Network Working Group
Internet-Draft
Intended status: Informational
Expires: 1 September 2026

Y. Ni
C. P. Liu
Q. Gao
Huawei
Z. Li

28 February 2026

Security Requirements for AI Agents
draft-ni-a2a-ai-agent-security-requirements-01

Abstract

This document discusses security requirements for AI agents, covering different stages of security interactions. These include provisioning, registration, discovery, cross-domain interconnection, and access control.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://liuchunchi.github.io/draft-ni-a2a-ai-agent-security-requirements/draft-ni-a2a-ai-agent-security-requirements.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ni-a2a-ai-agent-security-requirements/.

Discussion of this document takes place on the WG Working Group mailing list (mailto:WG@example.com), which is archived at https://example.com/WG.

Source for this draft and an issue tracker can be found at https://github.com/liuchunchi/draft-ni-a2a-ai-agent-security-requirements.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Copyright Notice

Table of Contents

1. Introduction

With the widespread application of agentic AI technology across various business scenarios, its security issues have become increasingly prominent.

This document aims to provide an architecture addressing security requirements across different stages of interactions of Agentic AI use cases. These include provisioning, registration, discovery, cross-domain interconnection, and access control. This document establishes a starting point to guide Agentic AI security design, development, and implementation consideration discussions.

The target audience of this document would be IETF security experts that wish to understand AI Agent's behaviorial patterns, so to evaluate if the proposed security requirements are worthy of further security designs.

2. Architecture

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

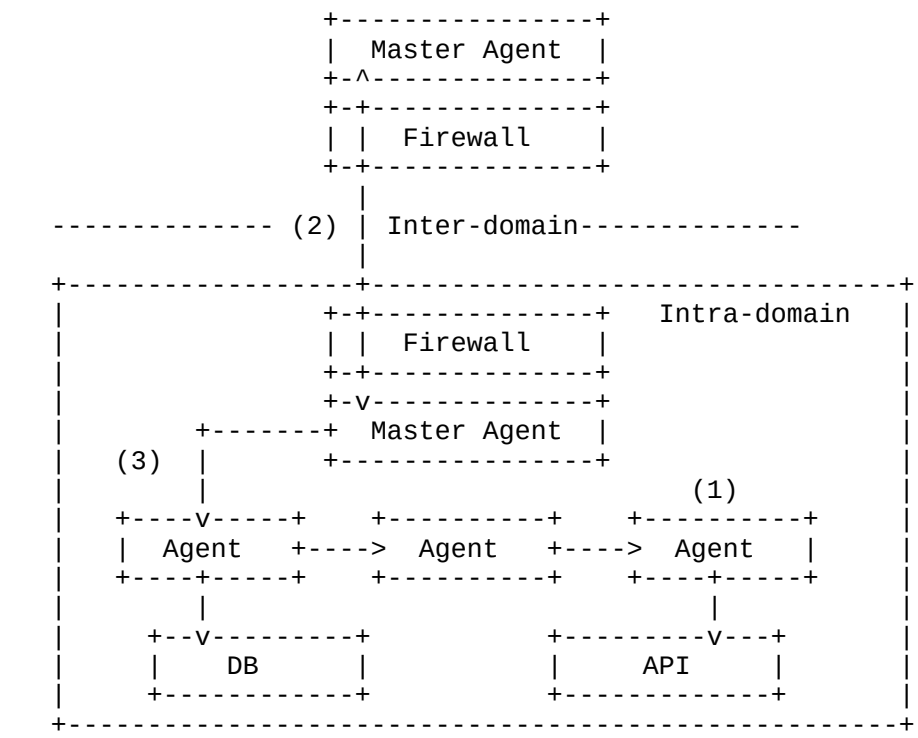

Figure 1. Architecture of Agent Security Control and Management

The architecture of agent security control and management is illustrated in Figure 1. There are four types of security interactions, in a sequential order:

1. Provisioning, Registration, and Discovery: Creating agent identity, establishing initial trust, provisioning agent secrets and credentials, onboarding agents to enable discovery.

2. Cross-domain Interconnection: Enabling secure, authenticated communication between agents across different trust domains.

3. Access Control: The Master Agent validates both intra-domain and inter-domain access tokens, creates internal workflow and manages different credentials for heterogeneous systems.

Therefore, the architecture includes four components:

1. Firewall: A network security device designed to monitor, filter, and control incoming and outgoing network traffic based on predetermined security rules.

2. Master Agent: The central orchestrating entity that manages multi-agent operations, including cross-domain communication, workflow coordination, credential management, and security policy management.

