red canary
a zscaler company

#61 | April 21, 2026

Welcome to SecOps Weekly!

Phil Hagen

PRINCIPAL SECURITY RESEARCHER

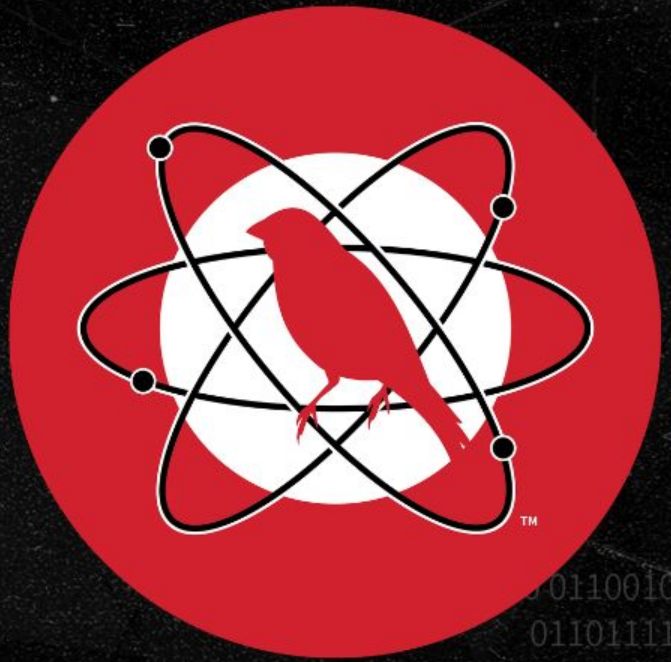
ZSCALER

Hare Sudhan

SECURITY ENGINEER &

ATOMIC RED TEAM MAINTAINER

Supercharging Atomic Red Team with AI

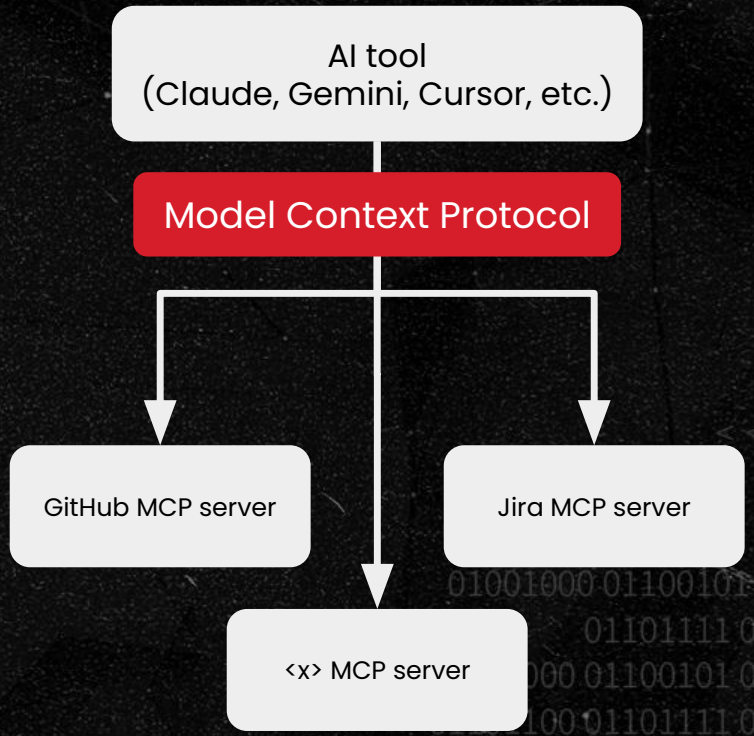


red canary
a zscaler company

Atomic Red Team Model Context Protocol (MCP) server

MCP explained

- A “USB-C port for AI”
- Universal interface between AI model and tools
- *Tools*: Functions the AI can call
- *Resources*: Data the API can read
- *Prompts*: Reusable workflow templates
- *Transports*: stdio, SSE, streamable HTTP



MCP architecture

red canary
a zscaler company

MCP host

AI application (Claude Desktop, Cursor, VSCode, Windsurf, Gemini, etc.)

