Engineer

Cut systemd sandbox exposure on every unit

Cut systemd sandbox exposure across every unit this platform installs — a dead-man's-switch service from 9.6 UNSAFE to 1.5, an internet-facing git forge from 8.3 EXPOSED to 1.5 — and added a converge-time parser check after finding a misspelled directive silently ignored in three unit templates.

September 2026

every unit sandboxed, and the typo that was ignored

Situation. Thirteen unit files were installed by this repository and run on the production host with the privileges the supervisor gives by default, which is most of them. A hardening score put the dead-man’s-switch service at 9.6 and called it unsafe; the internet-facing git forge scored 8.3 and exposed. Both numbers were accurate and neither had prompted anything, because a score with no threshold attached is a number people learn to skip.

Task. Every unit needed a sandbox proportional to what it actually does, and the sandboxes needed a check, because the failure mode of a wrong sandbox is that the unit starts and then breaks somewhere the parser cannot see.

Action. Privilege restrictions went into every unit template — a privilege ceiling fixed at the unit’s own, a system-call filter, restricted address families, and memory that is writable or executable but never both. The measured results: the dead-man’s switch went from 9.6 to 1.5, the forge from 8.3 to 1.5, and both backup units from 9.6 to 2.3 and 2.5. Writing them turned up something better than the scores. One directive had been misspelled in three unit templates for as long as those roles had existed — a plausible-looking name that does not exist, which the supervisor answers by logging that it does not recognise the key and starting the unit anyway. Grepping for a directive proves it was written; only the parser proves it took effect. So a verification role now runs every unit through the supervisor’s own verifier during the converge, and is included by every role that installs a unit. What that still cannot see is stated

plainly in the same documents: every directive capable of breaking these units breaks them at runtime, not at parse time, and a score cannot tell whether a script still reaches its last line.

Result. Every unit runs with the privileges it needs and not the ones it does not, the numbers are measured rather than asserted, and the class of defect where a typo becomes a silently absent control now fails a run. The limits of the measurement are recorded next to it, which is the part that keeps the next reader from over-trusting a green band.

Bring this experience to your project

Use “Cut systemd sandbox exposure on every unit” as a starting point for a similar challenge in your organisation. The services behind this work are available for your project. The whole setup written down as files, so it can be recreated exactly and never lives only in one person's head.

Explore our services Get in touch →

Look behind the result

Read the situation, decisions and result in this example. We can discuss which parts transfer to your constraints and which need another approach. Cut systemd sandbox exposure on every unit — every unit sandboxed, and the typo that was ignored

Explore our services Get in touch →

Do your services have more access than they need?

Talk through your project with the engineer responsible for delivery. We agree the scope, price basis and handover in writing before work begins.

Role context: Site Reliability Engineer. It draws on Infrastructure, Containers (Docker/Kubernetes), Automation & CI/CD, Linux & Servers, Security and Testing & QA. And while you are here, we are rather good at Infrastructure as Code, Containerization & Orchestration, System Administration and Security & Access Management.

Get in touch Explore our services

One piece of **a longer record** — and it is all on this site.

Every entry is written the same way: the situation, the task, what we did and what changed.

See the whole record

Define the improvement you need

Bring the deployment path, the operational risks and the people on call. We can scope a change with checks, a recovery route and clear ownership.

See what it costs Get in touch →

Do your services have more access than they need?

Ask about the decisions behind “Cut systemd sandbox exposure on every unit”. You will speak with the people who can explain the work and assess your own project.

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://engineer.company/portfolio/cut-systemd-sandbox-exposure-on-every-unit/