

hackerone



Bug Bounty Hunting with AI: A Beginner's Guide

ROOTCON 20 – Bug Bounty Village PH

Robin "Japz" Divino

Philippines HackerOne Brand Ambassador



> whoami

- ★ ROBIN "JAPZ" DIVINO - @JAPZDIVINO
- ★ PRINCIPAL OFFSEC ENGINEER @ [VIKINGCLOUD](#)
- ★ INDEPENDENT SECURITY RESEARCHER AT HACKERONE
- ★ OSCP, CWES
- ★ FATHER, GAMER



› Table of Contents

01

Bug Bounty Basics

What it is · the problem

02

AI & AI Agents

Chatbot vs agent · where it fits

03

AI-Assisted Workflow

Five stages, Discover → Report

04

AI Tools

Claude Code · Skills · opencode

05

MCP + Burp Suite

MCP · Burp MCP · DVWA

06

Demo & Q&A

Live walkthrough · open floor



What is Bug Bounty Hunting?

Ethical hackers get paid to find security flaws before attackers do. Companies invite them to test their systems and reward them based on how serious the vulnerabilities are.

Bug Bounty Platforms

HackerOne - <https://www.hackerone.com>

YesWeHack - <https://yeswehack.com>

Bugcrowd - <https://www.bugcrowd.com>

Immunefi - <https://immunefi.com>

Intigriti - <https://www.intigriti.com>

Learning Platforms

Hacker101 - <https://www.hacker101.com>

Portswigger Academy - <https://portswigger.net/web-security>

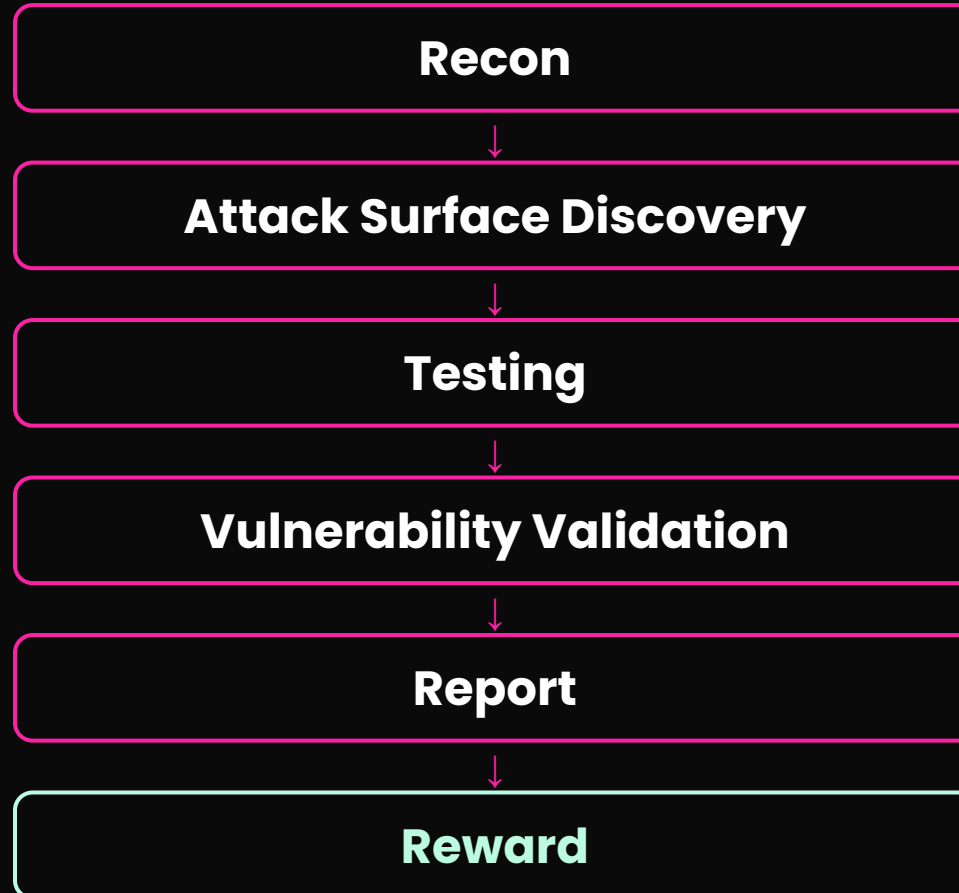
Hack The Box - <https://www.hackthebox.com>

