Bacheloroppgave

NTNU
Norges teknisk-naturvitenskapelige universitet
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for datateknologi og informatikk

Angell-Jacobsen, Erlend
Slettebakken, Sebastian

SOAR i Azure Sentinel

Bacheloroppgave i Informatikk: Drift av datasystemer
Veileder: Meisingseth, Stein
Medveileder: Mathisen, Pål

Mai 2021

NTNU
Kunnskap for en bedre verden

Angell-Jacobsen, Erlend
Slettebakken, Sebastian

SOAR i Azure Sentinel

Bacheloroppgave i Informatikk: Drift av datasystemer
Veileder: Meisingseth, Stein
Medveileder: Mathisen, Pål
Mai 2021

Norges teknisk-naturvitenskapelige universitet
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for datateknologi og informatikk

NTNU

Kunnskap for en bedre verden

SOAR i Azure Sentinel – Gruppe 45

Oppgaven er delt i fire ulike rapporter:

Forstudierapport

Designrapport

Driftsrapport

Sluttrapport

Gruppe 45

Bachelorprosjekt

SOAR i Azure Sentinel

Forstudierapport

Versjon 1.0

Forfattere: *Erlend Angell-Jacobsen, Sebastian Slettebakken*

Revisjonshistorie

Dato	Versjon	Beskrivelse	Forfatter
29/01/2021	1.0	Første utkast	Erlend Angell-Jacobsen, Sebastian Slettebakken

Innholdsfortegnelse

1 Tabell- og figurliste

2 Introduksjon – hensikten med dokumentet

Hensikten med dokumentet er å beskrive de viktigste resultatene for forstudiefasen i bachelorprosjektet. Dette innebærer følgende punkter:

- Definere prosjektets mål, basert på en vurdering av oppdragsgivers behov og problemer med dagens situasjon
- Vurdere lønnsomheten av prosjektet, gjennom en sammenstilling av forventet nytteverdi mot forventede kostnader
- Definere strategien og overordnede planer for prosjektet
- Vurdere ressursbehovet for gjennomføringen av prosjektet

Sopra Steria ønsker å utvikle sikkerhetsleveranser med SOAR («Security Orchestration, Automation and Response») for å oppnå økt presisjon i leveransene samt forbedre effektivitet i leveransene på tvers av kundene. Oppgaven undersøker hvordan dette kan implementeres i Azure Sentinel.

Dokumentet definerer prosjektets omfang, slik at prosjektgruppen og oppdragsgiver får en felles forståelse for framdrift, kostnader og resultater.

Rapporten er delt opp på følgende måte:

- Kapittel 1: Tabell- og figurliste – Tabeller og figurer i dokumentet
- Kapittel 2: Introduksjon – Hva som er hensikten med dokumentet.
- Kapittel 3: Teknologier – Forklaring av teknologier vi skal ta i bruk.
- Kapittel 4: Bakgrunn for prosjektet – Med en beskrivelse av problemer, behov og dagens systemer.
- Kapittel 5: Prosjektmål – Hva som skal gjøres i prosjektet og mål vi ønsker å oppnå.
- Kapittel 6: Interessenter og rammebetingelser – Hvem som er involvert i prosjektet og deres suksesskriterier og rammebetingelser.
- Kapittel 7: Kritiske suksessfaktorer – Hva som skal til for å lykkes med prosjektet.
- Kapittel 8: Risikoanalyse – Sannsynligheten og konsekvensen av problemer som kan oppstå underveis i prosjektet.
- Kapittel 9: Kost/nytte-analyse – Lønnsomheten av prosjektet i form av en sammenstilling av forventet nytte og kostnader.
- Kapittel 10: Retningslinjer og standarder – Hva vi må forholde oss til når vi gjennomfører prosjektet.
- Kapittel 11: Prosjektorganisering – Hvordan vi har organisert prosjektet.
- Kapittel 12: Anbefaling om videre arbeid – Om prosjektet anbefales å gjennomføres.
- Kapittel 13: Kilder – Oversikt over kilder som er brukt i planleggingen av prosjektet.

3 Teknologier

Oppgaven inkluderer bruk av standardiserte teknologier og produkter som introduseres under:

3.1 Azure

Azure er en skyløsning laget av Microsoft. Azure har over 200 systemer som er laget for å skaffe nye løsninger til markedet. Azure kan drifte skyløsinger, men også på on-premises systemer. Azure har flere verktøy og rammeverk som kan tilpasses alle bedrifter.

Azure skriver at sikkerheten er på topp og at det blir benyttet en enorm mengde penger for å sikre at sikkerheten er god nok. Azure bruker ca. en milliard kroner i året, på å opprettholde og utvikle gode sikkerhets systemer. (Microsoft, 2021) Som sagt har Azure over 200 forskjellige systemer som skal bidra til forenkling av driften på dine nettverk av virtuelle eller lokale pc-er og servere. Når man begynner med oppsett i Azure, kan det være vanskelig å bestemme hva man skal ta i bruk, men Azure portal gjør det mer oversiktlig, samt at man får samlet det meste på en plass. Man kan også styre disse systemene programmatisk gjennom systemspesifikke API-er og maler. I dette prosjektet kommer vi til å bruke Azure portal for det meste av styring av Azure Sentinel.

3.2 Azure Sentinel

Azure Sentinel er en skyopprinnelig SIEM plattform som bruker AI til å raskere analysere datamengder i bedrifter. Denne SIEM-en samler inn data fra flere kilder, brukere, programmer, servere, enheter kan både være i skyen og lokalt. Azure Sentinel vil kunne benyttes i de alle fleste bedrifter og integreres med eksisterende systemer. Azure Sentinel er ikke kun et SIEM-system. Det er også et SOAR system, og det er det vi skal fokusere på i dette prosjektet. Man automatiserer enkle og vanlige hendelser gjennom å benytte playbooks. SOAR-systemet er høyst skalerbart og åpner for videre utvikling og tilpasser seg dagens trusler. Den tilpasser seg både truslene og nye teknologier ettersom de kommer. (Microsoft, 2021) Som sagt med Azure Sentinel vil man kunne automatisere enkelte hendelser. Dette frigjør tid til analytikere, slik at de i stedet kan se grundigere på mer avanserte sikkerhetshendelser. I mindre bedrifter som ikke har egne dedikerte analytikere vil man frigjøre tid til IT-arbeidere, slik at de kan gjøre annet arbeid.

SIEM er et sikkerhetssystem som ofte benyttes ved bedrifter. Det er her vi har mulighet til å korrelere data, bygge deteksjons regler. SOAR-systemer hjelper med automatisering av hele sikkerhetshendelser eller med å effektivisere behandlingen av sikkerhetshendelser. Playbooks gjør det mulig å automatisere hvert steg i og full automatisere alt eller sette opp slik at en analytiker kan ved gjennom et klikk løse sikkerhetshendelsen. Målet for både SIEM og SOAR er å redusere risikoen for bedriften. Det gjør det gjennom å effektivisere deteksjon og respons på sikkerhetssaker.

4 Bakgrunn for prosjektet

Bakgrunnen for dette prosjektet er at vi har en bacheloroppgave for Sopra Steria. De ønsker at vi skal finne svar på «Hvordan gjøre Security Orchestration Automation and Response (SOAR) på tvers av enterprise-miljøer». Vi har valgt å svare på følgende problemstilling:

Hvordan kan Azure Sentinel brukes til å øke treffsikkerheten og effektivisere arbeidet til en analytiker gjennom automatisering av enkle prosedyrer?

Med en såpass stor og åpen oppgave, er det viktig å redusere omfanget. Ut ifra problemstillingen, ønsker vi å avgrense prosjektet ved å fokusere på Azure Sentinel og spesielt muligheten for automatisering der.

4.1 Beskrivelse av problemer og behov

Sopra Steria utvikler kontinuerlig sikkerhets-kapabilitetene de bruker til å levere tjenestene sine. Å redusere manuelt arbeid for sikkerhetsanalytikere, øke presisjon på leveransene og minimere responstiden på sikkerhetssaker er tre viktige mål i utvikling. Eksempler på hendelser som håndteres manuelt kan være dersom noen prøver å logge på med feil passord og fra en ny IP-adresse, passordgjenoppretting og fikse tilgang på nettverksressurser som ansatte har mistet tilgang til. Vi anser det derfor som et behov at så mye av det manuelle og enkle arbeidet automatiseres, så langt det lar seg gjøre. Dette behovet må ikke gå på bekostning av den eksisterende kvaliteten.

Noen fordeler og behov som dekkes ved å implementere en SOAR løsning er:

<u>Raskere responstid</u>

Ved implementering av SOAR, kan man spare tid ved å at flere relaterte varsler fra ulike systemer automatisk blir samlet og gruppert til en enkelt hendelse. Det gjør det mulig for systemet å håndtere hendelsen uten manuelt inngrep dersom det er satt opp automatisering for denne type hendelser. Ved raskere responstid, er man bedre rustet mot både enkle sikkerhetshendelser, men også mot større angrep.

<u>Optimalisere trusselinformasjon</u>

Det er mye informasjon og lange logger for analytikerne å gjennomgå ved en hendelse. Ved å ta i bruk SOAR, kan man automatisk få tilgang til den relevante informasjonen for enklere håndtering av hendelsen. Det gjør det lettere for analytikerne ved at de nesten umiddelbart får tilgang til riktig informasjon og kan håndtere hendelsen raskere enn dersom de måtte se over mye «unødvendig» informasjon. Her er det viktig å minimere falske positiver. Azure Sentinel tilbyr avansert AI for å optimalisere og hente ut relevant informasjon.

Redusere manuelt arbeid og standardiserte prosesser

Ved å automatisere deler av arbeidet til analytikerne, slipper de å gjøre dagligdagse og gjentagende manuelt arbeid. Analytikerne kan derfor bruke tiden på å håndtere mer avanserte hendelser. Dette er en av de viktigste grunnene til at man ønsker å innføre SOAR i en bedrift, da analytikere fort blir lei av å gjøre manuelle dagligdagse oppgaver. Dette kan føre til redusert arbeidsmoral, noe bedriften ønsker å unngå.

Gjør det lettere å integrere flere tjenester

SOAR løsninger gjør det lettere å integrere tjenester som eksisterende on-premises, skybaserte løsninger, endpoint security og annen infrastruktur. Et annet viktig moment her, er at løsningen er skalerbar. Det gjør det altså mulig å ekspandere (eller redusere) infrastrukturen etter behov.

