CCC response to NIST-2025-0035

Submission

This document contains the response from the Confidential Computing Consortium to NIST 2026-0035. The following questions are addressed:
1a, 1b, 1c, 1d, 1e
2a, 2b, 2c, 2d, 2e
3a
4d
5a, 5b, 5e

Submission contact: Mike Bursell, Executive Director, Confidential Computing Consortium.
mbursell@contractor.linuxfoundation.org

About the CCC

The Confidential Computing Consortium (CCC) brings together hardware vendors, cloud providers, and software developers to accelerate the adoption of Trusted Execution Environment (TEE) technologies and standards.

CCC is a project community at the Linux Foundation dedicated to defining and accelerating the adoption of Confidential Computing. It embodies open governance and open collaboration that has aided the success of similarly ambitious efforts. The effort includes commitments from numerous member organizations and contributions from several open source projects.

Members bring expertise from multiple areas and disciplines, including hardware and software architecture, design and development, AI, operational experience and regulatory and standards involvement.

Introduction to Confidential Computing

The Confidential Computing Consortium defines Confidential Computing as “the protection of data in use by performing computation in a hardware-based, attested Trusted Execution Environment”, and identifies three primary attributes for what constitutes a Trusted Execution Environment: data integrity, data confidentiality, and code integrity. As described in “Confidential Computing: Hardware-Based Trusted Execution for Applications and Data”, four additional attributes may be present (code confidentiality, programmability, recoverability, and attestability) but only attestability is strictly necessary for a computational environment to be classified as Confidential Computing.

Confidential Computing (CC) capabilities protect against unauthorized accesses and data leaks while enabling collaboration and compliance. Encrypting data at rest, in transit and while processing using Confidential Computing technology further allows sensitive workloads to move to the cloud even without trusting the cloud provider (admins, hypervisors, etc.).

Responses to Questions

1. (a) **What are the unique security threats, risks, or vulnerabilities currently affecting AI agent systems, distinct from those affecting traditional software systems?**

- **MCP Specific Threats:** As a foundational enabler for the Agentic AI systems, MCP brings a new set of threats for Agentic AI systems, e.g. Shadow MCP servers, tool poisoning and confusion attacks. These risks are mitigated by running MCP servers in attested, hardware-based TEEs, eliminating the deployment of shadow servers, preventing the poisoning of the tools and configuration, and ensuring that the AI Agents interact with the right MCP servers.
- **Privacy-Specific Threats related to Context & Memory Scraping.** Unlike traditional software where data is transient, AI agents maintain long-running "context windows" or memories containing sensitive user data (PII) and retrieval results. A key vulnerability is the potential for **memory scraping** or **cold boot attacks** by a malicious host or hypervisor to extract this unencrypted context from RAM or system hardware.
- **Model protection:** If the model is sensitive, it must be encrypted while at rest. However, the software that runs the model must decrypt it before it can be used. As a result, the decryption key must be available to the application, often in cleartext within the application image, which creates the risk that an attacker could access the model in its decrypted form.

(b) How do security threats, risks, or vulnerabilities vary by model capability, agent scaffold software, tool use, deployment method, hosting context, use case, and otherwise?

- **Hosting Context** (Cloud vs. Edge) will have different threat modeling profiles. Agents hosted in **public clouds** face the "untrusted host" problem, where the model weights and inference data are vulnerable to cloud provider and operator access due to standard virtualization techniques. This risk is mitigated by **Confidential Computing (use of attested, hardware-based TEEs)** which protects data-in-use. Edge deployments may have fewer controls than a restricted datacenter and may benefit from some TEE protections, though TEEs do not provide blanket protection against physical attacks.
- **Tool Use:** Agents with access to **sensitive tools** (e.g., banking APIs) or **sensitive data** increase the "blast radius" of a key compromise. The risk scales with privilege levels of the cryptographic keys the agent holds. Confidential Computing isolates these components from other entities, providing confidentiality and integrity for sensitive operations and data.

- **Accelerators**: Agents deployed in multi-tenant environments are at risk from attacks by other tenants and infrastructure providers. This includes the shared use of accelerators like GPUs, which without proper security controls creates another attack vector to compromise confidentiality or integrity of data. Confidential Computing (TEE) capabilities are increasingly available in accelerators like GPUs.

(c) To what extent are security threats, risks, or vulnerabilities affecting AI agent systems creating barriers to wider adoption or use of AI agent systems?

