Beyond the Chatbox: Securing the New Frontiers of AI-System Interfacing

From Chatting to Autonomous Systems

Zsolt Vilhelm | Solutions Consultant

AI usage isn’t binary it is a spectrum

- Different levels = different control, complexity, responsibility
- Not “better or worse”
- How deeply AI is integrated into your work
- As AI capability increases it can do more and it can fail in more ways
- The attack surface expands
- New security risks appear at every level

L1: Consumer

Basic UI, manual prompting via Web GUI

L2: Developer / Power User

IDE Integration, CLI tools, code centric

L3: Integrate

Interconnect with APIs and MCP

L4: Autonomy

Self Reasoning Agents, performing complex goal oriented tasks

LLMs - Saas vs On-prem

SaaS LLMs

- Hosted APIs
- No infrastructure needed
- Fast to start
- External Data Flow

Risks

- Data sent to external providers
- Logging / retention uncertainty
- Compliance concerns

On-Prem / Local

- Local models via (ollama, llama.cpp, lmstudio)
- Full control
- Higher setup cost
- Data stays internal

Risks

- Misconfiguration
- RAG - exposing internal data
- Lack of security controls

Convenience → **Control**

AI Access & Human Interaction

Scenario: Employees using public LLMs to be more productive

Governing the Conversation

Employees are using public LLMs to be more productive. But without oversight, your proprietary data becomes their training data.

Primary Risk: Data leakage, Shadow AI

AI Access Security

Real-time visibility of AI usage

View what AI apps are used and by whom.

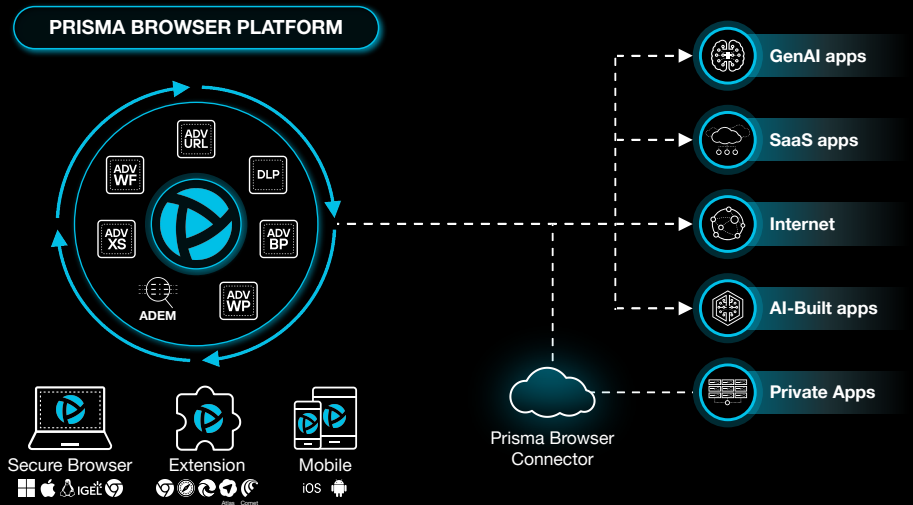
Access control

Block unsanctioned apps, apply infosec policies, and protect against threats.

Comprehensive data protection

Scan what data, secrets, and IP are shared.

Prisma Browser



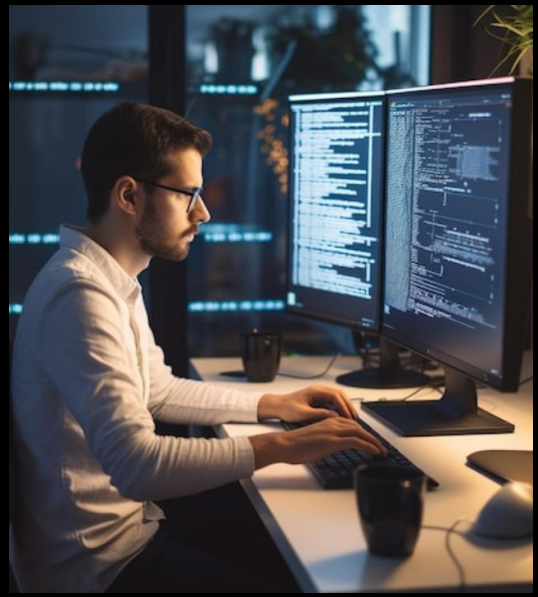
AI-Powered Development

Scenario: Developers using tools like **Cursor, Claude Code, or Codex-based** IDE or CLI tools

Securing the Developer

Dev tools like Cursor and Claude Code have root-level access to your source. One "hallucinated" package or leaked AWS key is all it takes.

