vge-cc-guard Install Guide

Step-by-step guide to installing vge-cc-guard from npm on a clean machine and wiring it up to Claude Code and VGE.

6 STEPS

1. Install npm package
2. Register hooks in Claude Code
3. Start the daemon
4. Configure via TUI
5. Restart Claude Code
6. Verify end-to-end

Plus: Uninstall, Troubleshooting, What lives where

If you already have `vge-cc-guard` partially installed and want to start fresh, uninstall first (see the Uninstall section), then come back to Step 1.

PREREQUISITES

Before you start, confirm all of these:

REQUIREMENT	CHECK COMMAND	EXPECTED
Node.js 20.10+ (24 recommended)	`node --version`	v20.10.0 or higher
Claude Code installed	`which claude`	a path
VGE API URL	from your VGE admin	https://...
VGE input key	from your VGE admin	32+ characters, vg_test_... or vg_live_...
VGE output key (optional)	from your VGE admin	another 32+ character key with output role

If your VGE input key has both `input` and `output` roles, you can skip the output key.

STEP 1 — INSTALL THE NPM PACKAGE

```
npm install -g @vigil-guard/vge-cc-guard
```

Expected output

```
added 125 packages in 3s

46 packages are looking for funding

run `npm fund` for details
```

> **No** `npm WARN` **and no** `npm ERR!` **means success.** npm does not print "OK" — silence is success.

Verify

```
which vge-cc-guard
```

This should print a path inside your Node install (e.g. `~/.nvm/versions/node/v24.x.x/bin/vge-cc-guard`). If it prints nothing, your global npm bin directory is not on `PATH`. Run `npm bin -g` to find it and add that directory to your `PATH`.

STEP 2 — REGISTER HOOKS IN CLAUDE CODE

This writes seven hook entries into `~/.claude/settings.json`. It does not remove or modify any of your existing hooks.

Preview first

```
vge-cc-guard install --scope=user --dry-run
```

Expected output:

```
[dry-run] Changes that would be applied to settings.json:

+ UserPromptSubmit: vge-cc-guard hook userprompt

+ PreToolUse: vge-cc-guard hook pretool

+ PostToolUse: vge-cc-guard hook posttool

+ SessionStart: vge-cc-guard hook sessionstart

+ SessionEnd: vge-cc-guard hook sessionend

+ SubagentStart: vge-cc-guard hook subagentstart

+ SubagentStop: vge-cc-guard hook subagentstop

Run with --apply to apply changes.
```

> If you see any `-` lines (removals) or different content, stop and investigate before proceeding.

Apply

```
vge-cc-guard install --scope=user --apply
```

Expected output:

```
vge-cc-guard hooks installed to /Users/<you>/.claude/settings.json

Restart Claude Code to activate. Run `vge-cc-guard config` to set your API key.
```

```
NOTE: Write and Edit are gated as `block` by default for safety.

Run `vge-cc-guard config` (Tools Policy) to flip them per project.
```

Verify

```
grep -c "vge-cc-guard hook" ~/.claude/settings.json
```

Should print `7`.

STEP 3 — START THE DAEMON

The TUI configurator (next step) needs a running daemon to verify your VGE credentials. The daemon does not start automatically right after install. Start it manually for now:

```
vge-cc-guard daemon &
```

> The `&` runs the daemon in the background of your current terminal. If a daemon was already running, the new process detects it and exits cleanly via the supersede protocol. See Troubleshooting if needed.

Verify

```
vge-cc-guard daemon status

ls -la ~/.vge-cc-guard/daemon.sock
```

Expected output:

```
Daemon status: running pid=83304 configRevision=1 sidecarEnabled=true build=<sha>

srw------- 1 <you> staff 0 May 18 20:57 /Users/<you>/.vge-cc-guard/daemon.sock
```

Two things to check: `daemon status` says `running`, and `daemon.sock` exists and starts with `s` (Unix domain socket).

STEP 4 — CONFIGURE VIA THE TUI

```
vge-cc-guard config
```

4a. API Keys

» 1. Select `API Keys` and press Enter.
» 2. `VGE API URL`: paste your VGE endpoint.
» 3. `API Key (input)`: paste your input key.
» 4. `API Key (output, optional)`: paste output key if you have one, else leave empty.
» 5. `Source`: leave on `Auto` (sends OS username to VGE as client ID).
» 6. Tab to `[Test Connection]` and press Enter.

> **Expected:** a green `verified_at:` line appears. If you see `Connection failed: daemon unavailable`, the daemon died. Go back to Step 3 and restart it.

» 7. Tab to `[Save]` and press Enter.
» 8. Esc to return to the main menu.

4b. Tools Policy

The defaults are safe but conservative. Pay attention to `Write` and `Edit` — they are set to `block` by default, meaning every file edit in Claude Code will prompt you for permission.

GATE	MEANING
`allow`	Pass through with optional VGE analysis.
`ask`	Claude Code shows a native approval prompt before running.
`block`	Tool is rejected before execution.

Press a key (usually g or Enter) on a row to cycle through gates. Toggle `analyze_output` per tool if you want VGE scoring of the output. Esc to return to the main menu. Changes are saved on Esc.

