Engineer

Härdade applikationen med nonce-baserad CSP

Härdade applikationen med nonce-baserad CSP, HSTS, SameSite-cookies, databasroller med minsta behörighet och förnyade kontroller av åtkomsträttigheter på serversidan.

2025

försvar som håller när webbläsaren är fientlig

Situation. Plattformen lagrar yrkes- och organisationsdata, den sortens data som folk förväntar sig ska hanteras ordentligt, så en enda försvarslinje skulle aldrig räcka. Arbetsantagandet måste vara att klienten är fientlig — att allt som webbläsaren upprätthåller kan kopplas bort av den som sitter vid webbläsaren — och säkerheten måste hålla ändå.

Uppgift. Plattformen behövde härdas i varje lager — frontend, API, databas — så att säkerheten upprätthölls av servern oberoende av vad gränssnittet råkade tillåta.

Handling. I frontenden sätter middlewaren i Next.js — `proxy.ts` — en Content-Security-Policy med en nonce per anrop och strict-dynamic, plus HSTS och SameSite-cookies, så att webbläsaren är låst när det gäller vad den kör och skickar. I API:et finns hastighetsbegränsning, CORS, storleksgränser för anrop, validering av indata innan något når databasen och loggning av säkerhetsrelevanta händelser. I databasen loggar API:et in som en roll med minsta möjliga behörighet som bara får anropa appens funktioner med EXECUTE, funktionerna körs som SECURITY DEFINER och allt är parametriserat. Och rättigheterna — nivå, roll, organisation, avgränsning per fartyg — kontrolleras på nytt på servern vid varje anrop, medan spärrarna i frontenden enbart behandlas som UX. Spärrarna avgör vad du ser; servern avgör vad du får göra.

Resultat. Säkerheten hänger inte på att gränssnittet beter sig. Skydden ligger i lager för att undvika beroende av en enda kontroll. Auktoriseringen på servern är utformad för att fungera oberoende av kontrollerna på klienten, så att gränssnittet inte avgör vem som får åtkomst.

Använd erfarenheten i ert projekt

Använd ”Härdade applikationen med nonce-baserad CSP” som utgångspunkt för en liknande utmaning i er organisation. Tjänsterna bakom arbetet finns tillgängliga för ert projekt. Den del användarna aldrig ser: där data lagras, vilka reglerna är och hur andra system får kommunicera med den.

Se våra tjänster Hör av er →

Se bakom resultatet

Läs om situationen, besluten och resultatet i exemplet. Vi kan diskutera vad som passar era förutsättningar och vad som kräver ett annat angreppssätt. Härdade applikationen med nonce-baserad CSP — försvar som håller när webbläsaren är fientlig

Se våra tjänster Hör av er →

Är din applikation verkligen säker?

Prata om ert projekt med ingenjören som ansvarar för leveransen. Vi avtalar omfattning, hur priset beräknas och överlämning skriftligt innan arbetet börjar.

Roll i uppdraget: Säkerhetsingenjör. Det bygger på Plattformsarkitektur, Backendutveckling, Frontendutveckling, API:er och integration, Databaser och Säkerhet. Och nu när ni är här: vi är ganska bra på Backend- och API-utveckling och Säkerhet och behörighetshantering.

Hör av er Se våra tjänster

En del av **en längre historik** — och den finns i sin helhet här.

Varje post är skriven på samma sätt: situationen, uppgiften, vad vi gjorde och vad som förändrades.

Se hela historiken

Definiera förbättringen ni behöver

Beskriv arbetsflödet, befintliga gränssnitt och användarna som behöver förändringen. Vi kan definiera en testbar leverans och en överlämning som gör att teamet kan underhålla lösningen.

Se vad det kostar Hör av er →

Är din applikation verkligen säker?

Fråga om besluten bakom ”Härdade applikationen med nonce-baserad CSP”. Ni talar med personer som kan förklara arbetet och bedöma ert eget projekt.

Se våra tjänster

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

Ta med dig den här sidan

Visitkort

Omsorgsfullt skapad med kärlek av D.

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

Upphovsrätt © 2025 – nu · Engineer ApS · Alla rättigheter förbehållna.

Utskrivet från https://engineer.company/sv/portfolio/hardened-the-application-with-nonce-based-csp/