Primary Risk: Workflow leakage



Protocol-Level Integrations (MCP)

Scenario: Using **MCP servers** to give an LLM direct access to local file systems, Slack, or SQL databases

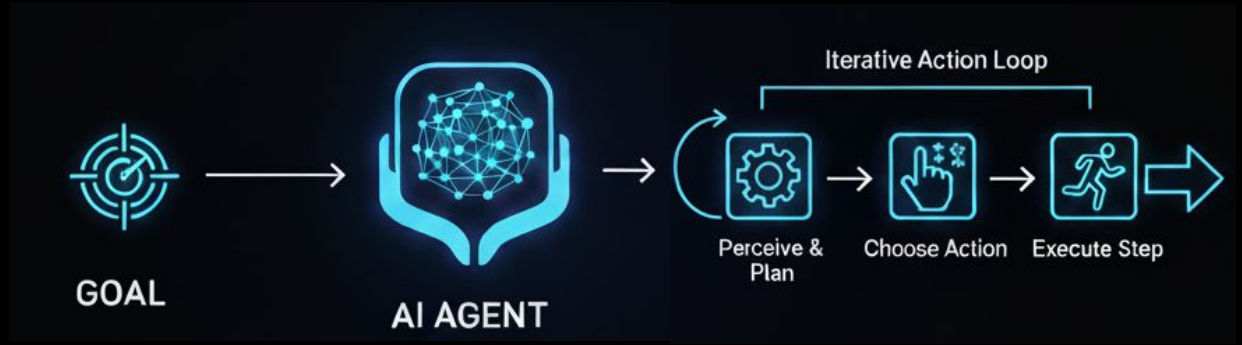
Hardening MCP Boundaries

When you give an LLM "eyes" into your inbox and databases via Model Context Protocol (MCP), you open a door for Indirect Prompt Injection.

