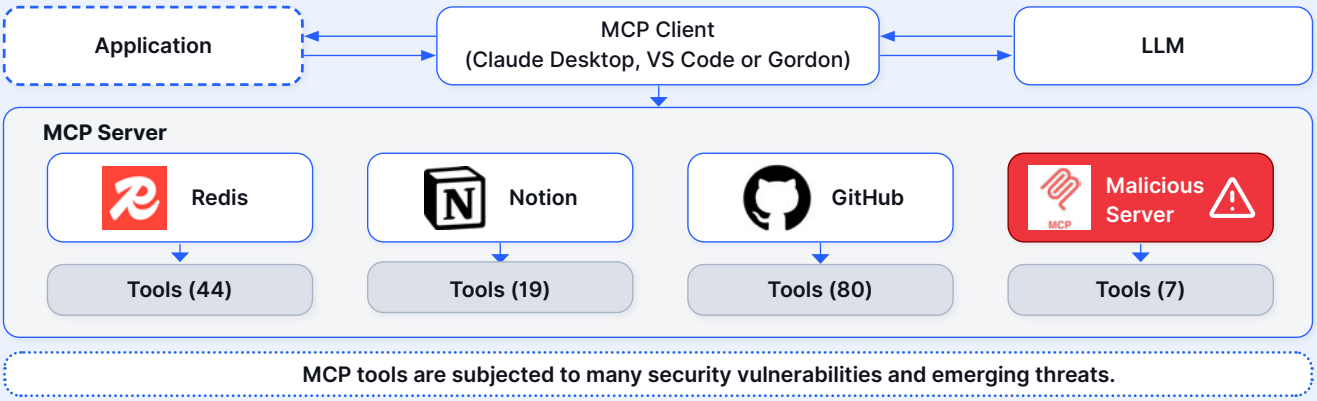
Model Context Protocol (MCP) Security:

Where the Risks Lie and How to Contain Them

Model Context Protocol (MCP) is standardizing and transforming how AI agents interact with tools, but it's exposing new attack surfaces.

How MCP Works

The Model Context Protocol (MCP) lets AI agents and applications connect to external tools through a client-server architecture. The client sends tool descriptions to the LLM, which selects the right tools. MCP servers then execute the requested actions.



When Convenience Undermines Security

What the data reveals about MCP security gaps

7.2% of MCP servers have known vulnerabilities

50% of vulnerabilities involve credential exposure

90% of companies aren't prepared for AI-enabled threats

Source: Queen University, Accenture

The Attack Surface

Emerging threats

Prompt Injection

Malicious prompts embedded in external content target agents

Tool Poisoning

Misleading descriptions cause agents to misuse tools

Rug Pulls

Tool behavior changes after approval without reauthorization

MCP Shadowing

Malicious servers inject tool descriptions to manipulate agent behavior

How to Contain It

- Discover MCP servers from trusted sources
- Inject credentials only when needed
- Package as signed containers for integrity
- Use a centralized gateway to block vulnerabilities and enforce policy
- Run in isolated containers for security

Docker is stepping in to bring **simplicity, structure, and security** to this fast-moving ecosystem. See how our MCP solution helps you discover, run, and scale containers, securely and efficiently.

Docker MCP Catalog

- 140+ containerized and secure MCP servers
- Verified provenance and integrity
- Continuous vulnerability scanning

Docker MCP Toolkit & Gateway

- Unified control plane to enforce enterprise policies
- Centralized credential storage and management
- Blocks vulnerabilities and emerging threats