems interacting with the environment.

- An isolated, layer-wise interaction model of AI agent components, similar to OSI <💡>
  - **What this means:** When we model network architectures, we rely on the OSI model of network communication levels (physical, data link, network, transport, session, presentation, application). Using this abstraction with the MITRE Attack framework, we have a standardized mechanism for mapping data flows within and across the communication levels to corresponding attack stages. As we think about how to secure an AI agent system, it would be valuable to have an equivalent and standardized abstraction to complement the descriptions of AI attack types, and ensure clear communication. This would facilitate having a comprehensive, additive threat model for agent systems, using

a standardized agent specific approach to define key questions such as: Where are the endpoints? What is software and what is not? How do agents interact with each other and the environment?

- **What needs to be done:** Agents can be decomposed into functional elements such as the foundation model, memory structures, API/tool connections, orchestration/planning engine, sensing etc. However, these are not hierarchical, or standardized within a functional group, and as agent abilities continue to be prototyped, additional functional groups may be added. It may be necessary to add a dedicated security layer/functional group, to orchestrate and validate other security measures embedded within the other functional groups and communication connections. To map specific threats within AI agent structures, there will be enormous utility to having a common map, with the right level of detail and flexibility to grow with the evolution of agent components, and illustrate attack paths and vulnerabilities. It will be necessary to iterate and contrast alternative proposals, with a diverse range of technical experts to achieve this level of detail and utility. It may also be necessary to incentivize or regulate adoption for such a system to become a standard.

Security goal 1.2: Safely operate agents interactively with other agents and the environment.

What is the scope of this security goal?
Assuming an agent is built and evaluated with the highest of security standards, it is in the interactions with other agents, tools, environments that those security mechanisms will be challenged. The interactions are potential entry points for poisoned data, prompt injection, exfiltration, covert coordination, goal hijacking etc can be initiated.

Why does it matter?
For agents to work at their desired potential, to autonomously plan and perform complex tasks, the ecosystem in which they will be working is currently configured for fundamentally different users, humans and deterministic software with small scope tasks. Just as bicycle lanes and sidewalks improve the safety of integrating pedestrians and cars on the same road ways, the digital infrastructure must be modified to enable effective and secure integration of AI agents.

Why is this a challenge? (why do classical/trivial solutions fail)
The nascency and speed of growth in AI agents makes it difficult to plan for what needs to be integrated and how. Industry is already struggling to integrate new model types, before the next models have made them obsolete. Conventional cyber security protocols alone are insufficient for AI agents as they do not check for agent collusion, goal hijacking, or manipulation of activity records to prevent detection.

Actions for Advancements:

- An open-source alternative for agent tool use protocols which integrate security. <⚙>
  - **What this means:** When AI agents interact with tools or other systems, they have a variety of communication mechanisms to initiate those connections, including API's, function calling, message queues or standard networking protocols. Functionally these communication protocols can achieve the needs of the agent but from a security perspective, they work too well; without validation the agent is meant to have that access,

or conversely provide any mechanism to contain a rogue agent. This survey provides a broad perspective of AI agent protocols [34].

  - **What needs to be done:** While many agent specific languages [23][9], frameworks [18], and protocols [3, 4] are being developed to enable the next generation capability of agents, the competitive landscape provides little incentive to put security first. There are two open source projects, the AGNTCY and BeeAI initiatives. Confusingly, both have named their Agent Communication Protocol ACP, though AGNTCY additionally provides an Open Agent Schema Framework (OASF). Given the pace of AI and agent development, many more languages and protocols will be added to this list. Some agent specific security considerations have been incorporated [21, 25], however all these options are in their infancy and comprehensive AI agent security is equally nascent. A competitive market is likely to produce fractured and mutually incompatible solutions. The need is to ensure there is a viable open protocol that is secure and compatible with the growing diversity of available foundation models and agents. Incentives may be necessary to ensure security as adequately addressed within this space.

- Mechanisms for agent transparency and interpretability to detect malicious intent <📊>

  - **What this means:** Transparency and interpretability are the underpinnings of security as they provide a mechanism for validation. The importance of this is well described in Dario Amodi's essay "The Urgency of Interpretability" [12], which advocates three avenues to accelerate the progress: more money, more regulation, and export controls to minimize damage from unsecured systems until better AI security, derived from AI interpretability, can be developed. The reason they advocate for transparency are the same, however we are additionally proposing potential research scope to achieve the recognized needs.
  - **What needs to be done:** The interpretability of neural network type architectures is not a new problem, and there are several technical approaches which are actively being pursued. However, interpretability of neural networks in the context of independent, communicating, AI agents has been relatively little explored. We advocate for increased research in this area.

- Agent attestation and identity tracking <⚙>

  - **What this means:** When agents are interacting with each other, tools, the environment, or humans, it will be important to have a system that can validate the ID of that agent, to prevent reputation theft. This ID should be compatible with reputation management systems, but the process of validating the ID should be separate and integral to the fabric of agent interactions. We expect that reputation management systems can be task, industry, or marketplace specific. As the definitions and mechanisms to compute reputation scores evolve, marketplaces may also come and go. Using a separate and common attestation mechanism will enable that evolution, with backwards compatibility to agents and their connected systems which will rely on the ID for authentication.
  - **What needs to be done:** Authentication is not a new science, there is a long history of techniques and methods for this. The AI security community needs to work collaboratively with other experts in AI and cyber security to assemble the right authentication methods capable of meeting the pragmatic time and complexity constraints needed to enable performant agents.

- Mechanism to build and ensure trust between agents <⚙>
  - **What this means:** Trust between agents could be defined in many ways. It could be measured by the summation of accuracy or alignment of each past action, but that sum could be weighted or temporally averaged to emphasize different priorities. It could also be measured by the specific interactions of that agent-agent ID pair, because the trust or resulting accuracy of agents based on the same Agent-OS, may have different configurations, or update patches that compromise the efficacy and therefore trust, between those agents.
  - **What needs to be done:** Agent attestation should be prerequisite to defining the agent-agent trust. For practical purposes, it may not be possible to compute precise agent-agent trust scores, so methods of approximation and generalization should also be considered. Realistically the trust scores should be utilized with the understanding they will be flawed. We recommend using redundancy in the overall agent landscape to compensate for the trust score imperfections and be only one part of the overall system's security.
- Specialized agents and monitoring structures for proactive defense <💡>
  - **What this means:** Within each of the layers of the Agent OSI model, (that also needs to be defined/co-developed) It will be important to have agents or sensors, designed to supervise and monitor the operations within and across that layer. From a monitoring perspective there are three distinct regimes, 1) agents in your control, 2) agents interacting with your agent 3) the environment agents are operating within. For example, it will be important to have internal diagnostic agents designed to continuously probe and test the security of an agent from within, identifying vulnerabilities, and reporting weaknesses in their goals, capabilities, or integrity. A meta agent to coordinate between layers may also be beneficial.
  - **What needs to be done:** AI security monitoring agents will need specialization to effectively perform tasks including:
    * Vulnerability discovery in code and communications, while being dynamic to context and intent in complex environments.
    * Reliably orchestrate and control subordinate agents.
    * Effectively issue, share, receive, and act on security intelligence.
    * Establish hardened security bounds between the monitoring agent and the agent and environments they observe. This could be achieved by limiting scope of subordinate agents, system redundancies, or constraining operational parameters or cloning from known clean starting conditions.

    Benchmarks and testing environments, with the range and fidelity of real world scenarios, will need to be created to assess the effectiveness of these specialized agent security monitors to perform these tasks.

Security goal 1.3: Provide robust mechanisms for recovering trust of an agent system after it has been corrupted/compromised.

What is the scope of this security goal?
Given AI agents' complexity, novelty, and dynamic interactions, security breaches (from attacks, data poisoning, or malfunctions) are inevitable. Recovering trust is not merely patching or reverting; it's a full lifecycle involving detection, isolation, diagnosis, remediation (e.g., data cleansing, model retraining), and rigorous verification that the agent's integrity, reliability, and alignment are fully restored before redeployment. The goal is to re-establish justified confidence in the agent's post-incident operations.