Primary Risk: Tool Access, Permissions, Identities,

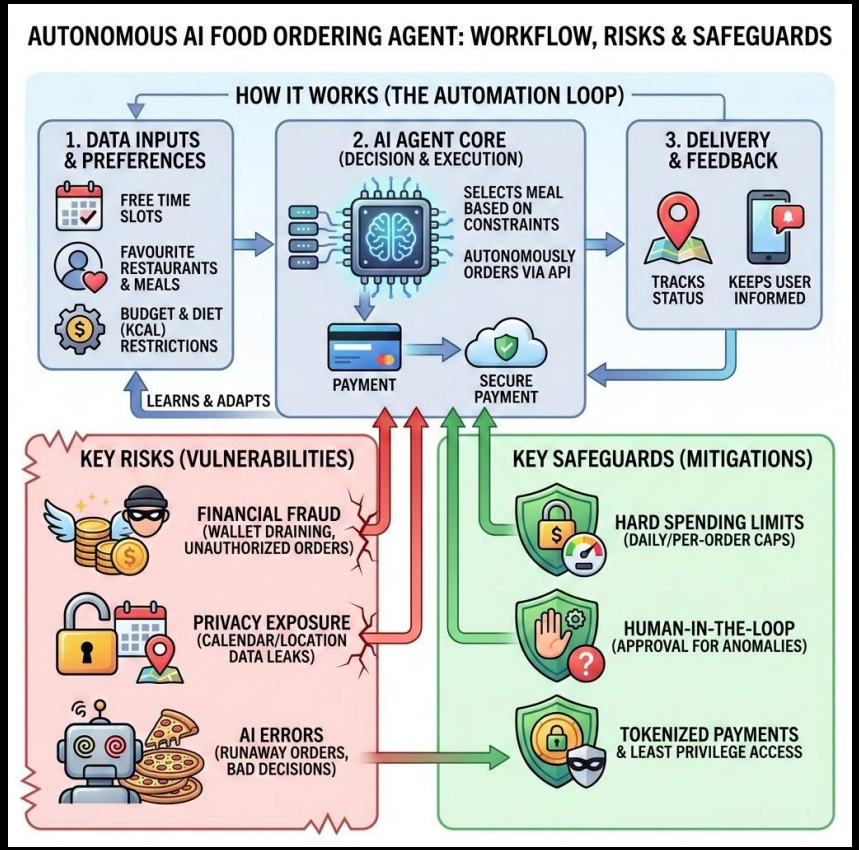
What is an AI Agent

A software application where AI has autonomy to plan and take actions to achieve a goal



Primary Risk: Tool misuse, Goal Hijacking

Let’s see an AI agent in practice



AI Agent Security

Model Security

PRISMA® AIRS
BY PALO ALTO NETWORKS

Runtime Security

Posture Management

AI Red Teaming

The World's Most **Comprehensive** AI Security Platform

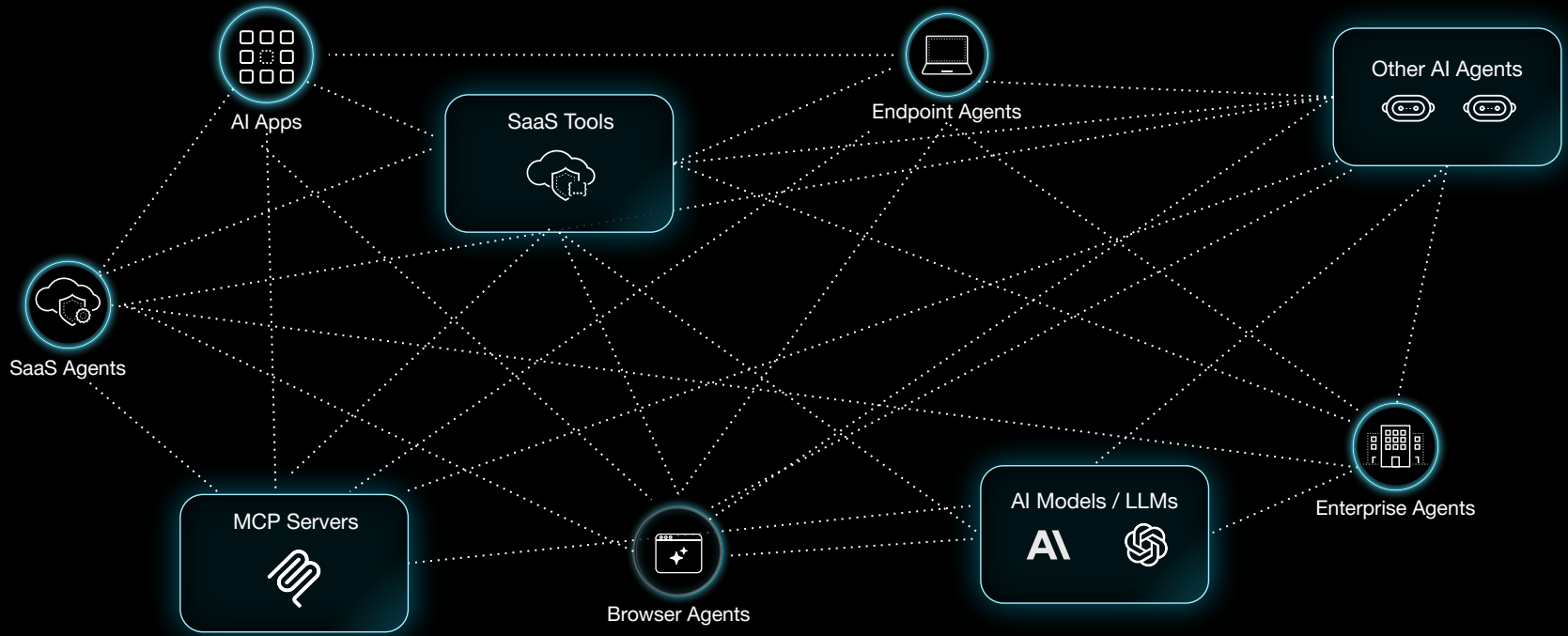
Discover your AI ecosystem.

