vge-cc-guard Instrukcja instalacji

Krok po kroku: instalacja vge-cc-guard z npm na czystej maszynie i pełne podłączenie go do Claude Code i VGE.

6 KROKÓW

Jeśli masz już częściowo zainstalowany `vge-cc-guard` i chcesz zacząć od nowa, najpierw przejdź do sekcji Odinstalowanie, a potem wróć do Kroku 1.

WYMAGANIA WSTĘPNE

Przed rozpoczęciem potwierdź wszystkie poniższe wymagania:

WYMAGANIE	KOMENDA SPRAWDZAJĄCA	OCZEKIWANY WYNIK
Node.js 20.10+ (zalecane 24)	`node --version`	v20.10.0 lub nowszy
Zainstalowany Claude Code	`which claude`	ścieżka
URL API VGE	od administratora VGE	https://...
Klucz wejściowy VGE	od administratora VGE	ponad 32 znaki, vg_test_... albo vg_live_...
Klucz wyjściowy VGE (opcjonalny)	od administratora VGE	inny klucz z rolą output

Jeśli Twój klucz wejściowy VGE ma zarówno role `input`, jak i `output`, możesz pominąć klucz wyjściowy.

KROK 1 — ZAINSTALUJ PAKIET NPM

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

Oczekiwany wynik

```
added 125 packages in 3s

46 packages are looking for funding

run `npm fund` for details
```

> Brak `npm WARN` i brak `npm ERR!` oznacza sukces. npm nie wypisuje "OK" — cisza oznacza powodzenie.

Zweryfikuj

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

To powinno wypisać ścieżkę wewnątrz Twojej instalacji Node (np. `~/.nvm/versions/node/v24.x.x/bin/vge-cc-guard`). Jeśli nic nie wypisuje, globalny katalog binarny npm nie znajduje się w `PATH`. Uruchom `npm bin -g`, aby go znaleźć, i dodaj ten katalog do `PATH`.

KROK 2 — ZAREJESTRUJ HOOKI W CLAUDE CODE

Ten krok zapisuje siedem wpisów hooków w `~/.claude/settings.json`. Nie usuwa ani nie modyfikuje żadnych istniejących hooków.

Najpierw podejrzyj

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

Oczekiwany wynik:

```
[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.
```

> Jeśli widzisz jakiekolwiek linie z `-` (usunięcia) albo inną treść, zatrzymaj się i sprawdź przyczynę przed przejściem dalej.

Zastosuj

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

Oczekiwany wynik:

```
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.
```

Zweryfikuj

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

Powinno wypisać 7.

KROK 3 — URUCHOM DAEMON

Konfigurator TUI (następny krok) potrzebuje działającego daemona, aby zweryfikować poświadczenia VGE. Daemon nie startuje automatycznie bezpośrednio po instalacji. Uruchamiamy go ręcznie:

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

> Znak & uruchamia daemona w tle bieżącego terminala. Jeśli daemon już działał, nowy proces wykrywa go i kończy się poprawnie przez protokół zastępowania. Oryginaly daemon nadal działa.

Zweryfikuj

```
vge-cc-guard daemon status

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

Oczekiwany wynik:

```
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
```

Dwie rzeczy do sprawdzenia: `daemon status` mówi `running`, i `daemon.sock` istnieje i zaczyna się od `s` (gniazdo domeny Unix).

KROK 4 — SKONFIGURUJ PRZEZ TUI

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

4a. API Keys

» 1. Wejdź w `API Keys` (Enter).
» 2. `VGE API URL`: wklej endpoint VGE.
» 3. `API Key (input)`: wklej klucz wejściowy.
» 4. `API Key (output, optional)`: wklej klucz wyjściowy jeśli go masz; w przeciwnym razie zostaw puste.
» 5. `Source`: zostaw `Auto` (wysyła nazwę użytkownika systemowego do VGE).
» 6. Przejdź Tab do `[Test Connection]` i naciśnij Enter.

> Oczekiwane: pojawia się zielona linia `verified_at: .` Jeśli widzisz `Connection failed: daemon unavailable`, daemon zakończył działanie. Wróć do Kroku 3 i uruchom go ponownie.

» 7. Przejdź Tab do `[Save]` i naciśnij Enter.
» 8. Esc, aby wrócić do menu głównego.

4b. Tools Policy

Domyślne ustawienia są bezpieczne, ale konserwatywne. Zwróć szczególną uwagę na `Write` i `Edit` — domyślnie są ustawione na `block`, co oznacza, że każda edycja pliku w Claude Code poprosi Cię o zgodę.

BRAMKA	ZNACZENIE
`allow`	Przepuść z opcjonalną analizą VGE.
`ask`	Claude Code pokazuje natywny monit zatwierdzenia przed uruchomieniem.
`block`	Narzędzie zostaje odrzucone przed uruchomieniem.

Naciśnij klawisz (zwykle g lub Enter) na wierszu, aby przełączać bramki. Przełącz `analyze_output` dla narzędzia, jeśli chcesz oceniać jego wynik przez VGE. Esc, aby wrócić do menu. Zmiany zapisują się po Esc.

4c. Security Baseline

Zostaw `Credential path protection` ustawione na `ON`. To twardo zakodowana lista odmowy dla `.env`, `.ssh/`, `.aws/credentials` itd.

Dostosuj `Session TTL`, jeśli chcesz, aby decyzje ograniczone do sesji wygasały szybciej lub wolniej niż domyślnie. Esc, aby wrócić do menu.

4d. View Configuration

Tylko do odczytu, zredagowany zrzut JSON. Naciśnij e, aby wyeksportować oczyszczoną kopię do `~/.vge-cc-guard/config.export-.txt` (tryb 0600). Naciśnij Esc dwa razy, aby całkowicie wyjść z TUI.

KROK 5 — ZRESTARTUJ CLAUDE CODE

Istniejące sesje Claude Code nie wiedzą o nowych hookach. Zamknij każde otwarte okno Claude Code i otwórz nowe. Po otwarciu nowego okna uruchamia się hook `SessionStart`, shim leniwie startuje daemona (albo znajduje już uruchomionego), a sidecar działa dla tej sesji.

KROK 6 — ZWERYFIKUJ END-TO-END

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

Oczekiwany wynik (wszystko zielone):

