OS2rollekatalog

Sikring af driftsmiljøet

Version: 1.1.0
Date: 07.05.2018
Author: BSG

Indhold

1 Indledning

Dette dokument beskriver de sikkerhedsmæssige aspekter af OS2rollekatalog, såvel de applikationsmæssige som driftsmæssige, der alle indgår i den samlede driftssikkerhed.

2 Applikationssikkerhed

Applikationssikkerhed dækker over sikkerhedsmæssig funktionalitet der er en del af selve løsningen, uafhængig af driftsmiljøet.

2.1 Sikring af data

Sikring af data dækker over to forskellige områder, sikring mod uautoriseret ændring af data, og sikring mod uautoriseret adgang til data.

OS2rollekatalog indeholder ikke stærkt følsomme data, men opbevarer dog data som har karakter af persondata. Dette er medarbejdernes navne. OS2rollekatalog skal sikre uautoriseret adgang til disse data.

OS2rollekatalog opbevarer oplysninger om adgange til andre systemer, hvilket vurderes at være oplysninger, hvor sikring mod uautoriseret ændring af data er kritisk.

2.1.1 Sikring mod uautoriseret adgang til data

OS2rollekatalogs snitflader (brugergrænseflade såvel som API) udstilles via en TLS sikret forbindelse, hvilket beskytter mod at 3.part kan lytte med, og dermed få uautoriseret adgang til data via disse kanaler.

Bemærk at applikationsdesignet blot sikrer at TLS er krævet under drift, men i en normal driftssituation vil TLS forbindelsen offloades til en loadbalancer, så sikring af korrekt valg af understøttede krypterings- og signeringsalgoritmer ligger som en arbejdsopgave i konfigurationen af ens Load Balancer.

OS2rollekatalogs snitflader (brugergrænseflade såvel som API) kræver autentifikation af brugeren, inden denne kan tilgå snitfladerne, hvilket er beskrevet i afsnit 2.2 og 2.3.

2.1.2 Sikring mod uautoriseret ændring af data

OS2rollekatalogs applikationsarkitektur er lagdelt, og består af 4 lag

- **Brugergrænseflade**. Dette lag udstiller den web-baserede brugergrænseflade, og kommunikerer alene med det underlæggende Controller lag. Brugergrænsefladen anvender Spring Security[1] til at beskytte imod angreb som Cross Site Request Forgery, og andre web-baserede angreb.
- **Controller**. Dette lag udstiller alle tilgængelige operationer mod omverdenen, og er indkapslet i et autorisations- og autentifikationssikkerhedslag baseret på Spring Security. Kald der ikke er autentificeret og autoriseret afvises inden de når ned til dette lag. Controller laget kommunikerer alene med det underlæggende Service lag.
- **Service**. Dette lag står for forretningslogikken i OS2rollekatalog, og håndhæver alle forretningsregler og data-integritetsregler, inden der kommunikeres videre til Data laget.

[1] https://projects.spring.io/spring-security/

- **Data**. Dette lag er et abstraktionslag ovenpå den underlæggende database, og sikrer at kun foruddefinerede SQL statements kan afvikles, og beskytter bl.a. mod SQL Injection angreb. Data laget er det eneste lag der kan tilgå SQL databasen.

Applikationsarkitekturen er designet efter security-by-design principper, og hvert lag er ansvarlig for at beskytte imod angreb rettet mod dette lag, med passende overlap i beskyttelsesmekanismerne, hvor princippet om 'Defence in depth'[2] finder anvendelse.

2.2 Adgangskontrol, Brugergrænseflade

Adgang til brugergrænsefladen gives via fødereret brugerstyring, og der anvendes SAML 2.0 som teknologi til denne sikring.

Der er udarbejdet retningslinjer for opsætning af adgang til OS2rollekatalog, hvor det anbefales at man styrer adgangen i ens lokale Identity Provider (typisk AD FS), hvor kun udvalgte medarbejdere for adgang til OS2rollekatalog, og at denne adgang kun gives på det interne netværk, så adgang til OS2rollekatalog via internettet afvises.