Try Hack Me - <https://tryhackme.com>

PentesterLab - <https://pentesterlab.com>

Company → Program → Hacker → Vulnerability Report → Fix → Reward

Traditional Bug Bounty Workflow



The Problem

Modern applications have:

- Hundreds/thousands of endpoints
- APIs
- JavaScript
- Authentication flows
- Complex business logic
- Large amounts of HTTP traffic

“What if we could have AI help us analyze all of this?”

What is AI?

Artificial Intelligence (AI) is technology that allows computers to perform tasks that normally require human intelligence.

AI can understand, summarize, reason about, and generate information.

HTTP Request



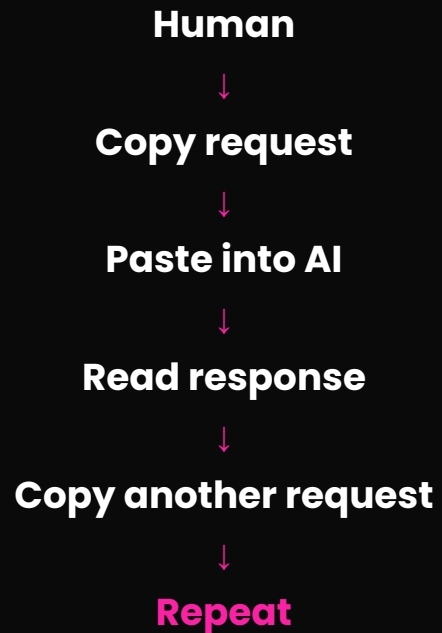
AI



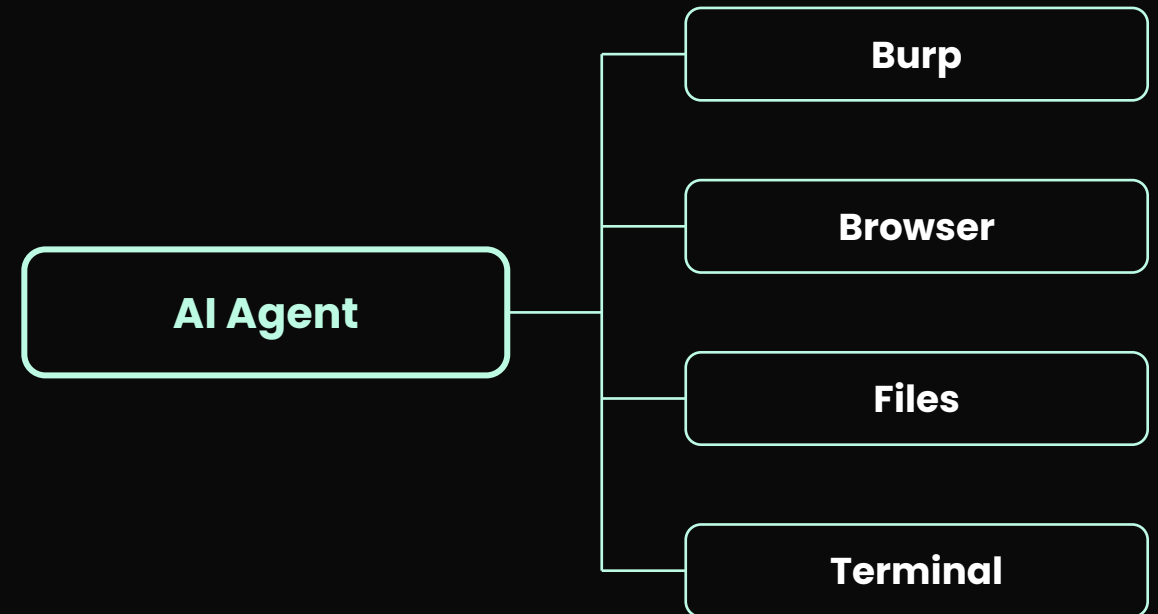
"Interesting parameter detected"

Chatbot vs AI Agent

Traditional AI chatbot



AI Agent



The agent can interact with tools, rather than simply answering questions.

