1.8 Sikkerhetsløsninger

Hvordan sikre ekstern datatrafikk og API-tilgang.

Dette kapittelet er mest ment for teknisk IKT-folk, men greit for prosjektdeltakere å ha noe begrepskunnskap om saken.

En må forvente at kommersielle standard API, tilhørende SKY-system eller OnPrem-system (lokalt installert hos kunde/driftspartner), har tilstrekkelig sikkerhet for normale krav. Men en kan utvide sikkerheten med 2-faktor-pålogging, varighet til tilgang eller VPN. På linje med flere låser på ei dør, vil økt sikkerhet normalt komplisere bruk og vedlikehold av en løsning, noe som kan bety mindre brukervennlig. Det kan igjen føre til at komplekse passord og krevende tilkoblinger føres på post-it-lapper eller notat-bok på PC, noe som gjøre at sikkerheten blir dårligere enn noen gang.

Behovet for sikkerhet
Systemet eller integrasjonen er vel ivaretatt mtp sikkerhet spør gjerne brukere og kunder, hvor de blir lite beroliget når svaret er at det kommer an på i hvilken grad du krever av sikkerhet.

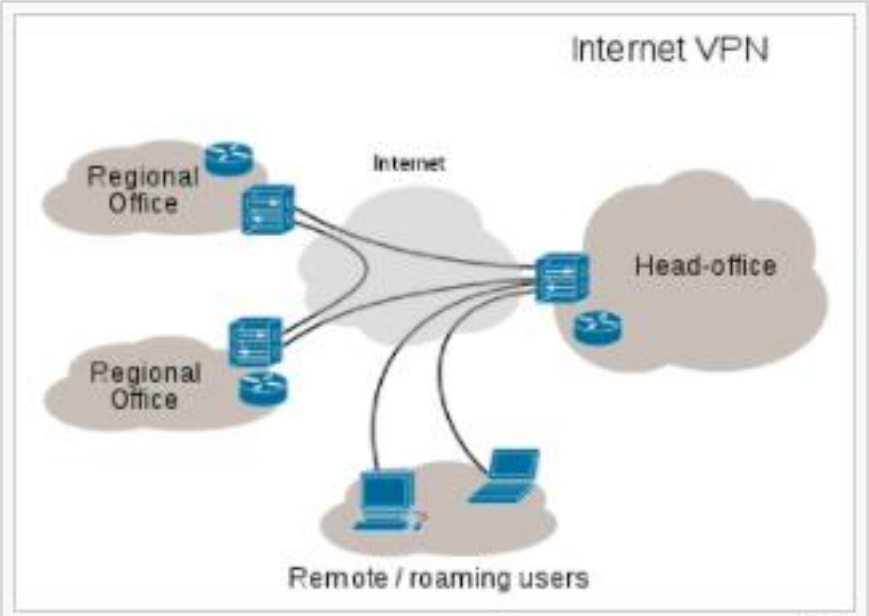
En må derfor analysere behovet og graden av sikkerhet, snakker vi om sensitive helseopplysninger, bankoverføringer, patenter, forsvarshemmeligheter, etc eller tilgangen til å føre timer på en ansatt? Hva er konsekvensen for at informasjon misbrukes, ødelegges eller feilaktig registreres og hvilken pris ønsker vi å betale for økt sikkerhet i form av kostnader og ekstra tid med å få tilgang til system og data?
Overdreven sikkerhet kan være like skadelig, som manglende sikkerhet. Må du en lege låse opp fire låser på vei til akutten, kan pasienten dø pga overdreven sikkerhet. Hvis politi må gjennom for mange prosedyrer og ikle seg sikkerhetsutstyr før de kan aksjonere kan det få katastrofale følger (ref terrorsaker på Utøya og Kongsberg). Det viktigste sikkerhetstiltaket kan være å beslutte graden av sikkerhet, men skal en åpne for eksterne integrasjoner er ett minimum av type sikkerhetsløsninger som en kan anse som standard.

Type Sikkerhet
Vi skal her ta for oss følgende sikkerhetsbegrene WEB, VPN, 2-faktor, token, https, endpoints, DMZ og Proxy.

WEB-applikasjoner
En sak som er svært viktig å være klar over med applikasjoner som går via internett i form av WEB-applikasjoner mot et API, er at scriptene du sender over nettet, er helt åpen. Du kan ta høyre mustast på en WEB-applikasjon og velge inspiser for å se datainnhold og store deler av programkoden (HTML). En må derfor ALDRI lagre/vise/sende ukrypterte passord fra en slik applikasjon, selv om trafikken sendes kryptert. Passord kan mao ikke hentes opp fra en database/API, men må skje innenfor brannmur, helst i et API.

