Engineer Plattform

{ Sikkerhet og tilgangsstyring }

Bestemme hvem som får tilgang til hva

Kontoer, passord og tilganger ordnet slik at folk har nøyaktig den tilgangen jobben krever, og de som slutter, mister den samme dag.

Avklarer dere fortsatt hva dere trenger?

Fortell oss hva som går galt, eller hva dere prøver å bygge. Vi sier hva det vil kreve, omtrent hva det vil koste, og om vi er de rette til oppgaven.

Fra 1 400 DKK timen, eksklusive mva. — og mindre per time ved et lengre oppdrag.

Ta kontakt Se hva det koster

Utvalgt arbeid som viser denne tjenesten.

Gjør tjenesten til et resultat

La «Sikkerhet og tilgangsstyring» støtte et forretningsmål. Vi kan knytte det tekniske arbeidet til brukerne, driften og beslutningene det skal støtte. Kontoer, passord og tilganger ordnet slik at folk har nøyaktig den tilgangen jobben krever, og de som slutter, mister den samme dag.

Ta kontakt

Undersøk det relevante arbeidet

Bruk arbeidet på denne siden til å undersøke tilnærmingen vår. Spør om avveiningene og forutsetningene bak hvert resultat. Skrev 1 000 RBAC-regler for å sikre Linux-servere — 1,000 tilgangsregler skrevet for å tette et sikkerhetshull

Skrev 1 000 RBAC-regler for å sikre Linux-servere Ta kontakt →

Skrev 1 000 RBAC-regler for å sikre Linux-servere

Styrket datasikkerheten ved å innføre 1 000 RBAC-regler for utviklere, applikasjonsinstanser, PostgreSQL, MS SQL og andre Linux-servere, og forhindret uautorisert tilgang; dokumentert med Ansible-automatisering.

Ledet tilgangsstyring på tvers av M365 og SSO

Forhindret sikkerhetsbrudd ved å lede tiltak for tilgangsstyring med M365, 1Password, Red Hat SSO og OKTA SSO.

Administrerte og feilsøkte 8 VPN-forbindelser

Administrerte og feilsøkte 8 VPN-forbindelser med WireGuard og IPsec, og sikret trygg kommunikasjon på tvers av Google Cloud og Linux-systemer.

Forbedret kommunikasjonen og samarbeidet i teamet

Forbedret kommunikasjonen og samarbeidet i teamet ved å innføre Slack, Mattermost, 1Password og Jira, og sparte 8 000 arbeidstimer.

Administrerte nettverksinfrastrukturen

Administrerte nettverksinfrastrukturen for over 1 000 servere og sikret optimal produksjonssetting, sikkerhet og feilsøking av systemene.

Automatiserte SSL/TLS for 100 Docker-apper

Automatiserte opprettelsen av SSL/TLS-sertifikater for 100 Docker-applikasjoner og sikret trygge forbindelser på tvers av Ubuntu Linux-verter.

Bygde bytte av organisasjonskontekst for brukerne

Designet et system for å bytte organisasjonskontekst med localStorage på klienten og speiling i informasjonskapsler på serversiden, slik at brukere kan opptre som organisasjoner de administrerer, med autorisasjon etter prinsippet om minste privilegium.

Bygde e-post med failover og leveringslogging

Bygde e-post som en plattformfunksjon — tre leverandører med failover, leveringswebhooks, logging av sending og levering, maler og kampanjer — bak en oppstartskontroll som ikke lar plattformen starte uten minst én leverandør.

Klargjorde Azure-infrastruktur som kode med Bicep

Klargjorde Azure-infrastruktur som kode med Bicep — Container Apps, PostgreSQL Flexible Server, Front Door/WAF og nettverk — i utviklings-, staging- og produksjonsmiljøene.

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.

Automatiserte samarbeid i teamet

Automatiserte samarbeid i teamet, passordhåndtering, oppgave- og tidsstyring, og bygde et halvautomatisk system for å vise frem prosjekter, noe som økte teamets produktivitet.

Integrerte en passordbehandler i hele selskapet

Integrerte et passordhåndteringssystem for hele selskapet, styrket sikkerheten og effektiviserte tilgangskontrollen.

Bygde tilgangsrettigheter for Stripe og appkjøp

Bygde laget for betaling og tilgangsrettigheter — Stripe sammen med kjøp i appen hos Apple og Google — som styrer tilgangen til katalogen, søket og eksporten gjennom en tilgangsmodell på 11 tabeller som kontrolleres på serveren.

Bygde fail-closed hastighetsgrenser mot misbruk

Bygde misbruksvern som stenger ved feil — 22 Redis-baserte hastighetsbegrensere, Cloudflare Turnstile, idempotente forespørsler og en origin-lås — slik at plattformen avviser boter og flom i stedet for å stole på dem som kaller den.

Bygde eierskapskrav og avgjørelse for selskaper

Bygde løsningen for eierskapskrav til selskaper fra ende til annen — en bruker gjør krav på et selskap, en administrator avgjør saken, og en godkjenning skriver om autorisasjonsgrafen som bestemmer hvem som får redigere hva.

Drev et løpende program for sikkerhetsskanning

Etablerte et kontinuerlig sikkerhetsprogram — kodeskanning, DAST, kontroll av avhengigheter og sårbarheter, SBOM-generering, skanning etter hemmeligheter og SHA-låste actions — sammen med 21 skriftlige sikkerhetsrevisjoner.

Tre låser på døren, ingen av dem låst

Fant og rettet tre SSH-vern mot brute force som aldri hadde virket: en jail for utestenging som overvåket port 22 mens daemonen lyttet på 1986, en hastighetsgrense skygget av en bredere regel over den, og en utestengingshandling der binærfilen aldri ble funnet, slik at ingen utestenging noen gang hadde trådt i kraft.