Samtidig er OS2rollekatalog konfigureret til at kræve aktivt login (SAML forceAuth) ved adgang til OS2rollekatalog, så adgang til en medarbejders PC ikke automatisk giver adgang til OS2rollekatalog.

Hele brugergrænseflade er beskyttet af den funktionalitet som Spring Security tilbyder, herunder beskyttelse mod klassiske angreb som Cross Site Request Forgery og lignende.

Spring Security står også for at blænde dele af brugergrænsefladen af, hvis man ikke har adgang til denne, om end dette gøres alene af brugervenlighedsmæssige årsager.

Controller laget står for den egentlige afvisning af uautoriserede kald.

2.3 Adgangskontrol, API

OS2ollekatalog udstiller en række API'er, som er beskyttet med HTTP Basic Auth[3], baseret på genererede API nøgler med 122 bits entropi (UUID v4).

Der anvendes samme sikringslag på bagsiden af API'et som der ligger på Brugergrænsefladen, da der er tale om samme applikationslag og kode som kaldene bevæger sig gennem.

2.4 Auditlogning

OS2rollekatalog logger alle sikkerhedshændelser, samt opdaterende operationer. Sikkerhedshændelser håndterer af Spring Security, og Data laget håndterer alle logninger på data-niveauet.

Ved at ligge denne logik i Data laget, sikres det at alle adgange logges, samt at dette sker på en ensartet måde.

Auditloggen gemmes i OS2rollekatalogs database som struktureret data i de 6 måneder der er tilladt af it-sikkerhedsbekendtgørelsens §19.

[2] https://en.wikipedia.org/wiki/Defense_in_depth_(computing)
[3] https://en.wikipedia.org/wiki/Basic_access_authentication

2.5 Docker

OS2rollekatalog er designet til deployment via Docker[4]. Selvom dette formelt set er en del af driftsmiljøet, er valget af Docker tænkt ind i applikationsdesignet, så der ikke er nogen restriktioner i anvendelsen af Docker som deployment strategi.

Dockers primære funktion er at sikre let, ensartet deployment af høj kvalitet, men en side-effekt af at bruge Docker, er en øget adskillelse mellem de forskellige komponenter i det samlede driftssetup.

Hver komponent i driftsmiljøet deployes i sin egen Docker container, og via konfiguration styres det præcist hvilke container der må kommunikere med hvem, og hvordan de må kommunikere (porte m.m.).

En sårbarhed i et backup script (som kører i sin egen Container), vil ikke påvirke den container der kører OS2rollekatalog, osv.

Det er her vigtigt at bemærke at Container teknologi ikke giver samme adskillelse mellem komponenter som en fuld virtualisering vil gøre, men fordi Containere er letvægts by-design, vil man naturligt pakke hver del-komponent ind i sin egen Container, modsat virtualiserede miljøer, hvor scripts, proxy'er, firewalls, load balancers, applikationer m.m. kører på samme virtuelle maskine, med mulighed for at påvirke hinandens sikkerhed utilsigtet.

2.6 Security Advisories

OS2rollekatalog baserer sig alene på Spring stakken, hvilket både sikrer at de enkelte frameworks spiller sammen, og er testet og kvalitetssikret af Pivotal, der står bag den største og mest udbredte applikations-stak til Java (Spring).

Pivotal udstiller security advisories[5] både for egne frameworks i Spring stakken, men også for 3.parts frameworks, så der er et samlet overblik over kendte sikkerheds-issues, og leverandøren af OS2ollekatalog følger aktivt med i de advisories der udsendes, og vurderer relevansen for OS2rollekatalog.

3 Driftsmiljøet

Dette afsnit beskriver det driftsmiljø som driftsleverandøren anvender til drift af OS2rollekatalog. Hvis OS2rollekatalog driftes lokalt, skal man lave lignende overvejelser om ens driftsmiljø I forhold til det sikkerhedsniveau man ønsker.

