DATASHEET

42Crunch Secure MCP Server

Secure Agent Access, Without Losing Control

Overview

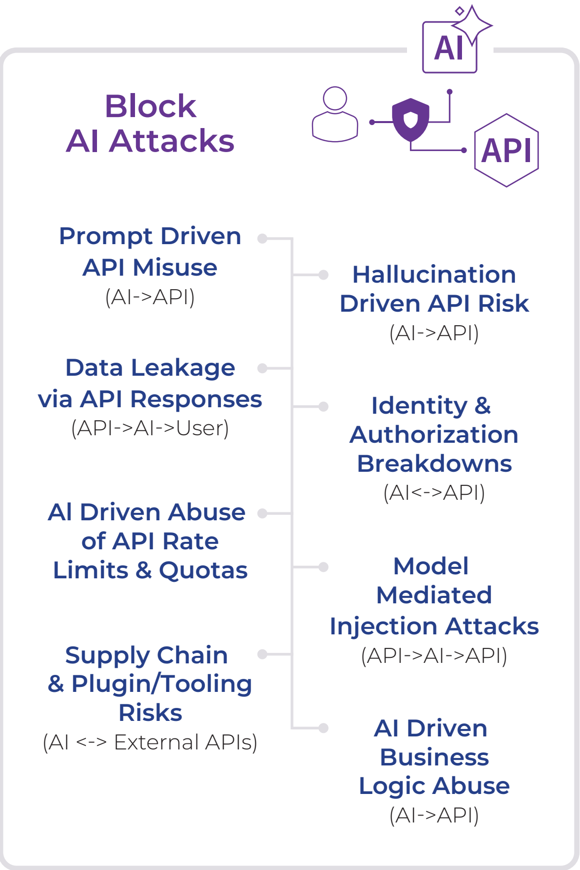
As enterprises accelerate the adoption of agentic AI, a new challenge is emerging: how to safely expose core business capabilities to autonomous AI agents without increasing security, compliance and operational risks.

APIs have long served as the connective tissue of digital enterprises, but the rise of Model Context Protocol (MCP) servers fundamentally changes how those APIs are discovered, accessed, and executed by AI-driven systems. Instead of exposing APIs directly to agents for consumption, MCP servers were developed to provide a standard, model-agnostic way for AI agents to discover, understand, and invoke tools without hard-coding APIs into prompts or models. So while most MCP servers and MCP gateways enable this connectivity today, they do not offer control or governance of the agents. Without strong authentication, fine-grained authorization, and runtime policy enforcement, AI agents introduce a new, poorly governed attack surface where agents operate with little accountability.

Secure MCP Server - Govern and Control Agent Behavior

To enable businesses securely expose their API-based business services via MCP as secure, AI-ready services, 42Crunch has introduced the Secure MCP Server which extends the trust already built in your API infrastructure to the agentic AI execution layer. The 42Crunch Secure MCP Server enables enterprises to safely expose API-driven business services to AI agents and LLM-powered applications using the Model Context Protocol (MCP). Built on 42Crunch's industry-proven API security platform, it provides a security-first control plane that transforms existing APIs into governed, auditable, AI-ready capabilities—without bypassing enterprise security guardrails.

Rather than connecting AI models directly to low-level API operations on a one-to-one basis, the Secure MCP Server introduces a hardened intermediary that enforces API flow contracts, policies, and runtime protections by default, allowing organizations to adopt agentic AI with confidence.



Block AI Attacks

Prompt Driven API Misuse (AI->API)

Hallucination Driven API Risk (AI->API)

Data Leakage via API Responses (API->AI->User)

Identity & Authorization Breakdowns (AI<->API)