Where AI Fits in Bug Bounty Hunting

How can we use AI in Bug Bounty Hunting

01 Reconnaissance planning

02 Understanding application functionality

03 Analyzing HTTP requests/responses

04 Writing scripts

05 Understanding unfamiliar technologies

06 Analyzing JavaScript

07 Finding interesting
parameters/endpoints

08 Creating wordlists

09 Explaining error messages

10 Report writing

Where AI Fits in Bug Bounty Hunting

From Human-Led Hunting to AI-Powered Bug Discovery

AI-Assisted Hunting

 **Human leads → AI assists**

Human identifies the target; AI analyzes the code and generates test cases and payloads.

AI-Augmented Hunting

 **Human discovers → AI expands**

Human finds one bug; AI discovers other endpoints with the same vulnerability pattern.

Autonomous AI Hunting

 **Human defines the goal → AI executes**

Human defines the goal; AI autonomously performs the end-to-end hunting workflow.

AI-Assisted Bug Bounty Workflow

How AI compresses the traditional hunting loop



1. DISCOVER

AI can help analyze:

- **Domains**
- **Subdomains**
- **URLs**
- **APIs**
- **JavaScript**
- **Endpoints**
- **Parameters**
- **Technologies**

DISCOVER · UNDERSTAND · HYPOTHEZIZE · VALIDATE · REPORT

2. UNDERSTAND

AI can help answer:

“What does this application actually do?”

For example:

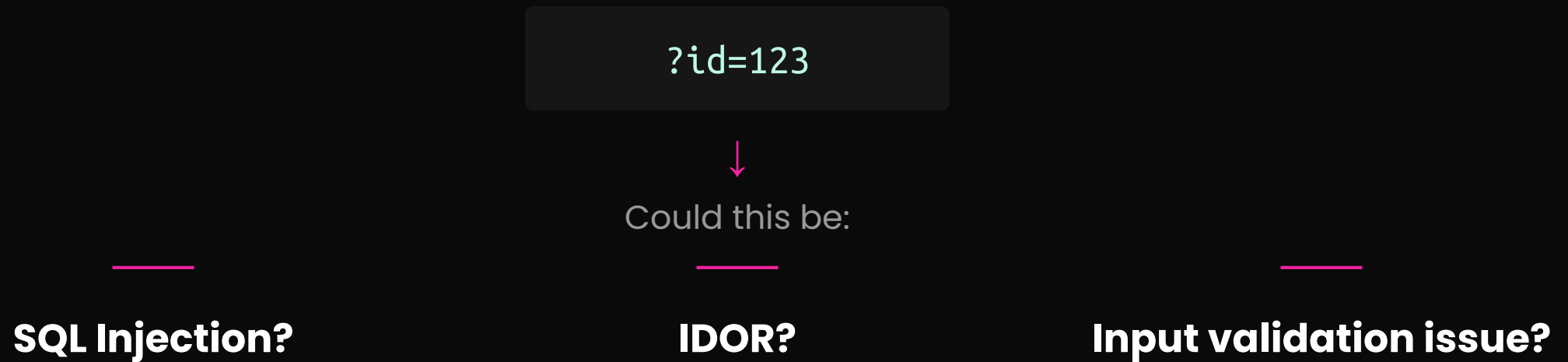
```
/api/user?id=123
```

AI might recognize:

id appears to reference a user object.

3. HYPOTHESIZE

AI can generate vulnerability hypotheses:

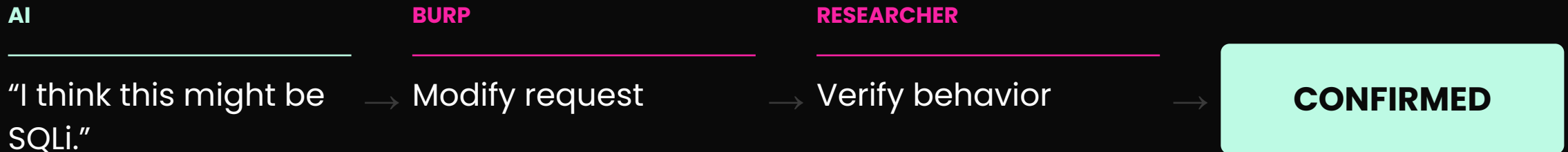


Potential ≠ Confirmed

DISCOVER · UNDERSTAND · **HYPOTHESIZE** · VALIDATE · REPORT

4. VALIDATE

This is where the security researcher comes in.