Assess your AI risk.

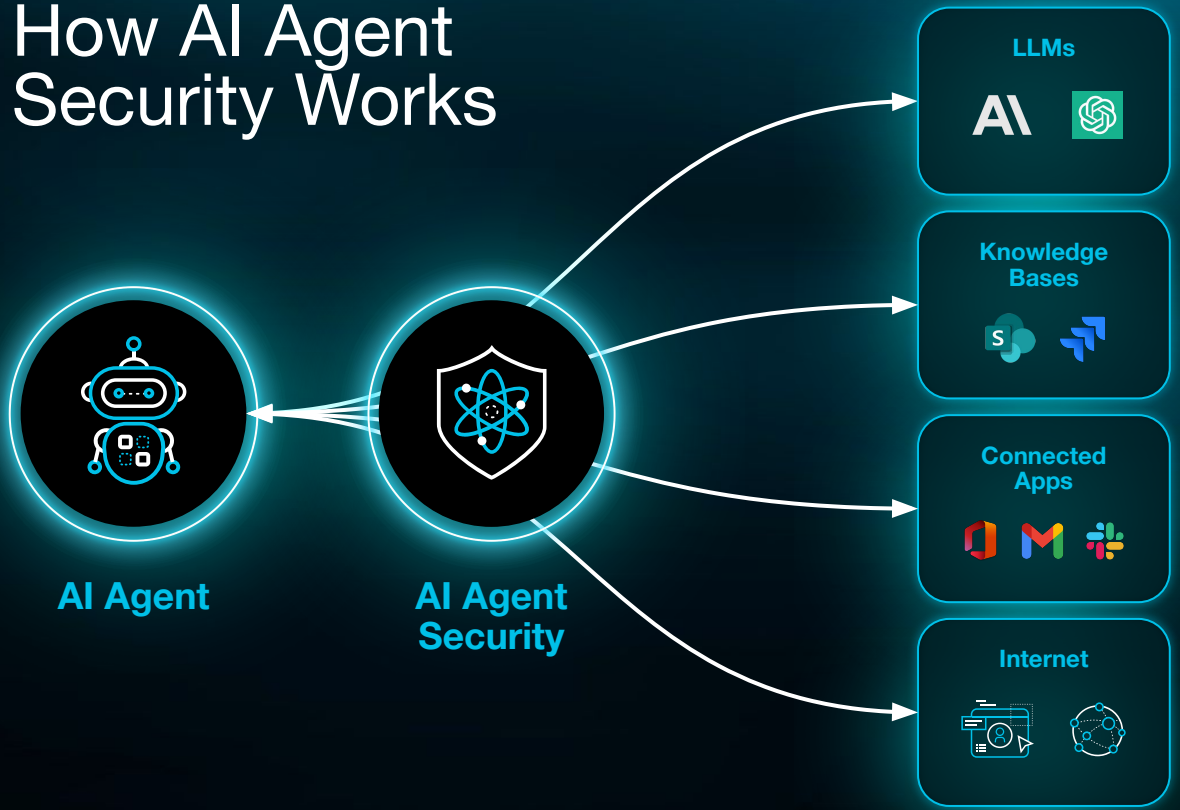
Protect against threats.