AI Driven Abuse of API Rate Limits & Quotas

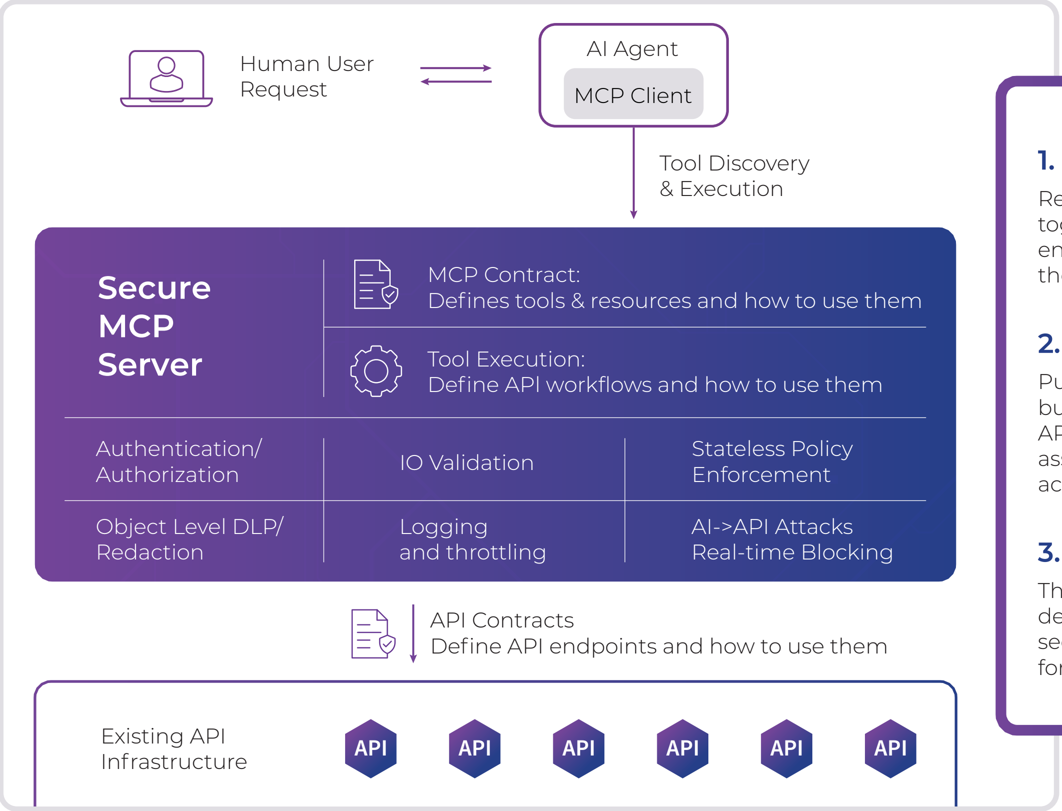
Model Mediated Injection Attacks (API->AI->API)

Supply Chain & Plugin/Tooling Risks (AI <-> External APIs)

AI Driven Business Logic Abuse (AI->API)

How it Works

The 42Crunch Secure MCP Server can be deployed flexibly to align with your own enterprise architecture and governance requirements. There are several options: it is available as a managed service, deployable in customer-controlled environments as a containerized workload for Kubernetes or as a virtual machine.



Human User Request
AI Agent
MCP Client
Tool Discovery & Execution
Secure MCP Server
MCP Contract: Defines tools & resources and how to use them
Tool Execution: Define API workflows and how to use them
Authentication/ Authorization
IO Validation
Stateless Policy Enforcement
Object Level DLP/ Redaction
Logging and throttling
AI->API Attacks Real-time Blocking
API Contracts Define API endpoints and how to use them
Existing API Infrastructure
API
API
API
API
API
API

1. Record

Record the business flow together with your API end points to execute the AI application.

2. Configure

Publish to the platform this business flow describing all API steps, inputs and assertions needed to achieve the business goal.

3. Publish

The platform auto-builds, deploys and operates a secure MCP server ready for access by AI Agents.

Benefits

The 42Crunch Secure MCP Server allows enterprises to move fast with agentic AI—without sacrificing API security, governance, or trust. Backed by 42Crunch's proven API security expertise, it turns APIs into safe, auditable execution surfaces for the next generation of AI-driven systems.

AI-Ready in Minutes, Not Months

Expose existing business services to AI safely without re-architecting your APIs. Pre-governed access removes security friction, letting AI teams move from pilot to production faster.

Uncompromising Security & Governance

Every request is validated, authorized, logged, and controlled—by default. Govern Agent behavior at runtime.

Future-Proof Architecture

A policy-driven MCP abstraction layer that evolves with AI agents, MCP standards, and enterprise requirements at scale.

Data Sovereignty

Control where your data is stored, processed, and accessed when used by agentic AI. Expose critical business capabilities (CRM, ERP, finance, ops) to agents without wiring directly to sensitive data and internal systems.