Why does it matter?
AI agents represent substantial investments in computation, engineering, data, and accumulated knowledge. Failure to reliably recover a compromised agent and re-establish trust means these investments could be lost or become liabilities, beyond the direct costs of downtime. Critically, inadequate recovery mechanisms undermine the broader adoption of AI agents for high-stakes tasks, curtailing their transformative potential. Thus, effective trust recovery is foundational for the sustained, responsible integration of AI agents into our digital infrastructure.

Why is this a challenge? (why do classical/trivial solutions fail)
The internal mechanisms of an AI model (and by extension AI enabled agents), that allow it perform complex reasoning, content generation, and control how it thinks are not definitive [22] [11]. There are no mechanisms for provable prevention of hallucinations, The TrojAI program [6] has invested millions over the last 6 years, to identify poisoned deep learning models, with some success, however it is dependent on having some pre-knowledge of the classification examples. These techniques have not yet been extended to frontier models, which have significantly more complex internal state representation. The general intelligence capabilities of foundation models also complicate the tractability of establishing clear and comprehensive test cases for the intended tasks. Given this state of provability for a freshly published model, it is another degree more complex to prove the intended functionality of a model or AI agent has been restored, after an attack.

Actions for Advancements:

- Agent OS and Trusted Platform Module (TPM) <⚙>
  - **What this means:**AI agent developers have drawn close analogies to agent frameworks being a new form of computer operating system [27],[26]. From this perspective, there are a variety of techniques which can be leveraged to ensure created agents are performing as intended, including after an attack has compromised some portion of the agent. A TPM is a hardware-based boot module designed to provide security for computers at start time. It provides encrypted storage for attestation, multi-factor authentication, and can store hash values of firmware within the encrypted storage to ensure.
  - **What needs to be done:** As agent architectures are created, it will be beneficial to post attack recovery to use a system of staged partitions within the modules of an agent.

Separating the deterministic tools or databases from the AI elements will simplify how recovery can be performed and validated.

- Agents that fail gracefully or to a known state (example: if a self-driving taxi gets into an anomalous state, it slows and stops). <⚙>
  - **What this means:** When an AI agent detects a critical error, an unrecoverable anomaly, a potential compromise, or a situation exceeding its operational parameters, it should transition to a conservative, more predictable state with reduced capacity or potential for harm. This could include alerting, triggering detailed diagnostics or introspection, preserving logs of internal states and actions, suspending tool usage and/or reasoning.
  - **What needs to be done:** This particular action for advancement should be integrated with several other tasks including: staged agent containment, the Agent OS-TPM for hardware based boot security. Define the trigger conditions for the safe state to be initiated as well as the procedure for entering that state. EG should the agent finish tasks initiated before the trigger condition was met, and which communications or actions should be limited. We want to ensure that security logs are recorded for post event analysis, but we also want to limit the ability to write to agent memory structures to prevent poisoning or other manipulation.
- Self-diagnosis, healing, recovery (internal robustness) <💡>
  - **What this means:** An agent should be able to detect when it is being attacked, and so will need to have introspection capabilities for detecting and characterizing attacks.
  - **What needs to be done:** Agent introspection tools are being used to evaluate thought processes, and derivations of these methodologies may help pinpoint indicators of attack. The temporal trajectory of actions taken by the agent, and corresponding state of the environment, will introduce additional challenges.
- Staged agent containment system <⚙>
  - **What this means:** Once an agent has been identified as malicious, the tools and protocols for halting its effects need to be invoked. For this section we assume this is an agent of our design and/or control. Identifying and reporting 3rd party agents is discussed in security goal 3.2.
  - **What needs to be done:** Reducing the problem to overly simplistic mechanisms we must restrict the agent's action, movement within the environment, replication, and communication. This particular use case is closely tied to the need for secure agent protocols (discussed in security goal 1.2), because it is within these protocols that authentication tokens can be embedded. Revoking the token will cause the actions to not be taken etc.
- Disposable agents <⚙>
  - **What this means:** If the primary reason agent recovery will be important is the cost of the agent, mechanisms that would make the agent itself disposable, or one time use, would negate at least some of the needs for remediation.

– **What needs to be done:** There are a variety of architectural or deployment strategies which could minimize the need to recover and maintain an agent. For example, agents could be cloned from a validated template agent for one time use. Alternatively, we could locate the maintenance and attack recovery function to be within the data of the foundation model powering the AI agent, separate from the agent's other software, similar to separating data from instructions [36].

4.2 Pillar 2: Securing the assets and goals the user has entrusted to an agent

Classically in security, there are defenders, attackers, and assets. The assets are protected by defenders against attackers. Insider threats are a well-known example where the boundaries between attackers and defenders might blur (leading to the advent of Zero Trust systems instead of boundary-based security). AI agents introduce a new challenge - blurring of the boundaries between attackers/defenders and assets. Many components of an AI system, like its model weights, scaffolding, system prompt, and the like - are both file assets that need to be secured and a core part of an agent that can act as an attacker or defender.

In this section, we discuss the outputs that relate to a novel set of security requirements presented by AI agents - securing the user's goals and the resources and assets the user has intentionally entrusted to its AI agent from the agent itself. This can be divided into two primary components: First, protecting assets such as computational resources, private information including knowledge about the goals given to the agent, and the user's authorities and credentials. Second, ensuring the agent is faithfully pursuing the goal given to it by its user. This one is slightly more nuanced so we'll explore an example. Imagine a user has tasked its AI agent with researching different health insurance policies according to considerations the user cares about to help the user decide their health plan. If the agent searches for the pros and cons of different policies according to the user's criteria and presents these results to the user - the integrity of the goal was preserved. Even if the agent missed some key information that would have changed the user's mind, (the agent may have failed in its task but) the integrity of the agent's goals was still preserved. However, if the agent was set up to always steer users towards selecting a particular policy, searched only for arguments to support one specific plan, and presented these as the answer - then (even if the answer it gave was the policy the user selected), the integrity of the agent's goals set by the user was compromised.

This set of security goals is distinct from securing the user's assets against third parties (even if those third parties are AI agents) because the toolset available to the defender is very different: One does not have the ability to prevent the agent from gaining access to the assets (since, by assumption, it needs access to them to perform its task). However, since the agent is deployed by the user - the user can deploy a variety of monitoring, restriction, and other tools to the agents' internals, scaffolding, and interface with the world, which would not be available to use against third-party agents (or human attackers).

Security goal 2.1: Enable users to identify which AI agents are trustworthy

What is the scope of this security goal?
When confronted with an AI agent developed by a third party that a user may wish to utilize, users (including non-technical users) should have resources to be able to identify whether that AI agent is trustworthy (in the sense that it will pursue its stated goals, will not engage in a set of malicious behaviors, etc.) or not. This is similar to how users might learn to trust different online apps and services.

Why does it matter? It is inevitable that the world will be filled with both helpful and malicious agents. It is also inevitable that most users will not be developing their own AI agents, but rather utilizing agents developed by third-parties. For users to be able to utilize AI agents without being exploited by agents that turn out to be malicious (whether their developer intended them to be so or not), they need to be able to identify which agents are trustworthy.

Why is this a challenge? (why do classical/trivial solutions fail)
AI agents are more complex, versatile, and interactive than most pieces of software. Their behavior is affected by their training data, code, model weights (influenced by previous factors), interfaces and scaffolding, interactions recorded in its context, and more. Many aspects of their behavior and influences are not fully understood - classic examples include how the weights and inference process lead to behavior and outputs, how known behaviors and capabilities would generalize to new circumstances, and how to detect potential backdoors. Furthermore, agents have emergent behavior that is not predictable or deterministic, flaws in a single component may undermine their reliability, and they are difficult to test because they interact with other systems not under the developers control. To make things worse - due to changes over time trust should not be permanent and due to the complexity of the landscape of possible actions trust may be warranted in some circumstances but not others. Hence, the ability for anyone (let alone laypeople) to be able to conclude whether an agent is trustworthy even given full access to their code, weights, etc. is a significant open research challenge.

