Engineer

{ Säkerhet och behörighetshantering }

Bestämmer vem som når vad

Konton, lösenord och behörigheter ordnade så att alla har exakt den åtkomst deras arbete kräver, och de som slutar förlorar den samma dag.

Håller ni fortfarande på att reda ut vad ni behöver?

Berätta vad som inte fungerar eller vad ni försöker bygga. Vi säger vad det skulle kräva, ungefär vad det skulle kosta och om vi är rätt personer för uppgiften.

Från 1 400 DKK i timmen, exklusive moms — och mindre per timme vid en längre bokning.

Hör av er Se vad det kostar

Utvalt arbete som visar den här tjänsten.

Gör tjänsten till ett resultat

Låt ”Säkerhet och behörighetshantering” stödja ett affärsmål. Vi kan knyta det tekniska arbetet till användarna, driften och besluten det ska stödja. Konton, lösenord och behörigheter ordnade så att alla har exakt den åtkomst deras arbete kräver, och de som slutar förlorar den samma dag.

Hör av er

Granska det relevanta arbetet

Använd arbetet på sidan för att granska vårt angreppssätt. Fråga om avvägningarna och förutsättningarna bakom varje resultat. Skrev 1 000 RBAC-regler som säkrar Linux-servrar — 1,000 åtkomstregler, skrivna för att täppa till en säkerhetslucka

Skrev 1 000 RBAC-regler som säkrar Linux-servrar Hör av er →

Skrev 1 000 RBAC-regler som säkrar Linux-servrar

Stärkte datasäkerheten genom att införa 1 000 RBAC-regler för utvecklare, applikationsinstanser, PostgreSQL, MS SQL och andra Linux-servrar, vilket förhindrade obehörig åtkomst; dokumenterade reglerna som automatisering i Ansible.

Ledde behörighetshanteringen i M365 och SSO

Förhindrade säkerhetsintrång genom att leda initiativ för behörighetshantering med M365, 1Password, Red Hat SSO och OKTA SSO.

Förvaltade och felsökte 8 VPN-anslutningar

Förvaltade och felsökte 8 VPN-anslutningar över WireGuard och IPsec och säkerställde säker kommunikation i Google Cloud och på Linux-system.

Förbättrade teamets kommunikation och samarbete

Förbättrade teamets kommunikation och samarbete genom att införa Slack, Mattermost, 1Password och Jira, vilket sparade 8 000 arbetstimmar.

Administrerade nätverksinfrastrukturen

Administrerade nätverksinfrastrukturen för över 1 000 servrar och säkerställde optimal driftsättning, säkerhet och felsökning av systemen.

Automatiserade SSL/TLS-cert för 100 Docker-appar

Automatiserade skapandet av SSL/TLS-certifikat för 100 Docker-applikationer och säkerställde säkra anslutningar på alla värdar med Ubuntu Linux.

Byggde växling av organisationskontext

Utformade ett system för att växla organisationskontext med localStorage hos klienten och en speglad cookie på serversidan, så att användare kan agera som de organisationer de förvaltar medan auktorisering enligt minsta behörighet upprätthålls.

Byggde e-post med failover och leveransloggning

Byggde e-post som en förmåga i plattformen — tre leverantörer med failover, webhooks för leveranser, loggning av utskick och leveranser, mallar och kampanjer — bakom en startkontroll som vägrar starta utan en leverantör.

Provisionerade Azure-infrastruktur som kod

Provisionerade Azure-infrastruktur som kod med Bicep — Container Apps, PostgreSQL Flexible Server, Front Door/WAF och nätverk — i utvecklings-, staging- och produktionsmiljöerna.

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.

Automatiserade teamets samarbete

Automatiserade teamets samarbete, lösenordshantering samt uppgifts- och tidshantering och byggde ett halvautomatiskt system för att visa upp projekt, vilket höjde teamets produktivitet.

Införde en företagsgemensam lösenordshanterare

Integrerade ett företagsövergripande system för lösenordshantering, vilket stärkte säkerheten och effektiviserade åtkomstkontrollen.

Byggde åtkomsträttigheter via Stripe och appköp

Byggde skiktet för betalningar och åtkomsträttigheter — Stripe vid sidan av köp i appen via Apple och Google — som spärrar katalogen, sökningen och exporten bakom en åtkomstmodell i 11 tabeller som kontrolleras på servern.

Byggde hastighetsbegränsning som stänger vid fel

Byggde skydd mot missbruk som stänger vid fel — 22 hastighetsbegränsare i Redis, Cloudflare Turnstile, idempotenta anrop och ett ursprungslås — så att plattformen avvisar botar och anropsfloder i stället för att lita på sina anropare.

Byggde anspråk på företagsägarskap och prövning

Byggde anspråk på företagsägarskap från början till slut — en användare gör anspråk på ett företag, en administratör avgör, och ett godkännande skriver om auktoriseringsgrafen som bestämmer vem som får redigera vad.

Drev ett löpande program för säkerhetsskanning

Etablerade ett kontinuerligt säkerhetsprogram — kodskanning, DAST, kontroller av beroenden och sårbarheter, generering av SBOM, skanning efter hemligheter och SHA-låsta actions — vid sidan av 21 skriftliga säkerhetsgranskningar.

Tre lås på dörren, inget av dem låst

Hittade och åtgärdade tre skydd mot brute force-attacker över SSH som aldrig hade fungerat: en jail som bevakade port 22 medan daemonen lyssnade på 1986, en hastighetsbegränsning som skuggades av en bredare regel ovanför den och en spärråtgärd vars binärfil aldrig gick att hitta, så att ingen spärr någonsin hade tillämpats.