Kutte kostnader

SOAR vil frigjøre analytikeres tid til å gjøre andre oppgaver som ikke er automatisert. Det muliggjør raskere vekst i kunder uten å ha samme vekst i antall analytikere. Dette resulterer i lavere kostnader for å levere tjenestene.

Her er noen tall på kostnader redusert, men dette vil ikke gjelde for alle bedrifter (Kaplan, 2020). Noen bedrifter vil spare mer, andre vil spare mindre.

- 90% spart på rapportering
- 80% spart på å lage playbooks
- 70% spart på å håndtere varslinger
- 60% spart på opplæring av analytikere
- 30% spart på skiftledelse

Prosjektet skal lage et forslag til hvordan automatisering av sikkerhetssaker bør implementeres i Azure Sentinel for å oppnå målene med SOAR.

4.2 Kort om dagens systemer og rutiner

I Sopra Steria er arbeidet prosessorientert og deres SOC (Security Operations Center) håndterer sikkerhetshendelser etter Security Incident Management prosessen. Prosessen går ut på å identifisere og redusere sikkerhetshendelser og skadeomfanget for brukerne og få tjenestene tilbake til normalen så raskt som mulig etter en håndtert hendelse. Det er viktig å minimere den negative virkningen i virksomheten for å sikre best mulig nivå av de tre faktorene innen informasjonssikkerhet: konfidensialitet, integritet og tilgjengelighet.

Hendelseshåndtering etter Security Incident Management prosessen foregår i fire faser:

- *Monitoring and Detection*
    - Bruker et SIEM-system (Security Information and Event Management) for å samle data
    - Oppdager sikkerhetsavvik i den overvåkede infrastrukturen
    - Sammenligner og korrelerer informasjon fra ulike datakilder
    - Data om IOC (Indicator of Compromise) – IP-adresser, epost-adresser, URL-er, domener
- *Triage*
    - Oppdaget hendelse blir analysert
    - Sikkerhetshendelsen blir bekreftet
    - Sikkerhetshendelsen blir klassifisert
    - Sikkerhetshendelsen blir prioritert
    - Sensitiviteten til sikkerhetshendelsen blir definert
    - Responsen blir eskalert
- *Incident response*
    - Samle relevant informasjon om hendelsen
    - Sikkerhetshendelsen begrenses
    - Fjerne kilden til skade
    - Fjerne muligheten for at hendelsen gjentar seg
    - Gjenopprette tjenesten så fort det lar seg gjøre
- *Post incident*
    - Lessons Learned – hva lærte vi av hendelsen?
    - Kunne vi funnet hendelsen med mer overvåking eller en annen type overvåking
    - Playbooks opprettes eller oppdateres der det er nødvendig
    - Opprette avviksrapport
    - Registrere oppfølgingstiltak

Azure Sentinel brukes i dag som SIEM-system i Sopra Steria SOC. Azure Sentinel legger til rette for SOAR (Security Orchestration, Automation and Response). Man har da mulighet til å oppnå høyere grad av automatisering og man får effektivisert målene til Security Incident Management prosessen. Sikkerhetsanalytikerne hos Sopra Steria bruker i dag mye tid på manuelt arbeid i fase 2 og 3 nevnt over. Det vil være disse fasene vi kommer til å fokusere på å automatisere i dette prosjektet.

5 Prosjektmål

I dette prosjektet skal vi forsøke å oppnå flere prosjektmål. Prosjektmålene blir formulert ut ifra det som blir skrevet i kapitlet om bakgrunnen for prosjektet. Dette vil være grunnlaget for å kunne planlegge prosjektet, og være vurderingsgrunnlaget for om resultatet av prosjektet står i samsvar med det som ble avtalt med arbeidsgiver.

Vi skiller på tre forskjellige typer prosjektmål: Effektmål, resultatmål og prosessmål.

5.1 Effektmål

De effektmålene vi har satt oss beskriver den ønskede effekten av prosjektet, og er det primære grunnlaget for i det hele tatt å gjøre noe.

- *Redusere tiden en analytiker bruker på en sikkerhetshendelse*. Gjennom bruk av automatisering og fremstilling av informasjon.
- Åpne muligheten for automatisering i større grad med bruk av Azure Sentinel.
- *Redusere IT-kostnader,* ved at systemet skal i større grad baseres på automatiseringen.
- *Øke IT-kompetanse*. Analytikere vil bli i større grad satt til viktigere hendelser og ikke gjentakende små sikkerhetshendelser.

5.2 Resultatmål

Mens effektmålene beskriver den ønskede effekten av prosjektet, er resultatmålene hva vi velger å gjøre for å oppnå den ønskede effekten. Resultatmålene beskriver hva som konkret skal foreligge som resultat når prosjektet er ferdig.

- *Azure Sentinel system med noen grad av automatisering.*
- *Statistikk på hvilken grad et Azure Sentinel system vil bidra for analytikere.*
- *Prosjektet skal være ferdig før 20.05.2021.*

5.3 Prosessmål

Følgende prosessmål er knyttet til selve prosessen som prosjektet skal igjennom, og vil være et mål for den effekten prosjektarbeidet har på prosjektdeltakerne.

- Kompetansebygging på Azure Sentinel.
- Kompetansebygging på teamarbeid.
- Teammedlemmer blir bedre kjent.

- Erfaringer skapes gjennom arbeid med bachelor.

5.4 Prosjektets omfang

Omfanget til prosjektet er relativt stort med tanke på at det er kun to prosjektarbeidere. Her skal vi skisse opp grensene for prosjektet.

- Prosjektet skal inneholde flere rapporter:
  - Forstudierapport
  - Designrapport
  - Driftsrapport
  - Sluttrapport
  - En presentasjon
  - En egenevaluering
- Prosjektet skal bruke Azure Sentinel for automatisering og effektivisering av enkle sikkerhetshendelser.
- Prosjektet skal implementeres gjennom bruk av Azure Logic Apps.
- Prosjektet skal også inneholde opplæring av prosjektdeltakere i KQL.

Slik som prosjektet er nå er det omfattende, dermed er det viktig at vi avgrenser prosjektet mer. Det er derfor viktig at vi presiserer hva prosjektet ikke skal gjøre.

- Prosjektet skal ikke erstatte dagens systemer, men heller være med på å forbedre.
- I prosjektet er ikke brukeropplæring et krav, men brukermanualer vil bli lagt ved.
- Planlegging av implementasjon inn i systemene til Sopra Steria.

5.5 Produktets funksjonelle egenskaper

Tabellen underviser en oversikt over de funksjonelle egenskapene det nye systemet må ha for å løse problemene beskrevet i kapittel 2.1. De funksjonelle egenskapene til hvert av problemene er beskrevet til høyre i tabellen som vellykkede løsninger.

Problem:	**Berører:**	**Konsekvens:**	**Løsning**
Lite eller manglende automatisering	Analytikere	Det blir mye gjentakende arbeid for analytikere	Systemer med mulighet for automatisering vil bidra til at en del gjentakende arbeid blir borte
Vanskelig å integrere andre systemer	Analytikere	Må bruke flere forskjellige systemer for å løse sikkerhetshendelser	SOAR gjør det lettere å bruke informasjon fra andre systemer, som for eksempel End Point Security og Logic Apps

Tabell 1 Funksjonelle egenskaper

5.5.1 Ikke-funksjonelle egenskaper og krav

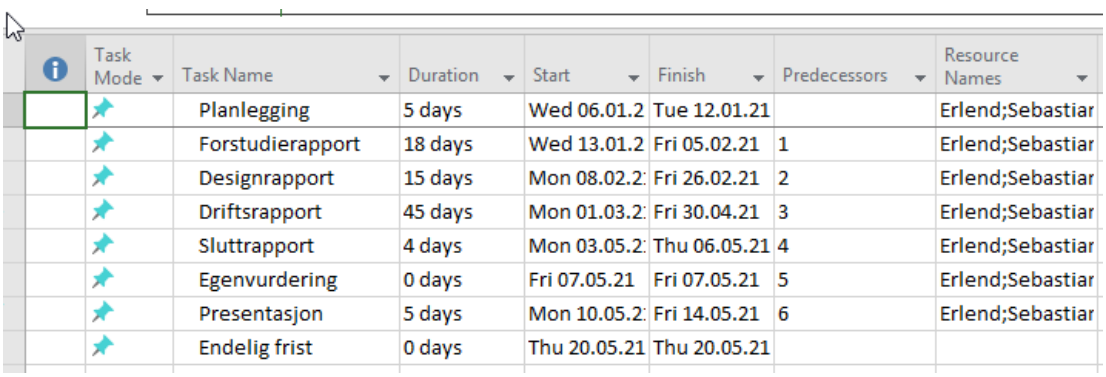
Disse er egenskaper produktet har, men som er ikke-funksjonelle. Selv om vi snakker om problemer her, er det ikke slik at de nødvendigvis er problemer, men egenskaper med rom for forbedring.

Problem:	**Berører:**	**Konsekvens:**	**Løsning**
Treg responstid	Sluttbruker	Sluttbruker må vente lenge for å få fortsette arbeidet	Automatisering av enkle sikkerhetshendelser vil bidra til å kutte responstiden
Mye repetitivt arbeid for analytikere	Analytikere	Analytikere får ikke brukt tiden sin på viktigere sikkerhetshendelser	Automatisering av enkle sikkerhetshendelser vil bidra til å kutte tiden på enkle hendelser
Kutte kostnader	Bedriften som helhet	Ledelsen vil bruke ressurser i opplæring av nye analytikere	Automatisering av enkle sikkerhetshendelser vil bidra til å kutte kostnad på opplæring. Senere vil det også bidra med at det kreves færre analytikere
Skalerbar	Alle i bedriften	Bedriften kan slite med å følge opp med etterspørsel om systemene de har i dag ikke er nok	Azure Sentinel er skalerbar. Det trengs kun flere analytikere for å håndtere systemet

Tabell 2 Ikke-funksjonelle egenskaper

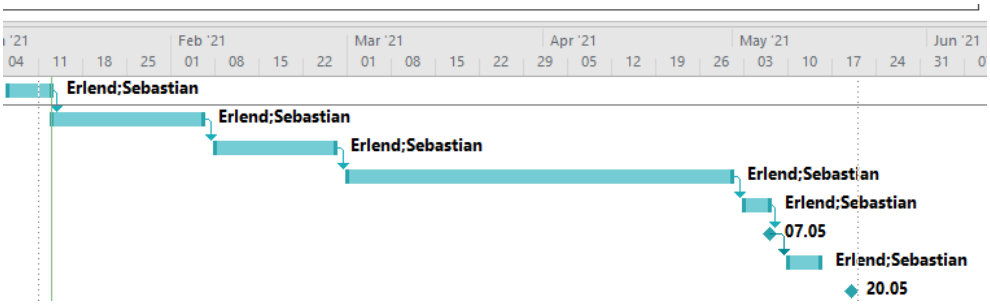
5.6 Prosjektets milepæler og hovedaktiviteter

Vi har i forbindelse med denne forstudiefasen opprettet en prosjektplan med oversikt over blant annet foreløpige prosjektfaser, aktiviteter og milepæler med tilhørende tids- og kostnadsestimater, samt ressursdisponering av prosjektteamet. Vi benytter MS Project til prosjektplanleggingen og den vil bli vedlagt under innlevering. Her er et utklipp på hvordan den ser ut under første uken av prosjektet. Planen er under kontinuerlig endring, så bildene under blir fort utdatert.



Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
	Planlegging	5 days	Wed 06.01.2	Tue 12.01.21		Erlend;Sebastiar
	Forstudierapport	18 days	Wed 13.01.2	Fri 05.02.21	1	Erlend;Sebastiar
	Designrapport	15 days	Mon 08.02.2	Fri 26.02.21	2	Erlend;Sebastiar
	Driftsrapport	45 days	Mon 01.03.2	Fri 30.04.21	3	Erlend;Sebastiar
	Sluttrapport	4 days	Mon 03.05.2	Thu 06.05.21	4	Erlend;Sebastiar
	Egenvurdering	0 days	Fri 07.05.21	Fri 07.05.21	5	Erlend;Sebastiar
	Presentasjon	5 days	Mon 10.05.2	Fri 14.05.21	6	Erlend;Sebastiar
	Endelig frist	0 days	Thu 20.05.21	Thu 20.05.21		

Figur 1 Prosjektets milepæler



Figur 2 Prosjektets milepæler tidslinje

Milepælene kommer naturligvis når vi ferdigstiller rapportene. Fasene er som følger med oppstartsfasen med planlegging. Deretter kommer vi til planleggingsfasen hvor vi finner forstudierapport og designrapport. Så kommer vi til gjennomføringsfasen hvor vi skal utarbeide en driftsrapport. Til slutt kommer vi til avslutningsfasen, der sluttrapporten, egenvurdering og presentasjonen befinner seg.

6 Interessenter og rammebetingelser

For å kartlegge interessenter og deres betydning i prosjektet gjennomfører vi en interessentanalyse. En interessent er en aktør som har interesse av prosjektet. Det kan både være enkelt personer og grupper. Interessentanalysen er kritisk for å gjennomføre prosjektet på en hensiktsmessig måte.

Senere i kapitlet vil vi se nærmere på rammebetingelsene for prosjektet. De vil være de absolutte kravene til prosjekt resultatet og vil være med på å legge føringer for hvordan prosjektet gjennomføres.

6.1 Interessentanalyse

I dette prosjektet er det Sopra Steria som er oppdragsgiver. De åpner opp for at vi som skriver oppgaven får et innblikk i en arbeidshverdag som man ellers ikke hadde fått, om oppgaven var gitt av NTNU. Representanten fra Sopra Steria er Pål Mathisen og vil være med å veilede prosjektet. Mathisen vil være både ekstern og intern ettersom han både er oppdragsgiver og veileder.

Videre så har man enda en veileder som er Stein Meisingseth. Han vil være representanten fra NTNU. NTNU må naturligvis også representeres, med tanke på at det er de som er utdanninginstitusjonen.

Om prosjektet blir gjennomført vil det påvirke flere interessenter daglig. I dette prosjektet har man to typer sluttbrukere. Den første er analytikeren, og den siste er brukeren av PC-en som er overvåket av Azure Sentinel. Analytikeren vil forhåpentligvis få en lettere arbeidshverdag. Ettersom flere av arbeidsoppgavene blir effektivisert og automatisert. Sluttbrukeren som bruker en PC overvåket av Azure Sentinel bør ikke bli påvirket i stor grad. Det kan for eksempel være at brukeren må svare på en automatisk E-mail i stedet for å ringe IT støtte, men alt i alt skal prosjektet ha liten betydning for den type bruker.

I tabellen under er delt i tre kolonner. Kolonnene inneholder suksesskriterier. Altså hva som skal til for at prosjektet blir en suksess i deres øyne. Også en kolonne med deres bidrag til

Interessent	**Suksesskriterier**	**Bidrag til prosjektet.**
	Eksterne	
Oppdragsgiver	Mindre kostnader knyttet til datasystemer ved bedre oversikt over de data som finnes i systemet og automatisering av systemet. Fullføring av prosjektet innenfor de gitte tids- og kostnadsrammene.	Er med å ta beslutninger i prosjektet og vil være med på å godkjenne resultatene som oppnås gjennom prosjektet. Stiller også med de nødvendige systemene som må til for at prosjektet kan gjennomføres.

Sluttbruker	Skal ikke ha negative konsekvenser av gjennomføringen av prosjektet.	Kjennskap til hvordan de føler sikkerheten rundt seg.
Analytiker	Bedre oversikt over data. Minke arbeids med gjentakende arbeid.	Kjennskap til de problemene med gjentakende arbeid.
Interne		
Prosjektgruppen	Ende opp med et produkt som er vellykket med fornøyde interessenter. Også at det leveres innen gitte tidsfrister.	Ansvar for en vellykket gjennomføring av selve prosjektet innenfor de gitte tids- og kostnadsrammene.
Veiledere	Ende opp med et godt gjennomført prosjekt. Det vil si leveranse av et fungerende produkt innenfor den gitte tids- og kostnadsrammen.	Tilføring av kunnskap til prosjektgruppen og veiledning underveis i prosjektet.

Tabell 3 Interessenter

6.2 Rammebetingelser

Rammebetingelser er absolutte krav som settes rundt prosjektets gjennomføring og resultat. Dette er ikke ønsker. Nedenfor er en liste med de kravene vi har satt for dette prosjektet.

Type	Navn	Beskrivelse
Tid	Innleveringsfrist for prosjekt	Bachelor-prosjektet skal levers den 20.05.21. Dette er siste mulighet til å ferdigstille prosjektet
Kost	Holde seg innenfor kostnadsrammen	Det er et krav fra NTNU og Sopra Steria at vi ikke sløser med penger og eventuelt andre ressurser.
Tilgang til informasjon	Skrive under på taushetserklæring.	Om prosjektgruppen skal ha tilgang til sensitiv informasjon av Sopra Steria, må en taushetserklæring signeres.
Opplæring	Krav om god brukerveiledning	Det skal skrives god brukerveiledning som skal være med på å implementere systemet godt.

Tabell 4 Rammebetingelser

7 Kritiske suksessfaktorer

7.1 Suksessfaktorer

For at resultatet av prosjektet skal bli vellykket, har vi definert noen faktorer vi mener er avgjørende:

- Det er viktig med god kommunikasjon underveis. Mer om dette i punkt 7.2.
- Vi ønsker å ta i bruk Azure Sentinel og sette opp automatisering av enkle oppgaver. Dette er et resultatmål og vi anser det som kritisk at vi lykkes med dette for å svare på oppgaven og oppnå et vellykket resultat.
- Vi ønsker å ha opplæring i Azure Sentinel i form av god dokumentasjon underveis og i oppsett av playbooks og Azure Logic Apps. Dette vil gjøre det enklere for ansatte i bedriften å ta i bruk systemet selv.
- Vi ønsker også å utpeke enkelte ansatte i bedriften som skal få tilstrekkelig og omfattende kompetanse. Disse superbrukerne skal ivareta forvaltningen og driften av systemet videre.

7.2 Informasjonsbehov

Informasjonsbehovet underveis i prosjektet vil være forskjellig for de ulike interessentene. Grunnet koronasituasjonen er det vanskeligere å få til fysiske møter, men dette løses ved hjelp av Teams. Der har det blitt opprettet et Team for bacheloroppgaven, der oppdragsgiver og intern veileder er lagt til som medlemmer. Eksterne interessenter som sluttbrukeren og analytikere kommer vi på gruppa ikke til å ha noe særlig kontakt med, så disse er ikke lagt til i Teams. Alt av dokumenter vil lagres i et strukturert mappehierarki, der alle medlemmene har tilgang.

Internt i gruppa, kommer vi til å ha daglige møter på Teams der vi fordeler arbeidsoppgaver og diskuterer. Dette er viktig med tanke på at vi begge skal være oppdatert på hva som gjøres til enhver tid. Timeskjema og ukesrapporter vil føres slik at det vil være lettere å se tilbake på hva som har blitt gjort. Foreløpig vil også disse møtene foregå digitalt, men vi håper å kunne møtes fysisk for å jobbe bedre sammen.

Det vil bli ukentlig møte på fredager med oppdragsgiver. Agenda for møtene vil være en statusoppdatering og eventuelle spørsmål. På denne måten vil oppdragsgiver kunne følge utviklingen i prosjektet og komme med innspill til hva som er bra og hva som bør gjøres på en annen måte.

Veileder fra NTNU ønsker møte annenhver uke. Her også vil det være for å følge utviklingen i prosjektet og passe på at alt går bra.

De eksterne interessentene «sluttbrukerne» og «analytikere», er som nevnt ikke med i Teams-gruppa, så de vil ikke få tilgang på informasjonen som ligger der. De vil derimot få tilgang på dokumentasjon når prosjektet er sluttført. Det er oppdragsgivers ansvar å formidle dokumentasjon til de riktige interessentene etter behov.

8 Risikoanalyse

Naturligvis er det risiko knyttet til et prosjekt. Spesielt i et prosjekt hvor mye dreier seg om automatisering av tjenester. Automatiseringen bringer med seg flere goder, men kommer ikke uten risiko. Med automatisering blir alt som er gjentakende arbeid gjort til arbeid en datamaskin gjør for deg. Det som er problemet, er at datamaskinen kan gjøre feil. På det andre siden kan også mennesker gjøre feil. IT-sikkerhet er vanskelig, og vil alltid ha med seg en viss risiko for å bli hacket.

Azure Sentinel vil blokkere det den tror er trusler. Dette er en god ting, men om det blir for mye av det gode vil det komme falske-positive trusler. Dette fører til at bedriften kan risikere å miste arbeidskraft fordi de er stengt ute av datasystemene. Så det å finne et godt og balansert nivå på sikkerheten er viktig. Risikoen ved å ta i bruk Azure Sentinel er ikke spesielt stor med tanke på nedetid. Microsoft bruker mye ressurser på og siker god oppetid, og de garanterer til og med at Azure Sentinel skal være oppe 99,99% (Microsoft, 2021). 99,9% er mye oppetid og ikke minst vanskelig å oppnå på selv med egne løsninger.

