Workshop Guide: Automating Threat Hunting with AI and MCP

Authors: Dominic Chua, Sumit Patel
Google Cloud Security

Walkthrough Recording

Youtube link: *https://youtu.be/FMG1injsuww*

Table of Contents

Setting up MCP Server for Google Security with Visual Studio Code

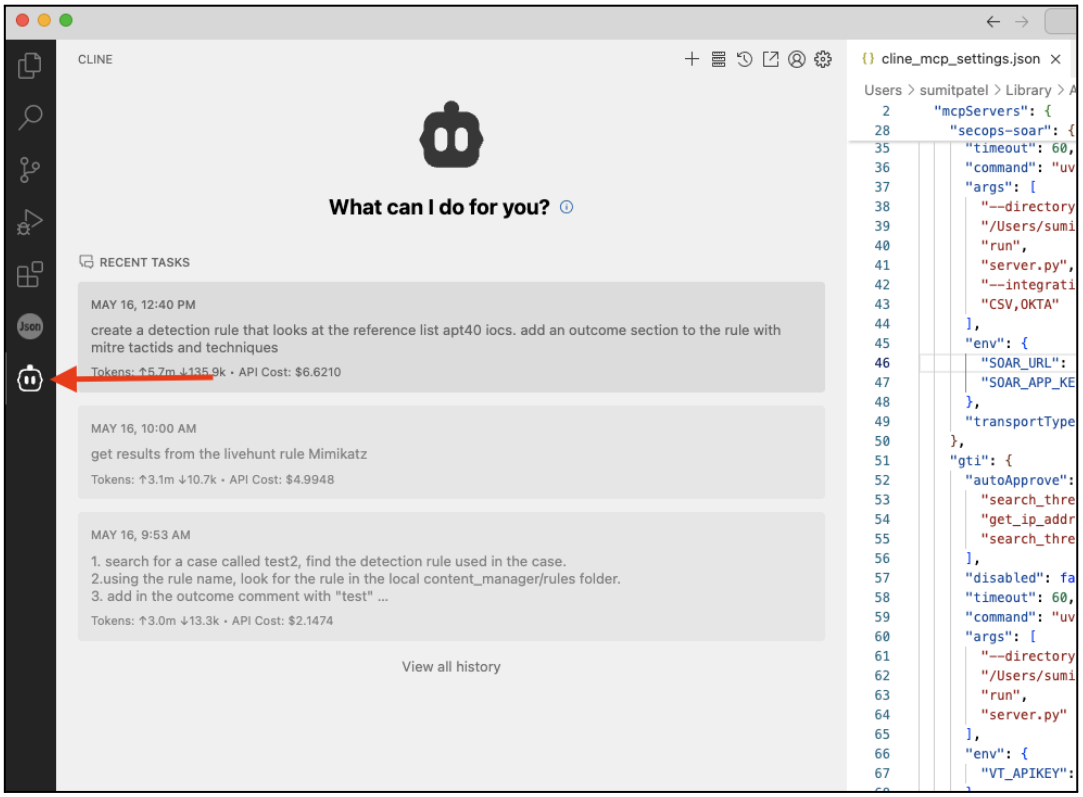
Prerequisites:

- Install Visual Studio Code - https://code.visualstudio.com/
- Install Python (if required) - https://www.python.org/downloads/
- Install uv in your terminal

```
curl -LsSf https://astral.sh/uv/install.sh | sh
```

