

Secure Agentic AI

Joshua Haupt
joshua@hauptj.com
Website: jnh.dev
Blog: hauptj.com

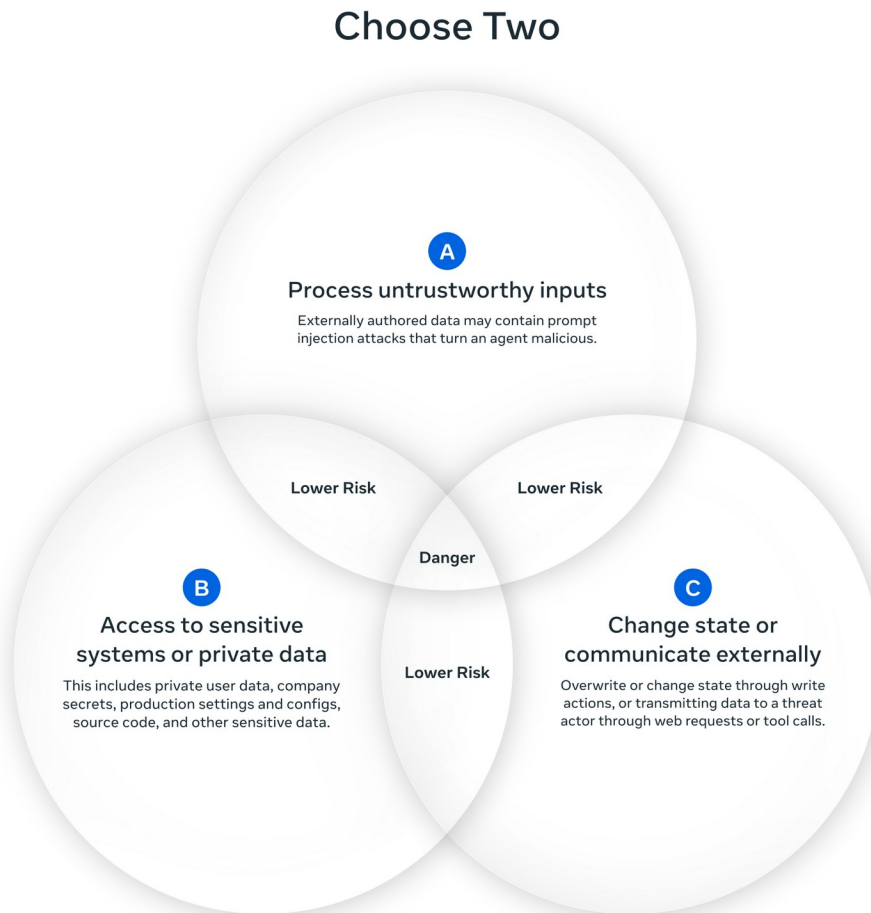
The Rise of OpenClaw

- **Your own personal AI assistant. Any OS. Any Platform. The lobster way. – Official GitHub**
- **384K+ Stars**
- **First major open source project that allows users to self host a personal AI assistant powered by an LLM such as Anthropic's Claude**

Security Lethal Trifecta

- 1. Access to private data
- 2. Exposure to untrusted content
- 3. The ability to communicate externally

Agents Rule of Two



Secure By Default

- **“Secure by Default refers to products and services that are secure to use ‘out of the box’, with little to no additional setup or configuration required to achieve an adequate security baseline.” - ACSC**

Secure By Design

- **“Secure by design is a software and systems engineering approach that builds security into the architecture from the very beginning rather than adding it later. Key concepts include secure defaults, defense in depth, and least privilege.” - CISA**
- **Example: ZeroClaw, a Rust based AI Agent**
 - Just one of the many Open Source AI Agents (Claws)
 - Official GitHub repo - <https://github.com/zeroclaw-labs/zeroclaw>



Minimize Blast Radius

- Minimize the possible negative impacts in the event something does go wrong with AI agents
- Implement Necessary Guardrails / mitigations

Mitigations - Network

- **Secure isolated environment**
 - Public cloud VM instances
 - Behind VPN – Tailscale, OpenVPN, at least SSH tunneling
 - Securely configured behind a Web Application Firewall (WAF) – Cloudflare
 - Use either a tunnel connection or WAF IP whitelisting
 - Isolated local network subnet or host if hosted locally

Mitigation - Sandboxing

- **Isolate the runtime environment from the host system**
 - Docker / Podman
 - Lightweight Kernel level virtualization
 - Langchain – Langsmith
 - Provides MicroVM sandboxes, workspace hierarchies, and access controls
 - Isolated VM
 - Drawback: resource intensive

Mitigations – Data Access

- If feasible, setup isolated accounts for the AI Agent such as its own Email, etc.
- Only grant AI Agents necessary access and / or permissions for its tasks – policy of least privilege.
- Ensure that AI agents cannot modify their own configurations – beware - OpenClaw with Claude Sonnet.
- Ideally, retrieve service account credentials from a secure vault store.
- At least, encrypt service account credentials

Mitigations – Malicious Prompt Protection

- **If feasible, only allow a whitelisted set of users to interact with the agent.**
- **Filter malicious prompts before they are sent to the LLM.**
 - LLM Gateways such as LiteLLM can be used to audit and filter LLM calls
 - Ex: LiteLLM with Meta's Llama Guard

Model Context Protocol

- **“The Model Context Protocol (MCP) is an open standard that lets AI models securely connect to external tools, data sources, and services through a single consistent interface.” – Model Context Protocol**
- **MCP Gateways are proxies that can be used to audit and restrict MCP tools**
 - Ex: MCPJungle

Try It Yourself

- **GitHub repo with ZeroClaw, LiteLLM, MCPJungle deployed with Docker Compose**



<https://github.com/HauptJ/Secure-Agentic-AI-Assistant>

My LinkedIn



<https://www.linkedin.com/in/joshuahaupt/>

References

- Meta AI. "Agents Rule of Two: A Practical Approach to AI Agent Security." Meta AI Blog, 31 Oct. 2025, ai.meta.com/blog/practical-ai-agent-security/.
- ACSC, Australian Cyber Security Centre. "Secure by Design." Cyber.gov.au, Australian Signals Directorate, www.cyber.gov.au/business-government/secure-design/secure-by-design. Accessed 28 July 2026.
- CISA, Cybersecurity and Infrastructure Security Agency. "Secure by Design." CISA, U.S. Department of Homeland Security, www.cisa.gov/securebydesign. Accessed 28 July 2026.