- **Privacy Barrier:** Assurances that "data in use" is private and confidentiality protected (including from cloud provider insiders) is an important driver in regulators sectors. The wide scale deployment of **Confidential Computing** across these industries for deploying agents that process PHI or Material Non-Public Information is slowed because assurance is hard to demonstrate consistently across environments. The rise in requirements for digital sovereignty capabilities exacerbates this issue.
- **IP Barrier:** Model owners are hesitant to deploy proprietary agents on third-party infrastructure such as enterprise, edge or third-party clouds without **model weight protection** provided by Confidential Computing.
- Data loss and irrevocable decisions - Customers have concerns about data exfiltration and loss due to compromised AI Agent components like MCP servers, and tools with which MCP servers communicate. Customers also have concerns about irrevocable or damaging decisions and execution that AI Agents can perform due the new class of threats/risks, which will/is inhibiting adoption and deployment of AI agent systems.
- The lack of a comprehensive, end-to-end authentication framework that addresses all trust boundaries in the MCP ecosystem exposes critical attack vectors if an Agent AI workload is running in an attested, hardware-based TEE. MCP Authentication is optional by design (and instead should be mandatory and standardized) while vital data structures signing procedures, such as authenticity proofs for MCP Bundles, are at-best platform-specific instead of based on universal and uniform requirements. Some package managers, for example, do not require any form of package signing for MCP packages: a vulnerability that could allow MCP servers to install dependencies at runtime rendering every startup an actual supply chain risk. If such vulnerable MCP endpoints terminate within an attested, hardware-based TEE, the confidentiality of an entire Agentic AI workload is constantly at risk.

(d) How have these threats, risks, or vulnerabilities changed over time? How are they likely to evolve in the future?

- **Evolution:** Threats have shifted from simple input manipulation (prompt injection) to complex **state exfiltration, agent scheming,** and **guardrail misalignment**.
- **Future:** We expect attacks to target the **cryptographic identity** of agents. As agents become autonomous economic actors (paying for services), threats will evolve toward stealing the **wallet keys, decryption keys** or **signing keys** held in the agent's memory. Future mitigation will require hardware-rooted identities (e.g., TPM/TEE-backed keys).

(e) What unique security threats, risks, or vulnerabilities currently affect multi-agent systems, distinct

from those affecting singular AI agent systems?

- Inter-Agent Trust and “confused deputy” attacks are going to become the next set of challenges within multi-agent systems. In multi-agent systems, a malicious or compromised agent can trick another into **sharing sensitive data**. The fundamental problem is the lack of **proper authentication protocols extended with attestation**. Without a mechanism for one agent to cryptographically verify the code and integrity of another agent before sharing a secret or data, the system is vulnerable to **"confused deputy"** attacks.

2. Security Practices for AI Agent Systems

(a) What technical controls, processes, and other practices could ensure or improve the security of AI agent systems in development and deployment?

- **Confidential Computing (TEEs):** Deploying agents within attested, hardware-based Trusted Execution Environments to ensure data is encrypted in memory during processing.
- **Cryptographically Assured Workload Identity:** Assigning unique, ephemeral identities to agents, rooted in hardware attestation, rather than using static API keys.
- **Key and Secret Management:** Using **Key Broker Services** that only release decryption keys or API credentials to an agent *after* successful Remote Attestation of the corresponding workload.
- **Sandboxing and Runtime Isolation**: Agents should be deployed in sandboxing environments that limit their operational scope and abilities. Utilizing open source technologies like Kata Containers (e.g., https://agent-sandbox.sigs.k8s.io/) ensures that agent activities remain isolated from the host system. Because agents frequently handle credentials and sensitive data, these sandbox environments must also be hardened using projects like CNCF Confidential Containers (https://confidentialcontainers.org/) to take advantage of Confidential Computing capabilities.
- **Cryptographically Assured Workload Identity:** An ephemeral identity document represents a short-lived, dynamically-issued cryptographic identity that exists only for the duration of a workload's execution (seconds to minutes) and is derived via remote attestation of platform and workload rather than being a pre-shared secret.
- **Verifiable Identity:** Each agent should have a cryptographic identity bound to its code via remote attestation of the TEE in which it is executing, preventing "rogue" agents from impersonating legitimate ones. This can also allow the operator of the system hosting the agent to prove that they have no ability to tamper or inspect the agent during execution.