VPN – Virtual Private Network

For å sikre at trafikken mellom 2 endepunkt i ett nettverk, kan en benytte en VPN-løsning. Da vil sammenkoblingen i ett intranett, via Internett, oppleves som et utvidet nettverk som vi kan kalles Ekstranett. Hvis vi har en API-API synk-integrasjon mellom 2 system via en slik VPN-kanal, kan dette komme i tillegg til øvrig sikkerhet eller redusere kravet for sikkerhetsløsningen.



Skjema over et VPN-nettverk

Et virtual private network, er et datanettverk hvor «punkt-til-punkt»-forbindelser, såkalte «tunneler», sluttes gjennom et annet datanett (som for eksempel internett).

En VPN-tunnel kan være kryptert, noe som er viktig når man ikke kjenner, eller er usikker på sikkerheten gjennom et eventuelt offentlig datanett, som for eksempel internett. Det gjelder særlig ved besøk av ukrypterte sider med http-tilkobling.

VPN ble først utviklet som et verktøy for bedrifter. Hensikten var å øke sikkerheten ved overføring av data mellom servere, samt ved kommunikasjon mellom ansatte.

Kilde: https://no.wikipedia.org/wiki/Virtuelt_privat_nettverk

2 – faktor

For å kunne utvide sikkerheten ved pålogging med flere låser eller flere dører som må låses opp, kan en bruke 2-faktor pålogging (autentifisering). Dette er løsninger de fleste av oss i dag er kjent med ift Altinn, Nettbank, etc, som betyr at påloggingen må skje i 2 steg, normalt da vet at du må bekrefte med pinkode på mobil. Selv elv om vi har aktivert to-faktor-autentisering er vi ikke helt sikre. Noen kan faktisk lage falske nettsider som også får oss til å oppgi engangskoden samtidig med passordet (et eksempel her). Det er litt for omfattende å gå i detaljer her, men også BankID-mobil kan i teorien også svindles. 2-faktor gjelder primært til bruk i slutt-bruker applikasjoner om WEB og APP (mobil).

Image courtesy of Stuart Miles at FreeDigitalPhotos.net

I teorien kan en maskinell 2-trinns pålogging også benyttes i en API-API integrasjon, men da er nok VPN-løsninger å foretrekke. I tillegg ligger url-linken, bruker og passord (endpoints) skjult inne i applikasjoner eller API, så det skal litt til at en uvedkommende å finne denne informasjonen. En må skille mellom integrasjonsbrukere og sluttbrukere (skjermbilder), hvor en sluttbruker ikke skal kunne logge seg på API og hvor en integrasjonsbruker ikke når funksjoner i selve applikasjonen.

Etter hvert som vi ser at overdreven sikkerhet kan være negativt og lite brukervennlig, vil forenklet pålogging med kun pinkode (4 siffer) erstatte komplekse passord, da enten vi at en alt har gjort en passord/2-faktor-pålogging med en varighet (timer/dager), hvor kun pin-kode brukes for rask tilgang eller at hele påloggingen kun skjer med pinkode. Det siste kan være for typisk timeregistrering på mobil, hvor en ugyldig tilgang har lav risiko eller konsekvenser. Normalt vil brukernavn ble lagret i nettleseren, slik at kun passord eller pinkode må angis ved pålogging.

Kilde: https://www.datasikkerhetsboka.no/blogg/2016/08/04/to-faktor-autentisering/

Token

Ved pålogging utføres en generering av nøkkel til videre bruk mot API for å kunne lese og skrive mot databasen/applikasjonen, noe vi kaller «Token eller Ticket» med bruk av en service/tjeneste i API som kan hete «PostToken». Med en slik nøkkel, trenger en kun en pålogging, og sender med nøkkelen sammen med tjenester du vil utføre (Request). Denne e-nøkkelen i form av lang kryptisk streng kan ha en levetid på minutt, timer eller dager, styrt i fra avsender (API). Når nøkkelen sin levetid er utløpt, vil API-et returnere en «Expire Token» melding som medfører krav om ny pålogging. Dette er type programvarelogikk utvikler må ta med i sin applikasjon eller integrasjon.

Https

Kryptert trafikk over internett går via URL med https, hvor «s» betyr at krypteringssertifikat blir benyttet. Sertifikat må bestilles fra autorisert sertifikatutsteder og kan tilhøre bedriften som har lokal serverpark eller hos driftsleverandør i form av Wildcard-sertifikat til bruk for sine driftskunder.

Endpoints

URL som henviser til API sin IP-adresse kaller vi endpoint for API-et, normalt da nok med kun en slik URL-link, brukernavn og passord for å logge seg på et API, med mindre en har utvidet sikkerhet.

DMZ og Proxy

Et API som er installert i ett servermiljø kan nås via åpen port i brannmur, hvor trafikken styres på en server plassert utenfor brannmur i det vi kaller DMZ (de-militær sone). På denne serveren legger en da en «proxy-tjeneste» for styre tilgang, kryptering av trafikken, noe som da kan være en delt ressurs for flere. Dvs sette opp proxy-tjenester for flere aktører på samme web-server i et driftsmiljø som omfatter mange kunder/bedrifter.