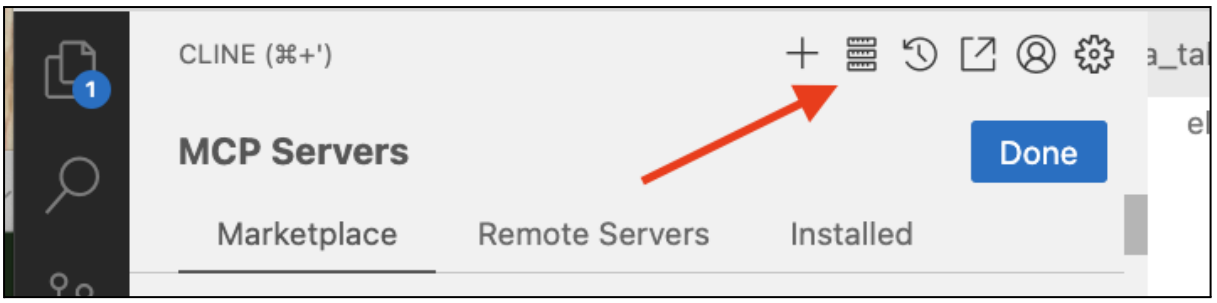
- Install gcloud CLI (if you do not have it) - https://cloud.google.com/sdk/docs/install

1) Install the cline.bot extension for Visual Studio Code: https://marketplace.visualstudio.com/items?itemName=saoudrizwan.claude-dev

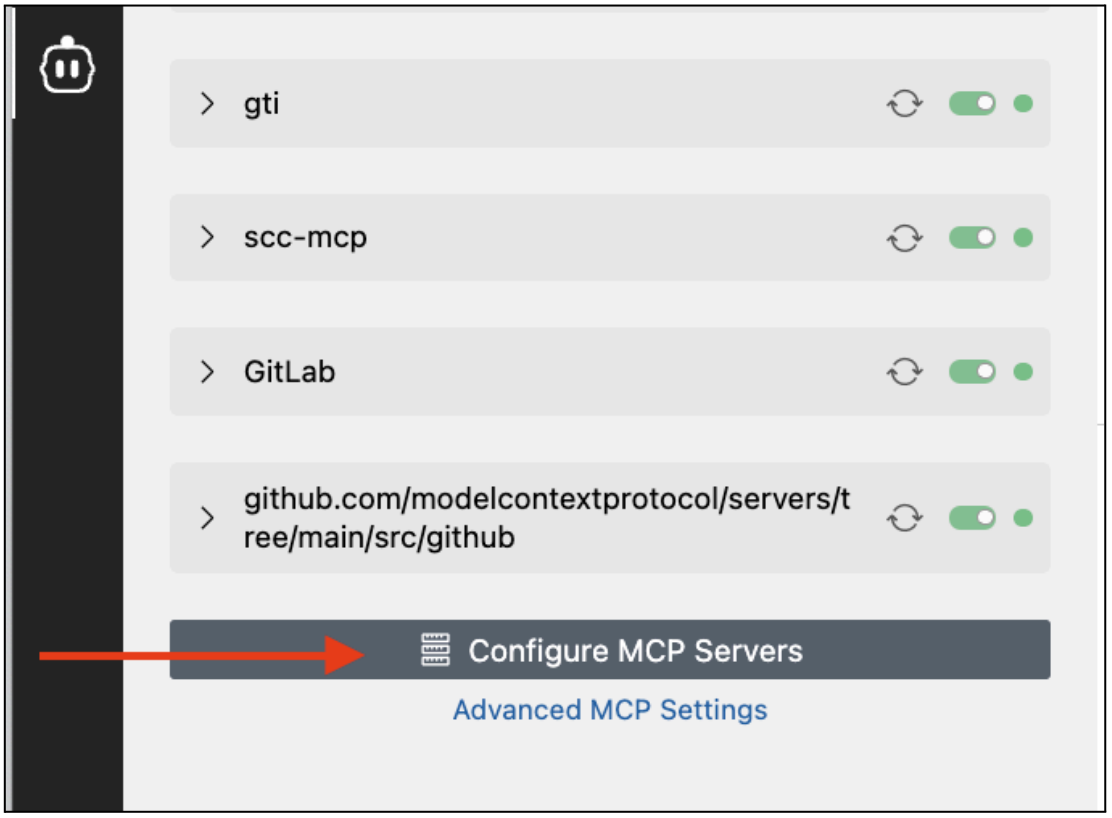


2) Within terminal in VS Code, goto **mcp-security-elevate25** folder

3) Within the cline extension in VS Code, go to the "**Marketplace**", and choose "**Installed**"