3.1 Den overordnede driftsstrategi

Driftsleverandøren anvender AWS (Amazon Web Services) som driftsmiljø, og har indgået en underdatabehandleraftale med AWS. AWS følger det kommende GDPR direktiv nøje, og har udarbejdet en databehandleraftale som understøtter direktivet[6].

Hvor muligt anvendes altid en managed, compliant, og SLA-belagt service frem for egen-installerede komponenter, dette dækker bl.a. overvågning, load balancer, firewall m.m.

AWS har et stort sikkerhedsteam, hvis primære funktion er at sikre disse managed services, både mht patch-management, overvågning og aktiv respons i tilfælde af angreb. AWS

[4] https://www.docker.com/
[5] https://pivotal.io/security
[6] https://aws.amazon.com/compliance/eu-data-protection/

foretager løbende sikkerhedsreviews og penetrationstests m.m. af deres managed services, og vil kunne sikre et højere sikkerhedsniveau end tilsvarende komponenter, som kan installeres af driftsleverandøren.

AWS platformen stiller både funktionelle services til rådighed (applikationsservere, databaseservere, load balancers, m.m.) samt egentlige sikkerhedskomponenter (DDOS skjold, auditlogning, logovervågning, m.m.), som anvendes aktivt af driftsleverandøren til at sikre både et tilgængeligt og sikkert driftsmiljø.

3.2 Compliance

Driftsleverandøren anvender alene cloud-komponenter i AWS der er både ISO 27001, ISO 27017 og ISO 27018 certificeret[7].

Listen af anvendte AWS komponenter vurderes løbende, men er på skrivende tidspunkt

- AWS ELB (https://aws.amazon.com/elasticloadbalancing/)
- AWS EC2 (https://aws.amazon.com/ec2/)
- AWS ECS (https://aws.amazon.com/ecs/)
- AWS RDS (https://aws.amazon.com/rds/)
- AWS CloudWatch (https://aws.amazon.com/cloudwatch/)
- AWS S3 (https://aws.amazon.com/s3/)

3.3 Sikkerhedskomponenter

Endvidere anvendes en række sikkerhedskomponenter i AWS, herunder AWS Security Groups og AWS Shield som beskrevet nedenfor.

De sikkerhedskomponenter fra AWS platformen der anvendes er på skrivende tidspunkt

- AWS Shield (https://aws.amazon.com/shield/)
- AWS Certificate Manager (https://aws.amazon.com/certificate-manager/)
- AWS Security Groups
- AWS IAM (https://aws.amazon.com/iam/)
- AWS CloudTrail (https://aws.amazon.com/cloudtrail/)
- AWS Inspector (https://aws.amazon.com/inspector/)

3.4 Adgang til driftsmiljøet

Leverandøren følger AWS retningslinje for sikker adskillelse af driftsmiljøer, hvilket bl.a. betyder at AWS flere-konto strategien[8] følges.

Adgangen til administration af OS2rollekatalogs driftsmiljø sikres via den funktionalitet som AWS stiller til rådighed, hvilket på nuværende tidspunkt betyder

- IP filtrering på hvorfra man kan tilgå miljøet
- 2-faktor login til den administrative konsol
- SSH-nøgle beskyttelse af alt adgang til servere

Som udgangspunkt er alle komponenter i AWS miljøet låst ned, så de kun kan tilgås fra internettet via AWS ELB load balanceren og firewall. Denne load balancer tillader kun trafik på

[7] https://aws.amazon.com/compliance/services-in-scope/
[8] https://aws.amazon.com/answers/account-management/aws-multi-account-security-strategy/

port 443 (HTTPS beskyttet), og dirigerer trafikken til netop de services som er konfigureret ELB og ikke andre.

Når der skal udføres egentlig server vedligehold, som ikke kan udføres via AWS management konsollen, åbnes for port 22 (SSH) for driftsleverandørens IP adresse, så serveren kan tilgås, og server management kan udføres. Adgangen lukkes efter afsluttet vedligeholde.

3.4.1 Logning af adkomster og handlinger i driftsmiljøet

AWS platformen stiller et drifts-audit komponent til rådighed ved navn CloudTrail, der anvendes til at logge alle administrative handlinger på driftsmiljøet.