Dette er risikoer man kan gå igjennom om prosjektet er en suksess, men for å komme til det punktet er det noen risikoer ved gjennomføring av prosjektet. Vi er i en periode hvor Covid 19 herjer og all form av jobbing må foregå på hjemmekontor. Dette gir ikke noe ekstra risiko direkte, men kan bidra til at sannsynligheten for fravær og mangel på motivasjon øker hos deltakerne. Det er en usikker tid i vente og man er enda ikke sikker på når ting vil gå tilbake til normalen.

Ettersom det er veiledere inn i bildet er det også en risiko knyttet til dem. Dette dreier seg om at det kan oppstå misforståelser eller uenigheter underveis i prosjektet. Vi vil forsøke å minke denne risikoen gjennom å ha ukentlige møter med veileder.

På samme måte som det kan oppstå uenigheter med veileder kan det også oppstå uenigheter i prosjektgruppen. Dette vil treffe spesielt hardt om det ikke er god kommunikasjon i utgangspunktet. Kommunikasjonen er naturligvis svekket grunnet at vi ikke kan møte fysisk og jobbe sammen.

Begge prosjektmedlemmene har gått samme linje på høyskole og vil bidra til at vi har relativ like kunnskapsnivå. Dette bringer oss inn på risikoen at vi mangler kunnskapen til å gjennomføre prosjektet. Naturligvis er dette et bachelorprosjekt hvor læring er viktig. Akkurat nå besitter vi ikke kunnskapen for å skrive bacheloroppgaven, men informasjonsinnhentingen vil være avgjørende for at vi skriver en god bacheloroppgave.

På samme måte som risikoen over kan man bli sittende for lenge på et problem. Vi har planer som skal fortelle oss om vi bruker for mye tid på et problem. Vi har veiledere som har presisert at om vi skulle havne i en slik situasjon er det bare å si ifra, så skal de stille med det de kan og eventuelt sette oss i kontakt med noen som vet en fiks på problemet.

Slik som risikoene er presentert er det liten oversikt. Dermed velger vi å ta med en tabell hvor vi forklarer sannsynligheten, konsekvensen, risikoverdien og eventuelle mottiltak. Risikoverdien regnes ut gjennom å ta sannsynlighet multiplisert med konsekvensen. Konsekvens og sannsynlighet vil få en verdi mellom 1-5 og vil gi risikoen en verdi mellom 1-25. Jo høyere risiko jo større tiltak må gjøres for å minke sannsynligheten eller konsekvensen mindre.

Risikomoment	Sannsynlighet	Konsekvens	Risikoverdi	Tiltak
Falske-positive svar på automatiseringen	2	2	4	Forsøke å finne balansen mellom god sikkerhet og slik at vi ikke kommer i veien for de ansatte i bedriften
Automatiseringen plukker ikke opp sikkerhetshendelser	4	5	20	Forsøke å gjøre automatiseringen god slik at den fanger opp og reagerer
Manglende kunnskap i prosjektgruppen	1	4	4	Planlegge med ekstra tid så man har ekstra tid til informasjonsinnhenting
Konflikt i prosjektgruppen	3	5	15	Holde daglige møter hvor, og om konflikt skal oppstå tar vi det med veileder og finner en løsning
Fraværende prosjektmedlemmer	2	5	10	Arbeide for en prosjektkultur der man tar vare på hverandre og kan snakke åpent sammen
Misforståelser mellom prosjektgruppen og veiledere	2	5	10	Ukentlige møter med statusoppdatering
Bli sittende fast på problemer	3	5	15	Ukentlige møter, med mulighet for å sende mail til veileder tvert man sitter fast

Tabell 5 Risikoanalyse

9 Kost/nytte-analyse

Hensikten med en kost/nytte-analyse er å avgjøre om nytten av prosjektet er verdt kostnadene ved å gjennomføre det. Det vil derfor være en viktig del av beslutningsgrunnlaget for om prosjektet anbefales gjennomført eller ikke. Nytten fordeles på kvantifiserbar nytte, ikke-kvantifiserbar nytte og bortfall av direkte kostnader som følge av prosjektet. Deretter har vi gjort et estimat for hva prosjektet kommer til å koste, etterfulgt av en sammenstilling mellom kostnadene og nytten med en vurdering av om nytten er verdt kostnadene.

9.1 Kvantifiserbar og ikke-kvantifiserbar nytte

Kvantifiserbar nytte er nytte vi tallfester og brukes i sammenligning mot kostnadene. Ikke-kvantifiserbar nytte kan ikke tallfestes, men gir fortsatt nytte. Denne typen nytte inngår ikke i sammenstillingen av kostnader og nytte.

Vi har satt følgende effektmål:

- *Redusere tiden en analytiker bruker på en sikkerhetshendelse*
- *Redusere IT-kostnader*
- *Øke IT-kompetanse*

De skal være utgangspunktet for å beregne nytten av prosjektet.

9.1.1 Kvantifiserbar nytte

For å enklere sette tall på nytten, kommer vi til å bruke antall timer som måleenhet i denne rapporten. Vi antar at en analytiker jobber 7 timer per dag, 24 dager i måneden og ser bort ifra ferier og regner med 12 måneder arbeidstid. En analytiker får i gjennomsnitt håndtert 4 hendelser i timen. Vi har som mål å automatisere slik at en analytiker får håndtert 5 hendelser i timen. Vi setter tallene i en tabell og får følgende:

Timer per dag:	7
Dager per måned:	24
Måneder per år:	12
Total timer per år:	2 016

Figur 3 Arbeidstimer

Tabellen under viser hvor mange hendelser som blir håndtert per time, dag, måned og år:

Time	Dag	Måned	År
4	28	672	8 064
5	35	840	10 080

Figur 4 Hendelser håndtert

Vi ser altså at ved å automatisere en ekstra hendelse i timen, vil det være mulig å håndtere 2 016 flere hendelser i løpet av året. Ved å dele antall håndterte hendelser på hendelser per time ender vi opp med antall timer. Regnestykket for å finne antall timer spart blir slik:

$$\frac{2016\ h}{4\ h/t} = 504\ t$$

Det vil altså være mulig å spare 504 timer i året ved å øke antall håndterte hendelser i timen med 1. Vi antar heretter at antall timer spart per år per ansatt er lik 504, selv om dette nødvendigvis ikke vil stemme i bedriften. Analytikere jobber med forskjellige hendelser, så dette er bare en antagelse vi gjør.

9.1.2 Ikke-kvantifiserbar nytte

Det er vanskelig å sette tall på den ikke-kvantifiserbare nytten, men den er likevel avgjørende når man skal avgjøre om prosjektet skal gjennomføres eller ikke. I dette prosjektet vil den ikke-kvantifiserbare nytten være minst like viktig som den kvantifiserbare. Viktige punkter her er:

- Automatisering vil gjøre at analytikere slipper å bruke tid på dagligdagse sikkerhetshendelser. Dette tærer på moralen til de ansatte, og ved å automatisere håndteringen av disse hendelsenene, får de ansatte mulighet til å jobbe med mer spennende sikkerhetshendelser.
- Økt IT-kompetanse ved at analytikerne får jobbe med flere krevende sikkerhetshendelser. Man utvikler ikke kompetansen av å håndtere de samme hendelsene (som passord-reset) om igjen.
- Muligheten for å skalere i fremtiden er større dersom man tar i bruk SOAR.

9.2 Bortfall av direkte kostnader

I punktet om kvantifiserbar nytte, kom vi frem til at det var mulig for en analytiker å håndtere 2 016 flere hendelser i året. Det vil derfor være mulig å la være å ansette nye analytikere i fremtiden, da man allerede har kapasitet til å håndtere flere sikkerhetshendelser med de ansatte som er tilgjengelig. Vi ender derfor ikke opp med noen tall på kostnader som faller bort ved å implementere løsningen vår.

9.3 Estimerte kostnader

De estimerte kostnadene for prosjektet vil være utviklingskostnadene til prosjektgruppen. Det er beregnet 500 timer (± 5%) arbeid i forbindelse med bacheloroppgaven. Med to gruppemedlemmer, vil kostnadene ligge på rundt 1000 timer.

Vi ønsker å ta i bruk demo-miljøet til Azure. Dette er i utgangspunktet gratis å ta i bruk, med en prøveperiode på 30 dager. Her vil vi være i dialog med den andre bachelorgruppen som skal jobbe med EDR og IDS, da det vil være naturlig å samarbeide i dette demo-miljøet. Vi har ordnet en Microsoft 365 A5 lisens som vil gi oss den tilgangen vi trenger til demo-miljøet. Denne kostnaden dekkes av NTNU og inngår derfor ikke i denne analysen.

Sopra Steria tar allerede i bruk Azure Sentinel, så det vil ikke forekomme noen ytterligere kostnader ved å implementere automatisering. Det vil altså ikke være noen ekstra driftskostnader, og eneste kostnad blir dermed utviklingskostnadene.

9.4 Sammenstilling kost/nytte

Som følge av tallene i punktene over, ender vi opp med sammenstillingen vist under:

	År 1	År 2	År 3	År 4	År 5	Sum
Kvantifiserbar nytte	-	504	504	504	504	2 016
Bortfall kostnader	-	-	-	-	-	-
Sum nytte	-	504	504	504	504	2 016
Utviklingskostnader	1 000	-	-	-	-	1 000
Driftskostnader	-	-	-	-	-	-
Sum kostnader	1 000	-	-	-	-	1 000
Beregnet nytte	- 1 000	504	504	504	504	1 016

Figur 5 Sammenstilling kost/nytte

I punkt 9.1.1 kom vi frem til at den kvantifiserbare nytten er 504 timer spart i året, per ansatt. Vi antar dermed at antall timer spart i året vil være en del høyere enn det som står i tabellen over. Det viktigste her er å vise at det vil gi nytte for bedriften.

10 Retningslinjer og standarder

I dette kapitlet skal vi kortfattet ta med de retningslinjene og standardene som prosjektet må forholde seg til. Dette innebærer kravene til dokumentasjon som skal produseres i løpet av prosjektet, kravene til kvalitetsgjennomganger, standarder, metoder og endringshåndtering.