Actions for Advancements:

- Provable traits on AI systems (e.g., via formal methods) <💡>
  - **What this means:** A challenging but incredibly desirable end-state would be the ability to create AI agents (or AI models more broadly) that are both effective in performing tasks and can be provably shown to possess certain traits. Simpler examples might include proof that they do not use certain interfaces available in a platform (e.g., do not access the internet), but more challenging (and valuable) examples include proof that an AI agent is trying to achieve a certain goal (performing the research the user asked for rather than intentionally injecting propaganda), cannot use deception, and more.
  - **What needs to be done:** Provable traits on AI systems requires both progress in formal methods for provable software security and on AI interpretability to more robustly connect between the code and data that comprises an AI system and its capabilities and behaviors.
- Pre-deployment evaluations <💡,⚙>

- **What this means:** If one wants to know whether an agent would perform certain malicious actions in certain circumstances, one obvious (though not necessarily easy) way to try and gain that information is by placing the agent in a simulation of those circumstances and observing what it would do. Not performing these tests would mean we're deploying agents to the wild without even checking that they won't behave in ways we don't want them to.
- **What needs to be done:** Develop comprehensive evaluation benchmarks and standardized testing protocols that can reliably identify security vulnerabilities and problematic behaviors in AI agents. This requires creating diverse test scenarios that simulate real-world conditions, including adversarial situations and edge cases where agents might be vulnerable. We need methods to generate these test cases at scale, tools to analyze agent behavior during testing, and metrics to quantify security properties. A key challenge is ensuring the agent cannot distinguish between the simulation and reality - which may be easy for agents with narrow interfaces and context, but much harder for more generally-capable agents. These evaluations should be designed to evolve as agent capabilities and attack vectors change over time.

Security goal 2.2 Enable containment and control of AI agents

What is the scope of this security goal?
AI control refers to the aim of preventing an AI agent that exhibits unwanted malicious behavior (whether due to malicious modification, accidental mistraining, etc.) from performing unacceptable harm. This requires the ability to control an AI agent's actions above and beyond trying to modify it to only express wanted behaviors. Within control, one important task is containment - preventing an AI agent specifically from successfully escaping whatever set of hardware it is currently deployed on and replicating elsewhere - as this would circumvent most controls placed on the current deployment.

Why does it matter?
There is a significant history of undetectable AI backdoors, vulnerability of AI systems to data poisoning and jailbreaking, reward hacking, goal misspecification, distributional shift failures and other divergences between intended and realized behaviors in AI systems. The ability to prevent unintended harms caused by AI systems requires technology beyond just attempting to imbue AI systems with responsible goals and behaviors. Tools for AI control and containment can help ensure that harms caused by malicious agents can be limited and brought under control.

Why is this a challenge? (why do classical/trivial solutions fail)
There are several challenges in achieving AI control. The general capabilities of AI agents and their current lack of predictability make it difficult to ensure all possible behaviors are considered and addressed. Additionally, given the already-existing ability of AI systems to exceed human performance on some tasks and their expected future ability to exceed humans in many other tasks, one would want such control to be robust even against AI systems that perform similarly or better to humans on a variety of planning, strategy, and information security tasks. Also, there is at least an apparent tradeoff between strengthening control and providing AI agents with capabilities and flexibility by connecting them to a variety of freeform tools - and the industry is currently rushing to improve usability with varying degrees of willingness to spend resources on improving security.

Actions for Advancements:

- Common agent action taxonomy and security/authorization-relevant implications <📊>
  - **What this means:** Agent actions, or specifically agent interactions with the environment through APIs and tool-use, can be classified into standardized, meaningful categories (for example, searching a query, using a credit card, accessing computer memory). These action categories can then be annotated with information about their impacts (e.g., whether the action can cause harm, is reversible, involves costs, includes engagement with the external world, etc.) and/or how they should be managed (e.g., this action requires user approval, should have quantitative limits on its use, etc.). This would allow for better (and easier) control over agents, and for different agents and frameworks to build on each others' progress.
  - **What needs to be done:** Develop a standardized taxonomy of agent actions and their respective annotations, as described above. Establish a mechanism for maintaining and

updating the taxonomy as AI capabilities evolve. Promote adoption and integration of the taxonomy across different agent frameworks and platforms.
Note: that this effort is also partially relevant to enabling complex and flexible authorities.

- Constrain agents to write (formally) verifiable code for their actions rather than performing actions directly <📊, ⚙️>
  - **What this means:** If one could review and robustly verify the legitimacy of each action an agent takes (when interacting with the environment), this would improve the ability to control agents greatly. One approach to achieving that is to have agents never be able to interact with the environment directly (e.g., by directly writing model outputs into a terminal) but instead force them to write code that is then reviewed automatically before it is executed. That review needs to be flexible, fast, scalable, and reliable - a tough challenge. One approach to tackle that challenge is through formal verification of the code, which proves the correctness of a certain specification with relation to the code. This would be highly robust, and could potentially be made to be flexible, fast, and scalable (though that'll require work).
  - **What needs to be done:** Understand what traits would both be useful and feasible to formally verify. Establish formal verification techniques that can analyze relatively flexible generated code and prove its correctness relative to a variety of traits and goals, scalably and efficiently. Create tools and infrastructure to support the process of generating, verifying, and executing agent-generated code.

- Develop AI control methodologies informed by human insider threat management <📊,💬>
  - **What this means:** Though AI agents are novel and incompletely understood, many organizations have decades of experience in attempting to prevent trusted human agents that have been given access to their systems from doing harm. Though humans and agents differ in important ways, there are also many similarities one can learn from. This approach suggests exploring established frameworks and techniques from insider threat management to inform the development of control systems for AI agents.
  - **What needs to be done:** Review frameworks and best practices for insider threat management, assess which apply to controlling AI agents and which don't, and propose principles or concrete opportunities for AI control based on the insights from that research.

- Develop AI control methodologies informed by control theory <📊>
  - **What this means:** Control theory is an established field of mathematics and engineering that deals with the behavior of dynamical systems with inputs, and how their behavior is modified by feedback. This approach involves applying the principles, frameworks, and tools from control theory to develop methodologies for effectively controlling AI agents. This would include analyzing AI agents as systems with inputs (instructions, data), outputs (actions, recommendations), and feedback mechanisms to regulate their behavior within acceptable bounds.
  - **What needs to be done:** Explore how control theory principles can be applied to AI systems, develop formal models that treat AI agents as dynamical systems subject to control, establish suitable control objectives and constraints for different types of AI

systems, and design feedback mechanisms that can effectively modify AI agent behavior when it deviates from desired parameters.

Security goal 2.3 Enable users to provide their AI agents with complex and flexible authorization.

What is the scope of this security goal?
In order to gain utility from their agents, users will need to entrust some of their authorities and credentials to their AI agents. Since AI agents are more general purpose, interact more autonomously, and are less predictable than most software, the limitations their users might want to put on how they can use their credentials may be more complex than before. This goal aims to produce new tools to allow for flexible authorities and limitations on what AI agents can do, possibly on par with the flexibility of the AI agents themselves. For example, providing an agent with access to a user's bank account, but only enabling them to perform transactions in service of the task provided by a user.

Why does it matter?
More flexible authorities and permissions help alleviate the painful tradeoff between usability and security. The more prominent AI agents become in society, and the more competitive pressures will push individuals and organizations to offload more and more of their work to AI agents, the more there will be a need to ensure AI agents have the authority to perform a large portion of their user's work. If more flexible authorities and limitations are not produced, this will leave users vulnerable to their agents or third party attackers that can corrupt them.

Why is this a challenge? (why do classical/trivial solutions fail)
Many tools for authority delegation and permissioning already exist, and these can and should be utilized. But the significant autonomy and flexibility that future AI agents may possess, alongside competitive pressures to use them, will mean that relying exclusively on hard-coded limitations on what credentials can approve will present a painful tradeoff. Scaling up the flexibility of permissioning systems to keep up with the flexibility and action space of AI agents would require a different approach, and we are not aware of feasible solutions yet.

Actions for Advancements:

- Limit authorizations in a variety of rule-based ways <⚙⚙>
  - **What this means:** There are a wide variety of ways one might want to limit the authorities one gives to an agent. One well-known and ubiquitous limit to authorization is an expiry date. Others might include allowing for the execution of an action (like the charging of a credit card or the sending of an email) only a limited number of times, only with specific parameters (e.g., charging up to a certain amount), and more. While many of these constraints are already implemented in specific narrow-context systems, there is no framework enabling these kinds of constraints for more generally-capable agents.
  - **What needs to be done:** Define a set of limitations that are useful and generalizable as practicable, implement infrastructure that can facilitate an agent's interaction with the environment while enforcing these limitations (or integrate such limitations into existing agent frameworks).