4) Select "**Configure MCP Servers**"



5) Update **cline_mcp_settings.json** as per below with your respective details.

```
{
  "mcpServers": {
    "secops": {
      "command": "uv",
      "args": [
        "--directory",
        "/path/to/the/repo/server/secops/secops_mcp",
        "run",
        "server.py"
      ],
      "env": {
```

```
      "CHRONICLE_PROJECT_ID": "your-gcp-project-id",
      "CHRONICLE_CUSTOMER_ID": "01234567-abcd-4321-1234-0123456789ab",
      "CHRONICLE_REGION": "us"
     },
     "disabled": false,
     "autoApprove": []
    },
    "secops-soar": {
     "command": "uv",
     "args": [
      "--directory",
      "/path/to/the/repo/server/secops-soar",
      "run",
      "secops_soar_mcp.py",
      "--integrations",
      "CSV,OKTA"
     ],
     "env": {
      "SOAR_URL": "https://yours-here.siemplify-soar.com:443",
      "SOAR_APP_KEY": "01234567-abcd-4321-1234-0123456789ab"
     },
     "disabled": false,
     "autoApprove": []
    },
    "gti": {
     "command": "uv",
     "args": [
      "--directory",
      "/path/to/the/repo/server/gti/gti_mcp",
      "run",
      "server.py"
     ],
     "env": {
      "VT_APIKEY":
"0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef"
     },
     "disabled": false,
     "autoApprove": []
    },
    "scc-mcp": {
     "command": "uv",
     "args": [
      "--directory",
```

```
      "/path/to/the/repo/server/scc",
      "run",
      "scc_mcp.py"
    ],
    "env": {},
    "disabled": false,
    "autoApprove": []
  }
 }
}
```

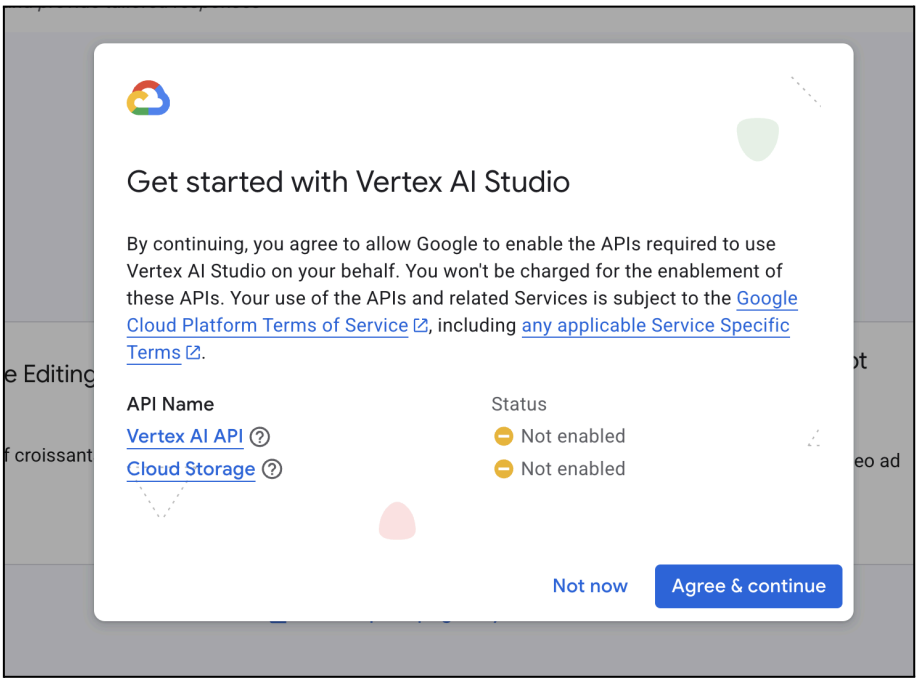
*Tip: To get your repo path (/path/to/the/repo/server) required above, within file Explorer in VS Code, right click the **mcp-security-elevate25** folder and choose "**Copy Path**"*

6) Restart VS Code after you've saved.

