Engineer

{ Sikkerhed & adgangsstyring }

At bestemme, hvem der har adgang til hvad

Konti, adgangskoder og rettigheder sat op, så folk har præcis den adgang, jobbet kræver, og fratrådte mister den samme dag.

Er I stadig ved at finde ud af, hvad I har brug for?

Fortæl os, hvad der går galt, eller hvad I prøver at bygge. Så siger vi, hvad det vil kræve, cirka hvad det vil koste, og om vi er de rigtige til det.

Fra 1.400 DKK i timen, ekskl. moms — og mindre pr. time ved en længere booking.

Kontakt os Se, hvad det koster

Udvalgte resultater, der demonstrerer denne ydelse.

Gør ydelsen til et resultat

Lad »Sikkerhed & adgangsstyring« understøtte et forretningsmål. Vi forbinder det tekniske arbejde med de brugere, den drift og de beslutninger, det skal støtte. Konti, adgangskoder og rettigheder sat op, så folk har præcis den adgang, jobbet kræver, og fratrådte mister den samme dag.

Kontakt os

Undersøg det relevante arbejde

Brug arbejdet på siden til at undersøge vores tilgang. Spørg til afvejningerne og forudsætningerne bag hvert resultat. Har skrevet 1.000 RBAC-regler til Linux-servere — 1,000 adgangsregler skrevet for at lukke et sikkerhedshul

Har skrevet 1.000 RBAC-regler til Linux-servere Kontakt os →

Har skrevet 1.000 RBAC-regler til Linux-servere

Har forbedret datasikkerheden ved at implementere 1.000 RBAC-regler for udviklere, applikationsinstanser, PostgreSQL, MS SQL og andre Linux-servere for at forhindre uautoriseret adgang; dokumenteret med Ansible-automatisering.

Har ledet adgangsstyring med M365 og SSO

Har forhindret sikkerhedsbrud ved at lede initiativer inden for adgangsstyring med M365, 1Password, Red Hat SSO og OKTA SSO.

Har administreret og fejlsøgt 8 VPN-forbindelser

Har administreret og fejlsøgt 8 WireGuard VPN- og IPsec VPN-forbindelser og sikret beskyttet kommunikation mellem Google Cloud og Linux-systemer.

Har forbedret teamets kommunikation og samarbejde

Har forbedret teamets kommunikation og samarbejde ved at implementere Slack, Mattermost, 1Password og Jira og sparet 8.000 arbejdstimer.

Har administreret netværksinfrastruktur

Har administreret netværksinfrastruktur for over 1.000 servere og sikret optimal systemudrulning, sikkerhed og fejlfinding.

Har automatiseret SSL/TLS til 100 Docker-apps

Har automatiseret oprettelsen af SSL/TLS-certifikater til 100 Docker-applikationer og dermed sikret forbindelserne på alle Ubuntu Linux-værter.

Har bygget skift af organisationskontekst

Har designet et system til skift af organisationskontekst med localStorage i klienten og en spejlet cookie på serveren, så brugere kan optræde som de organisationer, de administrerer, mens autorisationen følger mindste privilegium.

Har bygget e-mail med failover og leveringslog

Har bygget e-mail som en funktion i selve platformen — tre udbydere med failover, webhooks for levering, logning af afsendelse og levering, skabeloner og kampagner — sikret af et opstartstjek, så platformen ikke starter uden mindst én udbyder.

Har provisioneret Azure-infrastruktur som kode

Har provisioneret Azure-infrastruktur som kode med Bicep — Container Apps, PostgreSQL Flexible Server, Front Door/WAF og netværk — i udviklings-, staging- og produktionsmiljøet.

Har hærdet applikationen med nonce-baseret CSP

Har hærdet applikationen med nonce-baseret CSP, HSTS, SameSite-cookies, least-privilege-databaseroller og fornyet kontrol af rettigheder på serveren.