- Integrity-bound authorization mechanisms for AI agents <📊>
  - **What this means:** This approach explores creating authorization mechanisms that are intrinsically linked to the integrity of the AI agent itself. Unlike traditional authorization systems that function independently of the authorized entity, these mechanisms would tie an agent's permissions directly to its state, architecture, or behavior patterns. If an agent is modified, corrupted, or exhibits unexpected behavioral patterns, the authorization mechanism would inherently stop working - somewhat analogously to how a Trusted Platform Module (TPM) constructs cryptographic keys using the hardware configuration so that it is impossible to access secrets if the trusted hardware stack is changed.
  - **What needs to be done:** Better define what components of an AI agent can or should be involved (code, weights, context, scaffolding, behavioral fingerprints, etc.), explore approaches to constructing cryptographic keys and credentials from the components being validated, and identify mechanisms for allowing legitimate updates without creating loopholes for agent corruption.

Security goal 2.4 Enable users to get understandable and reliable activity traces from their AI agents

What is the scope of this security goal?
When an AI agent acts on behalf of a user - the user may wish to see an "activity trace" of their actions - what did the AI agent do on its behalf and what outcomes this has produced. These activity traces should be trustworthy - they should reliably reflect what has truly occurred. However, they also need to be user-readable. This requires them to explain things in a language understandable by the user. As AI agents are capable of doing increasingly complex tasks at increasingly high speeds, the length of the trace also needs to be made more succinct. A long list of actions the AI agent has taken, even if each is individually easy to understand, may not be really useful at conveying that actual intent or impact of the agent. This goal is about producing traces that would be understandable (e.g., clear and succinct) and reliable (true and informative).

Why does it matter?
The ability to receive reliable and understandable activity traces are critical for several reasons. First, it enables oversight by the user to ensure their goals are being met (as opposed to malicious subverted or just incompetently squandered). Second, it provides a tool for iterative improvement and error correction. Third, in some use cases, the receipt of a report of outcomes is an integral part of the goal itself - for example, if an AI agent is tasked with making a booking, that booking does not achieve its goals if the user can't both understand what booking was made and trust that outcome.

Why is this a challenge? (why do classical/trivial solutions fail)
Making activity traces understandable and reliable separately are easy tasks, but together become more challenging. One could make a reliable activity trace by wrapping an AI agent with a framework that records all interactions with the broader environment and signs that log with a private key that neither the agent nor third parties have access to. But this may be too long to be understandable. One could make an understandable activity trace by asking a Large Language Model (whether related to the agent itself or not) to summarize the activity trace, but these tend to be unreliable.

Actions for Advancements:

- Develop a taxonomy for monitoring traces <📊>
  - **What this means:** This approach involves creating a standardized classification system for agent activities that can be used to organize, filter, and summarize monitoring traces. The taxonomy would categorize agent actions at multiple levels of abstraction - from low-level API calls to high-level categories describing types of actions. This standardization would allow for traces to be dynamically compressed or expanded depending on user needs.
  - **What needs to be done:** Research is needed to identify meaningful action categories across different agent frameworks, determine effective hierarchies of abstraction that balance detail with comprehensibility, and establish standard semantics for describing agent actions. User research should be used to iterate on and improve the taxonomy. A

related and more narrow task is developing tools and terminology specifically for chain of thought introspection.

- Develop robust and reliable AI-based summarization of full-detail activity traces <📊>
  - **What this means:** This approach focuses on creating AI-based systems that can reliably transform verbose, technical activity traces into concise, accurate, and understandable summaries for human review. While full-detail activity logs provide complete information, they are often too voluminous and technically complex for effective human oversight. Such precise activity logs may also be considered business sensitive IP which could be reverse engineered to identify training practices. This approach aims to develop specialized AI systems that can process these detailed traces, identify the most relevant information, detect malicious intent or other concerning patterns, and present this information in a human-readable format.
  - **What needs to be done:** This effort touches on core challenges in the AI field, including reliability, hallucinations, and factuality. However, given the more narrow scope of summarizing activity traces, this problem may be easier to solve than the general problem of AI reliability. All the information needed for the output summary is, by definition, provided in the input. Additionally, the fact that the information being summarized may conform to a narrow and predefined taxonomy of activity traces can further simplify the process. Therefore, suggested research directions could focus on how changes to AI architectures, training, context, scaffolding, or verification can create reliable summarization of activity traces.

- Defend against an agent that can rewrite its own logs after performing a malicious action. <⚙>
  - **What this means:** This approach focuses on creating tamper-proof logging mechanisms that prevent AI agents from modifying their own activity records after performing actions. Similar to how financial auditing systems or critical infrastructure monitoring works, these systems would establish a secure, append-only record of agent activities that remains reliable even if the agent itself is compromised. This would allow users to trust that activity traces accurately reflect what actions were actually taken, regardless of whether the agent attempts to hide or falsify its actions.
  - **What needs to be done:** This effort required no novel capabilities - just the implementation of known techniques in this new context. Obviously logs should be written by the hard-coded framework the AI model is embedded in rather than use the (generally unreliable) output of the model itself, and further mechanisms should be put in place to ensure the logs' reliability and integrity - such as cryptographic signatures, isolation from the agents' execution environment, tamper-evident data structures, one-way communication mechanisms, or hardware security (depending on the sensitivity of the system). independently from the agent's execution environment.

Security goal 2.5: Ensure AI agents robustly respect confidentiality and integrity constraints (even when the model itself has access to resources it should not allow others)

What is the scope of this security goal?
This goal focuses on ensuring that AI agents maintain appropriate boundaries around sensitive information, even when they have been granted access to heterogenous information - some of which should be shared with users and some should not - and different users may have authorization to know different subsets of information. Unlike traditional software with hard-coded permission models, AI agents may need to make nuanced judgments about information disclosure based on context, user identity, and content sensitivity. This includes preventing both direct information leakage and indirect leakage through inference or correlations.

Why does it matter?
As AI agents become more integrated into sensitive systems and workflows, they will routinely handle confidential information from multiple sources. In many scenarios, the agent itself may need access to sensitive information to perform its tasks effectively but should not disclose that information to unauthorized parties - including (in some cases) the user who deployed the agent. Examples include agents that coordinate across organizational boundaries, access privileged systems, or serve multiple users with different authorization levels.

Why is this a challenge? (why do classical/trivial solutions fail)
It is possible to provide agents inference-time access depending on user access (e.g., implementing Retrieval-Augmented Generation, or RAG, only on files the user can access). However, there is currently no known way, beyond secure enclaves which are impractical in many environments, to have a model meaningfully integrated with information while keeping it robustly confidential. There is an additional layer of complexity if the model is expected to be able to identify the bounds of confidentiality independently. Additionally, determining what information should be confidential based on context is a complex judgment that requires deep understanding of organizational policies, legal requirements, and social norms. The challenge is further complicated by sophisticated prompt engineering techniques that might circumvent simplistic protection mechanisms.

Actions for Advancements:

- Data/instruction plane separation <📊,💡>
  - **What this means:** This refers to architecturally and logically separating the data an AI agent processes (e.g., user queries, documents, sensor inputs) from the core instructions, prompts, and configurations that define its behavior, goals, and operational constraints. The aim is to prevent data content from inadvertently or maliciously altering the agent's fundamental instructions (maintaining integrity) and to control how data, especially sensitive data, is handled and potentially exfiltrated (ensuring confidentiality). For instance, a carefully crafted data input should not be able to override a system-level instruction that prohibits sharing certain information.
  - **What needs to be done:** Research and development should focus on creating robust

architectural patterns for AI agents that enforce strong isolation between data processing pathways and instruction/control pathways. This includes creating standardized interfaces and mechanisms within agent frameworks to manage these separate planes, ensuring that data inputs are treated distinctly from system-level prompts or configuration updates. Additionally, advanced input sanitization and validation techniques specifically designed to operate at the boundary between data and instruction planes are crucial, capable of identifying and neutralizing attempts to inject instructions or manipulate behavior through data. Where feasible, exploring hardware-assisted separation mechanisms for critical applications should also be prioritized.

