/security-audit

Source: `~/.claude/skills/security-audit/SKILL.md`

---

name: security-audit version: 2.0 description: > Run comprehensive security audit following OWASP and ALAI LAWS. Use for: "security review", "audit this code", "check for vulnerabilities", "OWASP check", "before deploying", "security scan", "/security-audit". level: 3 company: Securion

/security-audit — Security Review

Purpose

Systematic security review covering OWASP Top 10, ALAI internal LAWS, and code-specific vulnerabilities.

Variables

Variable	Type	Description	Default
`target`	path/scope	File, directory, or "full system"	current project
`depth`	string	quick | standard | deep	standard
`focus`	string	owasp | laws | auth | api | all	all

Workflow

Step 1: Scope

- Read $ARGUMENTS to determine target and depth
- if no target → audit current working directory
- if depth=quick → run only LAWS + auth checks
- if depth=deep → run all + tob-* skill checks

Step 2: ALAI LAWS Compliance

Check each LAW:

- **ZAKON 0 (Tajnost)**: No secrets in code, no internal URLs exposed, no employee data hardcoded
- **ZAKON 1 (Ne škodi)**: No destructive ops without confirm, backups exist for critical data
- **ZAKON 2 (Slušaj)**: Auth on all endpoints, RBAC, admin routes protected
- **ZAKON 3 (Čuvaj sebe)**: Error handling, graceful degradation, no crash on bad input

Step 3: OWASP Top 10 Check

For each category, scan target:

1. Injection (SQL, NoSQL, command injection)
2. Broken Authentication (weak JWT, no rate limit, session issues)
3. Sensitive Data Exposure (logs, responses, hardcoded secrets)
4. Security Misconfiguration (CORS, headers, default credentials)

5. XSS (reflected, stored, DOM-based)
6. Broken Access Control (IDOR, privilege escalation)
7. Vulnerable Dependencies (`npm audit` or equivalent)
8. Insecure Deserialization
9. Logging & Monitoring gaps
10. SSRF

Step 4: Run Available Tools

- if tob-static-analysis available → run on target
- if tob-insecure-defaults available → check configs
- if tob-sharp-edges available → check dangerous patterns
- `npm audit --audit-level=high` if package.json exists

Step 5: Report

- if CRITICAL found → flag for immediate fix, offer to create MC task
- if HIGH found → list with recommended fixes
- if depth=deep → include code snippets for each finding

Report Format

```
SECURITY AUDIT REPORT
Target: [scope]
Depth: [quick|standard|deep]
Date: [timestamp]

CRITICAL (block deployment):
  [C1] [finding] — [file:line] — [fix]

HIGH (fix before next release):
  [H1] [finding] — [fix]

MEDIUM:
  [M1] [finding]

LAWS: [PASS|FAIL — list failures]
OWASP: [X/10 categories clean]
Tools run: [list]
```

$ARGUMENTS

Revision #3
Created 2026-04-16 21:45:06 UTC by John
Updated 2026-06-21 20:02:52 UTC by John