3. Agents: Autonomous software entities deployed in various domains to perform specific tasks.

4. Heterogeneous systems: API endpoints, microservices, tools, and databases.

The above architecture is from the perspective of a service flow. From the identity management perspective, we recommend reusing IETF works like WIMSE. This draft [I-D.draft-ni-wimse-ai-agent-identity-01] discusses WIMSE applicability to Agentic AI.

3. Provisioning, Registration, and Discovery

Figure 2 shows the diagram of provisioning and registration, which includes Agent Certificate Authority (ACA) and Agent Registry Service (ARS):

1. ACA (Agent Credential Authority): A trusted third party that issues and manages credentials for agents. Credential formats include but not limited to: X.509 certificates, identity tokens, etc.

2. ARS (Agent Registry Service): A system responsible for agent identity registration and discovery-matching.

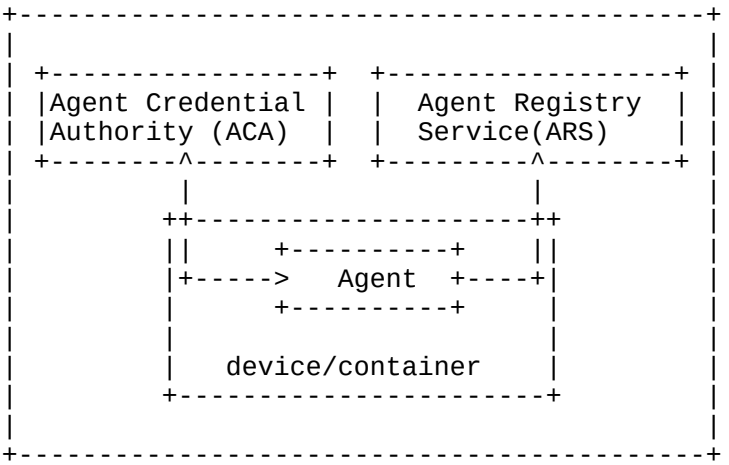

Figure 2. Diagram for Provisioning and Registration

3.1. Identity Provisioning and Management

Identity provisioning and management are the process of creating and assigning a verifiable digital identity to an agent.

* Initial Trust Establishment: Initial trust can be established through one or more of the following trust anchors, including, but not limited to: a manufacturer-embedded immutable credential like an IDevID certificate; a hardware root of trust like a Trusted Platform Module (TPM) or Hardware Security Module (HSM); identity documents like an AWS Instance Identity Document or an Azure Managed Service Identity token. This step verifies the agent's execution environment (device, container, etc.) as trustworthy, allows the device or container to join the network, thereby enabling secure operations for all subsequent steps.

* Credential Request: During a credential request, the agent must provide multiple proofs of its legitimacy, could include, for example, but not limited to:

  - Proof of Possession (PoP): A Certificate Signing Request (CSR) or other PoP forms signed with the agent's private key, demonstrating that the agent holds the private key corresponding to the requested identity.

  - Remote Attestation Evidence or Result: A set of security-relevant claims about the Target Environment submitted to a RATS Verifier (could be the ACA), which reveals operational status, health, configuration, or construction.

  - AI Bill of Materials (AIBOM): A comprehensive inventory that details the agent's supply chain, including models, datasets, configurations, dependencies, and related infrastructure. This prevents the use of vulnerable AI components.

  - Provider Endorsement: A digital signature or credential from the Agent Provider, ensuring the agent originated from a trusted source.

  - Identity Binding: A cryptographic binding to a specific human user or an organizational role to specify on whose behalf the agent operates and its authorized scope.

* Credential Issuance: The ACA validates proofs and requests from the above two steps, if passed, it issues an agent-specific credential that may include its owner or requester identity, capabilities, locator, acceptable validation methods for the ARS.

* Credential Lifecycle Management: The ACA not only issues credentials but also defines and enforces revocation policies. These policies are triggered by specific events, such as a detected security compromise, the agent's scheduled decommissioning, or a key rotation.

3.2. Secret Management

AI Agents SHOULD NOT have direct access to secrets due to new threats like Prompt Injection. AI Agents SHOULD reuse secret management modules on the platform it operates on, for example, cloud secret managers or TEE/keystore/keychains on smart devices. Best practices like secret/credential generation, rotation and revocation apply. Agents SHOULD only obtain temporary access tokens or signed messages via a secure API or other kind of trusted intermediary. Guardrails also apply for general secret information exfiltration prevention.