(b) To what degree, if any, could the effectiveness of technical controls vary with changes to deployment method use case and otherwise?

- Remote attestation - of platform and Agentic AI workload - is a requirement to provide assurances of protection and isolation by TEEs: Confidential Computing, by definition, requires use of attested, hardware-based TEEs, where the phrase “hardware-based” specifies that the isolation mechanism is provided by hardware (rather than, for instance, hardware only providing a Root of Trust for Reporting).

- Mechanisms to allow remote attestation to be used both to provide assurances of the suitability of a platform for a particular Agentic AI workload and, subsequently, that the correct Agentic AI workload has been instantiated, need to be put in place.
- Agents making use of or being deployed across accelerators, e.g., GPUs or NPUs, in multi-tenant environments inherit and take on the attack vectors on that shared hardware. In such deployments Confidential Computing accelerators should be used. These devices can build secure connections with workloads deployed in Confidential Computing virtual machines.
- TEEs are highly effective in **cloud and enterprise deployments** for isolating agents from the provider. However, their effectiveness on physically vulnerable **edge devices** (IoT) depends on the specific hardware support for physical tamper resistance.
- Agents with high performance requirements might face latency overhead from memory encryption, though modern hardware (hardware-accelerated TEEs) minimizes this. Start-up times may also be impacted by set-up and remote attestation requirements.

(c) How might technical controls need to change, in response to the likely future evolution of AI agent system capabilities?

- As AI agents increasingly distribute workloads across specialized accelerators, the primary security risk evolves from centralized host vulnerabilities to distributed lateral movement and data exposure across internal buses and accelerators. A lack of hardware-enforced isolation between the GPU (reasoning), DPU/IPU (infrastructure/networking), and CPU (logic) allows a compromise in one domain to jeopardize the agent's entire execution context, including its KV cache, proprietary weights, and identity keys. To mitigate this, technical controls must shift toward composite Confidential Computing models. These models utilize a unified attested, hardware-based Trusted Execution Environment (TEE) that extends protection across the entire multi-accelerator pipeline. Organizations can verify the integrity of the full computational path before any sensitive agentic state is processed.
- As agents become more autonomous, controls must shift from "perimeter security" to "Agentic Zero Trust." Every inter-agent interaction must employ an authentication protocol extended with remote attestation, such as attested TLS, ensuring that "who the agent is" (identity) is cryptographically bound to "what the agent is running" (code measurement) and policies as to “what the agent can do”(authorization).

(d) What are the methods, risks, and other considerations relevant for patching or updating AI agent systems throughout the lifecycle?

- Patching stateful agents is complex in all cases, and use of Confidential Computing requires particularly careful design and implementation, given the protections that are provided in terms of confidentiality and integrity and the importance of maintaining stateful identity across the patching process. Some approaches to patching a stateful agent in a TEE require "Secure State Migration." This may require memory state to be encrypted, transferred to a new (patched) enclave, and decrypted. This requires a "Migration Enclave" to manage the key exchange and ensure the state is never exposed to the host during the update.

2 (e) Which cybersecurity guidelines, frameworks, and best practices are most relevant to the security of AI agent systems?

i. What is the extent of adoption by AI agent system developers and deployers of these relevant guidelines, frameworks, and best practices?

- **Confidential Computing Consortium (CCC):** Guidelines on attestation and data-in-use protection.

ii. What are impediments, challenges, or misconceptions about adopting these kinds of guidelines, frameworks, or best practices?

- **Remote Attestation for Device Assignation:** the secure mapping of hardware-isolated physical and virtual resources plays a critical role for Confidential AI workloads where, for example, a dedicated physical accelerator is assured to be in the same Confidentiality Computing system boundary as the execution environment the AI workload is running in. Standards for this are currently not fully defined.
- **Attested TLS protocols:** the quality of existing attested TLS protocols varies significantly, with known examples of replay, relay, and diversion attacks [See references]. Further work is required to improve the quality of these protocol designs and implementations, though implementations do exist that allow TLS connections to be protected using remote attestation.
- Logging and monitoring: many standard mechanisms for logging and monitoring often rely on inspection of AI agent or workload memory by external components or parties. The isolation provided by Confidential Computing prevents naive implementations of this kind of approach, and so alternative mechanisms must be used.

iii. Are there ways in which existing cybersecurity best practices may not be appropriate for the security of AI agent systems?

