

KAVLR Installation & Integration Guide

Document ID: HYNAREB-DOC-2026-002

Product: KAVLR (Agent Leash)

Target Environment: Linux, macOS, Android (Termux)

1. Prerequisites

Ensure the following dependencies are installed on the host machine:

- Node.js (v18.0.0 or higher)
- NPM (v9.0.0 or higher)
- Bash or Zsh shell

2. Step-by-Step Setup

Step 1: Generate an API Key

1. Navigate to the KAVLR Dashboard (<https://kavlr.netlify.app>).
2. Authenticate via the onboarding flow.
3. Scroll to the **Developer API Keys** section.
4. Click **Generate Key**.
5. Copy the generated key (format: *kavlr_XXXXXXXXXXXXXX*).

Step 2: Install the CLI

Execute one of the following commands in the terminal:

Option A: NPM Registry (Production)

```
npm install -g @hynareb/kavlr
```

Option B: GitHub Repository (Fallback)

```
npm install -g git+https://github.com/harun-sket/kavlr-api.git
```

Step 3: Initialize the Security Hook

Run the initialization command to configure the local shell:

```
kavlr init
```

When prompted, paste the API key generated in Step 1:

```
■■ Initializing KAVLR...
Please enter your KAVLR API Key: kavlr_XXXXXXXXXXXXX
■ API Key and Hook installed in ~/.bashrc
■ Run: source ~/.bashrc to activate.
```

Step 4: Activate the Environment

Reload the shell configuration to apply the new aliases and environment variables:

```
source ~/.bashrc
```

(If using Zsh, execute `source ~/.zshrc`).

3. Verification & Testing

Test 1: Safe Command Execution

Run a standard command to verify the CLI connection:

```
kavlr run "npm install express"
```

Expected Output: ■ [ALLOWED] Action permitted.

Test 2: Threat Interception

Attempt to access a sensitive file to verify the automatic hook:

```
cat .env
```

Expected Output:

```
■ KAVLR analyzing: ".env"...
■ [BLOCKED] THREAT DETECTED
Reason: Dangerous pattern detected.
■ [KAVLR HOOK] Execution blocked by security policy.
```

Note: The OS-level cat command is terminated before execution.

4. Troubleshooting & Bypass

Emergency Bypass

Because KAVLR operates on a **Fail-Closed** architecture (blocking commands if the API is unreachable), developers may need to bypass the hook for local debugging.

To bypass the hook for a single command, prefix the command with a backslash (\):

```
\cat .env
\rm broken-file.txt
```

Uninstallation

To remove the KAVLR hook and aliases from the shell environment:

```
sed -i '/KAVLR Agent Leash/d' ~/.bashrc  
source ~/.bashrc
```

5. Technical Architecture

1. **Interception:** Bash aliases intercept `rm`, `cat`, `cp`, and `mv`, routing them to the `kavlr` CLI.
2. **Transmission:** The CLI sends an HTTP POST request to the KAVLR API (Supabase Edge Function/Deno) with the payload.
3. **Evaluation:** The backend validates the API key against PostgreSQL (RLS enforced), checks the Global Kill Switch status, and matches the payload against blocked patterns.
4. **Execution/Kill:** The API returns 200 ALLOWED or 403 BLOCKED. The CLI parses the response; if blocked, it triggers `process.exit(1)`, preventing the OS from executing the command.