5. REPORT

AI can help generate:

- Vulnerability title
- Severity
- Description
- Evidence
- Reproduction steps
- Impact
- Remediation

But:

Never allow AI to invent evidence.

The Bottom Line

AI-Assisted Bug Bounty Workflow

AI doesn't replace the workflow. It compresses it.

AI generates hypotheses. The hacker generates evidence.



AI TOOLING

AI Tools!

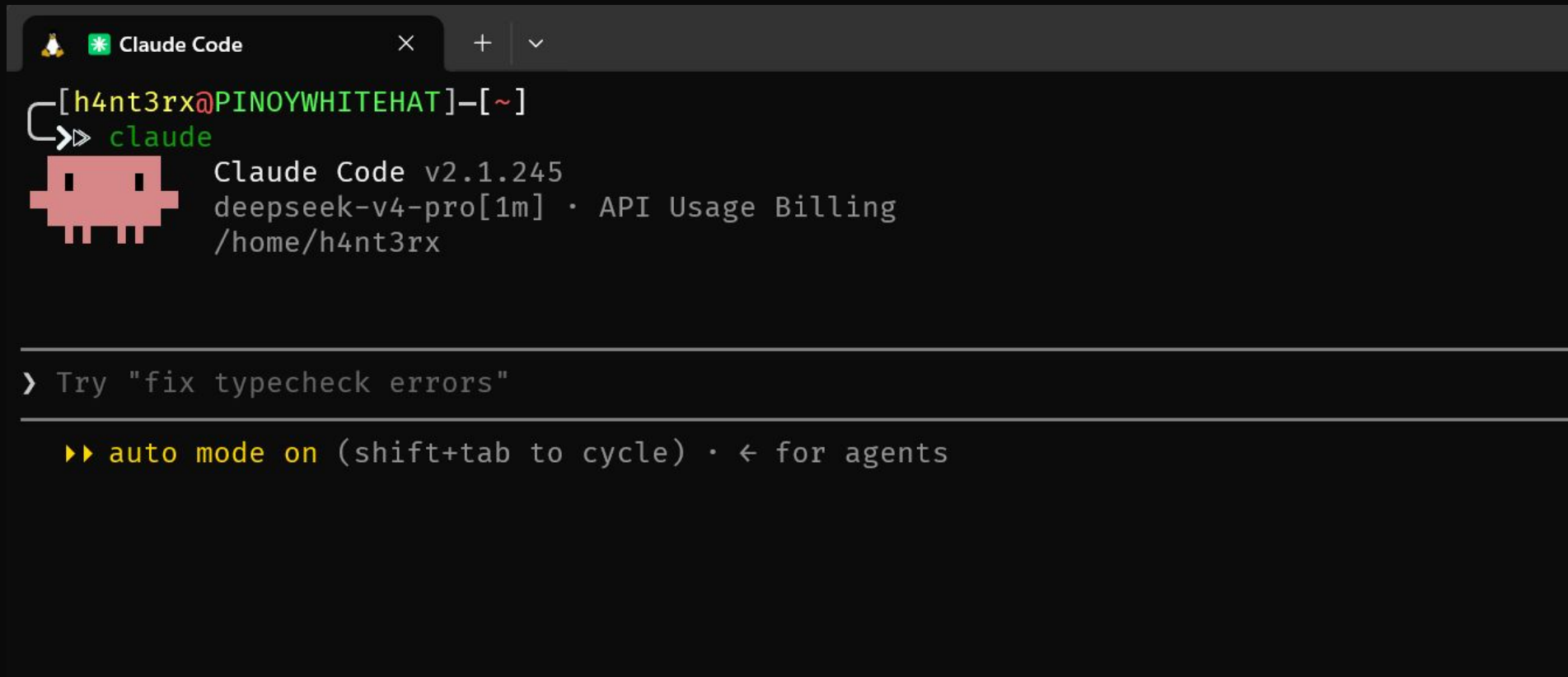
Bug Bounty · AI-assisted Tools

hackerone

claude code

installation

```
curl -fsSL https://claude.ai/install.sh | bash
```

A screenshot of the Claude Code application interface. At the top, there's a dark header bar with a terminal icon, a green star icon, and the text "Claude Code". Below this, the terminal shows a prompt "[h4nt3rx@PINOYWHITEHAT]-[~]" and the command "claude" has been entered. To the left of the command is a red pixelated robot icon. To the right of the command, the text "Claude Code v2.1.245" is displayed, followed by "deepseek-v4-pro[1m] · API Usage Billing" and the path "/home/h4nt3rx". Below the terminal output, there's a horizontal line, then a prompt "> Try 'fix typecheck errors'" followed by another horizontal line. Below that, there's a prompt "▶▶ auto mode on (shift+tab to cycle) · ← for agents".

```
[h4nt3rx@PINOYWHITEHAT]-[~]  
→ claude  
Claude Code v2.1.245  
deepseek-v4-pro[1m] · API Usage Billing  
/home/h4nt3rx  
  
> Try "fix typecheck errors"  
  
▶▶ auto mode on (shift+tab to cycle) · ← for agents
```

<https://code.claude.com/docs/en/quickstart>

Claude AI MODELS

Pick the model that fits the job

Haiku

Haiku 4.5

Fastest for quick answers

Sonnet

Sonnet 5

Efficient for routine tasks

Opus

Opus 5

Best for everyday, complex tasks

Fable

Fable 5.1

Most capable for your hardest and longest-running tasks

Requires usage credits

Mythos

Mythos 5.1

Specialized high-capability model

Available only to vetted organizations through Anthropic's trusted-access programs

What are Claude Skills?

Reusable expertise the agent loads on demand

A Skill is a **folder of instructions** Claude loads only when the task matches.