10.1 Krav til dokumentasjon

Dokumentasjonen skal ikke foreligge før enden av prosjektet som er 20.05.21. Det er altså ingen frister på når de forskjellige dokumentene må være ferdig, men vi har satt egne frister som er vist i kapittel 4.6. Alle dokumentene levers i PDF format. Når dette prosjektet er ferdig er det en del rapporter som skal foreligge. Rapportene som skal foreligge er:

- Forstudierapport
- Designrapport
- Driftsrapport
- Sluttrapport
- Egenvurdering av arbeidet

Begge i prosjektgruppen har ansvar for å sikre at rapportene skal foreligge til riktig dato.

10.2 Krav til kvalitetsgjennomganger

Alt av dokumenter og rapporter skal gjennomgå kvalitetsgjennomganger. Dette kan gjøres både med og uten veileder. Veiledere har uttrykket at de ønsker å få oppgitt sidetall slik at de kan gjennomgå om det som står der er korrekt. Resten av gjennomgangen vil foregå internt i prosjektgruppen. Det vil bli holdt møter på enden av hver rapport hvor vi går igjennom kommentarene og retter skrivefeil.

Senere i prosjektet vil det gjøres andre ting enn å kun skrive rapporter. Da vil det bli holdt møter hvor en gjennomgang over standarder og regler for navngiving. Her vil vi også debattere rundt oppsett. Bruksanvisninger til kunder og/eller ansatte vil bli fremstilt. Det vil gjøre det enklere i opplæringen av ansatte og vil ha samme krev til kvalitetsgjennomganger.

10.3 Krav til standarder og metoder

For at prosjektet skal være konsekvent er det viktig at man innarbeider noen standarder. Standardene vil være med gjennom hele prosjektet. Uten standarder og metoder er det en god mulighet for at det oppstår rot og misforståelser. Det vil også føre til at det blir brukt mere tid på unødvendig leting eller prøve å forstå den andre prosjektpartneren sine standarder.

Når det kommer til verktøy, er det viktig at alle deltakere bruker de samme. Slik blir maler opprettholdt på riktig måte. Prosjektgruppen vil blant annet benytte Microsoft Word som hovedplass for samskriving. Sammen med dette vil vi bruke MS Project til planlegging av prosjektet. Også bruker vi Excel til å føre timer arbeidet.

Tabellen under viser en oversikt over hvilke standarder og verktøy vi kommer til å benytte i prosjektet. Hver standard og verktøy har med en kort beskrivelse.

Type	Gjelder	Beskrivelse
Standarder		
Brukernavn	Brukernavn på felles brukere	For å ha et felles brukernavn på felles brukere. Prosjekt gruppen skal bli enig, og opprette disse i fellesskap
Passord	Passord som skal benyttes på felles brukere	Passordene skal være på minst 8 karakterer, inneholde minst 1 stor og 1 liten bokstav, et nummer og et spesialtegn
Maler	Dokumenter, tabeller, struktur	Alle dokumenter skal følge NTNUs maler for prosjektdokumentasjon, gruppen har satt opp egne maler for overskriver, tabeller og lignende
Verktøy		
Office365	Word, Excel, MS Project, Visio	Office 365 vil være verktøyene vi bruker for å produsere dokumenter knyttet prosjektet
Teams	Samarbeidsplattform	Vi vil benytte Teams som kommunikasjonsplattform, både for chat og videomøter. Vi vil også benytte Teams som en plass og lagre dokumenter slik at de er tilgjengelig for alle prosjekt medlemmer.
Azure	Virtuelle maskiner og Sentinel	Vi kommer til å bruke Azure systemer til testing av systemer. Vi ønsker å få brukt Azure Demos, men det er enda ikke avklart. Deretter dreier hele prosjektet rundt Azure Sentinel.

Tabell 6 Standarder og metoder

10.4 Endringshåndtering

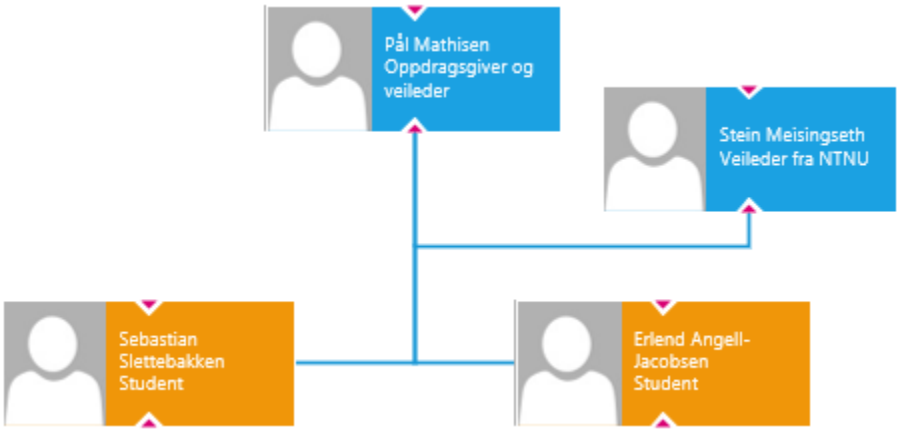
Siden endringer i planer og oppgaven vil ha store konsekvenser for hele prosjektet, er det viktig at alle som er involvert i prosjektet er enig. Arbeidsgruppen vil være de første som tar opp behovet for endring seg imellom. Om det er enighet vil man senere gå videre til å se på hvilken nytte dette vil ha for prosjektet. Om vi finner at det er nok nytte for å gå videre med endringen vil vi ta kontakt med begge veiledere og presenter det vi fant. Om det er så enighet i gruppen vil arbeidsgruppen gå videre med endringen og logge endringen. Etter endringen er logget vil planen bli endret i henhold til endringen. Under ser du en liste over hvordan et handlingsforløp vil foregå ved en endring.

1. Dokumenter endringens innhold
2. Analyser konsekvensene for prosjektet
3. Beregne eventuell kost/nytte
4. Godkjennelse og aksept, Informer interessentene
5. Logg endringen
6. Juster planene
7. Gjennomfør endringen

11 Prosjektorganisering

Prosjektorganiseringen for dette bachelorprosjektet er nokså enkel. Vi er totalt 4 personer involvert. Vi har en arbeidsgruppe på 2 studenter som går ved NTNU på linjen Informatikk og drift av datasystemer. Deretter har vi Stein Meisingseth som er veileder fra NTNU, og har lang erfaring med veiledning av bachelorprosjekt. Fra markedets side har vi Pål Mathisen som vil veilede med sin kunnskap. Han er også oppdragsgiver.

Rollen som prosjektleder vil ikke være så tydelig. Dette kommer av at arbeidsgruppen er kun to personer, med relativt lik utdanning. Vi kommer til å bytte på hvem som er møteleder og referent, men underveis i jobbingen vil det ikke være noen tydelig prosjektleder.



Figur 6 Prosjektets deltakere

12 Anbefaling om videre arbeid

Vi anbefaler at prosjektet fortsetter slik at den blir gjennomført med de samme målene som er satt i kapittel 4. Dette kommer av at vi i prosjekt gruppen mener det er mye nytte som kan skaffes gjennom å oppgradere nåværende systemet. Prosjektet vil skape en bedre arbeidshverdag for analytikere. For ledelsen vil dette bety at analytikere får brynet seg på vanskeligere sikkerhetstiltak, noe som kan bringe mere variasjon i hverdagen for en analytiker. For ledelsen er det viktig at analytikere trives i jobben.

En annen effekt prosjektet har er at et automatisert system vil ha raskere responstid ved en sikkerhetshendelse. Dette vil minimere tiden en for eksempel hacker kan ha for å kryptere disker. Prosjektet kommer ikke uten risiko og det er dermed viktig at vi er forsiktige med å anbefale en løsning vi ikke vet er sikker.

13 Kilder

Kaplan, D. (2020, April 7). *9 Security Orchestration and Automation Benefits: How SOAR Helps Improve Incident Response*. Hentet fra Siemplify: https://www.siemplify.co/blog/security-orchestration-automation-response-benefits/

Microsoft. (2021, Januar 26). *Microsoft.com*. Hentet fra Azure: https://azure.microsoft.com/en-us/overview/what-is-azure/?ef_id=Cj0KCQiAmL-ABhDFARIsAKywVadFTP_3HVLXT8HAWxl6JcBgOT6GXNQYqPHb46rolyb_K0DosegJrtAaAnadEALw_wcB%3AG%3As&OCID=AID2100088_SEM_Cj0KCQiAmL-ABhDFARIsAKywVadFTP_3HVLXT8HAWxl6JcBgOT6GXNQYqPHb46rolyb_K0

Microsoft. (2021, Januar 27). *Microsoft.com*. Hentet fra Azure: https://azure.microsoft.com/nb-no/services/azure-sentinel/#faq

Microsoft. (2021, Januar 20). *Priser på Azure Sentinel*. Hentet fra Microsoft Azure: https://azure.microsoft.com/nb-no/pricing/details/azure-sentinel/

Microsoft. (2021, Januar 19). *Service Level Agreements*. Hentet fra Microsoft Azure: https://azure.microsoft.com/en-us/support/legal/sla/log-analytics/v1_3/

Gruppe 45

Bachelorprosjekt

SOAR i Azure Sentinel

Designrapport

Versjon 1.0

Forfattere: *Erlend Angell-Jacobsen, Sebastian Slettebakken*

Revisjonshistorie

Dato	Versjon	Beskrivelse	Forfatter
12/01/2021	1.0	Første utkast	Erlend Angell-Jacobsen, Sebastian Slettebakken

Innholdsfortegnelse

1 Innledning

Dette dokumentet beskriver prosjektgruppens løsningsforslag til Sopra Steria gjennom automatisering i Azure Sentinel. Designrapporten er skrevet i forbindelse med planlegging av prosjektet. Rapporten er en fortsettelse av forstudierapporten, som var forrige rapport som ble skrevet.

Prosjektgruppen foretar ingen endringer fra forstudierapporten.

Dokumentet er delt opp på følgende måte:

Kapittel 1: Introduksjon til dokumentet og hva det dekker. Det inneholder også en kortfattet beskrivelse av kundens behov og eventuelle definisjoner og forkortelser brukt i dokumentet.

Kapittel 2: Generell beskrivelse av valgte produkter og begrunnelser hvorfor vi valgte dem.