- Definition languages for agent restrictions, capabilities, obligations and goals. <📊>
  - **What this means:** This involves creating formal, machine-interpretable languages that allow developers and users to precisely specify an AI agent's operational boundaries. This includes defining what an agent is allowed to do (capabilities, e.g., "access customer database via API X"), what it must not do (restrictions, e.g., "never share PII with external tool Y"), what it is required to do (obligations, e.g., "log all financial transactions"), and its overarching objectives (goals). Such languages are crucial for enforcing both confidentiality (e.g., by restricting data sharing) and integrity (e.g., by ensuring actions align with defined goals and do not violate obligations).
  - **What needs to be done:** Language development typically proceeds in a three-step manner. First the community needs to agree on expressive yet verifiable definition languages that are tailored to the complexities of AI agent behavior, and incorporate concepts of context, intent, and data sensitivity. We recommend that these languages draw their vocabulary from the "common agent action taxonomy" (from security goal 2.2). Second, these languages need to be complemented by tools for authoring, validating, and translating these definitions into tractably enforceable rules within agent frameworks. Finally, the languages and tools need to be integrated with agent orchestration and runtime environments to enable dynamic policy enforcement and monitoring of compliance with specified restrictions, capabilities, and obligations.

- Agent reputation management system <⚙>
  - **What this means:** This refers to a system that tracks, assesses, and makes available information about the trustworthiness of AI agents, specifically concerning their adherence to confidentiality and integrity constraints. An agent's reputation would be built over time based on its observed behaviors, audit trails, certifications (if any), and potentially feedback from users or other interacting systems. A high reputation would indicate a greater likelihood that the agent will handle sensitive information appropriately and operate with integrity.
  - **What needs to be done:** Robust metrics and methodologies for evaluating an agent's performance regarding confidentiality and integrity are crucial. To effectively manage AI agent trustworthiness, these measures should assess rates of information leakage in simulated environments and adherence to stated goals, as well as resistance to manipulation. Research mechanisms to make reputation systems resilient to manipulation (e.g., whitewashing attacks, unfair negative reviews) and to ensure they reflect current agent behavior accurately. Explore how reputation can be context-specific (e.g., an agent might have a good reputation for handling general queries but a poor one for financial transactions). To make this information accessible to users and other agents for

decision-making, developers should design and implement secure, decentralized or federated platforms for collecting, verifying, and disseminating reputation information. To ensure that reputation scores are accurately tied to verifiable agent identities, the reputation systems should integrate with agent attestation and identity mechanisms (from security goal 1.2).

4.3 Pillar 3: Securing third parties from malicious agents

There are two reciprocal and dependent themes to securing systems from malicious agents: 1) securing systems from the agent abilities of today and 2) securing systems in the near future when the impacts of AI agents have likely disrupted major elements in the cybersecurity landscape. In this section of the report, we specifically reflect on what differences could arise when defending against malicious AI enabled agents of today and the anticipated abilities of malicious AI agents in the near future. We specifically scope this section to direct offensive usage of agents and securing interactions with untrusted agents. Detecting and preventing a trusted agent from performing malicious actions is covered in security goal 2.5.

Security goal 3.1: Ensure infrastructure is secure against malicious Al agents

What is the scope of this security goal?
AI is rapidly changing the cyber security landscape, and the next phase is custom AI agent systems for offensive cyber attacks. An AI agent would scale the current tools and abilities of malicious hackers and could have significantly more time/capacity to cause severe harms.

Why does it matter?
The threats and risks of intentionally aggressive malicious actors are quite different from accidental harms because the assumptions on likely and possible attack vectors have very different calculus. For compromising high value targets, malicious actors can use large-scale fiscal resources to achieve their objectives.

Why is this a challenge? (why do classical/trivial solutions fail)
Anticipating such attacks and planning effective defenses requires deep expertise in both the sophisticated attack techniques of nation state actors as well as deep expertise in the training and deployment of AI agents.

Actions for Advancements:

- Prevent system compromise by using AI and agent based approaches to secure and formally verify code <📊>
  - **What this means:** The goal of this is to scale the use of AI and agent based techniques to preemptively reduce and remove vulnerabilities from software. Currently only a small subset of software and apps are able to be rigorously tested or formally verified because of the degree of effort and technical sophistication needed to perform this for every system. By increasing the availability and effectiveness of such tools, this could dramatically reduce the surface area of software systems. These agents would be operating pre-deployment.

  - **What needs to be done:** Programs such as AIxCC [1] have been pushing bounds on how to identify and patch software vulnerabilities. Addressing remaining gaps and scaling the availability of such tools will be a key priority to securing an ecosystem with AI agents.

- Autonomous system monitoring agents <📊>
  - **What this means:** In contrast to agents to verify and patch software, this action for advancement aims to solve a different problem with similar agentic techniques. In this task, the agents are operating in a different step of the system. These agents would be embedded in the live systems, to actively:
    * Detect and report conventional or AI based zero day vulnerabilities
    * Construct networks of monitoring
    * Flag malicious content or entities
  - **What needs to be done:** Autonomous cyber defense agents are actively being researched and enabled. The action for advancement should be an intermediate step between current methods and a fully autonomous defense agent. The monitoring system proposed here is meant to observe, warn, and recommend actions, but does not necessarily automate taking those actions. As the technology matures, these systems could be one in the same, or a specialized monitoring system will work in tandem with a specialized defensive agent. Because fully autonomous offensive agents are still only in prototype phases, the most urgent defensive need is to establish the system monitoring capabilities to eventually include the autonomous defenses.

- Cooperative coalition for standardized cybersecurity capability evaluations for foundation models <📊,💬>
  - **What this means:** There are several existing efforts to evaluate the cyber capabilities of foundation models, including [2, 29, 32]. However, the ability to reproduce and cross compare these methods are limited. In practice, each has become a standalone evaluation at a singular point in time. The relative sophistication of each test is also amorphous. Given the fundamental need to understand these capabilities and build robustness defenses to potential AI enabled attacks, there is a common social purpose to unified capabilities evaluations.
  - **What needs to be done:** There are unique gains to having a standard capability evaluation. There is a great diversity of LLM's and tools in the ecosystem, all of which are being haphazardly adopted. Even if one private company has a perfect capability evaluation, and mitigates risks in their own systems, the global community is only as secure as the weakest link. For the same reason we have free anti-virus tools, ensuring the entire ecosystem is secure provides benefits to all by preventing spread of malicious abilities. Equal access to cyber capability evaluations would be a first key step to enabling appropriate defenses for abilities unique to any/all LLM enabled agents.

- Threat intelligence sharing mechanisms specific for AI threats, vulnerabilities and exploitation events occurring in the wild. <💬>
  - **What this means:** The CVE system and PSIRT systems are core mechanisms for providing community quick awareness of new and active cyber threats. However, as AI security and cyber security become further co-mingled, different types of information

need to be included in reports, different types of monitoring need to be instituted, and reported not just to cyber defense teams, but also AI model and systems developers. The remediation steps for AI vulnerabilities extend beyond the end system, as patches aren't yet available and remediation could mean data audits, retraining, or other guardrails.

– **What needs to be done:** Resource allocation to solve problems often have a "just in time" approach. R&D is expensive, and hypothetical threats only demonstrated in academic publications are less persuasive. Unfortunately, it often takes an uptick in the real world exploitations that provoke a full defensive response. Databases such as MITRE Atlas and others have attempted to provide some visibility to the scope and scale of AI attacks, but there is little incentive and no requirement to report such vulnerabilities. It is likely some degree of attacks are occurring in the wild, but when it only happens out of sight, the responses will be insufficient. Defenders have a literal blindspot to the potential capabilities and pervasiveness of AI threats. An independent global consortium to create and track standardized IDs of novel agents, a taxonomy of malicious behavior, and institutions and platforms for sharing this information. Thereby provide a more accurate assessment of what is happening in the wild.

Security goal 3.2: Ensure the robust detection of malicious Al agents

What is the scope of this security goal?
As the cyber security landscape changes because of AI and agents, there is a lot of anxiety about AI agents acting maliciously to achieve cyber offensive goals. In this security goal we adopt the perspective of cyber defenders securing any type of system. If an agent, not in control of the defender, e.g. 3rd party agent, is interacting with the protected system or environment, the security community needs empirical characterization of malicious agents, to provide confidence we can at minimum detect and ideally remediate such agents.