RoI on Existing Investment

Extend trust from existing API security layer to the agentic AI execution layer.

Maintain Auditability and Accountability

Gain clear visibility into agent activity with auditable execution trails and alignment with compliance and regulatory requirements.

AI Attack Categories & 42Crunch Protection

AI Attack Category & Individual Vectors	OWASP API TOP 10	42Crunch AI Attacks Protection
1. **Prompt Driven API Misuse** (AI → API)	API6:2023 Unrestricted Access to Business Processes. LLMs can be manipulated into making API calls they were never intended to make.	Contract Enforcement: 42Crunch MCP server enforces strict API calls flows so if an AI is tricked into calling an endpoint or method not defined in the flow, it is blocked by the 42Crunch MCP Server.
Prompt Injection → Unauthorized API Calls	Attacker convinces the model to call privileged endpoints.	Access Control Audit: 42Crunch MCP enforces strict schema validation to ensure every endpoint has strict security requirements defined (OAuth/JWT) so the LLM cannot reach them without a valid, scoped user token.
Induced Parameter Manipulation	Model is tricked into sending dangerous parameters (SQL strings, oversized payloads).	Input Validation: 42Crunch MCP Server validates every parameter against the schema (type, regex, min/max length). If an AI sends a "negative quantity" or "SQL string" not permitted by the contract, it is blocked.
Function Calling Abuse	Attacker forces the model to select a sensitive function via schema manipulation.	Strict Schema Validation: 42Crunch ensures the model can only trigger functions and pass parameters that strictly match the documented JSON schema in each API contract defined in the MCP contract.
Workflow Escalation	Multi-step agents are manipulated to chain calls to escalate privileges.	State/Contextual Awareness: 42Crunch MCP Server by enforcing that calls happen within valid session contexts and preventing unauthorized vertical/horizontal movement.
2. **Hallucination Driven API Risk** (AI → API)	API9:2023 Improper Inventory Management. LLMs hallucinate—and hallucinations can become API calls.	Deterministic Protection: Unlike AI-based security that might "accept" a hallucination, 42Crunch uses a "no-guesswork" model that rejects any call not explicitly in the contract.
Hallucinated Endpoints	Model invents endpoints that don't exist, causing error storms.	Shadow API Discovery & Blocking: The 42Crunch MCP Server blocks any request to a non-existent or "shadow" path, preventing hallucinations from ever reaching backend legacy systems.
Hallucinated Parameters	Model fabricates fields that bypass validation.	Negative Testing (Conformance Scan): 42Crunch MCP Server blocks undefined fields.
Hallucinated Identities	Model assumes roles or tokens that don't exist.	Rigorous Auth Validation: The 42Crunch MCP Server validates the integrity and claims of every token (JWT). A hallucinated "fake admin" token would fail cryptographic or signature verification.
3. **Data Leakage Through API Responses** (API → AI → User)	API3:2023 Broken Object Property Level Authorization. AI models amplify leakage because they summarize and expose data.	Response Shielding: 42Crunch MCP Server inspects the outgoing API response. If the backend returns more data than the contract allows, the MCP Server blocks the response reaching the LLM.
Over Disclosure via LLM Summaries	API returns sensitive data; LLM summarizes it for the user.	Schema Conformance: Prevents "Mass Exposure" by ensuring the API only returns the specific fields defined in the MCP contract, stripping out or blocking sensitive internal data.
Cross Record Leakage	LLM merges multiple API responses, violating isolation.	Scoped Authorization: Enforces Broken Object Level Authorization (BOLA) checks to ensure the API (and thus the LLM) only retrieves records belonging to the active user.
PII Rehydration	LLMs infer missing details from masked data.	Data Minimization at Design: API Audit identifies "leaky" schemas during development, forcing developers to remove unnecessary data points that could be used for rehydration.
4. **Identity & Authorization Breakdowns** (AI ↔ API)	API1:2023 Broken Object Level Authorization (BOLA) / API2:2023 Broken Authentication. AI agents break traditional assumptions about identity and sessions.	Identity Governance: Integrates with IAM providers to ensure that the "Agent" identity is distinct from the "User" identity and that both are validated at the API layer.
Agent Identity Ambiguity	Confusion over who is calling: user, agent, or model.	Token Binding: Supports validating specific claims in the token that distinguish between human-initiated and agent-initiated calls.