Package

SKILL.md

Instructions, checklists, scripts and wordlists in one folder

On demand

auto-loaded

Claude opens it only when the work actually calls for it

Repeatable

every target

Your methodology, applied the same way every time

Skills in Practice

The bug bounty skills on my machine

TESTING

/sqli-test

SQL injection methodology, end to end

/xss-test

XSS discovery and payload selection

/webshell-detect

Webshell indicators in uploads and responses

Invoke it by name — or let Claude reach for it on its own:

```
> /sqli-test http://localhost/dvwa/vulnerabilities/sqli/?id=1
```

RECON & DATA

/wordlist

Target-specific wordlists, built on the fly

/api-keys

Secrets and keys hidden in JavaScript

seclists-*

Curated payload, fuzzing and username packs

Write the method once. Run it on every target.

opencode

installation

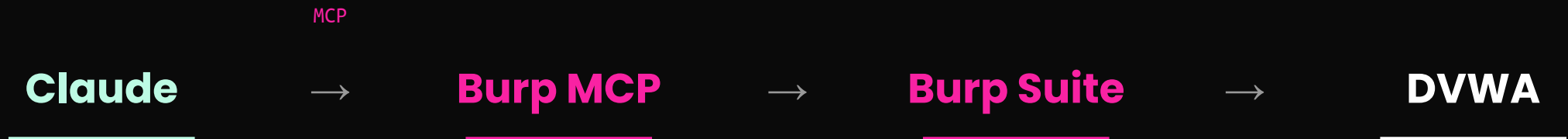
```
curl -fsSL https://opencode.ai/install | bash
```

The image displays two side-by-side screenshots of the OpenCode application interface. The left screenshot shows a terminal window with the title bar '~/opencode'. The terminal output includes the command `curl -fsSL https://opencode.ai/install | bash`, followed by 'Installing opencode version: 1.18.18' and a progress bar reaching 100%. Below this, it says 'Successfully added opencode to \$PATH in /home/h4nt3rx/.zshrc'. The 'opencode' logo is prominently displayed, followed by the text 'OpenCode includes free models, to start:'. A list of commands is shown: `cd <project>` for opening a directory and `opencode` for running a command. At the bottom, it directs the user to visit <https://opencode.ai/docs> for more information. The right screenshot shows the OpenCode application window with the title bar 'OpenCode'. It features the 'opencode' logo and a search bar with the placeholder text 'Ask anything... "Fix a TODO in the codebase"'. Below the search bar, there are buttons for 'Build auto', 'DeepSeek V4 Pro', and 'DeepSeek'. At the bottom of the window, it says 'tab agents ctrl+p commands' and '1.18.22'.