Har automatiseret teamsamarbejde

Har automatiseret teamsamarbejde, adgangskodehåndtering, opgave- og tidsstyring og bygget et halvautomatisk system til fremvisning af projekter og dermed øget teamets produktivitet.

Har integreret en fælles adgangskodeløsning

Har integreret et virksomhedsdækkende adgangskodehåndteringssystem, der styrkede sikkerheden og strømlinede adgangskontrollen.

Har bygget adgangsret via Stripe og in-app-køb

Har bygget betalings- og rettighedslaget — Stripe side om side med app-køb via Apple og Google — så adgangen til kataloget, søgningen og eksporten styres af en adgangsmodel på 11 tabeller, der kontrolleres på serveren.

Har bygget fail-closed rate limiting mod misbrug

Har bygget fail-closed misbrugskontroller — 22 Redis-baserede rate limiters, Cloudflare Turnstile, idempotens på requests og en origin-lås — så platformen afviser bots og flood-angreb i stedet for at stole på dem, der kalder den.

Har bygget ejerskabskrav og deres afgørelse

Har bygget ejerskabskrav på virksomheder fra ende til anden — en bruger gør krav på en virksomhed, en administrator afgør sagen, og en godkendelse omskriver den autorisationsgraf, der afgør, hvem der må redigere hvad.

Har kørt et løbende sikkerhedsscanningsprogram

Har etableret et løbende sikkerhedsprogram — code scanning, DAST, afhængigheds- og sårbarhedstjek, SBOM-generering, secret scanning og SHA-pinnede actions — sammen med 21 skriftlige sikkerhedsgennemgange.

Tre låse på døren, ingen af dem låst

Har fundet og rettet tre SSH-beskyttelser mod brute force, der aldrig havde virket: en jail, der holdt øje med port 22, mens dæmonen lyttede på 1986, en ratebegrænsning, som en bredere regel over den overskyggede, og en blokeringshandling, hvis binærfil aldrig blev fundet, så ingen blokering nogensinde var trådt i kraft.

Har hærdet SSH til 24 validerede direktiver

Har hærdet SSH til 24 direktiver, der alle kontrolleres med assertions, med validering i tre trin — kandidatfilen, den samlede konfiguration og til sidst dæmonens egen tilbagelæsning — efter at tilbagelæsningen afslørede, at den kørende server ubemærket tilsidesatte to af de fireogtyve.

Har bevist, at blokeringen virker ved hver kørsel

Har eftervist blokeringsvejen fra ende til anden ved hver hærdningskørsel ved at blokere en reserveret testadresse, læse den resulterende kerneregel og ophæve blokeringen i en oprydningsblok, der altid kører, så en jail, der holder op med at virke, får kørslen til at fejle i stedet for at melde alt i orden.

Har verificeret firewallregler efter position

Har verificeret firewallregler efter position snarere end tilstedeværelse ved at læse den nummererede regelliste og den kørende pakkefilterkæde, fordi det, at en regel findes, ikke betyder, at nogen pakke når frem til den.

Har tilføjet et preflight-tjek af hemmeligheder

Har tilføjet et preflight-play, der kører den samme kode som konvergensen mod operatørens lokale hemmeligheder på omkring et sekund, efter at en produktionskørsel var gået ned halvvejs, ved sin niende opgave, med swap-indstillingerne allerede skrevet til den kørende vært.

Har afstemt en DNS-zone med 20 poster deklarativt

Har afstemt en DNS-zone med 20 poster deklarativt mod Cloudflares API med separate indgange til gennemgang og BIND-eksport og slået CDN-proxyen fra igen af hensyn til privatlivet, efter at den var bygget.

Har strammet systemd-sandboxen på hver unit

Har reduceret eksponeringen i systemd-sandkassen for hver unit, platformen installerer — en dead man's switch-tjeneste fra 9,6 UNSAFE til 1,5, en git-forge med adgang fra internettet fra 8,3 EXPOSED til 1,5 — og tilføjet et parsertjek ved konvergens efter at have fundet et fejlstavet direktiv, der ubemærket blev ignoreret i tre unit-skabeloner.