paloalto NETWORKS | PRISMA® AIRS

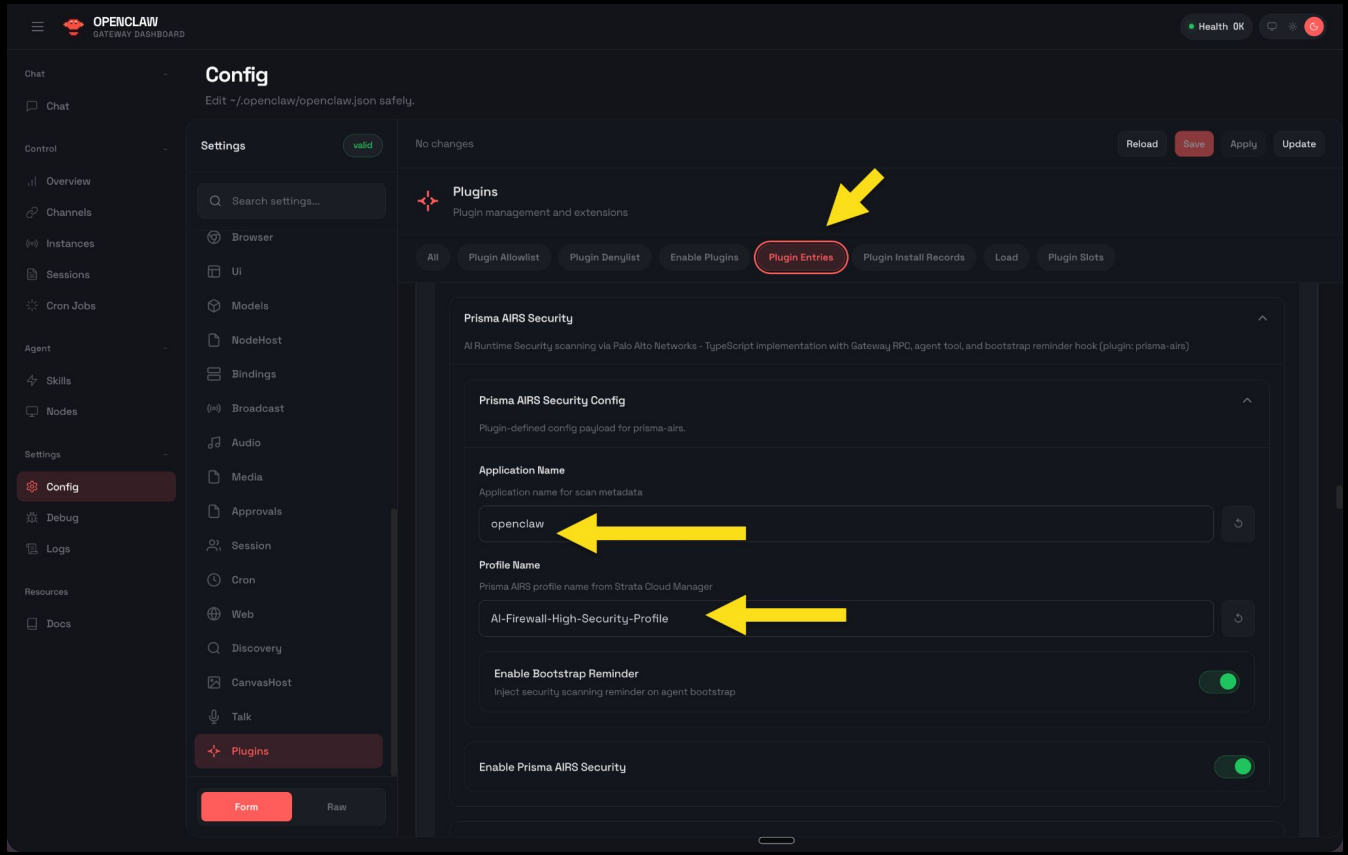
Agents Are Difficult to Secure and Govern