3.3. Agent Registration

After receiving a credential from the ACA, the agent then sends it to the ARS to authenticate itself and start the registration process.

* Authentication: The ARS must verify the legitimacy of the credential submitted by the agent. It must be signed or otherwise endorsed by the ACA.

* Registration: The ARS then checks if the information signed by the ACA, such as the agent's capabilities, exactly matches the registration request sent by the agent. Upon successful validation, the ARS assigns the agent a unique identifier and establishes an agent record that links the identifier to its attributes.

* Record Management: This step automatically removes expired credentials and synchronizes with the ACA to ensure timely revocation of credentials, preventing the use of invalid or compromised credentials.

3.4. Agent Onboarding

Agent onboarding differs between campus and cloud environments. On campus, agents use protocols like EAP-TLS for network access. In the cloud, the process involves injected sidecars, which register agents to the central service mesh registry automatically to enable communication and management.

3.5. Agent Discovery

After agent onboarding, the discovery process enables entities (e.g., a human user, an agent, etc.) to find and connect with registered agents.

* Authentication: The ARS must authenticate the entity initiating the discovery request. The requester is required to present a valid identity credential.

* Capability Filtering and Matching: The ARS performs dynamic filtering based on the requester’s identity and query and returns only agent records relevant to the query, enforcing the principle of least privilege at the discovery layer.

4. Cross-Domain Interconnection

4.1. Cross-Domain Identifier Interoperability

Different domains may use distinct identifier schemas. Possible methods include:

* pre-configured schema translation

* cross-domain identifier synchronization

* a universal parsing framework or system

4.2. Secure Cross-Domain Transmission

Mutual TLS (mTLS) connection starts from the external requesting agent to the master agent. The master agent terminates the mTLS connection and parses the application layer requests. In this case, the master agent functions as an OAuth resource server, and manages internal task orchestration.

4.3. Authenticating External Calls

The master agent then verifies the identity of the requesting agent, and whether or not it has permission to the requested service or agent. Different authentication methods might be possible:

* API keys
* Username-password
* Pre-shared secrets
* Assertions (for example, JWT Authorization Grant[I-D.draft-ietf-oauth-identity-chaining-06])

which can even be combined with AND/OR logic. During this process, the master agent might be able to identify the caller endpoint type:

* human user via browser or app
* human user via API
* AI agents
* Hardware or equipment via an IoT API

4.4. IAM Integration

Since the agent may inherit its access rights from its owner or user, when authenticating requests, the validation might require integration of IAM systems for redirected verification.

5. Access Control

5.1. Authorization Handling

The master agent acts as the OAuth 2.1 resource server and a Policy Enforcement Point (PEP). Its responsibilities are as follows:

* Token Validation: The master agent must validate access tokens as described in OAuth 2.1 Section 5.2. If validation fails, it must respond according to OAuth 2.1 Section 5.3 error handling requirements.
* Fine-Grained Policy Enforcement: The master agent serves as a PEP that queries a PDP (Policy Decision Point), such as Open Policy Agent (OPA), to evaluate the requester’s access request. The PDP functions by taking the master agent’s query, pre-configured

policies (supporting RBAC, ABAC, ReBAC models, etc.), and data as inputs to decide whether the requester is authorized for its intended action. The PDP then returns the final decision to the master agent for enforcement.

5.2. Authorization Models

In enterprise situation, Role-based Access Control (RBAC) Attribute-based Access Control (ABAC) or Adaptive Access Control (AdBAC) are different access control models used in practice. Regarding access control models, there are 2 ways forward:

1. whatever the authorization model used in the enterprise itself applies to AI Agents. This leaves 2 cases possible:

    1. The agent carry the identity and inherit access rights from its owner (a human or a department). Carring such human identity will help security control points make decisions with sufficient context, and to the discretion of its internal security policy plus access control model.

    2. The agent does not carry the identity from its owner. It carries independent security contexts rich enough for access control.

2. AI Agents require a new authorization model completely.

This section would require more discussion for best current practices.

5.3. Authorization Chaining Across Domains

In an agentic AI use case, a request may traverse multiple master agents in multiple trust domains before completing. It is common that the requesting agent from domain A needs to access the master agent of domain B. During this process, the following information should be preserved:

* Original requesting agent identity

* Authorization context

    - Scope

    - Resource

    - Audience

- Grant type

- Assertion

* Agent-to-Agent Context

The current best practice is [I-D.draft-ietf-oauth-identity-chaining-06], which can preserve the above information during a cross-domain token exchange process. This ensures that internal resource servers perform independent secondary authorization instead of blindly trusting the master agent's upstream validation, preventing the privilege abuse of the master agent and unauthorized lateral movement.

