LARAVEL CHEAT SHEET · SECURITY

How not to get hacked: the quarterly checklist

dominik-dev.pl
full article →

Server and network, application, CI/CD, data, detection, the first 72 hours

TL;DR: the order that pays off first

1	2	3	4	5	6
External scan, close all but 80/443	SSH keys only, over VPN	APP_DEBUG=false, web root on public/, composer audit	Policies + 403 tests on every ID	3-2-1 backups, restore test	Central logs, port diff, alerts

> **DOCKER** `ports: "6379:6379"` publishes on all interfaces and Docker's iptables rules run **before UFW**. `ufw status` lies; only an external `nmap -p-` tells the truth. Use `127.0.0.1:6379:6379` or no `ports` plus an `internal: true` network.

SERVER AND NETWORK

- ☐ External `nmap -Pn -p-` shows only 80 and 443
- ☐ Firewall denies incoming by default, in the cloud panel too
- ☐ No Compose `ports:` exposes DB, Redis or a panel on `0.0.0.0`
- ☐ SSH keys only, root off, access via VPN or bastion
- ☐ Security updates automatic, images rebuilt this month
- ☐ phpMyAdmin, Horizon, Telescope, Grafana not public

APPLICATION

- ☐ `APP_ENV=production`, `APP_DEBUG=false`
- ☐ Web root on `public/`; `/.env`, `/.git/` return 404
- ☐ Every endpoint with an ID: Policy + 403 test for another user
- ☐ Saves only from `validated()`; models define `$fillable`
- ☐ No `whereRaw` / `DB::raw` with unbound input
- ☐ Throttle on login, reset, registration, 2FA, API
- ☐ Uploads validated, private disk, no PHP execution
- ☐ HSTS, CSP, `nosniff`, `frame-ancestors`
- ☐ Session cookies `Secure`, `HttpOnly`, `SameSite`

CI/CD AND DEPENDENCIES

- ☐ `composer audit` + `npm audit` on every PR
- ☐ Dependabot or Renovate enabled
- ☐ Secret scanner (gitleaks) blocks commits with keys
- ☐ Pipeline blocks deploys with `APP_DEBUG=true`

DATA AND BACKUPS

- ☐ 3-2-1 backups, one immutable or offline copy
- ☐ Full restore tested within the last quarter
- ☐ Sensitive fields encrypted at rest (`encrypted` cast)
- ☐ Secrets inventory + rotation procedure written down

DETECTION

- ☐ App, web server and system logs in one place
- ☐ Alerts: failed logins, `/.env` probes, SSH outside VPN, new port, outbound spike
- ☐ Daily external port diff against a baseline
- ☐ File integrity monitoring (Wazuh / AIDE)

Daily port diff (run from outside the network)

```
current="$(nmap -Pn -p- --min-rate 1000 -oG - "$host" | grep -oE '[0-9]+/open' | sort -u)"
baseline="$(cat ~/port-diff/$host.baseline)"
[[ "$current" == "$baseline" ]] || echo "$current" | mail -s "port-diff: $host" ops@example.com
```

Incident: the first 72 hours

1. **Don't wipe:** snapshot the disk, preserve logs. **Isolate** the host, keep it running if possible.
2. **Close the way in**, then **rotate every secret**: DB, API keys, `APP_KEY`, CI tokens, SSH keys.
3. **Rebuild from a clean image**, restore data from a pre-attack backup. Assess what leaked, whose, since when.
4. **Report** (clock starts when you became aware), inform users, write a blameless post-mortem.

Regulation	Notify	Deadline
GDPR: personal data at risk	data protection authority (+ people if risk is high)	72 h
NIS2: essential/important entity, significant incident	national CSIRT	24 h warning · 72 h notification · 1 month final report