What is MCP?

Model Context Protocol

MCP allows AI models to interact with **external tools and data** through a standardized interface.



Without MCP vs With MCP

What changes for the hunter

WITHOUT MCP

Burp → **Copy** → **Paste** → **AI**

You move every request by hand.

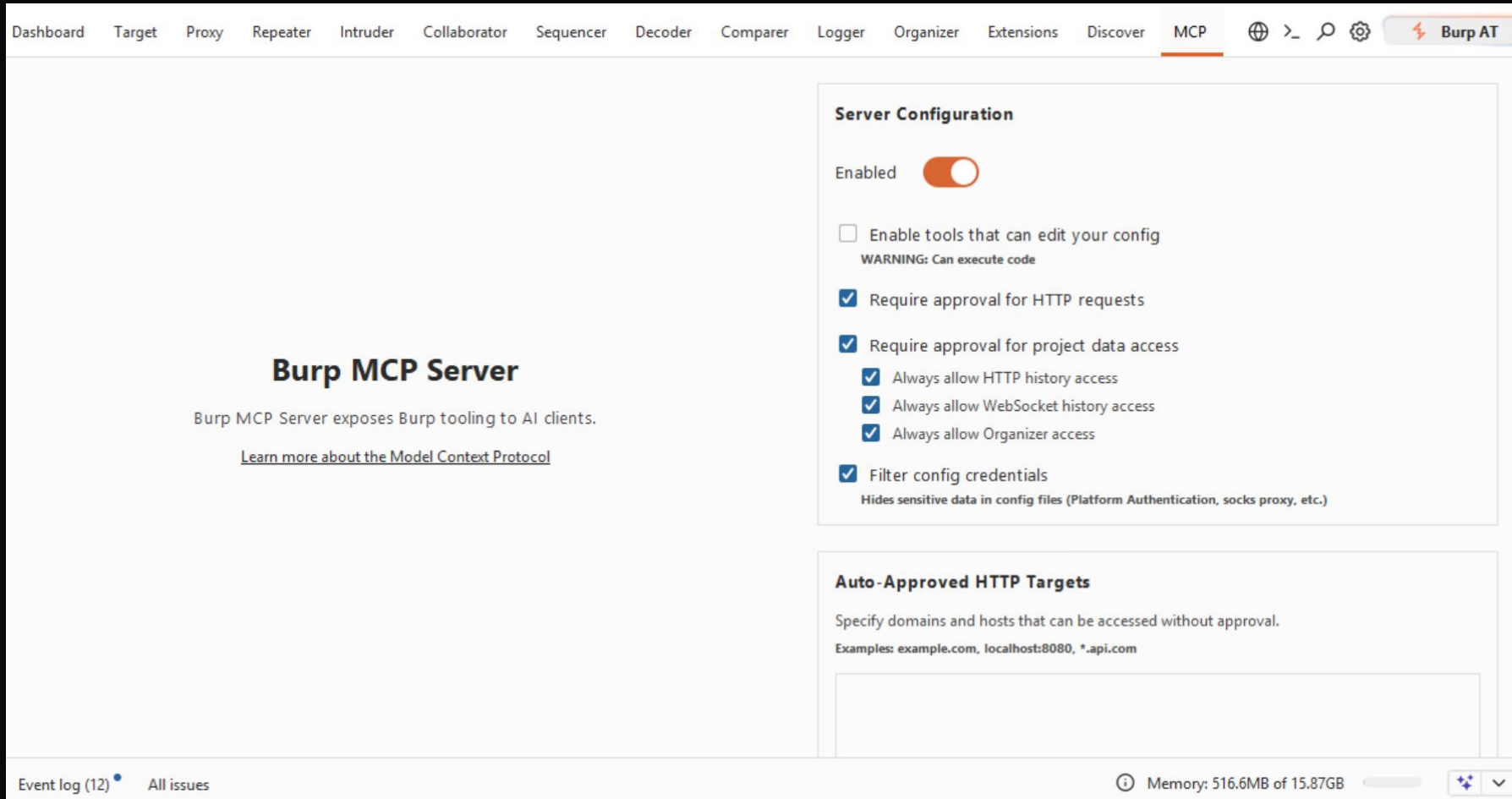
WITH MCP

Burp ↔ **AI Agent**

The agent reads and acts in Burp directly.

BurpSuite with MCP

burp MCP configuration



The screenshot shows the Burp Suite MCP configuration window. The top navigation bar includes tabs for Dashboard, Target, Proxy, Repeater, Intruder, Collaborator, Sequencer, Decoder, Comparer, Logger, Organizer, Extensions, Discover, and MCP (which is currently selected). On the right side of the navigation bar, there are icons for a globe, a magnifying glass, and a gear, along with a 'Burp AT' button.

The main content area is divided into two sections:

- Burp MCP Server**
Burp MCP Server exposes Burp tooling to AI clients.
[Learn more about the Model Context Protocol](#)
- Server Configuration**
 - Enabled ☒
 - ☐ Enable tools that can edit your config
WARNING: Can execute code
 - ☒ Require approval for HTTP requests
 - ☒ Require approval for project data access
 - ☒ Always allow HTTP history access
 - ☒ Always allow WebSocket history access
 - ☒ Always allow Organizer access
 - ☒ Filter config credentials
Hides sensitive data in config files (Platform Authentication, socks proxy, etc.)
- Auto-Approved HTTP Targets**
Specify domains and hosts that can be accessed without approval.
Examples: example.com, localhost:8080, *.api.com

The bottom status bar shows 'Event log (12)' with a blue dot, 'All issues', and 'Memory: 516.6MB of 15.87GB' with a slider and a dropdown menu.

BurpSuite with MCP

```
claude mcp add burp --transport sse http://127.0.0.1:PORT/
```

```
[h4nt3rx@PINOYWHITEHAT]~  
└─> claude  
 Claude Code v2.1.258  
Sonnet 5 · Claude Team  
/home/h4nt3rx
```