5.4. Converting to Internal Workflow

* Workflow Generation: Complex tasks often require multi-agent collaboration. The master agent receives, parses, and extracts the original job request from the external requesting agent, then creates sequential workflows or parallel calls. This requires the master agent to have information of all callable internal API assets, agent capabilities, etc.

* Downscoping: If the master agent intends to use a workflow, it extracts the original caller's identity and authorization context, and initiates a new internal workflow. It should follow the current least privilege best practice of downscoping-Transaction Tokens as specified in [I-D.draft-tulshibagwale-oauth-transaction-tokens-05]. The access rights to each downstream workload decrease.

* Agent-to-Agent Context: the Agent-to-Agent context and intent of the original requester must be preserved and propagated throughout the workflow to avoid authorization drift and context poisoning as specified in [I-D.draft-liu-oauth-a2a-profile-00].

5.5. Interoperability for Heterogeneous Systems

Within a domain, there might exist different types of heterogeneous systems or legacy systems that require different authentication methods. They could be API endpoints, microservices, tools or databases. The exact authentication methods are determined by the service itself, for example,

* identity tokens

* API keys

* pre-shared secrets

* username-passwords

* X.509 certificates, etc.

As a result, the master agent also works as an intermediary credential manager that converts the formats, scopes, identity of the credential, bridging the gap between heterogeneous systems and platforms.

Examples include:

* Static secrets (API keys) to be exchanged to short-lived, on demand credentials (identity tokens)

5.6. Zero Trust Analysis

The above information can be used as rich context that allows zero trust access control. There are three additional aspects can be implemented to enhance the zero trust framework:

* Remote Attestation Results: For the PEP at the master agent or the internal resource server, Remote attestation results could also be part of the inputs, which could include the following information:

  - RoT and trust anchors

  - Identifiers

  - Affiliations

  - Posture assessment results

  - Capabilities

* Continuous Observability: The system should utilize OpenTelemetry (OTel) to track each call across agents, sending OTel’s telemetry, which records call frequency, error rates, and behavioral anomalies, etc. to the PDP for real-time assessment.

* Microsegmentation: Based on the telemetry data, PDP can issue software-defined security policies to PEP at the perimeter of each segment to enforce microsegmentation, in order to prevent lateral movement of security risks. Possible granularity of microsegmentation includes:

  - per IP segment/subnet

- per each workload

- per tags and attributes (of workload), etc.

6. IANA Considerations

This document has no IANA actions.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

7.2. Informative References

[I-D.draft-ietf-oauth-identity-chaining-06] Schwenkschuster, A., Kasselman, P., Burgin, K., Jenkins, M. J., and B. Campbell, "OAuth Identity and Authorization Chaining Across Domains", Work in Progress, Internet-Draft, draft-ietf-oauth-identity-chaining-06, 12 September 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-oauth-identity-chaining-06>.

[I-D.draft-liu-oauth-a2a-profile-00] Liu, P. C. and N. Yuan, "Agent-to-Agent (A2A) Profile for OAuth Transaction Tokens", Work in Progress, Internet-Draft, draft-liu-oauth-a2a-profile-00, 20 October 2025, <https://datatracker.ietf.org/doc/html/draft-liu-oauth-a2a-profile-00>.

[I-D.draft-ni-wimse-ai-agent-identity-01] Yuan, N. and P. C. Liu, "WIMSE Applicability for AI Agents", Work in Progress, Internet-Draft, draft-ni-wimse-ai-agent-identity-01, 20 October 2025, <https://datatracker.ietf.org/doc/html/draft-ni-wimse-ai-agent-identity-01>.

[I-D.draft-tulshibagwale-oauth-transaction-tokens-05] Tulshibagwale, A., Fletcher, G., and P. Kasselman, "Transaction Tokens", Work in Progress, Internet-Draft,

draft-tulshibagwale-oauth-transaction-tokens-05, 20 October 2023, <https://datatracker.ietf.org/doc/html/draft-tulshibagwale-oauth-transaction-tokens-05>.

Acknowledgments

TODO acknowledge.

Authors' Addresses

Yuan Ni
Huawei
Email: niyuan1@huawei.com

Chunchi Peter Liu
Huawei
Email: liuchunchi@huawei.com

Qiangzhou Gao
Huawei
Email: gaoqiangzhou@huawei.com

Zhenbin Li
Email: robinli314@163.com