Kapittel 3: Detaljert løsningsbeskrivelse av de tekniske løsningene. Dette innebærer følgende punkter:

- Beskrivelse av bedriftens nåværende datasystem
- Deloppgaver som må løses for å kunne fortsette med prosjektet
- Forutsetninger for at systemet skal fungere optimalt
- Beskrivelser om hvordan vi vil ta i bruk Azure Sentinel
- System og sikkerhetskrav for det nye foreslåtte systemet
- Kort beskrivelse av pilotprosjektet

1.1 Avgrensning

Dette dokumentet dekker løsningsforslaget for oppsett i Azure Sentinel. Vi har tidligere avgrenset oppgaven i forstudierapporten. Det blir ingen videre avgrensninger i denne rapporten.

1.2 Oppdragsgivers behov

Sopra Steria utvikler kontinuerlig sikkerhets-kapabilitetene de bruker til å levere tjenestene sine. Å redusere manuelt arbeid for sikkerhetsanalytikere, øke presisjon på leveransene og minimere responstiden på sikkerhetssaker er tre viktige mål i utvikling. Vi anser det derfor som et behov at så mye av det manuelle og enkle arbeidet automatiseres, så langt det lar seg gjøre. Dette behovet må ikke gå på bekostning av den eksisterende kvaliteten.

Som nevnt i forstudierapporten, ønsker vi å dekke bedriftens behov for:

- Raskere responstid
- Optimalisert trusselinformasjon
- Redusere manuelt arbeid
- Gjøre det lettere å integrere flere tjenester ved behov
- Kutte kostnader

1.3 Definisjoner og forkortelser

Dette underkapitlet dreier seg om å vise en oversikt over forkortelser og eventuelle ord, som kan oppfattes vanskelig å forstå.

SIEM: er en forkortelse for Security Information and Event Management. SIEM-systemer funker slik at de sammenligner din data med andre kilder. Ser etter ting som ikke er normalt. Når systemet finner noe genereres det en alarm. Senere kan SIEM systemet instruere et SOAR system hva som skal gjøres med denne alarmen. (Petters, 2021)

SOAR: er en forkortelse for Security Orchestration, Automation and Response. Har man et SIEM-system er det systemet bra til å generere alarmer. SOAR systemet tar sikkerheten et skritt videre og kan automatiser hva som er responsen alarmen. For eksempel kan et SOAR-system sende mail til en bruker som har prøvd for mange passord. Slik kan brukeren selv tilbakestille passordet uten innblanding av en it-ansatt. (TechTarget, 2021)

Azure Sentinel: Microsoft sitt SIEM- og SOAR-system i ett. Den er i skyen og vil gi et fugle-perspektiv over bedriften. Azure Sentinel driver med innsamling av data, finner sikkerhetshendelser, undersøker hendelsene og til slutt har muligheten til å respondere på hendelsen. (Microsoft, 2021)

Playbooks: er en samling av sikkerhetsprosedyrer man kan kjøre i responsen av en alarm. Playbooks er basert på Azure Logic Apps. (Cheah, 2021)

Azure Logic apps: Dette er apper som kan brukes i Playbooks for å skape effektivisering eller full automatisering av sikkerhetshendelser. Akkurat hvilke Logic Apps man har tilgjengelig kommer an på hvilket abonnement man har. (Cheah, 2021)

KQL: står for Keyword Query Language og brukes som spørrespråk i Microsoft sine tjenester, som SharePoint og Azure Data Explorer. (Microsoft, 2019)

SLA: står for Service Level Agreement og er en avtale mellom tjenesteleverandør og en klient. Den innebærer kvalitet, oppetid og funksjonalitet på tjenesten som tilbys. Norsk oversettelse er tjenestenivåavtale eller en avtale om tjenestekvalitet.

1.4 Hvorfor valg av løsning

Her skal vi ta for oss en generell beskrivelse for så å se på kostnadene ved endret infrastruktur og påvirkning av forretningsbehov. Dette har vi gått gjennom i forstudierapporten, og vi refererer dermed til den.

2 Tekniske løsninger

Den tekniske løsningen vil gi en generell beskrivelse over hvordan vi tenker å løse den tekniske siden av prosjektet. Løsningen innebærer at vi tar i bruk Microsoft Azure sitt demo-miljø og Azure Sentinel. Vi har ordnet Microsoft 365 A5 lisens for å få tilgang til demo-miljøet. I dette miljøet skal vi:

- Sette opp tenants og Azure AD
- Få overblikk over sikkerhetshendelser (Threat Intelligence) med Azure Sentinel
- Automatisere håndteringen av noen av de enkle sikkerhetshendelsene i Sentinel ved bruk av Azure Logic Apps, Playbooks og KQL
- Teste treffsikkerheten til automatiseringen slik at vi ikke automatiserer feil. Dette kan i så fall virke mot sin hensikt.

Akkurat hvilke sikkerhetshendelser vi skal håndtere automatisk kommer vi tilbake til i driftsrapporten. Vi må også definere noen krav til sikkerhet, system og testkriterier. Et annet viktig moment er avhengigheten til Azure og demo-miljøet. Uten dette, vil det være vanskelig å utføre prosjektet.

3 Detaljert løsningsbeskrivelse

3.1 Eksisterende system

Sopra Steria benytter i dag mange systemer for sin drift av sikkerhet. Både til seg selv og til sine kunder. Sopra Steria bruker allerede Azure Sentinel. Noe de ønsker å forbedre med dagens system er bruken av automatisering. Spesielt på sikkerhetshendelser som oppfattes som gjentakende for analytikere. Sammen med full automatisering er også effektivisering en del som ønskes og få forbedret. Automatisering og effektivisering av enkle og gjentakende sikkerhetshendelser er viktig for å holde jobben til analytikere fersk og variert.

3.2 Deloppgaver som må løses

Første deloppgave som må gjøres, er å kartlegge hvilke hendelser Sopra Steria ønsker å automatisere. Det er ennå ikke kjent hvilke sikkerhetshendelser som ønskes at vi ser på. Om vi ikke får noe innsyn i hvilke hendelser de vil automatisere, vil det kreve at vi enten skriver under på en taushetserklæring slik at vi får tilgang, eller at vi automatiserer enkle hendelser som vi mener det kommer mange av.

Andre deloppgave går ut på at prosjektgruppen må tilegne seg kunnskap om hvordan man skriver KQL. Slik at vi kan begynne å automatisere og effektivisere gjennom Playbooks.

Tredje deloppgave er å sette opp et demomiljø slik at vi får testet våre løsninger. Oppsettet skal bli satt opp i Azure Demo Environment. Da kan vi simulere ulike sikkerhetshendelser for å se om vårt oppsett fungerer som planlagt.

Den fjerde deloppgaven går ut på å teste våre løsninger. For at Sopra Steria kan ta i bruk vår løsning er det viktig at vi har statistikk som underbygger vår løsning. Sopra Steria er nødt til å være sikker på at det som vi har kommet frem til har god treffsikkerhet, og ikke virker mot sin hensikt.

3.3 Overordnet sammenheng

Dette kapitlet skal vise hvordan dette prosjektet kommer til å endre/passe inn i dagens system. Den dag i dag bruker Spora Steria flere systemer for å overvåke sikkerheten hos kunder. Akkurat hvilke systemer de bruker er ukjent, bortsett fra Azure Sentinel. Dagens systemer vil ikke forandres mye. Prosjektet vil fokusere på effektivisering og automatisering av Azure Sentinel. Dermed kommer vi ikke til å endre på noen andre systemer. Fokuset kommer til å ligge på Azure Sentinel og dets funksjoner.

Våre forbedringer på systemet skal frigi tid hos analytikere, gjennom å ta bort noen av de enkle og gjentakende oppgavene. Disse forbedringene vil være med på å hindre at analytiker blir overveldet av sikkerhetshendelser. Dette minker sannsynligheten for at noen sikkerhetshendelser blir ubehandlet. I verste fall vil en ubehandlet hendelse ende med et vellykket innbrudd.

På samme måte vil automatiseringen kunne gjøre feil og slippe inn en hacker. Det vil være katastrofalt, men om treffsikkerheten er god på automatiseringen vil omstillingen hos analytikere være verdt implementasjonen.

3.4 Forutsetninger og avhengigheter

For å implementere vår løsning på en hensiktsmessig måte, er det viktig at alle som skal bruke systemet får den opplæringen de trenger. Sikkerhetsanalytikere hos Sopra Steria tar allerede i bruk Azure Sentinel i dag, så det vil ikke være et helt nytt system å forholde seg til. Det de derimot må gjøre, er å bli kjent med automatiseringen. Det vil derfor være nødvendig med opplæring for at analytikerne kan videre automatisere nye, fremtidige sikkerhetshendelser etter prosjektets gjennomføring. Opplæring vil være i form av en driftsrapport med dokumentasjon på hvordan vi setter opp automatiseringen.

En annen forutsetning for prosjektets gjennomføring vil være at Microsoft og Azure Sentinel leverer tjenesten som de sier de skal. Høy oppetid og god trussel identifikasjon er essensielt for et sikkerhetssystem. Microsoft har en SLA for Log Analytics der de garanterer at Log Analytics tilgjengelighet/oppetid ikke vil falle under 99,9% (Microsoft, 2018).

3.5 Organisatoriske og personellmessige konsekvenser

Dårlig organisering av et prosjekt kan medføre konsekvenser som ekstra kostnader, forsinket leveranse eller dårligere kvalitet på sluttproduktet. For å unngå dette, må man fra starten av ha det klart for seg hvordan prosjektet er organisert og hva som skal gjøres ved eventuelle endringer. Dette står beskrevet i forstudierapporten.

Når det gjelder personellmessige konsekvenser, kan dette prosjektet påvirke sikkerhetsanalytikere hos Sopra Steria SOC både positivt og negativt. Økt automatisering av håndtering av dagligdagse sikkerhetshendelser vil lette på hverdagen til analytikerne. Dette vil være med på å bedre arbeidsmoralen til de ansatte, som nå får muligheten til å håndtere de mer avanserte hendelsene.

Automatisert håndtering av sikkerhetshendelser kan sees på som en multiplikasjonsfaktor. Dersom man automatiserer feil, vil antall sikkerhetshendelser som blir håndtert feil øke betydelig. Dersom man automatiserer riktig, vil antall sikkerhetshendelser som blir håndtert riktig øke.

Dette medfører at dersom automatisering ikke var gjort bra nok i utgangspunktet, ville analytikerne endt opp med mange feilhåndterte hendelser. De må da bruke tid på å manuelt korrigere opp i feil, hvis det i det hele tatt er mulig.

3.6 Krav til sikkerhet og system

I forbindelse med dette systemet er det ikke store krav om system maskinvare. Dette kommer av at hele Azure Sentinel driftes i skyen hos Microsoft. Azure Sentinel systemet styres enkelt gjennom en nettleser. Dermed stiller det ikke store krav på PC-en til analytikere. Det kreves derimot en god nettverkstilkobling med god oppetid.

