Engineer

How we work securely

How Engineer ApS builds and operates securely: the practices we hold to, how this site is hardened, and where to report a vulnerability.

Security is not a service we sell separately. It is how the work is done: the threat model is part of the design, the hardening is part of the build, and the proof is part of the handover.

Engineer the controls from the start

Make access, deployment and recovery part of the engineering from the start.

Explore our services Get in touch →

Look behind a security claim

Ask for the configuration and tests behind a security claim, with its scope made explicit.

Cases Get in touch →

How we build

- **Least privilege by default.** Services run as their own unprivileged users, with the narrowest access that lets them work. Access boundaries are designed to limit movement between components if one is compromised.
- **Secrets never reach the repository.** Credentials live in a secret store or in the deployment layer, never in source and never in a build artefact. Automated scanning runs on every commit, and a hit stops the commit.
- **Infrastructure described as code.** Servers are built from a description that can be read, reviewed and rebuilt — so the configuration of a machine is a document under version control rather than a memory in somebody's head.
- **Dependencies pinned and audited.** What a system is made of is written down, and the list is checked against published advisories rather than trusted because it worked last month.
- **Encryption in transit, everywhere.** Certificates are issued and renewed automatically, and the renewal is monitored, because the common way TLS fails is a certificate nobody was watching.

- **Backup automation with separate restore tests.** A backup is a claim until somebody restores from it. We implement backup and restore-test workflows separately; verification rests on the data actually restored and checked.

Every line above is **an ordinary working day** — not a phase that gets added at the end.

Agree security boundaries, responsibilities and checks as part of the design.

How this site is built

We hold our own site to the same standard, and you can verify every line of this from your own browser.

- **No backend, no database, no admin login.** Every page is a static file written before it is requested. This avoids application sessions and database query endpoints, reducing the site's application attack surface.
- **Zero executable JavaScript.** Pages, menus and links use native HTML and CSS; the structured data is inert JSON-LD. There is no browser script to make a request or store a preference.
- **A strict Content Security Policy**, declared in the page itself, with `form-action 'none'` — the site posts no forms anywhere.
- **Subresource Integrity on the stylesheet.** The bundle is fingerprinted and hashed at build time, so your browser can verify it arrived unmodified.
- **No third-party requests.** No third-party analytics, font services or map widgets, and no consent banner. Open your browser's network panel and count the hosts: there is one, and it is ours.
- **No cookies.** The whole of what is and is not stored is on the cookies page.
- **A machine-readable disclosure policy** at `/.well-known/security.txt`, so a researcher who finds something knows where to send it without asking.

How it is operated

One small server in Toronto, described in code, with the access log masked as it is written and kept about thirty days. The credits page names every piece of software on it; the privacy page says exactly what it records while it answers.

Reporting a vulnerability

If you have found a weakness in this site or in anything we operate, write to welcome@engineer.company and say what you found and how to reproduce it. We will confirm receipt, tell you what we are doing about it, and credit you if you would like to be credited. There is no bounty programme and no legal threat — just an address that reaches an engineer.

Want the same rigour applied to your own systems? ask us to look.

This page is a claim **you can audit** — from the browser you are reading it in.

No backend, no cookies, zero executable JavaScript, and a policy the page declares in itself.

Ask us to look at yours

Identify what needs protection

Start with the assets, trust boundaries and permissions in the system you need to protect.

See what it costs Get in touch →

Agree a focused security engagement

Discuss a focused review or implementation with concrete controls and acceptance checks.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://ditte.engineer.company/security/