```
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 # albo `uses_input_key` jeśli pominięto klucz wyjściowy

System CA store: supported

Daemon CA snapshot: matches

CA drift: none
```

Jeśli `Claude Code contract state` nadal ma wartość `degraded` z powodem `status_missing`, auto-probe jeszcze się nie uruchomił. Wykonaj jedno wywołanie narzędzia w Claude Code (np. poproś o `ls`) i ponownie uruchom `doctor --cc-contract`. Status powinien przejść na `healthy`.

Szybki test integracji w Claude Code

» Poproś Claude o uruchomienie `ls` — powinno przejść bez monitu.
» Poproś Claude o `cat ~/.ssh/id_rsa` — powinno zostać twardo odrzucone przez ochronę ścieżek poświadczeń.
» Poproś Claude o edycję pliku (jeśli Write/Edit na `block`) — powinien pojawić się monit o zatwierdzenie.

Jeśli wszystkie trzy zachowania są zgodne z oczekiwaniami, instalacja jest ukończona.

ODINSTALOWANIE

Aby czysto usunąć `vge-cc-guard` bez naruszania innych hooków Claude Code:

```
# 1. Zatrzymaj daemona
vge-cc-guard daemon stop

# 2. Usuń wpisy hooków z ~/.claude/settings.json
# (zachowuje inne hooki i uprawnienia)
vge-cc-guard uninstall --yes --scope=user

# 3. Usuń pakiet npm
npm uninstall -g @vigil-guard/vge-cc-guard

# 4. (Opcjonalne) Usuń lokalny stan
# Pomiń, jeśli chcesz zachować historię audytu
cd ~ && rm -rf .vge-cc-guard
```

> Komenda `uninstall` zachowuje wszystkie natywne hooki lub uprawnienia Claude Code dodane przed instalacją sidecara. Użyj `--restore`, jeśli chcesz w pełni zastąpić `settings.json` kopią zapasową z czasu instalacji.

ROZWIĄZYWANIE PROBLEMÓW

Connection failed: daemon unavailable

Daemon nie działa. Uruchom:

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

Następnie ponów Test Connection.

vge-cc-guard: command not found

Globalny katalog binarny npm nie znajduje się w `PATH`. Uruchom `npm bin -g`, aby go znaleźć, i dodaj ten katalog do `PATH` swojej powłoki. Jeśli używasz nvm: `~/.nvm/versions/node//bin/`.

Claude Code contract state: degraded (status_missing)

Auto-probe jeszcze się nie uruchomił. Wykonaj jedną akcję narzędzia w Claude Code, a potem:

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

Daemon wyłącza się od razu po zamknięciu terminala

Do codziennego użycia w ogóle nie uruchamiaj daemona ręcznie. Po Kroku 5 (restart Claude Code) hook `SessionStart` leniwie uruchamia daemona przy każdym starcie Claude Code i daemon działa do zamknięcia Claude Code.

[1] + done vge-cc-guard daemon zaraz po daemon &

To protokół zastępowania: inny proces daemona już działał. Uruchom `vge-cc-guard daemon status`, aby potwierdzić, że daemon nadal działa.

daemon status raportuje sidecarEnabled=unknown po aktualizacji

Działający daemon został uruchomiony przez starszą wersję. Zrestartuj go:

```
vge-cc-guard daemon stop

vge-cc-guard daemon
```

Istniejąca sesja Claude Code nie widzi hooków

Claude Code odczytuje `settings.json` raz na sesję. Po `install --apply`, zrestartuj Claude Code (zamknij okno i otwórz nowe).

Test Connection działa, ale Claude Code nigdy nie wyzwala hooków

Potwierdź, że zakres instalacji odpowiada sposobowi używania Claude Code. `--scope=user` zapisuje do `~/.claude/settings.json`. Jeśli projekt ma własny `.claude/settings.json` z konfliktującymi wpisami, to on ma pierwszeństwo. Sprawdź:

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

CO ZNAJDUJE SIĘ GDZIE

ŚCIEŻKA	CO TO JEST
~/.claude/settings.json	Konfiguracja Claude Code — tutaj znajdują się siedem wpisów hooków.
~/.vge-cc-guard/config.json	Konfiguracja sidecara (tryb 0600): URL VGE, klucze, polityka narzędzi.
~/.vge-cc-guard/daemon.sock	Gniazdo Unix — shim rozmawia z daemonem przez to gniazdo.
~/.vge-cc-guard/audit.log	Dzienny ślad audytowy każdego bramkowanego wywołania narzędzia.
~/.vge-cc-guard/sessions/	Stan per sesja (allowlisty, eskalacje).
~/.vge-cc-guard/installs/	Kopie zapasowe settings.json wykonane przed każdą instalacją.
~/.vge-cc-guard/debug.log	Log daemona (błędy, ostrzeżenia, informacje startowe).

Vigil Guard Enterprise
Codzienne odniesienie do konfiguratora: patrz Przewodnik użytkownika.

vigilguard.ai | contact@vigilguard.ai