MCP · CAPABILITY SECURITY · PROGRAMMING LANGUAGES

Can MCP tools carry security semantics?

Abstract. Tool descriptions tell a model what a tool does. They do not, by themselves, say what the tool is allowed to do. The system separates read/search from action tools and enforces a consent-token gate for actions; the challenge is making that contract portable and hard to misdescribe.

THE PROBLEM

What minimal capability schema lets a host verify the difference between observe, propose, execute, export, and mutate—before a model is allowed to choose a tool?

HARD CONSTRAINTS

01 A tool name or natural-language description is not an authorization boundary.

02 Provider-specific function schemas must not erase security-relevant fields.

03 Live discovery must not silently expand an agent's authority.

04 A connector must be able to explain the scope it needs before execution.

A FIRST CONTRIBUTION

Define a versioned capability manifest and a compatibility test suite for schema sanitizers, with deliberate malicious and ambiguous tool declarations.

BUILD IN THE OPEN

github.com/hushh-labs/hushh-research
hushh.ai/discord

PRIMARY SOURCE · hushh.ai/developers · docs/AGENT_ARCHITECTURE.md

The read/action split and consent gate exist; portable capability semantics are an open design problem.

Read the research
hushh.ai/research/open-problems