Härdade SSH till 24 validerade direktiv

Härdade SSH till 24 kontrollerade direktiv med validering i tre steg — kandidatfilen, den sammansatta konfigurationen och sedan daemonens egen återläsning — efter att återläsningen hade avslöjat att den körande servern tyst åsidosatte två av de tjugofyra.

Bevisade att intrångsspärren fungerar varje gång

Bevisade spärrkedjan mot intrång från början till slut vid varje härdningskörning genom att spärra en reserverad testadress, läsa den kärnregel som uppstod och häva spärren i ett städblock som alltid körs, så att en jail som slutar fungera fäller körningen i stället för att rapportera att allt är i ordning.

Verifierade brandväggsregler efter position

Verifierade brandväggsregler efter position snarare än förekomst och läste den numrerade regellistan och den aktiva paketfilterkedjan, eftersom en regel som finns inte är detsamma som en regel som något paket når.

Kontrollerade hemligheter i förväg på en sekund

Lade till en play för förhandskontroll som kör samma kod som konvergeringen mot operatörens lokala hemligheter på ungefär en sekund, efter att en halvt tillämpad produktionskörning kraschade vid sin nionde uppgift med swapinställningarna redan skrivna till värden i drift.

Stämde av en DNS-zon med 20 poster som kod

Stämde av en DNS-zon med 20 poster deklarativt mot Cloudflare-API:t med separata ingångar för granskning och BIND-export, och stängde av CDN-proxyn igen av integritetsskäl efter att ha byggt den.

Minskade systemd-exponeringen för varje enhet

Minskade exponeringen i systemd-sandlådan för varje enhet som plattformen installerar — en tjänst för dödmansgrepp från 9,6 UNSAFE till 1,5, en internetexponerad git-forge från 8,3 EXPOSED till 1,5 — och lade till en parserkontroll vid konvergering efter att ha hittat ett felstavat direktiv som tyst ignorerades i tre enhetsmallar.

Körde containrar på Podman och Quadlet

Driftsatte en containermiljö på Podman och Quadlet under systemd i stället för Docker, eftersom Dockers standardpublicering av portar i bridge-nätverk kan kringgå UFW-regler — och gav driftsättningarna en oprivilegierad användare med ett enda fast kommando i stället för root.

Provisionerade en andra server från kod

Automatiserade provisioneringen av en andra server hos en andra molnleverantör och skapade brandväggen före maskinen så att den föds bakom en, med båda leverantörernas brandväggar skrivna direkt mot deras REST-API:er för att undvika en collection från tredje part.

Skrev en avgränsningsregel efter läckta uppgifter

Skrev in en avgränsningsregel i repot efter att en omstrukturering hade fört in ett annat företags värdförteckning, brandväggsundantag och text i det — och höll de karantänsatta resterna kvar under skanningen efter hemligheter i stället för att undanta dem.

Delade upp varje driftshemlighet per värd

Delade upp de fyra driftshemligheterna per värd efter att ha konstaterat att två värdar som delar ett dödmansgrepp larmar mindre än två separata grepp, inte mer, och att en delad lösenfras för säkerhetskopior kan ge tillgång till båda värdarnas kopior, även i separata repon, om någon får tag på den och kan läsa båda reponas krypterade data och nyckelfiler.

Tog i bruk den oanvända halvan av kärnan

Tog i bruk den hälft av meddelandekärnan som applikationen aldrig hade använt — överföring av säkerhetskopior, försvinnande meddelanden, redigering och omsändning av meddelanden, verifierade inbjudningar, proxyer och krypteringspolicy — och körde varje test mot det riktiga biblioteket utan mockar.

Känn till licensen för allt ni levererar

Granskade licenskompatibiliteten hos 644 Rust-crates vid varje bygge och bevisade att kontrollen slår till genom att skriva om licensen för en crate och genom att flytta den låsta kärnrevisionen utan att generera om.

Mätte 13 adresser helt utan spårning

Mätte tretton adresser enbart utifrån webbserverns egen åtkomstlogg — ingen cookie, ingen spårare och ingen tjänst från tredje part — maskerade klientens nätverk när varje rad skrivs och förde in resultatet i ett permanent arkiv med 21 dagliga räkningar och 20 månatliga fasetter.

Publicerade en läsbar onion-spegel i Tor

Publicerade webbplatsen som en onion-spegel i Tor på en läsbar adress som börjar med engineer, med nyckeln genererad utanför värden så att den aldrig nådde detta repo eller en bärbar dator, och lät den förbli den enda av fyra ytterdörrar som aldrig räknas.

Letar ni efter Säkerhet och behörighetshantering?

Använd de länkade arbetena för att bedöma relevant erfarenhet. Berätta vad ni behöver bygga; vi kan diskutera omfattning, arbetssätt och överlämning.

Hör av er Hela portföljen

Avgränsa uppdraget före beställning

Ta med driftsättningsvägen, driftsriskerna och dem som har jour. Vi kan avgränsa en förändring med kontroller, en väg tillbaka och tydligt ansvar.

Se vad det kostar Hör av er →

Gå igenom era krav med oss

Berätta vad ni behöver inom ”Säkerhet och behörighetshantering”, vilken tidsram ni önskar och vilket team som ska förvalta lösningen. Vi kan diskutera en praktisk väg framåt.

Hör av er

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/services/security-access/