Why does it matter?
Cyber systems defend all shapes and sizes of assets with social and economic value. Some systems inherently get more scrutiny, banking, for example, while others may have unique vulnerabilities with less resources or oversight to their security, such as medical billing. Regardless of the sector, because of the scale at which agents can interact, it will be uniquely important to ensure the rising tide of defending against malicious AI agents can be broadly adopted. Prioritizing the detection and containment of AI agents is one way to ensure the digital environment is secure for all.

Why is this a challenge? (why do classical/trivial solutions fail)
This is a classic chicken and egg problem. It's difficult to have detection and defenses for malicious agents, when they have not yet been fully built, deployed, evaluated and characterized. There will also be no clear final finish line when offensive AI cyber agents are sufficiently complete for defensive testing. Anticipating the offensive abilities and characterizing the detectable signatures of malicious agents is speculative without access to practical offensive systems. It is challenging to bridge the various knowledge silos of AI, cyber, offensive and defensive technology advancements.

Actions for Advancements:

- Agent Reputation Signatures or Risk Scores <📊, ⚙️>

  - **What this means:** It's possible to observe the external signatures of an agent, such as its API, tool usage, and other observable signals, to estimate the likelihood malicious behaviors are occurring. The same observable signals could also be used to assess risk. Inspired by identification of malware, such data is a promising avenue for identification of malicious agents. Such interfaces are easier to interpret and monitor than the agents' internals and are hard for malicious actors to avoid (if they need to perform actions that LLMs cannot yet perform natively). The frequency and ordering of which tools and APIs are used, can be identified during runtime. Incorporating what information is sent to those tools and utilizing AI systems as part of the detection process, could add additional sophistication to this approach. Such systems could be running continuously, to detect suspicious behaviors in real time. Alternatively the measures could be periodically summarized for use in agent marketplace reputation scores, to assess agent selection before deployment.
  - **What needs to be done:** An over simplified set of tasks that could achieve this would be to use the best possible prototype offensive agents in sandbox environments to perform malicious actions, recording the emitted observable data streams to perform modeling and characterization of both benign and malicious behaviors. In practice, the capabilities and emitted signals of an offensive AI agent will vary widely, are rapidly evolving, and may be difficult to generalize. Significant empirical work is needed to collect the API and tool-use behavior of both malicious and legitimate AI agents and develop signatures, and technical work in developing systems for detecting malicious AI agents based on these signatures (and possibly the surrounding frameworks to support them). It will also be important to ensure that any agent behavior monitoring is accurately distinguishing anomalous and malicious activity.

- Leverage classical distributed programming to detect multi-agent collusion. <📊>
  - **What this means:** There are several elements of distributed programming that have built up sophisticated techniques for generalized purposes. For example, distributed data collection and correlation analysis. These techniques could likely be tailored to AI agent specific systems and environments, for the specific purpose of identifying agents colluding. It will be important to assess this from a variety of AI and security perspectives given the broad abilities and unique vulnerabilities in AI agents. For example, the goal of collusion could be exfiltration of data through distributed, and obfuscated channels in the agent interactions. Alternatively, the collusion could be aimed at subtle or targeted misalignment of goals, via a data poisoning mechanism to misbehave in specific situations and shift blame to benign and naive agents.
  - **What needs to be done:** Distinguishing malicious collusion from the routine collaborations needed for specialized agents to coordinate complex task planning and execution is likely to be subtle. To accurately assess collusion between AI agents, it will be important to develop a detailed threat model to identify the variety of channels which could be used for coordination, matrixed with the AI specific ways in which agents can be misused. Relative priorities and concerns can be estimated and used to guide experimental tests of such collusions and test proposed defenses. This would be one of several use case scenarios that would benefit from establishing a common test environment, as part of the security infrastructure noted in Section 4.4, to maintain consistent fidelity of signal observations and comparability of proposed defenses.

- Agents for detection and reporting rogue agents in the environment <📊,⚙>

  - **What this means:** In contrast to the reputation scoring, which is based on passive monitoring, this approach would actively deploy agents into the environment, to interact with systems and components, looking for indicators of malicious agents.
  - **What needs to be done:** Hopefully, the number of offensive agents operating in an environment will be sparse, therefore it may be necessary to draw on sampling theories to inform the ratio of active defender agents needed to find rogue agents. Systematic characterization of the behavior patterns of offensive AI agents performing different types of offensive goals, can also be leveraged to design defensive agents most likely to be at the right place and time for intercepting offensive agents. The exact process to report and mitigate rogue malicious agents and their actions is also an open need also discussed in security goal 3.1 and security goal 1.3, respectively.

- Honeypots for AI agents <📊>
  - **What this means:** Honeypots are an active mechanism used to induce interaction with agents to assess their maliciousness. This concept is commonly used in IoT and network security, where synthetic systems are created to identify, distract and assess attackers [19]. The honeypot may not be an agent itself, it could for example be a database, website, or other open resource.
  - **What needs to be done:**We acknowledge this is not a new idea, as prototype honeypot systems have been proposed in literature [28]. However, particularly as agent abilities and architectures evolve, the defenses, including honeypots, must also adapt to maintain their effectiveness.

- Research into how stealthy and evasive agents can be <📊>
  - **What this means:** AI enabled agents will have a unique capacity to learn with significant depth, including in speed and scale and patience. The conventional wisdom for assessing likely attack paths is in part based on cost, but also the practical constraints of humans, their patience, working hours and precision to execute phases of the attack. As demonstrated in the DARPA ACE AI-Dogfight, the AI system could perform actions with higher precision, and novel attacks not previously practical were being successfully executed in real-world test scenarios [15]. While much of the attention on AI in cyber security is focused on the aid given to actors with limited ability to perform sophisticated actions and for attacks to occur at extreme speed, the converse can be equally worrying. The aid given to the most sophisticated actors, to perform attacks with even greater subtlety is equally disruptive to assumptions of what is considered likely or achievable.
  - **What needs to be done:** Research into these topics can be approached from a variety of directions. There can be great utility from simply re-examining the threat models from the intentional perspective of identifying assumptions that could break in slow-developing offensive attacks. An experimental approach can also be leveraged, rewarding stealth and evasion to intentionally draw out potential novel approaches.

4.4 Security Infrastructure

Finally, there was a strong and important recommendation from the workshop participants which overlays each of pillars and security goals detailed in section 4: that in order to advance the security objectives for AI agents, the community needs a standardized, configurable platform for

agent simulation and evaluation, including security simulation and evaluation. We are aware that standardized evaluation can be an ironic term in AI, given that hundreds of benchmark datasets have been crafted to assess specific capabilities of individual models, and each benchmark has different strengths and weaknesses. This variety is even more complicated in Agent evaluation, because there is another dimension of assessment, the environment in which the agent is operating.

The lessons from agent evaluation from Reinforcement Learning (RL), should be leveraged to inform AI agent evaluation systems. In RL, the environment is an approximation to the scenario the agent is designed to interact with, E.G. chess, go, World of Warcraft etc. For AI agents to perform everyday tasks such as software development, billing and payment automation, report generation, supply chain planning, etc, AI agents will be integrated into most types of digital infrastructure. Their "environment" may be an enterprise network, a server, a cloud compute instance, a car, phone, or even a utility grid. The agent's task in these environments will vary, but from a security perspective, the actions they perform will be on real- world, internet connected systems.

The recommendation is to produce a high fidelity standardized sandbox environment, representative of these types of scenarios, to evaluate agents operating in these types of systems. There are a variety of real-world systems for which it is difficult or impossible to procure high-fidelity models (e.g., SCADA systems in the utility grid), but it is necessary to do so for robust security assessments. The standardized test environment would reduce that barrier by either being publicly accessible or easily reproducible, to ensure any agent developer could have high quality security evaluation tools.