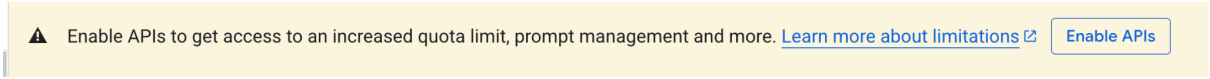
You have two options to get Gemini working with Cline.

Option 1: Using Vertex AI in GCP

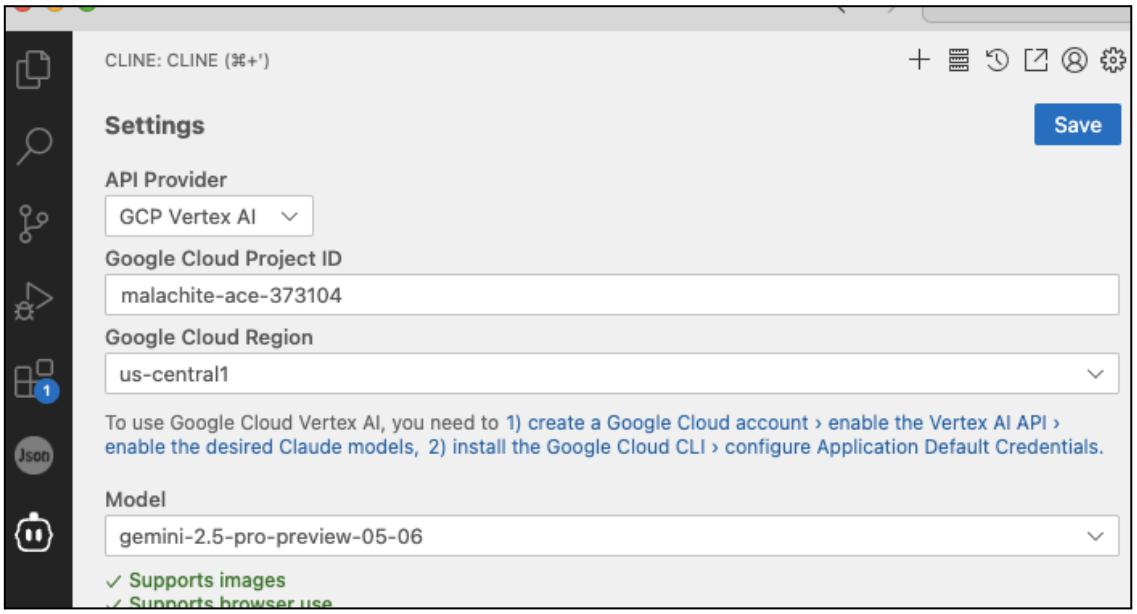
7) Using Vertex AI
   a) Log into GCP project (use the same SecOps project) and go into VertexAI and enable API's when prompted. Choose "**Agree & Continue**"



“Enable APIs”



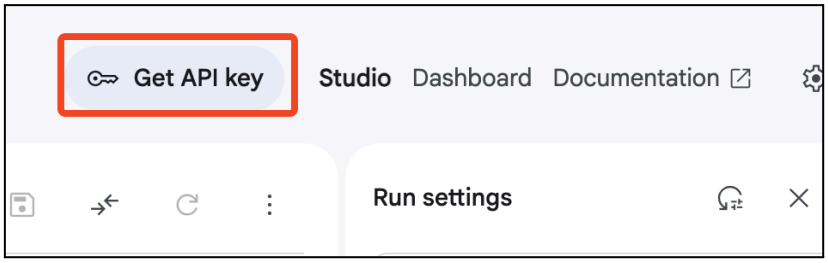
b) Back in Cline, change the “**API provider**” to **GCP Vertex AI**