CloudTrail opsamler alle handlinger, herunder SSH adgang til servere, ændringer i konfigurationer, sikkerhedsopsætninger m.m., og sender disse til en central auditlog, som kan inspiceres af AWS kontoholderen.

3.5 Vulnerability scanning

AWS Inspector anvendes til løbende at scanne servermiljøerne for kendte sårbarheder og dårlige konfigurationer. Amazon har udviklet en sårbarheds-scanner, der via agenter installeret på servermiljøet løbende scanner efter kendte sårbarheder og/eller dårlige konfigurationer, og alarmerer leverandøren i tilfælde af opdagede sårbarheder.

AWS Inspector er konfigureret til at afvikle en fuld sårbarhedsscanning ugentligt.

3.6 Docker + løbende redeployment af applikationsservere

Der anvendes standard Linux images fra Amazon til installation af OS2rollekatalog, og da hele opsætningen ligger i et Docker image, foretages der månedlig redeployment af OS2rollekatalog på friske server images (Amazon publicerer løbende opdaterede images, med nyeste sikkerhedspatches). Dette er en semi-automatiseret proces, der blot skal initieres af leverandøren.

Selve redeployment sker uden nedetid, da nye servere deployes først, meldes ind i load balanceren, og derefter tages de gamle servere ud af load balanceren.

3.7 Tilgængelighed / high-availability

Standard driftsmiljøet består af 2 applikationsservere og 2 databaseservere, hvor disse er placeret i geografisk adskilte driftscentre (1 applikationsserver og 1 databaseserver i hvert driftscenter).

AWS ELB sikrer distribution af trafik på tværs af de to driftscentre, baseret på kapacitet og tilgængelig af serverne.

Applikationsserverne er stateless, og et applikationsserver crash betyder blot at den dårlige server automatisk meldes ud af ELB, og at der startes en ny applikationsserver med Rollekataloget installeret.

Der er ikke slået auto-scaling til, så evt behov for øget kapacitet skal håndteres manuelt. Den tilgængelig kapacitet overvåges løbende af AWS miljøet, og hvis server-belastningen når over bestemte grænseværdier, alarmeres leverandøren.

Databaseserverene anvender et RAID disk setup, hvor data replikeres på tværs af 3 fysiske diske i hver af de 2 driftscentre (6 diske totalt), og nedbrud på diske håndteres automatisk af AWS, der skifter disken, og starter replikering op mod nye diske indenfor få minutter.

De to driftscentre ligger i Irland, og i driftsaftalen er der en option der kan aktiveres, for at sætte et tilsvarende miljø op i Frankfurt, med replikering på tværs af de to lande.

Hvis optionen aktiveres vil der i Frankfurt stå et stand-by miljø, der først modtager trafik hvis det primære miljø i Irland holder op med at svare (der er en tidsmæssige replikeringsfaktor i spil, der gør det uhensigtsmæssigt at load-balance aktivt på tværs af 2 lande).

Det er muligt at konfigurere driftsmiljøet til auto-scaling, så der i hvert driftscenter automatisk startes nye applikations- og databaseservere op hvis server-belastningen når bestemte grænseværdier. Dette vil I så tilfælde være tilkøb med en variabel driftspris, der afhænger af den aktiverede kapacitet.

3.8 Overvågning

OS2rollekatalog anvender Spring Actuator til selv-monitorering, og udstiller sin egen helbredsstatus til det omlæggende driftsmiljø. Denne status anvendes aktivt af AWS miljøet til at optimere oppetiden og tilgængeligheden af OS2rollekatalog. ELB load balanceren dirigerer trafik væk fra overbelastede servere, og lukket for alt trafik til servere der er gået ned, og alarmer sendes til leverandøren på alle negative hændelser i driftsmiljøet.

Der er opsat en række alarmer/triggers på OS2rollekatalogs systemlog, hvor bl.a. en overskridelse af grænseværdier for antal af fejl (timeouts, afvisninger, m.m.) indenfor en given tidsperiode vil resultere i alarmer der sendes til leverandøren.