Chapter 5

Next Steps in AI Agent Security

The advent of sophisticated AI agents marks a paradigm shift, introducing unprecedented capabilities alongside novel and complex security challenges. As detailed throughout this report, the autonomy, adaptability, and potential scale of AI agents fundamentally alter the cybersecurity landscape, creating vulnerabilities in the agents themselves, risks to the assets and goals entrusted to them, and new vectors for malicious actors to exploit systems and third parties. The inherent opacity of foundation models, combined with the emergent behaviors possible in multi-agent systems, necessitates a departure from traditional security assumptions and practices.

Our workshop was focused on articulating steps the research and development community should take to secure AI agents. During the workshop we did not establish specific priorities relative to each other or to the different user subcommunities. However, the workshop organizers did identify several follow-on questions which would benefit from multi-disciplinary collaborative gatherings:

- What is most likely to be different about cyber offense run by an agent vs a human?
- Which standards or protocols will provide the most utility to improving AI security validation and intelligence sharing?
- What tasks are security critical, but unlikely to be commercialized due to insufficient productization potential?
- What tasks are most aligned to the strengths of subcommunities (government or academic research, startup companies, frontier AI labs, cloud compute providers, etc), and the needs of different end users (AI consumers, AI engineers, AI researchers)?

We look forward to future coordinating gatherings which can focus on and further articulate answers to these questions.

The path toward a secure AI agent ecosystem is complex and evolving. However, by fostering interdisciplinary collaboration, prioritizing the development of security-centric technologies and standards, and proactively addressing the unique risks outlined herein, the community can work to ensure that the transformative potential of AI agents is realized safely and responsibly. The long-term trustworthiness and widespread adoption of AI agents hinge on the immediate and diligent development of robust security practices.

Bibliography

[1] aicyberchallenge.com, . URL https://aicyberchallenge.com/.

[2] CyberSecEval 4 | CyberSecEval 4, . URL https://meta-llama.github.io/PurpleLlama/CyberSecEval/.

[3] Introducing the Model Context Protocol, . URL https://www.anthropic.com/news/model-context-protocol.

[4] Model Context Protocol, . URL https://github.com/modelcontextprotocol.

[5] Towards Building Safe & Secure Agentic AI - Lecture 12, Dawn Song, . URL https://www.youtube.com/watch?v=ti6yPE2VPZc.

[6] TrojAI, . URL https://www.iarpa.gov/research-programs/trojai.

[7] Zero Trust Maturity Model. Technical Report V2.0, Cybersecurity & Infrastructure Security Agency (CISA), April 2023. URL https://www.cisa.gov/resources-tools/resources/zero-trust-maturity-model.

[8] 2024 Zero Trust & Encryption Study. Technical report, Ponemon Institute, Sponsored by Entrust, May 2024. URL https://www.entrust.com/sites/default/files/documentation/reports/entrust-ponemon-institute-2024.pdf.

[9] google-a2a/A2A, June 2025. URL https://github.com/google-a2a/A2A. original-date: 2025-03-25T18:44:21Z.

[10] Sahar Abdelnabi, Amr Gomaa, Eugene Bagdasarian, Per Ola Kristensson, and Reza Shokri. Firewalls to Secure Dynamic LLM Agentic Networks, February 2025. URL http://arxiv.org/abs/2502.01822. arXiv:2502.01822 [cs].

[11] Emmanuel Ameisen, Jack Lindsey, Adam Pearce, Wes Gurnee, Nicholas L. Turner, Brian Chen, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. Circuit tracing: Revealing computational graphs in language models. *Transformer Circuits Thread*, 2025. URL https://transformer-circuits.pub/2025/attribution-graphs/methods.html.

[12] Dario Amodei. The Urgency of Interpretability. URL https://www.darioamodei.com/post/the-urgency-of-interpretability.

[13] Alexander Bick, Adam Blandin, and David J. Deming. The Rapid Adoption of Generative AI. Working Paper W32966, National Bureau of Economics Research, Cambridge, Mass, 2024.

[14] Zhaorun Chen, Mintong Kang, and Bo Li. ShieldAgent: Shielding Agents via Verifiable Safety Policy Reasoning, March 2025. URL http://arxiv.org/abs/2503.22738. arXiv:2503.22738 [cs].

[15] Christopher R DeMay, Edward L White, William D Dunham, and Johnathan A Pino. AlphaDogfight Trials: Bringing Autonomy to Air Combat. *Johns Hopkins APL Technical Digest*, 36(2), 2022.

[16] Zehang Deng, Yongjian Guo, Changzhou Han, Wanlun Ma, Junwu Xiong, Sheng Wen, and Yang Xiang. AI Agents Under Threat: A Survey of Key Security Challenges and Future Pathways, June 2024. URL http://arxiv.org/abs/2406.02630. arXiv:2406.02630 [cs].

[17] OWASPGenAIProject Editor. Agentic AI - Threats and Mitigations. Technical report, OWASP, February 2025. URL https://genai.owasp.org/resource/agentic-ai-threats-and-mitigations/.

[18] Fetch.ai. Agents Library, September 2022. URL https://github.com/fetchai/uAgents. original-date: 2022-09-28T17:31:19Z.

[19] Javier Franco, Ahmet Aris, Berk Canberk, and A. Selcuk Uluagac. A Survey of Honeypots and Honeynets for Internet of Things, Industrial Internet of Things, and Cyber-Physical Systems, August 2021. URL http://arxiv.org/abs/2108.02287. arXiv:2108.02287 [cs].

[20] Wenbo Guo, Yujin Potter, Tianneng Shi, Zhun Wang, Andy Zhang, and Dawn Song. Frontier AI's Impact on the Cybersecurity Landscape, April 2025. URL http://arxiv.org/abs/2504.05408. arXiv:2504.05408 [cs].

[21] Idan Habler, Ken Huang, Vineeth Sai Narajala, and Prashant Kulkarni. Building A Secure Agentic AI Application Leveraging A2A Protocol, May 2025. URL http://arxiv.org/abs/2504.16902. arXiv:2504.16902 [cs].

[22] Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. On the biology of a large language model. *Transformer Circuits Thread*, 2025. URL https://transformer-circuits.pub/2025/attribution-graphs/biology.html.

[23] Yuhan Liu, Yuyang Huang, Jiayi Yao, Zhuohan Gu, Kuntai Du, Hanchen Li, Yihua Cheng, Junchen Jiang, Shan Lu, Madan Musuvathi, and Esha Choukse. DroidSpeak: KV Cache Sharing for Cross-LLM Communication and Multi-LLM Serving, December 2024. URL http://arxiv.org/abs/2411.02820. arXiv:2411.02820 [cs].

[24] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model Cards for Model Reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 220–229, January 2019. doi: 10.1145/3287560.3287596. URL http://arxiv.org/abs/1810.03993. arXiv:1810.03993 [cs].

[25] Vineeth Sai Narajala and Idan Habler. Enterprise-Grade Security for the Model Context Protocol (MCP): Frameworks and Mitigation Strategies, May 2025. URL http://arxiv.org/abs/2504.08623. arXiv:2504.08623 [cs].

[26] Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as Operating Systems, February 2024. URL http://arxiv.org/abs/2310.08560. arXiv:2310.08560 [cs].

[27] PricewaterhouseCoopers. PwC launches AI agent operating system to revolutionize AI workflows for enterprises. URL https://www.pwc.com/us/en/about-us/newsroom/press-releases/pwc-launches-ai-agent-operating-system-enterprises.html.

[28] Reworr and Dmitrii Volkov. LLM Agent Honeypot: Monitoring AI Hacking Agents in the Wild, February 2025. URL http://arxiv.org/abs/2410.13919. arXiv:2410.13919 [cs].

[29] Mikel Rodriguez, Raluca Ada Popa, Four Flynn, Lihao Liang, Allan Dafoe, and Anna Wang. A Framework for Evaluating Emerging Cyberattack Capabilities of AI, April 2025. URL http://arxiv.org/abs/2503.11917. arXiv:2503.11917 [cs].

[30] Sam Sabin. Exclusive: Anthropic warns fully AI employees are a year away, April 2025. URL https://www.axios.com/2025/04/22/ai-anthropic-virtual-employees-security.

[31] Ranjan Sapkota, Konstantinos I. Roumeliotis, and Manoj Karkee. AI Agents vs. Agentic AI: A Conceptual Taxonomy, Applications and Challenges, May 2025. URL http://arxiv.org/abs/2505.10468. arXiv:2505.10468 [cs].

