Engineer Programvare

Herdet applikasjonen med nonce-basert CSP

Herdet applikasjonen med nonce-basert CSP, HSTS, SameSite-informasjonskapsler, databaseroller etter prinsippet om minste privilegium og ny kontroll av tilgangsrettigheter på serversiden.

2025

forsvar som holder når nettleseren er fiendtlig

Situasjon. Plattformen inneholder data om fagpersoner og organisasjoner, den typen folk forventer blir håndtert ordentlig, så én enkelt forsvarslinje kom aldri til å være nok. Arbeidsantakelsen må være at klienten er fiendtlig — at den som sitter med nettleseren, kan slå av alt nettleseren håndhever — og sikkerheten må holde likevel.

Oppgave. Plattformen måtte herdes i hvert lag — frontend, API, database — slik at sikkerheten ble håndhevet av serveren uavhengig av hva grensesnittet tilfeldigvis tillot.

Handling. I frontenden setter proxy-mellomvaren i Next.js — proxy.ts — en Content-Security-Policy med en nonce per forespørsel og strict-dynamic, i tillegg til HSTS og SameSite-cookies, så nettleseren er låst ned med tanke på hva den vil kjøre og sende. I API-et finnes hastighetsbegrensning, CORS, grenser for forespørselsstørrelse, inndatavalidering før noe rører databasen, og logging av sikkerhetsrelevante hendelser. I databasen logger API-et inn som en rolle med minst mulig rettigheter, som bare har EXECUTE på appfunksjonene, funksjonene kjører som SECURITY DEFINER, og alt er parameterisert. Og rettighetene — abonnementsnivå, rolle, organisasjon, avgrensning per skip — kontrolleres på nytt på serveren ved hver forespørsel, og sperrene i frontenden behandles bare som UX. Sperrene avgjør hva man ser; serveren avgjør hva man kan gjøre.

Resultat. Sikkerheten er ikke avhengig av at brukergrensesnittet oppfører seg. Beskyttelsene ligger i lag for å unngå å være avhengig av én enkelt kontroll. Autorisasjonen på serveren er utformet for å fungere uavhengig av

kontrollene på klienten, slik at grensesnittet ikke avgjør hvem som får tilgang.

Bruk erfaringen i prosjektet deres

Bruk «Herdet applikasjonen med nonce-basert CSP» som utgangspunkt for en lignende utfordring i organisasjonen deres. Tjenestene bak arbeidet er tilgjengelige for prosjektet deres. Delen brukerne aldri ser: hvor dataene ligger, hvilke regler som gjelder, og hvordan andre systemer får lov til å snakke med den.

Se tjenestene våre Ta kontakt →

Se bak resultatet

Les om situasjonen, beslutningene og resultatet i eksemplet. Vi kan drøfte hva som passer deres rammer, og hva som krever en annen tilnærming. Herdet applikasjonen med nonce-basert CSP — forsvar som holder når nettleseren er fiendtlig

Se tjenestene våre Ta kontakt →

Er applikasjonen din faktisk sikker?

Snakk om prosjektet med ingeniøren som har ansvaret for leveransen. Vi avtaler omfang, prisgrunnlag og overlevering skriftlig før arbeidet begynner.

Rolle i oppdraget: Sikkerhetsingeniør. Det bygger på Plattformarkitektur, Backend-utvikling, Frontend-utvikling, API-er og integrasjon, Databaser og Sikkerhet. Og nå som dere er her: vi er ganske gode på Backend- og API-utvikling og Sikkerhet og tilgangsstyring.

Ta kontakt Se tjenestene våre

Ett stykke av **en lengre oversikt** — og den ligger i sin helhet her.

Hver oppføring er skrevet på samme måte: situasjonen, oppgaven, hva vi gjorde, og hva som endret seg.

Se hele oversikten

Definer forbedringen dere trenger

Beskriv arbeidsflyten, eksisterende grensesnitt og brukerne som trenger endringen. Vi kan definere en testbar leveranse og en overlevering som gjør teamet i stand til å vedlikeholde løsningen.

Se hva det koster Ta kontakt →

Er applikasjonen din faktisk sikker?

Spør om beslutningene bak «Herdet applikasjonen med nonce-basert CSP». Dere snakker med folk som kan forklare arbeidet og vurdere deres eget prosjekt.

Se tjenestene våre

welcome@engineer.company +45 50 10 61 84 København, Danmark

Ta med deg denne siden

Visittkort

Laget med kjærlighet av D.

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

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://software.engineer.company/nb/portfolio/hardened-the-application-with-nonce-based-csp/