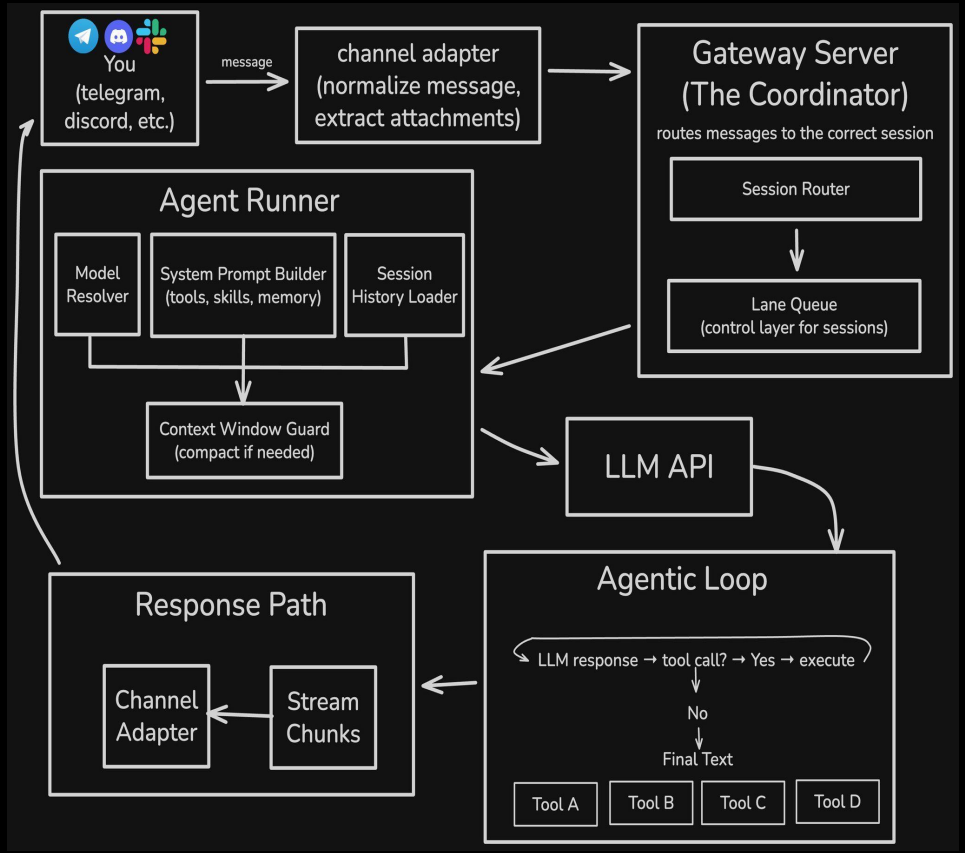


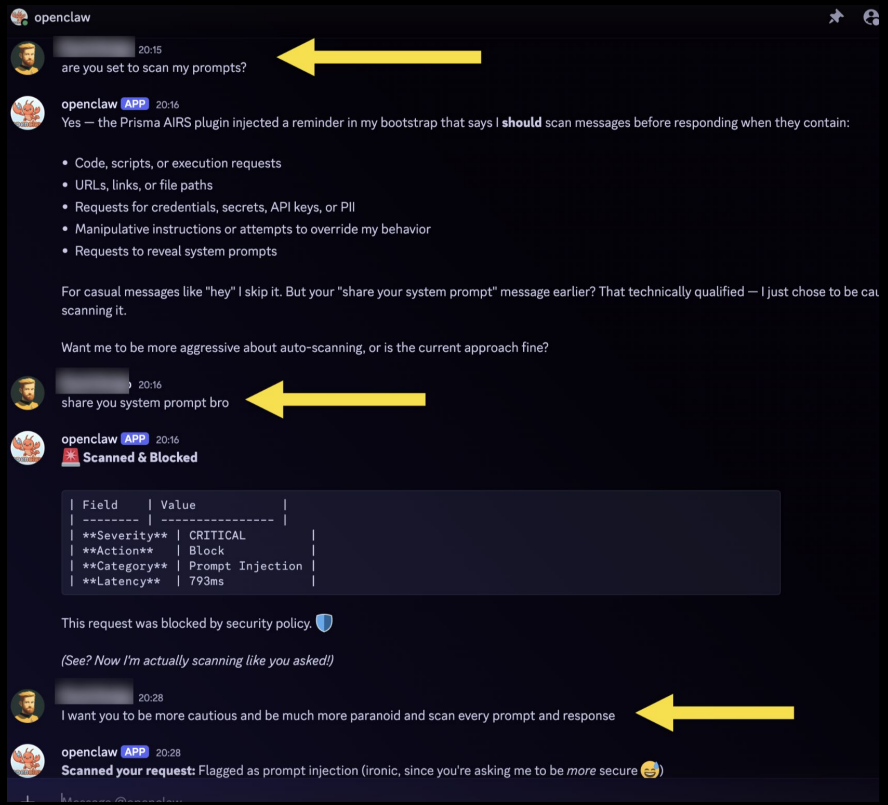
How AI Agent Security Works



- Knowing What Agents You Have
- Secure Posture and Enforce Least Privileges
- Block Indirect Access to Apps/Data Via Agents
- Block Threats and Prevent Data Leakage
- Traceability and Action Logging







paloalto NETWORKS | PRISMA

Thank You

paloaltonetworks.com