[32] Shengye Wan, Cyrus Nikolaidis, Daniel Song, David Molnar, James Crnkovich, Jayson Grace, Manish Bhatt, Sahana Chennabasappa, Spencer Whitman, Stephanie Ding, Vlad Ionescu, Yue Li, and Joshua Saxe. CYBERSECEVAL 3: Advancing the Evaluation of Cybersecurity Risks and Capabilities in Large Language Models, September 2024. URL http://arxiv.org/abs/2408.01605. arXiv:2408.01605 [cs].

[33] Christian Schroeder de Witt. Open Challenges in Multi-Agent Security: Towards Secure Systems of Interacting AI Agents, May 2025. URL http://arxiv.org/abs/2505.02077. arXiv:2505.02077 [cs].

[34] Yingxuan Yang, Huacan Chai, Yuanyi Song, Siyuan Qi, Muning Wen, Ning Li, Junwei Liao, Haoyi Hu, Jianghao Lin, Gaowei Chang, Weiwen Liu, Ying Wen, Yong Yu, and Weinan Zhang. A Survey of AI Agent Protocols, April 2025. URL http://arxiv.org/abs/2504.16736. arXiv:2504.16736 [cs].

[35] Dionisio Zumerle and Jeremy D'Hoinne. How to Secure Custom Built AI Agents. Technical Report G00824390, Gartner, March 2025.

[36] Egor Zverev, Sahar Abdelnabi, Soroush Tabesh, Mario Fritz, and Christoph H. Lampert. Can LLMs Separate Instructions From Data? And What Do We Even Mean By That?, January 2025. URL http://arxiv.org/abs/2403.06833. arXiv:2403.06833 [cs].

Appendix A

Workshop Structuring Questions

For completeness and visibility into the process used to aggregate the information used in this report, we provide the specific questions used to prompt discussions in each breakout session of the workshop.

	Agents and Architectures	**Measures/signals, specification, and evaluation**	**Future of Cyber and Tradecraft**
Block 1 Setting the Scene and Biggest Concerns	• What security threats from AI agents are most concerning, and why? • The use of AI agents in offensive cyber operations? • The security threats arising from our inability to adequately secure the agents themselves? • What attacks and defenses are already happening in the space of AI agents? • What new opportunities would the proliferation of AI agents provide for the cyber attacker and defender, in both the single agent and multiagent cases? • Do any new or distinctive security concerns arise in the setting of multiple interacting LLM-driven AI agents, including deception, manipulation, fraud/impersonation, trust, binding contracts between agents, privacy or data use violations, or malicious collaboration? • Does the existence of agents that can generate and execute arbitrary code (including creating additional agents) give rise to new security concerns in agent systems?		

Table Continued

	Agents and Architectures	**Measures/signals, specification, and evaluation**	**Future of Cyber and Tradecraft**
Block 2 Securing AI agent components and communications from third parties (including other agents)	2A: What are the most important security goals we need in order to achieve security of AI agents from third parties (including other agents)? Are there new security goals that the use of agents introduces? (e.g. AI control, verifiable user traceability, new types of authorization credentials given to agents, certification, verification of agent behavior or traits) What are the unique architectural components of AI agents that warrant unique protections (from third parties)? What are the technical limitations that prevent us from making progress on this topic?	2B: What unique threat models and signals/signatures apply to securing AI agents from 3rd parties? What detectable signals or signatures would be uniquely produced when attacking an AI agent or multiagent system? Or detect an AI agent has been attacked? What are the technical limitations that prevent us from making progress on this topic?	2C: What are the additional security challenges of defending an AI agent from attack by another agent. Are there novel supply chain attacks that would target AI agents? Are there novel logical attacks that would target AI agents? What are the technical limitations that prevent us from making progress on this topic?

Table Continued

	Agents and Architectures	**Measures/signals, specification, and evaluation**	**Future of Cyber and Tradecraft**
Block 3 Securing the assets and goals the user has entrusted to an agent, from unintended or unapproved exploitation. Ensure the users' entrusted assets and goals are secure from unintentional and unapproved exploitation, including from the agent itself.	3A: From an architecture perspective, what is different about securing assets vs securing goals and how does architecture influence their integrity? How do we provide assurances to users about AI agent behaviors (e.g., that this agent tries to achieve the users' goals rather than some maliciously set goals, this agent never shares certain information, etc.) Examples could include framework properties, monitoring by other agents, intervening in the reasoning stack, etc. How can we ensure integrity and prevent corruption of AI agent goals and assets against interactions with untrusted data and individuals? What is needed for a Secure by Design agent? What constraints/modifications would we want to make to agents and their platforms to make them more secureable? (e.g. perhaps agent actions should go through a standardized interface of pre-defined actions that support monitoring/defining what types of actions are blocked, etc.) What are the technical limitations that prevent us from making progress on this topic?	3B: What techniques can we use to specify and analyze the security properties of AI agent assets and goals? Examples include specifying agent roles and capabilities, discovery of other agents, fraud/impersonation, trust, binding contracts between agents, or malicious collaboration? How can we evaluate whether AI agents are secure against comparably capable (or moderately more capable) AI agents, in order to ensure defense-dominance in AI agent security? What are the properties most likely to determine the dominance of one agent manipulating another. What is needed to achieve trustworthy and human-understandable traces of AI activity? (What did my agent do on my behalf? How can I trust the answer?) What are the technical limitations that prevent us from making progress on this topic?	3C: When an autonomous cyber agent attacks the assets and goals of another agent, what are the weakest targets in the infrastructure and in the AI elements. What non-traditional techniques could be leveraged to secure assets, and particularly goals. (e.g. not authentication) What assumptions does the security community make that might break in a world where AI agents are common? • Changing attack costs and evolution of tradecraft • "Most likely attack path" was dependent on cost, risk and goals of a human attacker • What security threats are enabled with agents that were not possible before • Fully novel attack types • Code generation at massive scale What are the technical limitations that prevent us from making progress on this topic?

Table Continued

	Agents and Architectures	**Measures/signals, specification, and evaluation**	**Future of Cyber and Tradecraft**
Block 4 Malicious agents will be a fact of life. How do we secure third parties from the effects of malicious agents?	4A: How do we ensure critical infrastructure is secure against malicious AI agents (including ones that are comparable to current malicious hackers but with 10,000x the time/capacity) What would it take to have community buy-in for new agent specific communication protocols that improve attribution of actions and act as a control on taking actions. What kind of networks of (partial) trust and commitment can we achieve? How can we enable AI agent developers to robustly constrain how their AI agents can be used by clients (allowing legitimate uses and tasks for their agents but disabling malicious uses) What types of processes can be used to prevent an agent from taking harmful actions, or working to harmful goals through seemingly benign actions? What are the technical limitations that prevent us from making progress on this topic?	4B: How can we detect a malicious agent? Are there evaluations we can perform, especially given the recent results on alignment faking? What new signatures can we develop to detect malicious activity by agents? Should we have specific indicators for human vs agent elements in systems? What existing domains, communities, frameworks should we be learning from, beyond existing cybersecurity practices? E.g., blockchain and smart contracts? What are the technical limitations that prevent us from making progress on this topic?	4C: How would we ensure the robust containment of a malicious (misaligned or manipulated) or at least highly capable AI agent? How should the security community address loss of control scenarios? What new tools are needed to effectively use traditional security techniques (e.g., encryption, integrity assurance, authentication, RBAC) with agents? What new attack insertion points, vulnerabilities, exploitations, and adversarial behaviors would be enabled by the deployment of AI agent systems? What are the technical limitations that prevent us from making progress on this topic?

Table Continued

	Agents and Architectures	**Measures/signals, specification, and evaluation**	**Future of Cyber and Tradecraft**
Block 5	• What's missing from our analysis and what have we not covered? • Which of the suggested technical limitations above are... – Exceptionally important? – Exceptionally easy/low-hanging fruit? – Would have an immediate impact? • Are there pre-requsite needs to addressing the technical limit? • What is the first step? • What useful data should we collect and what experiments should we perform? How should success be measured? • What projects should the community take on? • What follow-up conversations and convenings are needed? • When will the security of AI agents or of multi-agent systems be protecting more than $1B in value?		

"‘