- **Remote Attestation of Tenants by the Platform:** due to the desired properties of isolation and confidentiality, a platform cannot conduct run-time integrity measurements of a confidential workload.

3. Assessing the Security of AI Agent Systems

3 (a) **What methods could be used during AI agent systems development to anticipate, identify, and assess security threats, risks, or vulnerabilities?**

- As much as possible of the TCB should be open source to allow analysis of possible threats, risks and vulnerabilities. Techniques such as hackathons, bug bounty programs and paying contributors and maintainers of this code can also be employed.
- Formal methods for high-assurance [Reference].

4. Limiting, Modifying, and Monitoring Deployment Environments

4 (d) What methods could be used to monitor deployment environments for security threats, risks, or vulnerabilities?

i. What challenges exist to deploying traditional methods of monitoring threats, risks, or vulnerabilities?

Many existing monitoring mechanisms rely on inspection of workload memory, which is explicitly prevented using Confidential Computing. So while this is one of key benefits of Confidential Computing, new or different mechanisms need to be employed to provide appropriate monitoring of AI agents.

ii. Are there legal and/or privacy challenges to monitoring deployment environments for security threats, risks, or vulnerabilities?

None known.

iii. What is the maturity of these methods in research and practice?

Confidential Computing is widely deployed in enterprise, government and defense settings globally.

(e) Are current AI agent systems widely deployed on the open internet, or in otherwise unbounded environments? How could the volume of traffic be tracked on the open internet or in otherwise unbounded environments over time?

5. Additional Considerations

5. (a) What methods, guidelines, resources, information, or tools would aid the AI ecosystem in the rapid adoption of security practices affecting AI agent systems and promoting the ecosystem of AI agent system security innovation?

- Promoting easy-to-use no code changes approaches that allow developers to "lift and shift" AI agents into TEEs without rewriting code.
- Ready-made SDKs, binaries, services and best practice guides for verifying attestation reports so non-experts can implement "Verify-then-Trust" logic.
- Enhance code-generators to include the option of generating code that runs in hardware-enabled TEEs, and enabled for remote attestation, with configurable policies and secure 'key and/or secret release' protocols implemented in the code. This sandboxing and isolation inclusion should be policy driven for the co-pilots/code-generation models.

(b) In which policy or practice areas is government collaboration most urgent?

- **Standardizing "Confidential AI" is a must.** The government should define a standard for "Level X Confidentiality" where the AI provider *technically cannot* see user data. This would unlock AI

adoption in government and defense, working closely with the Confidential Computing Consortium.

- **Key Management Standards should be part of the "Confidential AI" standards.** Policies requiring that AI agents capable of financial transactions must hold keys in FIPS 140-3 validated HSMs or TEEs.

(e) **Are there practices, norms, or empirical insights from fields outside of artificial intelligence and cybersecurity that might benefit our understanding or assessments of the security of AI agent systems?**

- **Hardware protected keys:** The practice of using Hardware Security Modules (HSMs) in security sensitive applications should be extended to AI Agents acting as autonomous entities that hold keys and sign transactions. Whereas in legacy systems, code might rely on physical HSM for key operations like signing, AI Agents could also employ "Soft HSMs" backed by hardware based Trusted Execution Environments. This will enable more dynamic deployment while still requiring the same rigor in key lifecycle management.

References

Confidential Computing: Hardware-Based Trusted Execution for Applications and Data - https://confidentialcomputing.io/wp-content/uploads/sites/10/2023/03/CCC_outreach_whitepaper_updated_November_2022.pdf

A Technical Analysis of Confidential Computing v1.3 - https://confidentialcomputing.io/wp-content/uploads/sites/10/2023/03/CCC-A-Technical-Analysis-of-Confidential-Computing-v1.3_unlocked.pdf

Common Terminology for Confidential Computing - https://confidentialcomputing.io/wp-content/uploads/sites/10/2023/03/Common-Terminology-for-Confidential-Computing.pdf

Protecting Agentic AI Workloads with Confidential Computing (CCC blog article) https://confidentialcomputing.io/2026/01/20/protecting-agentic-ai-workloads-with-confidential-computing/

Remote Attestation with Exported Authenticators https://datatracker.ietf.org/doc/draft-fossati-seat-expat/

RFC - 9334 - Remote ATtestation procedureS (RATS) Architecture - https://datatracker.ietf.org/doc/rfc9334/