Informasjonssikkerhet i Nettskjema

2022-03-15

Innholdsfortegnelse

1. Innledning

Nettskjema er et verktøy for utforming og gjennomføring av spørreundersøkelser på nett. Bruksområdet spenner vidt fra seminarpåmelding og faglig quiz, til forskningsundersøkelser og spesialtilpassede mobilapper for innsamling av helseopplysninger integrert mot Tjenester for Sensitive Data (TSD).

Nettskjema utvikles og driftes av Universitetets senter for informasjonsteknologi (USIT) ved UiO, og brukes av Universitets- og høyskolesektoren (etter avtale).

Det er en Spring-basert Java-applikasjon som er designet for svært høy oppetid og skalerbarhet.

Skjemadesign og svar på vanlige skjema lagres i en Oracle database. For forskningsundersøkelser som skal samle inn og behandle sensitive persondata, har Nettskjema integrasjon mot TSD med kryptert sikker lagring av sensitive data.

2. Avgrensing

Denne oversikten er avgrenset til sikkerhetsrelaterte forhold rundt Nettskjema og integrasjon mot relevante systemer som TSD, og går ikke dypt i selve programvaren eller sikringen av servere og infrastruktur. Dokumentasjon på sikring av servere og database kan fremlegges etter avtale.

3. informasjonssikkerhet og personopplysninger

Nettskjema er underlagt UiOs *ledelsessystem for informasjonssikkerhet* (LSIS)
https://www.uio.no/tjenester/it/sikkerhet/lsis/

3.1 Risiko- og sårbarhetsvurdering

Risiko- og sårbarhetsvurdering (ROS) gjennomføres minst annethvert år. Prosessen gjennomføres etter metodikk, rutiner og mal fra LSIS.

Siste versjon av ROS for Nettskjema er fra februar 2021. Det er mulig å få innsyn i siste ROS etter avtale.

3.2 Oppfølging av ROS

Vi følger opp risikoelementer fra ROS med moderat eller høy risiko i en risikohåndteringsplan, der det planlegges og gjennomføres sikkerhetstiltak slik at risiko reduseres til et nivå som kan aksepteres.

Risikohåndteringsplanen utvides med nye risikoelementer hvis det identifiseres nye hendelser som kan medføre moderat eller høy risiko.

Utviklingsarbeidet er i henhold til «best practice» for «OWASP topp 10»:
Top 10 Most Critical Web Application Security Risks