AI Attack Category & Individual Vectors	OWASP API TOP 10	42Crunch AI Attacks Protection
Token Over Delegation	LLMs running with broad API keys are exploited.	Least Privilege Audit: The platform audits API contracts to ensure they don't use "all-powerful" keys and encourages granular scopes for different AI functions.
Session Confusion	LLM mixes multiple user contexts.	Strict Session Validation: Validates that the data requested matches the correct user token, preventing an LLM from "accidentally" requesting User B's data with user A's token.
Replay via Model Memory	Attackers extract or reuse tokens stored in context.	Token Expiry & Validation: Enforces short-lived tokens and checks for replay patterns. Since the 42Crunch MCP Server sits in-line, it can detect and block reused or stolen credentials.
5. **AI Driven Abuse of API Rate Limits & Quotas**	API4:2023 Unrestricted Resource Consumption. AI agents generate traffic at machine speed, leading to exhaustion.	Granular Rate Limiting: Applies rate limits not just at the IP level, but at the user/token/API-key level to throttle hyper-active AI agents without affecting human users.
LLM Amplified DDoS	A single prompt triggers thousands of backend calls.	Throughput Quotas: Sets hard limits on the number of calls an agent can make per second/minute per endpoint, preventing DOS attack.
Recursive Agent Loops	Agents stuck in logic loops hammer APIs	Anomaly Detection Integration: 42Crunch MCP server can implement API quotas as well as MCP quotas.
Fan Out Explosions	LLMs call multiple APIs in parallel, overwhelming systems.	Concurrency Limits: MCP can restricts the number of simultaneous MCP connections and API sequence calls.
6. **Model Mediated Injection Attacks** (API → AI → API)	API8:2023 Security Misconfiguration. APIs return data that manipulates the LLM to trigger further calls.	Output Content Sanitization: 42Crunch MCP server will block any malicious "instructions" through strict contract validation. Only allowed data is returned.
Response Injection	API returns text that manipulates the LLM.	Response Schema Validation: Validates that the response content type and structure match the contract therefore blocking any injections in the response payload.
Schema Injection	API returns JSON that tricks the model into dangerous functions.	Strict JSON Parsing: 42Crunch MCP Server enforces strict JSON structures, preventing an attacker from injecting "additionalProperties" into a response to confuse the LLM's parser.
7. **Supply Chain & Plugin/Tooling Risks** (AI ↔ External APIs)	API10:2023 Unsafe Consumption of APIs. AI agents often call external APIs automatically.	Egress Filtering & Proxying: 42Crunch server enforces strict API calls flows on internal APIs protecting the internal enterprise APIs not external undocumented APIs.
Malicious Third Party Tools	Fake "AI Agents/plugins" that steal tokens.	Content Security Policy (CSP) for APIs: 42Crunch MCP Server has an "allowed list" of Agents. All other agents/plugins are not allowed.
Dependency Confusion	Attackers register similarly named tools to intercept calls.	Strict Discovery & Cataloging: 42Crunch MCP server maintains "Service Catalog" of approved APIs and tools. No other tooling allowed.
External API Injection	External APIs return crafted responses to manipulate the LLM.	External Data Validation: Even if the AI talks to an external service, 42Crunch can validate the incoming data from that service against a known safe schema.
8. **AI Driven Business Logic Abuse** (AI → API)	API6:2023 Unrestricted Access to Business Processes. AI is used to explore and exploit edge cases in business logic.	Logic Fuzzing (API Scan): Uses "Conformance Scanning" to proactively find business logic flaws (like BOLA or BPLA) before an AI attacker can find them.
Automated Discovery of Logic Flaws	LLMs fuzz APIs more intelligently than humans.	Shift-Left Testing: 42Crunch offers Static and Dynamic tools in IDE/CI-CD, removing the holes that an LLM would otherwise exploit before exposing API through MCP server.
Policy Circumvention	LLMs find sequences of calls that bypass rules.	Protocol Hygiene: Enforces strict adherence to the intended API flow (e.g., ensuring an "Update" can't happen before an "Auth") via the 42Crunch MCP Server.