```
> /mcp
```

Manage MCP servers

26 servers

```
Local MCPs (/home/h4nt3rx/.claude.json [project: /home/h4nt3rx])  
> burp ·  connected · 27 tools
```

Damn Vulnerable Web Application (DVWA)

The screenshot displays a Windows 10 x64 VM environment. The XAMPP Control Panel v3.3.0 is running, showing the status of various services. The MySQL service is highlighted with a blue border around its 'Stop' button. Below the service list, a log shows the following messages:

```
10:54:41 PM [main] All prerequisites found
10:54:41 PM [main] Initializing Modules
10:54:41 PM [main] Starting Check-Timer
```

A File Explorer window is open to the 'htdocs' directory, showing a list of files and folders:

Name	Date modified	Type
.github	1/3/2025 11:01 PM	File folder
config	1/3/2025 11:16 PM	File folder
dashboard	1/3/2025 10:54 PM	File folder
database	1/3/2025 11:01 PM	File folder
docs	1/3/2025 11:01 PM	File folder
dvwa	1/3/2025 11:06 PM	File folder
external	1/3/2025 11:01 PM	File folder

The web browser window shows the login page for the Damn Vulnerable Web Application (DVWA) at `localhost/login.php`. The page features the DVWA logo and a login form with the following fields:

- Username:
- Password:
- Login button

At the bottom of the page, there is a link: [Damn Vulnerable Web Application \(DVWA\)](#).

The taskbar at the bottom shows the system clock as 10:56 PM on 9/3/2026. A notification for 'Tropical cyclone warn...' is visible.

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.



LIVE WALKTHROUGH

DEMO

AI Agents · AI-assisted Workflow

hackerone

Demo 1 – Javascript analysis

The screenshot shows a web browser window with the URL `http://bugbounty.village`. The page displays the DVWA logo and a navigation menu on the left. The main content area is titled "Welcome to Damn Vulnerable Web Application!" and contains introductory text, general instructions, a warning, a disclaimer, and training resources.