3.3 Sikkerhetstesting

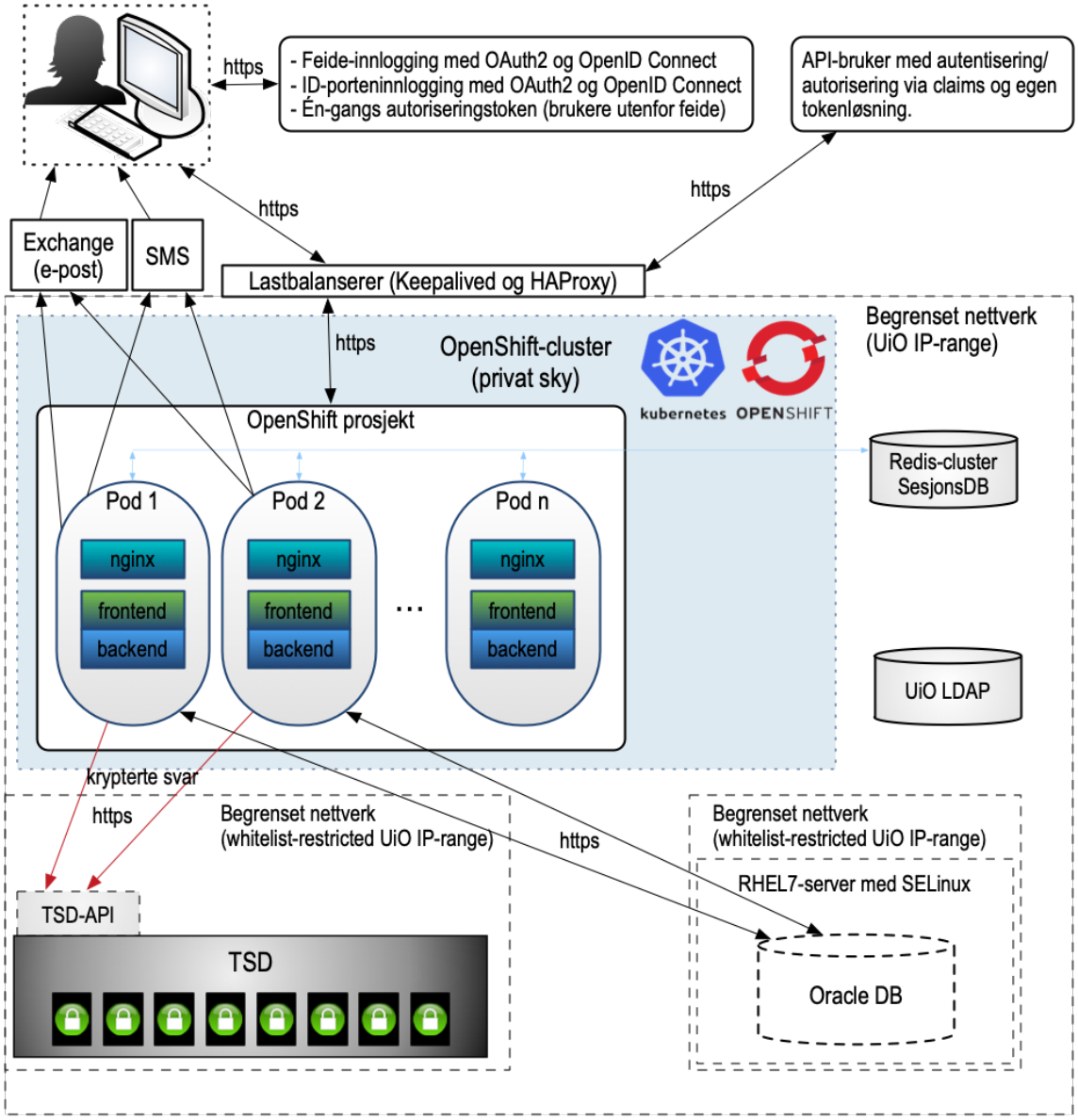
Som et tiltak for å motvirke OWASP **A06:2021-Vulnerable and Outdated Components** bruker vi «*OWASP dependency check*» til automatisk kontinuerlig sikkerhetstesting av sårbarheter i alle programvarekomponenter vi bruker. Alle alvorlige sårbarheter i komponenter følges opp umiddelbart.

4. Teknisk systembeskrivelse

4.1 Autentiseringsmuligheter for respondent

Hvert enkelt skjema må spesifisere en respondentgruppe som kan svare:

- Alle: Uautentisert, for anonyme svar
- UiO- og Feide-brukere: Feide (OpenID Connect/OAuth2)
- Inviterte: Man kan invitere en list av enkeltbrukere via e-postadresse
  - Autentisering via Feide for Feide-brukere
  - Autentisering via éngangstoken for andre brukere
- ID-porten nivå 3
- ID-porten nivå 4
- Signering med ID-porten nivå 3
- Signering med ID-porten nivå 4



4.2 Autentiseringsmuligheter for administrasjon og uthenting av data

Feidebrukere ved institusjoner som har avtale med Nettskjema har tilgang til å bruke Nettskjema.

Det er også mulig å gi tilgang til brukere som autentiserer seg med MinID og BankID.

Det er mulig å gi grupper ved UiO tilgang. Gruppetilgang er ikke implementert for andre institusjoner.

4.3 Kryptering

All kommunikasjon mellom klient og server skjer kryptert.

Nettskjema støtter PGP-kryptering av svar. Offentlig PGP-nøkkel må da

legges inn for skjemaet eller TSD-prosjekt. (Den private delen av nøkkelen skal aldri legges inn i Nettskjema). Kryptering av svaret skjer på server (ikke på klient). Klartekst-besvarelsen vil i dette tilfellet ikke lagres på web-server eller i database, men vil finnes i minne på serveren i en kort periode.