Manages connections, orchestrates tool calls

MCP client

Lives on host and maintains 1:1 connection with each server

Handles protocol negotiation, message routing

MCP server

Exposes *tools*, *resources*, and *prompts* to AI models

Can be local (stdio) or remote (HTTP)

Adversary emulation gap

Why security teams need AI-powered workflows

red canary
a zscaler company

TRADITIONAL WORKFLOW

😬 heavy manual workloads

search atomic tests for TTPS

copy YAML to editor

manually craft input arguments

debug execution failures

cross-reference detection results

write report manually

45+ minutes

THE AI ADVANTAGE

MCP WORKFLOW

😀 AI-powered efficiency

describe goal in natural language

AI searches, validates & executes

AI analyzes results & correlates

AI generates consolidated report

5-10 minutes

Atomic Red Team MCP server

Atomic Red Team is an open source library of 1,500+ small, focused security tests mapped to the MITRE ATT&CK framework

The MCP server puts all of those tests into your AI assistant's hands

Core Tools

`query_atomics`: Search by technique ID, name, platform

`execute_atomic`: Run tests (opt-in, lab only)

`validate_atomic`: Validate YAML schema

`refresh_atomics`: Update from GitHub

`get_validation_schema`: Get the atomic schema

`server_info`: Server status, platform info

Resources

`file:///documents/{technique_id}`:
Read raw YAML for any technique
Full atomic test content available to AI

Supported Clients

Claude Desktop, Claude Code, Cursor, Windsurf, VSCode, Gemini, any MCP client

Atomic Red Team MCP installation

Just five minutes to start!

Claude Code CLI

```
# Recommended using uvx
claude mcp add atomic-red-team \
  uvx atomic-red-team-mcp

# With execution enabled
claude mcp add atomic-red-team \
  -e ART_EXECUTION_ENABLED=true \
  uvx atomic-red-team-mcp

# Using Docker
claude mcp add atomic-red-team -- \
  docker run -- rm -i \
  ghcr.io/cyberbuff/atomic-red-team-mcp
```

Claude Desktop (claude_desktop_config.json)

```
{
  "mcpServers": {
    "atomic-red-team": {
      "command": "uvx",
      "args": [ "atomic-red-team-mcp" ],
      "env": {
        "ART_EXECUTION_ENABLED": "true"
      }
    }
  }
}
```

⚠️ Only enable execution in isolated lab environments!

Scenario 1 (demo): Threat intel → Playbook

A new malware campaign is identified; build an executable test playbook in minutes

Prompt: *Here's a report on APT41's recent campaign. Extract the TTPs and create an atomic test playbook.*

Claude's actions

1. Parse campaign report for TTPs
2. `query_atomics` for each TTP
3. Map to available atomic tests
4. `validate_atomic` for any gaps and generate new tests
5. Compile and execute playbook

Timeline

Step	Manual	With MCP
TTP extraction	15 min	30 sec
Library search	20 min	1 min
Gap analysis	10 min	1 min
Total	**45+ min**	**5 min**

Scenario 2: Detection rule validation

Validate a new Splunk detection rule for Cloudflare tunnel abuse

Prompt: *I wrote a Splunk detection for Cloudflare tunnel abuse (T1572). Validate it by running the relevant atomic tests.*

Automated workflow

```
1. query_atomics("cloudflare tunnel", platform="windows")
   → Found: T1572 - Protocol Tunneling tests

2. execute_atomic(guid, args) on atomic-windows server
   → Test executes, generates telemetry

3. Claude queries Splunk (via separate MCP)
   → "Did your detection fire?"

4. Analyzes gaps, suggests rule improvements
```

Total: ~3-5 minutes vs 30+ minutes manually

Other example scenarios

Multi-platform persistence

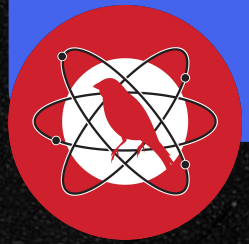
Task:
Simulate a threat actor establishing persistence across your entire fleet.

Prompt:
Simulate T1547.001 registry persistence on Windows and T1053.003 cron persistence on Linux simultaneously.

AI-assisted atomic test creation

Task:
A new CVE is released but no atomic test exists yet.

Prompt:
There's no atomic test for this technique. Create one of the following ATT&CK standards and validate it.



Get started!

red canary
a zscaler company

Install

```
# Claude Code

claude mcp add
atomic-red-team-mcp \
  uvx atomic-red-team-mcp
```

Explore

```
"Show me all T1059 atomics"

"Find Windows persistence tests"

"Search for credential dumping"
```

Execute

```
# Lab environments only!

ART_EXECUTION_ENABLED=true \
  uvx atomic-red-team-mcp
```

GitHub: github.com/cyberbuff/atomic-red-team-mcp
Blog: cyberbuff.substack.com
PyPI: uvx atomic-red-team-mcp
Docker: ghcr.io/cyberbuff/atomic-red-team-mcp

red canary
a zscaler company

Looking for more content?

REDCANARY.COM/BLOG

YouTube YOUTUBE.COM/REDCANARY

NEXT WEEK ⏩ APRIL INTELLIGENCE INSIGHTS