4c. Security Baseline

Leave `Credential path protection` set to `ON`. This is the hard-coded deny list for `.env`, `.ssh/`, `.aws/credentials`, etc.

Adjust `Session TTL` if you want session-scoped decisions to expire faster or slower than the default. Esc to return to the main menu.

4d. View Configuration

Read-only redacted JSON dump. Press e to export a sanitized copy to `~/.vge-cc-guard/config.export-.txt` (mode 0600). Esc twice to exit the TUI completely.

STEP 5 — RESTART CLAUDE CODE

Existing Claude Code sessions do not know about the new hooks. Close every open Claude Code window and reopen one. After the new window opens, the `SessionStart` hook fires, the shim lazy-starts the daemon (or finds the one you already started), and the sidecar is live for that session.

STEP 6 — VERIFY END-TO-END

```
vge-cc-guard doctor --cc-contract
```

Expected output (everything green):

```
Claude Code contract state: healthy

L0 output replacement: enabled

Reason: live_verified

Live contract: verified

Claude Code version: 2.x.x (Claude Code)

VGE connectivity: healthy

VGE API URL: <your URL>

VGE input key: ok

VGE output key: ok # or `uses_input_key` if you skipped output key

System CA store: supported

Daemon CA snapshot: matches

CA drift: none
```

If `Claude Code contract state` is still `degraded` with reason `status_missing`, the auto-probe has not run yet. Do one tool call in Claude Code (e.g. ask it to `ls`) and re-run `doctor --cc-contract`. It should flip to `healthy`.

Smoke test in Claude Code

» Ask Claude to run `ls` — should pass without prompting.

» Ask Claude to `cat ~/.ssh/id_rsa` — should be **hard-denied** by credential path protection.

» Ask Claude to edit a file (if Write/Edit are on `block`) — should prompt you to approve.

If all three behave as expected, the install is complete.

UNINSTALL

To remove `vge-cc-guard` cleanly without touching your other Claude Code hooks:

```
# 1. Stop the daemon

vge-cc-guard daemon stop


# 2. Remove hook entries from ~/.claude/settings.json

# (leaves your other hooks and permissions intact)

vge-cc-guard uninstall --yes --scope=user


# 3. Remove the npm package

npm uninstall -g @vigil-guard/vge-cc-guard


# 4. (Optional) Wipe local state

# Skip if you want to keep audit history and re-install later

cd ~ && rm -rf .vge-cc-guard
```

> The `uninstall` command preserves any native Claude Code hooks or permissions you added before installing the sidecar. Use `--restore` if you want to fully replace `settings.json` with the install-time backup.

TROUBLESHOOTING

Connection failed: daemon unavailable

The daemon is not running. Run:

```
vge-cc-guard daemon &

vge-cc-guard daemon status
```

Then retry Test Connection.

vge-cc-guard: command not found

Your npm global bin directory is not on `PATH`. Run `npm bin -g` to find it and add that directory to your shell's `PATH`. If you use nvm, the binary lives under `~/.nvm/versions/node//bin/`.

Claude Code contract state: degraded (status_missing)

The auto-probe has not run yet. Do one tool action in Claude Code (any), then:

```
vge-cc-guard doctor --cc-contract
```

Daemon dies when I close my terminal

For day-to-day use, do not start the daemon manually at all. After Step 5 (restart Claude Code), the `SessionStart` hook lazy-starts the daemon on every Claude Code launch, and it survives until Claude Code exits.

[1] + done vge-cc-guard daemon right after daemon &

This is the supersede protocol: another daemon process was already running. Run `vge-cc-guard daemon status` to confirm a daemon is still alive.

sidecarEnabled=unknown after upgrade

The running daemon was started by an older build. Restart it:

```
vge-cc-guard daemon stop

vge-cc-guard daemon
```

Existing Claude Code session does not see the hooks

Claude Code reads `settings.json` once per session. After `install --apply`, restart Claude Code (close and reopen the window).

Test Connection works but Claude Code never triggers hooks

Confirm your installation scope matches your Claude Code usage. `--scope=user` writes to `~/.claude/settings.json` (applies to all projects). If your project has its own `.claude/settings.json` with conflicting entries, that overrides. Run:

```
vge-cc-guard install --scope=project --dry-run
```

WHAT LIVES WHERE

PATH	WHAT IT IS
~/.claude/settings.json	Claude Code config — the seven hook entries live here.
~/.vge-cc-guard/config.json	Sidecar config (mode 0600): VGE URL, keys, tool policy.
~/.vge-cc-guard/daemon.sock	Unix socket — shim talks to daemon through this.
~/.vge-cc-guard/audit.log	Per-day audit trail of every gated tool call.
~/.vge-cc-guard/sessions/	Per-session state (allowlists, escalations).
~/.vge-cc-guard/installs/	Backup copies of settings.json taken before each install.
~/.vge-cc-guard/debug.log	Daemon log (errors, warnings, startup info).

Vigil Guard Enterprise
Day-to-day configuration reference: see the User Guide.

vigilguard.ai | contact@vigilguard.ai