EKSEMPEL SIKKERHETSKRAV BDO

Alle punkter skal besvares og vil være gjenstand for vurdering av i hvilken grad leverandør og løsning er egnet til å være en del av selskapets fremtidige systemporteføljе. En kan her merke seg at rutiner, organisering og forvaltning er sentralt, mens den rent tekniske løsningen langt mer anser som en selvfølge er tilfredsstillende. Det er det menneskelige aspektet rundt sikkerhet som gjerne er det svakeste punkt.

1. KRAV TIL STYRING	
1.1. Styringssystem for informasjonssikkerhet	Dokumenter et fungerende styringssystem gjennom en kortfattet beskrivelse og ev. relevante sertifiseringer. Sertifisering er ikke et «må»-krav, men det må som et minimum kunne vises til: • En sikkerhetsorganisasjon der roller og ansvar er tydelig beskrevet og dokumentert. • Styring- og ledelsesforankring av sikkerhetsrisiko • At det gjennomføres risikovurderinger som identifiserer sikkerhetsbehov, og gir prioritering av sikkerhetstiltak • At det er allokert tilstrekkelig med ressurser, med riktig kompetanse, til sikkerhetsarbeidet • At man har rutiner for å kontrollere at ansatte etterlever sikkerhetskrav
1.2. Personvern og taushetsplikt	Beskriv hvordan krav til oppbevaring og behandling av personopplysninger, og andre relevante lover og regler ivaretas. Dokumenter sikker praksis for informasjonsdeling, samt rutiner for å kontrollere at de ansatte følger denne praksisen.
1.3. Kontroll med særskilte tilganger	Dokumenter kontroll og begrensing på hvem som får særskilte tilganger, hvordan disse administreres og kontrolleres. Dokumenter prosedyrer for bakgrunnssjekk av ansatte. -
1.4. Kontroll med underleverandører	Dokumenter bruk av sertifiserte/evaluerte produkter.

2. KRAV TIL SIKKERHETSARKITEKTUR	
2.1. Sikker IKT-infrastruktur	
2.2. Kontroll med tilgjengelighet	Dokumenter hvordan tilgjengelighet sikres gjennom redundante systemkomponenter. Dokumenter hvordan tilgjengelighet ivaretas gjennom backupløsning.

2. KRAV TIL SIKKERHETSARKITEKTUR	
2.3. Kontroll med konfidensialitet og integritet	Dokumenter hvordan kundeinformasjon er beskyttet mot interne og eksterne trusselaktører, slik at ikke uvedkommende kan eksfiltrere (trekke ut) eller endre informasjon ved å oppnå uautorisert tilgang til applikasjoner, databaser eller serverinfrastruktur.
2.4. Prosess for testing og utvikling	Dokumenter separate miljøer for utvikling, test og produksjon. Dokumenter testing gjennom hele utviklingsløpet.

3. OPERASJONELLE KRAV	
3.1. God prosess for vedlikehold	Dokumenter og beskriv prosess for oppdatering av programvare for applikasjon og infrastruktur. Dokumenter utviklingsplan for sikkerhet i tjenesteproduksjon i tråd med utvikling i teknologi og trusselbildet over tid.
3.2. God endrings- og godkjenningsprosess	Dokumenter rutiner og prosesser for sikker drift, i form av ITIL-rammeverket eller tilsvarende. -
3.3. God prosess for hendelseshåndtering	Beskriv hvordan virksomheten håndterer en situasjon der det er oppdaget sikkerhetsbrudd. Beskriv hvordan BDO bli varslet og involvert ved et eventuelt sikkerhetsbrudd. .
3.4. God prosess for avvikshåndtering kontinuitet og beredskap	Beskriv hvordan avvik registreres og behandles. Dokumenter beredskapsplanverk *og* gjennomførte øvelser (årshjul).
3.5. Bruk av automatiserte verktøy for sårbarhetsovervåkning og oppdatering av sikringstiltak	Dokumenter at det gjennomføres regelmessig sårbarhetsskanning. Dokumentere prosess for oppdatering av programvare. Beskriv prosess for å varsle BDO hvis det blir oppdaget alvorlige sårbarheter.
3.6. Bruk av sikkerhetsovervåkning	Dokumenter og beskriv løsning for overvåkning, herunder: antivirus, IDS, logging, analyse av aktivitetslogger, mm. .
3.7. Gjennomføring av sikkerhetstester	Dokumenter og beskriv regelmessig sikkerhetstesting av applikasjon gjennom utviklingsløpet. Bekreft at det gjennomføres minst årlig penetrasjonstest av infrastruktur og applikasjon. .