Herdet SSH til 24 validerte direktiver

Herdet SSH til 24 kontrollerte direktiver med validering i tre trinn — kandidatfilen, den sammensatte konfigurasjonen og så daemonens egen tilbakelesing — etter at tilbakelesingen avslørte at den kjørende serveren i det stille overstyrte to av de tjuefire.

Beviste at utestengingen virker ved hver kjøring

Beviste at utestengingen virker fra ende til annen ved hver herdingskjøring ved å utestenge en reservert testadresse, lese den resulterende kjerneregelen og oppheve utestengingen i en garantert oppryddingsblokk, slik at en jail som slutter å virke, får kjøringen til å feile i stedet for å melde at alt er i orden.

Verifiserte brannmurregler etter plassering

Verifiserte brannmurregler etter plassering i stedet for tilstedeværelse ved å lese den nummererte regellisten og den aktive pakkefilterkjeden, fordi en regel som finnes, ikke er det samme som en regel en pakke faktisk når.

La til ett sekunds preflight for hemmeligheter

La til en preflight-play som kjører den samme koden som konvergeringen mot operatørens lokale hemmeligheter på omtrent ett sekund, etter at en halvveis gjennomført produksjonskjøring stoppet på den niende oppgaven med swap-innstillingene allerede skrevet til verten i drift.

Avstemte en DNS-sone med 20 poster som kode

Avstemte en DNS-sone med 20 poster deklarativt mot Cloudflare-API-et med egne inngangspunkter for revisjon og BIND-eksport, og slo CDN-proxyen av igjen av personvernhensyn etter å ha bygd den.

Strammet inn systemd-sandkassen for hver enhet

Reduserte eksponeringen i systemd-sandkassen for hver enhet plattformen installerer — tjenesten for dødmannsknappen fra 9,6 UNSAFE til 1,5, en internetteksponert git-forge fra 8,3 EXPOSED til 1,5 — og la til en parserkontroll ved konvergering etter å ha funnet et feilstavet direktiv som ble ignorert i det stille i tre enhetsmaler.

Kjørte containere på Podman og Quadlet

Satte et containerlag på Podman og Quadlet under systemd i produksjon i stedet for Docker, fordi Dockers standardpublisering av porter på bridge-nettverk kan omgå UFW-regler — og ga produksjonssettinger en uprivilegert bruker med én fast kommando i stedet for root.

Klargjorde en server nummer to fra kode

Automatiserte klargjøringen av en server nummer to hos en annen skyleverandør, der brannmuren opprettes før maskinen slik at den blir født bak en, og begge leverandørenes brannmurer er skrevet direkte mot deres REST-API-er for å slippe en collection fra tredjepart.

Skrev en avgrensningsregel etter tilgangslekkasje

Skrev en avgrensningsregel inn i repoet etter at en omstrukturering hadde ført et annet selskaps inventar, brannmuråpninger og tekst inn i det — og holdt de karantenesatte restene under skanneren for hemmeligheter i stedet for å unnta dem.

Delte opp hver driftshemmelighet per vert

Delte opp de fire driftshemmelighetene per vert etter å ha slått fast at to verter som deler én dødmannsknapp, varsler mindre enn to knapper, ikke mer, og at en felles passordfrase for sikkerhetskopier kan gi tilgang til begge vertenes kopier, selv i separate repoer, hvis noen får tak i den og kan lese de krypterte dataene og nøkkelfilene i begge repoene.

Nådde den ubrukte halvdelen av kjernen

Tok i bruk den halvdelen av meldingskjernen applikasjonen aldri hadde brukt — overføring av sikkerhetskopier, forsvinnende meldinger, redigering og ny sending av meldinger, verifiserte invitasjoner, proxyer og krypteringspolicy — og kjørte hver test mot det ekte biblioteket uten mocks.

Kjenn lisensen til alt dere leverer

Reviderte 644 Rust-crates for lisenskompatibilitet ved hvert bygg, og beviste at kontrollen slår ut ved å skrive om lisensen til én crate og ved å flytte den låste kjernerevisjonen uten å regenerere.

Målte 13 adresser uten sporing

Målte tretten adresser ut fra webserverens egen tilgangslogg alene — ingen informasjonskapsel, ikke noe sporingsverktøy og ingen tredjepartstjeneste — maskerte klientnettverket idet hver linje skrives, og samlet resultatet i et permanent arkiv med 21 daglige tellinger og 20 månedlige fasetter.

Publiserte et lesbart onion-speil på Tor

Publiserte nettstedet som et Tor onion-speil på en lesbar adresse som begynner med engineer, med nøkkelen generert utenfor verten slik at den aldri havnet i dette repoet eller på en bærbar PC, og lot det være den ene av fire inngangsdører som aldri telles.

Leter dere etter Sikkerhet og tilgangsstyring?

Bruk de tilknyttede arbeidene til å vurdere relevant erfaring. Fortell oss hva dere skal bygge; vi kan diskutere omfang, fremgangsmåte og overlevering.

Ta kontakt Hele porteføljen

Avgrens oppgaven før bestilling

Ta med utrullingsløpet, driftsrisikoene og dem som har vakt. Vi kan avgrense en endring med kontroller, en vei tilbake og tydelig ansvar.

Se hva det koster Ta kontakt →

Gå gjennom kravene med oss

Fortell hva dere trenger innen «Sikkerhet og tilgangsstyring», inkludert fristen og teamet som skal vedlikeholde løsningen. Vi kan drøfte en praktisk vei videre.

Ta kontakt

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://platform.engineer.company/nb/services/security-access/