Navigation Menu:

- Home
- Instructions
- Setup / Reset DB
- Brute Force
- Command Injection
- CSRF
- File Inclusion
- File Upload
- Insecure CAPTCHA
- SQL Injection
- SQL Injection (Blind)
- Weak Session IDs
- XSS (DOM)
- XSS (Reflected)
- XSS (Stored)
- CSP Bypass
- JavaScript
- Authorisation Bypass
- Open HTTP Redirect
- Cryptography
- DVWA Security
- PHP Info
- About
- Logout

Welcome to Damn Vulnerable Web Application!

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goal is to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and to aid both students & teachers to learn about web application security in a controlled class room environment.

The aim of DVWA is to practice some of the most common web vulnerabilities, with various levels of difficulty, with a simple straightforward interface.

General Instructions

It is up to the user how they approach DVWA. Either by working through every module at a fixed level, or selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module; however users should feel that they have successfully exploited the system as best as they possible could by using that particular vulnerability.

Please note, there are both documented and undocumented vulnerabilities with this software. This is intentional. You are encouraged to try and discover as many issues as possible.

There is a help button at the bottom of each page, which allows you to view hints & tips for that vulnerability. There are also additional links for further background reading, which relates to that security issue.

WARNING!

Damn Vulnerable Web Application is damn vulnerable! Do not upload it to your hosting provider's public html folder or any Internet facing servers, as they will be compromised. It is recommend using a virtual machine (such as [VirtualBox](#) or [VMware](#)), which is set to NAT networking mode. Inside a guest machine, you can download and install [XAMPP](#) for the web server and database.

Disclaimer

We do not take responsibility for the way in which any one uses this application (DVWA). We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

More Training Resources

DVWA aims to cover the most commonly seen vulnerabilities found in today's web applications. However there are plenty of other issues with web applications. Should you wish to explore any additional attack vectors, or want more difficult challenges, you may wish to look into the following other projects:

- [Mutillidae](#)
- [OWASP Vulnerable Web Applications Directory](#)

Demo 2 – Autonomous AI Agent

The screenshot displays a desktop environment with three main windows:

- Bug Bounty Village / live findings:** A web browser window showing a dashboard with three cards: "BOLA / IDOR", "Excessive Data Exposure", and "Mass Assignment". Each card has a status of "NOT YET FOUND". Below the cards is an "EVENT LOG" section with the text "Waiting for the first request that crosses a boundary..." and a "CLEAR BOARD" button.
- DVWA (Damn Vulnerable Web Application):** A web browser window showing the login page with fields for "Username" and "Password", and a "Login" button.
- VS Code:** A code editor window showing a terminal output. The terminal displays the command `[h4nt3rx@PINOWHITEHAT] - [mnt/./c/claude]` and the output `(20.19.2)(23.0.2)(WSL at 0) - [019,02:50]`.

The Windows taskbar at the bottom shows various application icons, including the Start button, File Explorer, and several web browsers. The system clock in the bottom right corner indicates the time is 2:50 AM on 9/19/2026.

Key Takeaways

What to take back to your next hunt

01 **AI as a Force Multiplier**

Accelerates analysis and reduces repetitive security tasks.

02 **AI + Security Tools**

MCP connects AI agents directly with tools like Burp Suite.

03 **Attack Surface First**

Understand endpoints, parameters, and application behavior before hunting for bugs.

04 **AI Generates Hypotheses**

AI can identify suspicious behavior and suggest potential vulnerabilities.

05 **Human Validation Matters**

Researchers must verify findings and produce real evidence.

06 **Fundamentals Still Matter**

Strong security fundamentals are essential for using AI effectively.

AI-Assisted, not AI-Replaced. AI brings the speed. You bring the expertise.

OPEN FLOOR

Questions?

Japz Divino

japz@wearehackerone.com



hackerone

ROOTCON XX / BUG BOUNTY VILLAGE PH

Thank you!

Japz Divino

japz@wearehackerone.com



hackerone