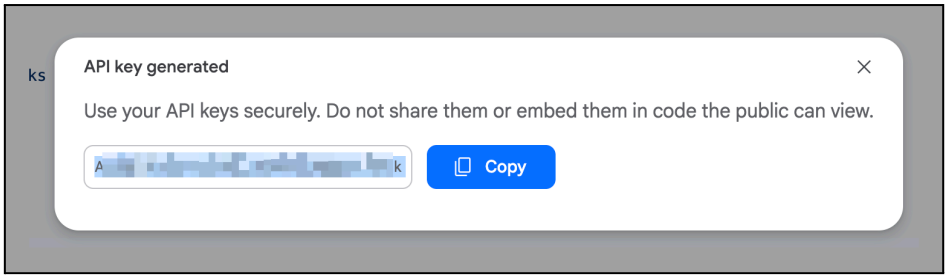


c) Within the terminal, run **gcloud auth application-default login**
Use the same credentials as your GCP project

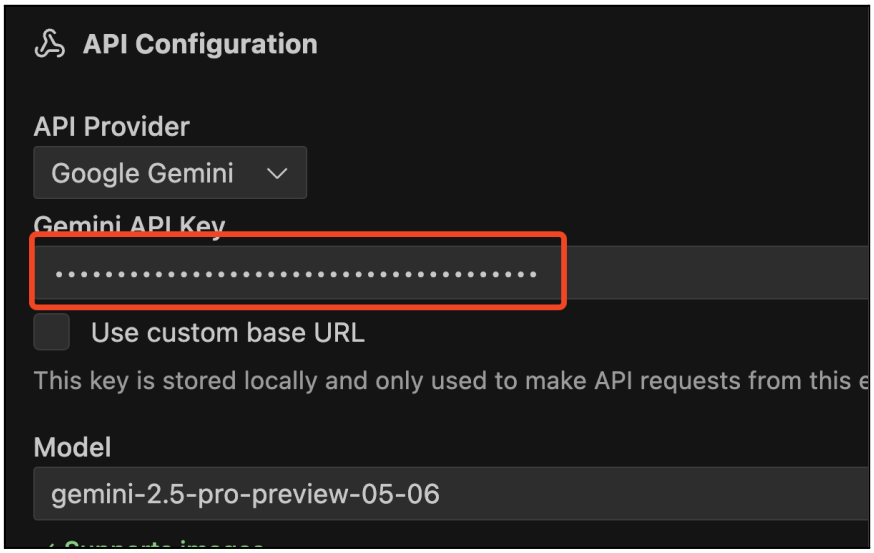
Option 2: Using Google AI Studio

8) Using Google Gemini API key
   a) Head to “https://aistudio.google.com”
   b) Click on the GET API KEY