Har kørt containere på Podman og Quadlet

Har udrullet et containerlag på Podman og Quadlet under systemd i stedet for Docker, fordi Dockers standardpublicering af porte på bridge-netværk kan omgå UFW-regler — og givet udrulninger en uprivilegeret bruger med én fast kommando i stedet for root.

Har provisioneret en ekstra server ud fra kode

Har automatiseret provisioneringen af en ekstra server hos en anden cloududbyder og oprettet firewallen før maskinen, så den står bag en firewall fra første øjeblik, med begge udbyderes firewalls skrevet direkte mod deres REST-API'er for at undgå en collection fra tredjepart.

Har skrevet en afgrænsningsregel efter et læk

Har skrevet en afgrænsningsregel ind i repositoryet, efter at en omstrukturering havde ført en anden virksomheds inventar, firewalltilladelser og tekst med ind i det — og holdt de karantænesatte rester under hemmelighedsscanneren i stedet for at udelukke dem.

Har adskilt alle driftshemmeligheder pr. vært

Har gjort de fire driftshemmeligheder individuelle pr. vært efter at have fastslået, at to værter, der deler én dead man's switch, giver færre alarmer end to switches, ikke flere, og at en fælles adgangssætning til backup kan blotlægge begge værters kopier, selv i separate lagre, hvis nogen får fat i den og kan læse begge lagres krypterede data og nøglefiler.

Har taget den ubrugte halvdel af kernen i brug

Har taget den halvdel af beskedkernen i brug, som applikationen aldrig havde brugt — backupoverførsel, forsvindende beskeder, redigering og gensendelse af beskeder, verificerede invitationer, proxyer og krypteringspolitik — og kørt hver test mod det rigtige bibliotek uden mocks.

Kend licensen på alt, hvad I sender ud

Har kontrolleret 644 Rust-crates for licenskompatibilitet ved hvert build og bevist, at tjekket slår til, ved at ændre licensen på én crate og ved at flytte den fastlåste kernerevision uden at regenerere.

Har målt 13 adresser uden tracking

Har målt tretten adresser alene ud fra webserverens egen adgangslog — ingen cookie, ingen tracker og ingen tredjepartstjeneste — hvor klientens netværk maskeres i samme øjeblik linjen skrives, og resultatet samles i et permanent arkiv med 21 daglige tællinger og 20 månedlige facetter.

Har udgivet et læsbart Tor-onion-spejl

Har udgivet websitet som et Tor-onion-spejl på en læsbar adresse, der begynder med engineer, med nøglen genereret uden for værten, så den aldrig nåede dette repository eller en laptop, og ladet den være den ene af fire hoveddøre, der aldrig tælles.

Leder I efter Sikkerhed & adgangsstyring?

Brug de tilknyttede opgaver til at vurdere relevant erfaring. Fortæl os, hvad I skal bygge; vi kan tale om omfang, fremgangsmåde og overdragelse.

Kontakt os Hele porteføljen

Afgræns opgaven før bestilling

Medbring udrulningsforløbet, driftsrisiciene og dem, der har vagten. Vi kan afgrænse en ændring med kontroller, en vej tilbage og tydeligt ansvar.

Se, hvad det koster Kontakt os →

Gennemgå jeres krav med os

Fortæl os, hvad I har brug for inden for »Sikkerhed & adgangsstyring«, herunder fristen og det team, der skal vedligeholde løsningen. Vi kan drøfte en praktisk vej frem.

Kontakt os

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

Tag denne side med dig

Visitkort

Lavet med kærlighed af D.

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

Copyright © 2025 – nu · Engineer ApS · Alle rettigheder forbeholdes.

Udskrevet fra https://engineer.company/da/services/security-access/