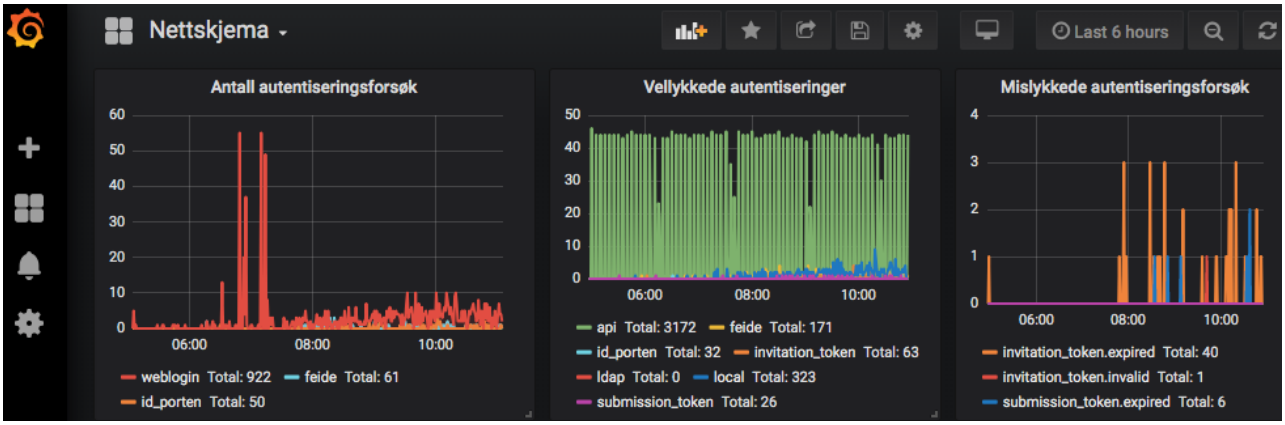
Krypterte TSD-besvarelser lagres i utgangspunktet kun i TSD, men hvis det oppstår en feil i kommunikasjon med TSD-API mellomlagres besvarelsen kryptert i Nettskjemas database.

4.4 Driftsmiljø, infrastruktur og overvåking

Applikasjons-server og utviklings-servere er en del av USITs regulære driftsregime. Her er det sentral overvåking og logging (ELK).

Alle tjenester i produksjon overvåkes. Vi logger helsedata fra applikasjonene som utgangspunkt for alarmer, med varsling via e-post og chat.

Vi teller og måler viktige måltall for applikasjonen, og har flere dashbord for målinger, tidsserier og trendanalyse (Grafana). Dette kan hjelpe oss til å se uregelmessigheter i drift.



4.6 Utvikling og kildekode

Gruppe for utvikling og forvaltning av webapplikasjoner (WAPP) jobber etter prinsippene for smidig utvikling og DevOps.

Et av hovedmålene er kort vei (tid) fra behov for ny funksjonalitet, eller behov for endring/retting, til produksjonssetting.

For å klare dette, og ha kontroll på risiko og sårbarhet, utvikler vi applikasjonen og tilhørende støttesystemer i små inkrementelle steg, og har fokus på automatisert testing og kontinuerlig kvalitetsforbedring i utviklingsstøttesystemene.

Med små endringer fra forrige versjon er det mindre som kan gå feil. Mer isolerte endringer påvirker vanligvis mindre del av applikasjonen. Dette gjør det enklere å oppdage feil og enklere å fikse eller rulle tilbake feil.

Kortere tid mellom utvikling/kodeskriving og prodsetting gjør at det er mindre sjanse for at miljøendringer rundt systemet gir feil. Utvikleren er inne i problemstillingen (i kontekst) og kan raskere/enklere oppdage og rette opp eventuelle feil. Utvikleren er ikke ferdig før utviklet funksjonalitet er i produksjon og virker som planlagt.

Kodekontroll

Alle prioriterte ønsker og behov for utvikling spesifiseres som brukerhistorier eller oppgaver.

Vi praktiserer streng kodekontroll. For all ny funksjonalitet skal det opprettes en egen kodegren (feature branch) i versjonskontrollsystemet (GIT: Bitbucket og GitHub Enterprise) som knyttes til en oppgave.

Tilordning av oppgaver til de enkelte utviklere og status for arbeidsflyt registreres i Jira eller GitHub Issues.

All kode som skal flettes inn må gjennom en kodegjennomgang (code review) i form av en «pull request», der andre utviklere må lese gjennom og godkjenne endringene. Det er dermed minst «4 øyne» på all ny kode som går ut i produksjon.

CI-server, testing og automatisering

Vi bruker CI/CD-server med automatisert bygging, enhetstester, funksjonelle tester og integrasjonstesting (Jenkins).

Kodekontrollen i Jira og Bitbucket er integrert med ulike pipelines i Jenkins.

I tillegg til full kjøring av automatiserte tester ved endringer i prod, har vi integrert automatisert testing så tidlig i prosessen som mulig, dvs. automatisert testing ved alle commit.