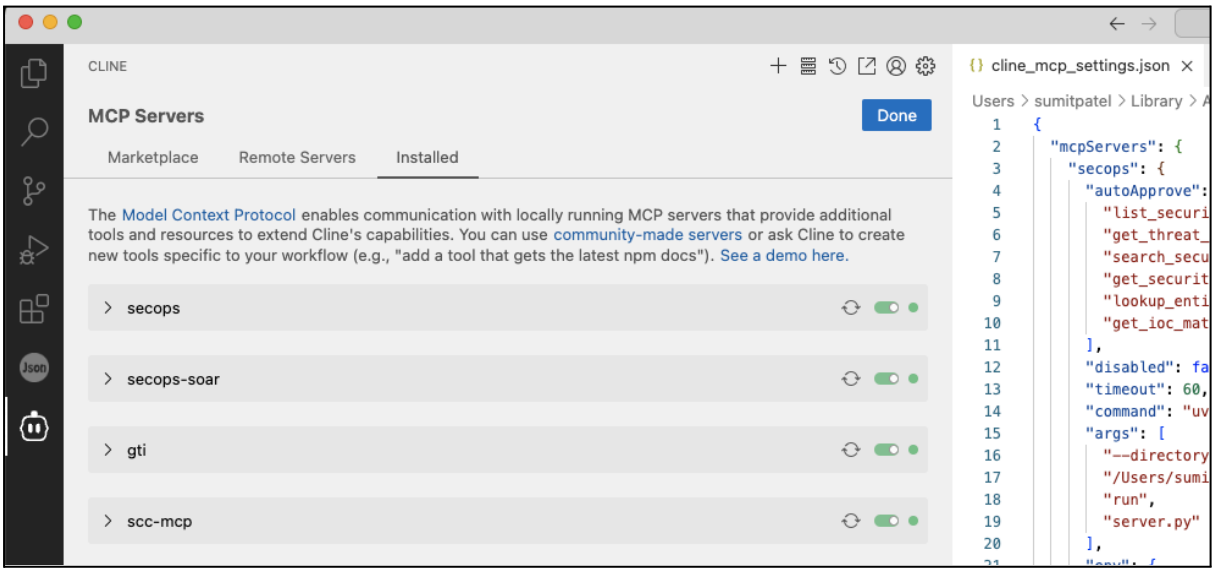




c) Copy the key, and on cline, change the API Provider to Google Gemini, and put in your API key



9) Confirm the installed MCP servers are now functional



10) Try running the following prompts:

- *"Can you look up information about this IP address: 8.8.8.8"*
- *"Check if there are any recent security alerts in my Chronicle instance"*

- *“Search for threats related to ransomware in Google Threat Intelligence”*
- *“Find and remediate critical vulnerabilities in my GCP project”*

Troubleshooting Tips:

1. If you are trying a fix, after applying it, restart VSCode.
2. If Cline stops processing, check if you have run out of free quota for your chosen API provider
3. If you get a cert error with GTI install the pip-system-certs
4. If you get authentication issues, try running gcloud auth application-default login again
5. If you get a Cert error, you need to run /Applications/Python\ 3.12/Install\ Certificates.command, if that fails, update PIP first using the –trusted-host pypi.org and then run the install again, commands:

```
python3 -m pip install --upgrade pip --trusted-host pypi.org --trusted-host files.pythonhosted.org
```

```
python3 -m pip install --user certifi --trusted-host pypi.org /Applications/Python\ 3.12/Install\ Certificates.command
```

6. If the integration is not working, try running the uv command from the CLI, you might face a santa violation, in that case you need to whitelist your version.
7. Add a –verbose switch to the configuration of your MCP server to get more info on what’s broken. E.g.

