

Security Hooks

- [Security Hooks \(Kotlin/GraalVM\)](#)

Security Hooks (Kotlin/GraalVM)

Security Hooks

All security hooks run as PreToolUse gates. Exit 2 = BLOCK, Exit 0 = ALLOW. Binary:

```
~/ .claude/hooks/alai-hooks
```

BashSecurityGate (alai-hooks bash)

Event: PreToolUse[Bash] | **ZAKON:** Multiple

Blocks dangerous shell commands:

- **NPM Audit Gate:** Blocks known malicious packages and dangerous flags
- **Destructive Commands:** DROP TABLE/DATABASE, DELETE without WHERE, dangerous git operations, recursive rm, chmod 777
- **Exfiltration Detection:** Blocks curl/wget to known exfil domains. Detects pipe-to-curl and DNS exfiltration
- **Shell Injection:** Blocks pipe to interpreter, eval, command substitution with dangerous commands
- **Inline SMTP:** Blocks inline email scripts (ZAKON #6)

WriteSecurityGate (alai-hooks write)

Event: PreToolUse[Write|Edit|MultiEdit]

Blocks writes to protected paths:

- ~/.ssh, ~/.gnupg, ~/.aws (credential theft)
- ~/Documents, ~/Desktop, ~/Downloads (security policy)
- Browser profiles, Keychains, Mail, Messages, Photos
- Advisory warning for secrets/API keys in file content

DeployGateZakon (alai-hooks deploy-gate)

Event: PreToolUse[Bash] | **ZAKON:** #2, #19

Blocks production deployments without CEO approval:

- `az containerapp update/create` blocked unless `/tmp/ceo-approved-deploy` exists
- `docker push` to production ACR blocked unless approved
- Strips heredoc content before pattern matching

BackendEditGuard (alai-hooks backend-guard)

Event: PreToolUse[Write|Edit|MultiEdit] | **ZAKON:** #20, #5

Prevents John from directly editing backend code:

- Detects `.java`, `.kt`, `.go` files in backend paths
- Skips subagent context (`/tmp/alai-subagent-context`)
- Warn mode (default) or strict mode (`/tmp/backend-edit-strict`)

HallucinationDetector (alai-hooks hallucination)

Event: PreToolUse[Write|Edit|MultiEdit] | **ZAKON:** #1

5-layer anti-hallucination defense:

1. **Known Wrong Facts:** Blocks known-incorrect values (wrong names, org numbers, API endpoints)
2. **Phantom Tools:** Blocks references to tools confirmed non-existent
3. **Wrong Ports:** Flags localhost ports not in known services map
4. **Phantom Endpoints:** Blocks known-invalid API endpoints for tracked services
5. **Phantom Paths:** Detects hardcoded file paths that don't exist on disk

Skips: `~/system/config/` files, `/tmp` paths, URLs, wildcards, template strings