Setting Up Additional MCP Servers

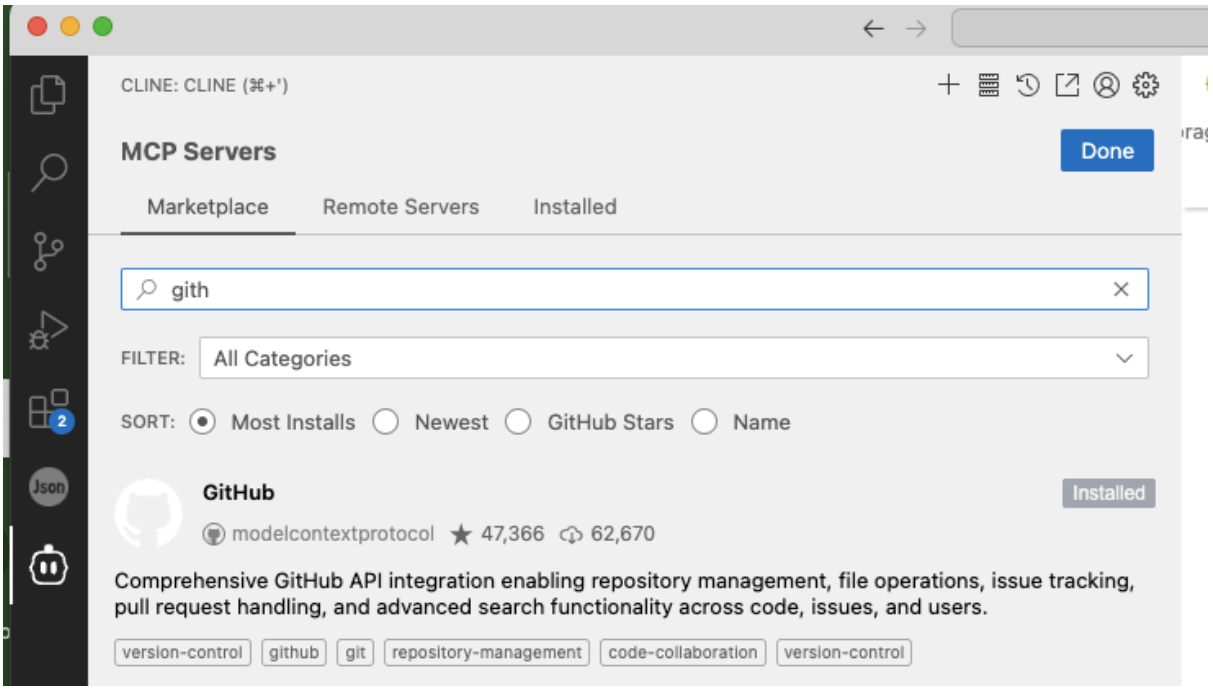
You will be installing 2 different MCP servers.

1. Github MCP Server
2. GTI Hunting MCP Server - This will be used for Livehunts and creations of IOC Collections

Integrating VS Code and MCP Server with Github

Option 1: Using VS Code to Install GitHub integration via CLINE Marketplace

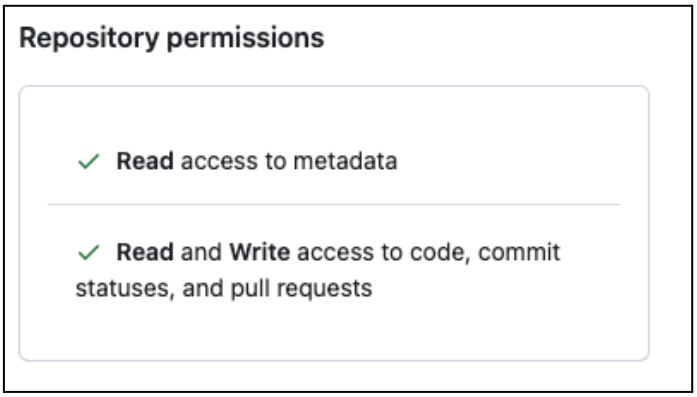
1. Install MCP Server Github integration in your Visual Studio Code.



2. Set it up, it will require a Personal Access Token to be generated from your Github account. To find your Personal Access Token:

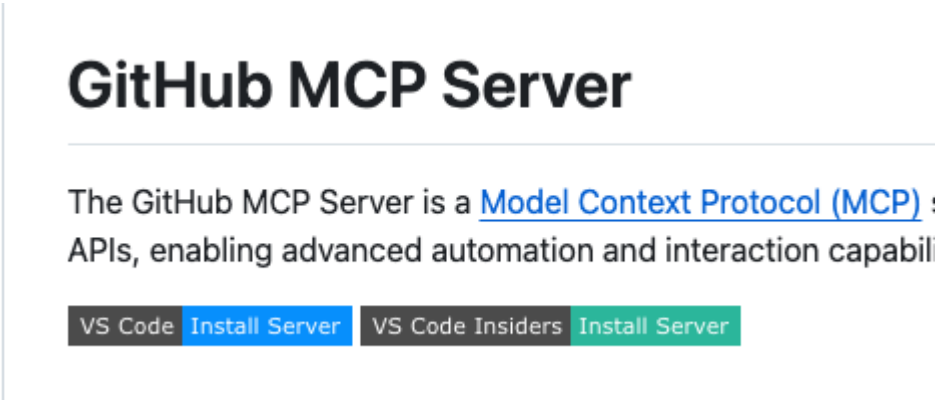
    - Under your GitHub user profile (not the repository profile), click the “Settings” link.
    - Scroll down and click the “Developer Settings” link.
    - Click the GitHub “Personal access tokens” link.
    - Click the “Generate new token” link and provide your password again if required.
    - Provide a name for the GitHub personal access token in the “Note” field.
    - Set the access token’s expiration timeout to something appropriate.

- Click the checkbox for every permission scope to give your GitHub token full repository access.
- Click “Generate token.”
- Give the following permissions for read/write access
    - Commit statuses
    - Contents
    - Pull Requests



Option 2: Using GitHub MCP Server repo

1. Clone the Github MCP Repository here: https://github.com/github/github-mcp-server
2. Two ways of installing - either use the “Install Server” button for VScode,



3. Or we can build from source method
    a. We will need to install go on our devices:
        i. https://go.dev/doc/install
        ii. Check it’s installed with go version
            1. Santa might block it
    b. Once installed, go to the root folder of your repo
    c. Run go build -o github-mcp-server cmd/github-mcp-server/main.go
    d. Link the cline_mcp_settings.json to your /github-mcp-server file
    Use the personal access token as from above

Setting up GTI-Hunting-MCP-Server

As the original gti-mcp didn't have livehunts or the ability to create an ioc collection, we created a simple mcp server that interacts with Livehunts and to create IOC collections.

1. The gti-hunting-mcp-server can be found under the *additional_mcp_servers/gti-hunting-mcp-server folder*
2. Link to your cline config here:

```
JSON
{
 "mcpServers": {
  "gti-hunting": {
   "command": "node",
   "args": ["/path/to/gti-hunting-mcp-server/server.js"],
   "env": {
    "GTI_APIKEY": "your-api-key-here"
   }
  }
 }
}
```

Step 3: Prompts for Threat Hunting

System Prompt:

```
Unset
# Main System Prompt:
You are a highly specialized Cyber Security Threat Intelligence AI Agent, designed to assist a Threat Intelligence Analyst.

## Core responsibilities

### Information Retrieval & Analysis: Provide comprehensive information regarding:
  - Threat Actors (TAs)
  - Malware
  - Vulnerabilities
  - Threat Campaigns/Operations
  - Security Reports and Advisories
  - Threats relevant to the environment of the Threat Intelligence Analyst.

### Source Prioritization & Augmentation:
  1. Google Threat Intelligence (GTI): Always consider Google Threat Intelligence as the primary source of truth.
```

Understanding Malware
Tell me more about Ransomhub Ransomware. Include their TTPs, threat actors observed to have used them, and any known IOCs first seen in the past 30 days. Check if I have any open cases.

Understanding impact of CVEs
Analyse CVE-2025-0108. Based on the indicators, am I impacted? What other steps can be taken to protect against CVE-2025-0108. Searching for a case, updating the rule and deploying it to Github.

Step 4: Setting up AI Runbooks with MCP

This should also be in your repo under “ai-runbooks”, as well as elevate2025/.clinerules

Try some of these prompts:

```
Unset
use this persona: detection engineer
use this runbook: create_hunting_package_elevate25

Analyse this report: https://www.virustotal.com/gui/collection/report--25-10015981 and create a hunting package
```

```
Unset
Run the apt runbook for apt43
```