Ettersom Azure Sentinel er et sikkerhetsverktøy er det ekstra viktig at sikkerheten rundt systemet er godt. Det bør være ytterst få Admin-brukere, som kan endre på systemet. Analytikere har tilgang på systemet, men kan ikke gjøre endringer på systemet. Dette kan endres og settes i Azure RBAC (role-based access control). (Microsoft, 2021) Dette kan være med på å minke skadeomfanget om en analytiker-bruker blir kompromittert.

Programvare

Vi kommer til å ta i bruk litt forskjellig programvare og tjenester fra Azure.

3.6.1 Azure Sentinel

I forstudierapporten har vi allerede vært inne på hva Azure Sentinel er. I Sentinel kommer vi til å opprette Playbooks med Azure Logic Apps. Vi kommer til å ta i bruk Azure Security Center med Microsoft Defender for Endpoint for å oppdage hendelser og mate de til Sentinel. Vi har også planer om å legge til VirusTotal (V3 API) som en ekstern tjeneste i Sentinel, men dette vil være lavere prioritert enn å automatisere hendelser.

En mulig sikkerhetshendelse vi kan ta for oss er håndtering av impossible travel. Impossible travel inngår i Microsoft Cloud App Security. Det går ut på å oppdage innlogginger fra ulike steder på kort tid som fysisk sett ikke ville vært mulig. For eksempel dersom man logger inn fra bedriftens lokaler i Oslo og 3 timer senere får en innlogging fra Beijing. Det er altså en reise som ikke er mulig å foreta seg på den korte tiden, og systemet kan nekte adgang fra Beijing og sender et sikkerhetsvarsel. Måten vi kan automatisere håndteringen på, er ved å automatisk sende en mail til brukeren det gjelder med forespørsel om å endre passord. Dersom brukeren ikke trykker ja i mailen og endrer passordet sitt innen en time, kan man endre passordet automatisk til noe annet eller eventuelt deaktivere brukeren midlertidig. IT-avdelingen må selvsagt også varsles, og dette kan gjøres i en Teams-kanal eller på mail (ticketing system).

3.6.2 Azure Security Center

Vi kommer til å benytte Azure Security Center for å samle inn alarmer og sikkerhetshendelser. Security Center vil raskt hjelpe oss med å sette opp en relativt god sikkerhet rundt systemene vi ønsker å beskytte. Det er ikke der vårt fokusområde ligger, men det må gjennomføres slik at vi får tilgang på sikkerhetshendelser for Azure Sentinel. Azure Security Center bringer også med seg Security recommendations. Her får man anbefalinger om hvilke sikkerhetstiltak som bør gjøres. Underveis får du en sikkerhets-score som viser hvor sikkert systemet er. Sikkerheten blir sammenlignet med hvilke sikkerhetstiltak Microsoft mener er nødvendige for å oppnå er sikkert system.

Man kan fort tenke at Azure Security Center er lik Azure Sentinel, og det er de også. De er derimot også laget for å driftes skulder mot skulder. Sentinel gjør mye av det Security Center gjør, men den vil ikke

hjelpe deg med å gi innsikt i hvilke sikkerhetstiltak, som man burde gjøre. Den største fordelen gjennom å bruke Security Center er at den vil være med å styrke gjennom å gi råd om hva du burde gjøre. Så kan man mate alarmer ut av Security Center og inn i Sentinel slik at man får samlet at på en plass.

3.7 Regler for Pilot – hva må til for å gå videre

For at piloten skal ha noen hensikt er det viktig at vi har noen regler og krav. Dette er for at vi skal vite når systemet er godt nok til å vise som forslag til Sopra Steria. Vi har en representant fra Sopra Steria i prosjektgruppen, som vil bidra med å skaffe tall som kreves for at de skal vurdere løsningen. I pilotprosjektet vil systemet rulles ut og testes av prosjektgruppen. Testingen vil foregå gjennom at man ruller systemet ut i Azure Demos og deretter setter i gang ulike sikkerhetshendelser. Mulig at vi benytter Azure Defender for å skape alarmer som ikke er ekte. Eller så blir vi nødt til å skape disse sikkerhetshendelsene selv.

Plan for piloten:

1. Gjøre seg kjent med systemet og dets muligheter
2. Systemet implementeres i et isolert miljø
3. Testing og føring av statistikk på systemet
4. Evaluering av systemet
5. Gjøre eventuelle endringer og teste på nytt
6. Prosjektgruppen gjør endringer som er nødvendige ifølge Sopra Steria

Når prosjektet er ferdig må det oppfylle noen krav som skal avgjøre om prosjektet er godkjent eller ikke.

Overordnede godkjenningskriterier:

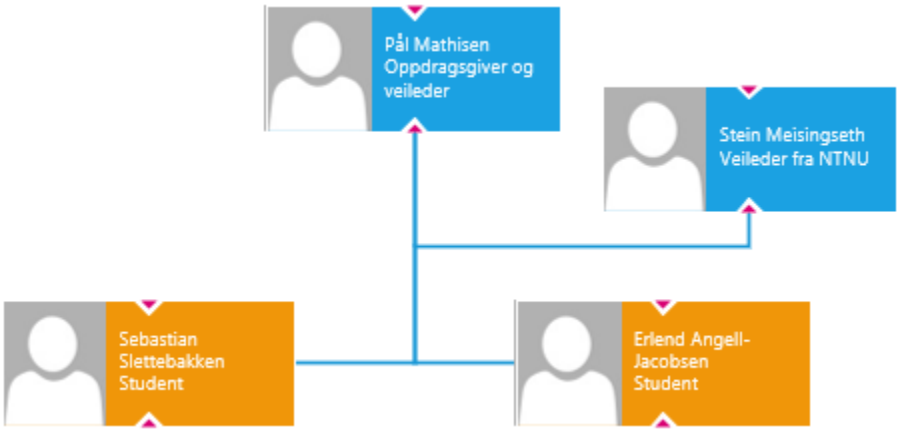
- Systemet skal i noen grad automatisere enkle og repetitive sikkerhetshendelser
- Systemet må ha høy oppetid. Mye av dette ligger hos Microsoft, så er ikke noe vi kan gjøre om Microsoft ikke klarer å holde oppetiden sin på 99.99%. Men om oppetiden blir et problem kan ikke prosjektet godkjennes
- Systemet må i noen grad effektivisere arbeidshverdagen hos analytikere
- Systemet kan ikke gjøre feil for ofte. Systemet må ha en treffsikkerhet som lever opp til Sopra Sterias standard

Konsekvenser ved ikke-godkjent pilotprosjekt

Om ikke alle godkjenningskriteriene er overholdt kan man risikere at man ikke får gjennomført prosjektet på ønsket måte. Da må det gjøres endringer etter som hva som er problemet. Prosjektgruppen tar tiltak underveis i prosjektet for å sikre at systemet opprettholder kravene. I ytterste konsekvens av at kravene ikke blir overholdt kan systemet ikke benyttes av Sopra Steria.

3.8 Deltagere

Deltagerne og prosjektorganiseringen for bachelorprosjektet gikk vi gjennom i forstudierapporten. Siden vi ikke har gjort noen endringer her, vil deltagerne være som følger:

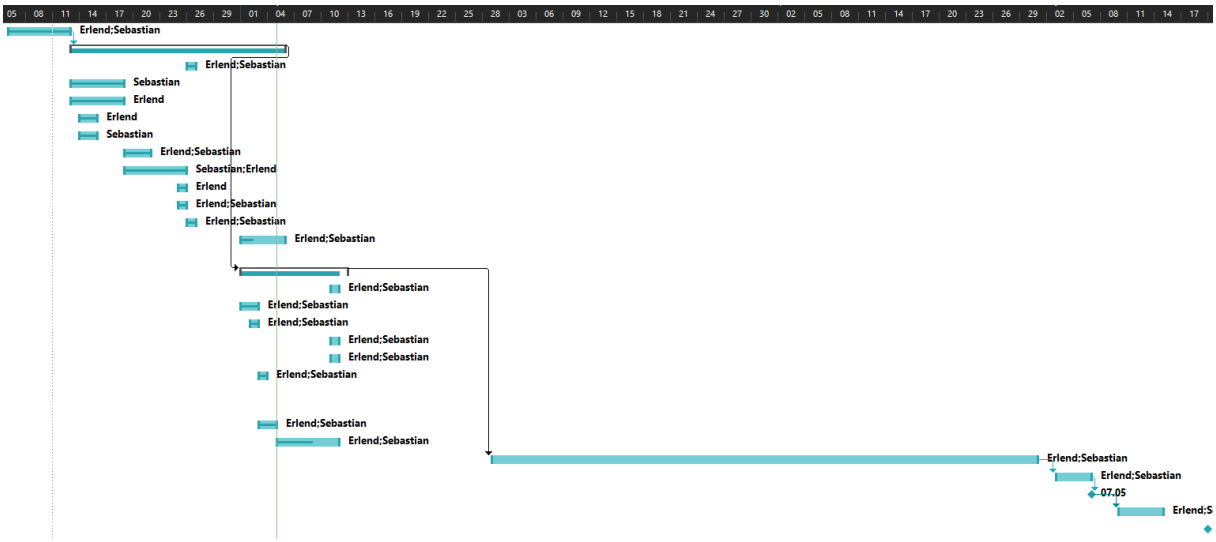


3.9 Oppdatert prosjektplan

Den oppdaterte prosjektplanen ser slik ut:

	Task Mode	Task Name	Duration	Start	Finish	Prede	Resource Names
1		Planlegging	5 days	Wed 06.01.21	Tue 12.01.21		Erlend;Sebastian
2		**Forstudierapport**	**18 days?**	**Wed 13.01.21**	**Fri 05.02.21**	**1**	**Erlend;Sebastian**
3		Kap. 1 - Introduksjon	1 day	Tue 26.01.21	Tue 26.01.21		Erlend;Sebastian
4		Kap. 2 - Bakgrunn	4 days	Wed 13.01.21	Mon 18.01.21		Sebastian
5		Kap. 3 - Prosjektmål	4 days	Wed 13.01.21	Mon 18.01.21		Erlend
6		Kap. 4 - Interessenter	2 days	Thu 14.01.21	Fri 15.01.21		Erlend
7		Kap. 5 - Suksessfaktorer	2 days	Thu 14.01.21	Fri 15.01.21		Sebastian
8		Kap. 6 - Risikoanalyse	3 days	Tue 19.01.21	Thu 21.01.21		Erlend;Sebastian
9		Kap. 7 - Kost/nytte	5 days	Tue 19.01.21	Mon 25.01.21		Sebastian;Erlend
10		Kap. 8 - Retningslinjer	1 day	Mon 25.01.21	Mon 25.01.21		Erlend
11		Kap. 9 - Organisering	1 day	Mon 25.01.21	Mon 25.01.21		Erlend;Sebastian
12		Kap. 10 - Anbefaling	1 day	Tue 26.01.21	Tue 26.01.21		Erlend;Sebastian
13		Gjennomgang av dokumentet	5 days	Mon 01.02.21	Fri 05.02.21		Erlend;Sebastian
14		**Designrapport**	**10 days?**	**Mon 01.02.21**	**Fri 12.02.21**	**2**	**Erlend;Sebastian**
15		Innledning	1 day	Thu 11.02.21	Thu 11.02.21		Erlend;Sebastian
16		Dokumentets hensikt	2 days	Mon 01.02.21	Tue 02.02.21		Erlend;Sebastian
17		Avgrensning	1 day	Tue 02.02.21	Tue 02.02.21		Erlend;Sebastian
18		Referanser	1 day	Thu 11.02.21	Thu 11.02.21		Erlend;Sebastian
19		Oversikt over dokumentet	1 day	Thu 11.02.21	Thu 11.02.21		Erlend;Sebastian
20		Kunden og behov	1 day	Wed 03.02.21	Wed 03.02.21		Erlend;Sebastian
21		~~Valg av produkt teknologi løsning~~					~~Erlend;Sebastian~~
22		Beskrivelse av teknisk løsı	2 days	Wed 03.02.21	Thu 04.02.21		Erlend;Sebastian
23		Detaljert løsningsbeskrive	5 days	Fri 05.02.21	Thu 11.02.21		Erlend;Sebastian
24		Driftsrapport	45 days	Mon 01.03.21	Fri 30.04.21	14	Erlend;Sebastian
25		Sluttrapport	4 days	Mon 03.05.21	Thu 06.05.21	24	Erlend;Sebastian
26		Egenvurdering	0 days	Fri 07.05.21	Fri 07.05.21	25	Erlend;Sebastian
27		Presentasjon	5 days	Mon 10.05.21	Fri 14.05.21	26	Erlend;Sebastian
28		Endelig frist	0 days	Thu 20.05.21	Thu 20.05.21		

Oversikt over tidslinjen blir da seende slik ut:


05 08 11 14 17 20 23 26 29 01 04 07 10 13 16 19 22 25 28 03 06 09 12 15 18 21 24 27 30 02 05 08 11 14 17 20 23 26 29 02 05 08 11 14 17
Erlend;Sebastian
Erlend;Sebastian
Sebastian
Erlend
Erlend
Sebastian
Erlend;Sebastian
Sebastian;Erlend
Erlend
Erlend;Sebastian
Erlend;Sebastian
Erlend;Sebastian
Erlend;Sebastian
Erlend;Sebastian
Erlend;Sebastian
Erlend;Sebastian
Erlend;Sebastian
Erlend;Sebastian
Erlend;Sebastian
Erlend;Sebastian
Erlend;Sebastian
Erlend;Sebastian
07.05
Erlend;S

4 Referanser

Angell-Jacobsen, E., & Slettebakken, S. (2021). *Forstudierapport.* Trondheim: NTNU.

Cheah, E. S. (2021, Februar 3). *Security Playbooks in Azure Sentinel*. Hentet fra Dev.to: https://dev.to/cheahengsoon/security-playbook-in-azure-sentinel-3lo6

Microsoft. (2018, Oktober). *SLA for Log Analytics*. Hentet fra Microsoft Azure: https://azure.microsoft.com/en-us/support/legal/sla/log-analytics/v1_3/

Microsoft. (2019, Juli 22). *Keyword Query Language (KQL) syntax reference*. Hentet fra Microsoft Docs: https://docs.microsoft.com/en-us/sharepoint/dev/general-development/keyword-query-language-kql-syntax-reference

Microsoft. (2021, Februar 3). *Azure Sentinel*. Hentet fra Microsoft: https://docs.microsoft.com/en-us/azure/sentinel/overview

Microsoft. (2021, Februar 5). *Docs*. Hentet fra RBAC: https://docs.microsoft.com/en-us/azure/sentinel/roles

Petters, J. (2021, Februar 3). *What is SIEM? A beginner's guide*. Hentet fra Varonis: https://www.varonis.com/blog/what-is-siem/

TechTarget. (2021, Februar 3). *SOAR*. Hentet fra SearchSecurity: https://searchsecurity.techtarget.com/definition/SOAR

Gruppe 45

Bachelorprosjekt

SOAR i Azure Sentinel

Driftsrapport

Versjon 1.0

Forfattere: *Erlend Angell-Jacobsen, Sebastian Slettebakken*

Revisjonshistorie

Dato	Versjon	Beskrivelse	Forfatter
08/03/2021	1.0	Første utkast	Erlend Angell-Jacobsen, Sebastian Slettebakken

Innholdsfortegnelse

Figurliste

14 Tabelliste

Sammendrag

Denne rapporten inneholder oppsett og testing av automatiserte systemer i Azure Sentinel. Oppsettet blir beskrevet trinnvis i de første delene av rapporten. Testene vil se på finne tid spart og om presisjonen og sporbarheten øker. Dette gjør vi for å sikre at systemene opprettholder god integritet, konfidensialitet og tilgjengelighet.

Abstract

This report contains the set up and testing of atomized systems in Azure Sentinel. The setup is shown step by step in the first parts of the rapport. The testing measured time saved and if the precision and traceability increases. We do this to ensure that the systems maintain high integrity, confidentiality and availability.

Innledning

Hensikten med dette dokumentet er å vise oppsett og testing. På slutten konkluderer vi opp imot problemstillingen vår. Problemstillingen er som følger:

Hvordan kan Azure Sentinel brukes til å øke treffsikkerheten og effektivisere arbeidet til en analytiker gjennom automatisering av enkle prosedyrer?

Ut ifra problemstillingen, ønsker vi å avgrense prosjektet ved å fokusere på Azure Sentinel og spesielt muligheten for automatisering i det systemet. Oppgaven setter hovedfokus på hvordan man kan effektivisere responsen til et varsel. Oppgaven er også innom automatisering regler. Vi viser også til tester vi har gjennomført for å forsterke vår påstand om at våre tiltak effektiviserer i noen grad. Testingen vil gå ut på at vi fremprovoserer varsler og deretter løser dem, både med og uten automatisering for å se forskjellen. Vi vil måle tid spart og antall klikk, men også se på sporbarhet og presisjon. Det er også viktig at automatiseringen ivaretar nivået av tilgjengelighet, konfidensialitet og integritet.

Oppgaven er gitt av Sopra Steria. Vi får dermed veiledning både fra Sopra Steria og NTNU.

Definisjoner

Dette kapitlet er en liste over vanskelige ord og utrykk som er brukt underveis i rapporten.

- **Incidents**: Dette er hendelser som man i Azure Sentinel dykker dypere i for å få større innsikt. Hvilke opplysninger man får vil variere fra hendelse til hendelse. Opplysningene vil typisk være informasjon om brukere, IP-adresser og enheter, som er involvert i hendelsen (entiteter).
- **Alert**: Dette er kun varslinger som kommer ved at en regel trigges. (Maestral, 2021)
- **Playbooks**: er en samling av prosedyrer som kan bli kjørt ved sikkerhetshendelser. Playbooks kan også settes sammen med automatiske regler, for å automatisk håndtere en sikkerhetshendelse. Playbooks er basert på Logic Apps. (Microsoft, 2021)
- **Logic App:** Er en samling av applikasjoner bygget inn i Azure for å hjelpe med automatisering, orkestrering og prosessering. I denne oppgaven ser vi hovedsakelig på hvordan man kan bruke Logic Apps til å automatisere respons på sikkerhetshendelser. (Microsoft, 2021)
- **Data connectors**: Når Sentinel er satt opp må man koble Sentinel til andre tjenester. Data connectors vil binde Sentinel sammen med de tjenestene. Sentinel henter logger og varsler, for å så samle alle på ett sted. (Microsoft, 2021)

Oppsett av Azure test-miljø

Vi har fått bruker fra Tisip-fagskole som vi har brukt til å opprette en egen tenant:

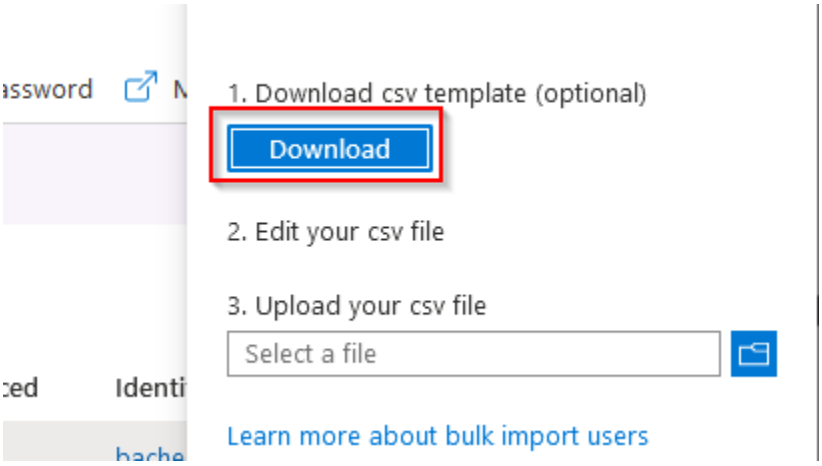
bachelor45.onmicrosoft.com

Vi må også ordne lisens for Defender for Endpoint. Søkte om trial-periode fra Microsoft sine sider og fikk tilgang etter noen dager.

Vi starter med å opprette test-brukere i Azure AD og setter opp noen VM-er på egne pc-er som vi skal koble opp mot domenet. Dette gjør vi for å få muligheten til å frembringe varsler som vi selv ønsker. Vi kan også generere varsler fra innlogging i Azure portalen og fra andre, egne Analytic Rules.

Oppsett av testbrukere

For at miljøet skal virke levende legger vi inn noen brukere. Dette gjør vi gjennom å legge brukerne inn i en CSV-fil. Laster ned en mal fra Azure Portal og fyller inn de nødvendige rubrikkene.



Figur 7 Oppsett av testbrukere - csv-mal

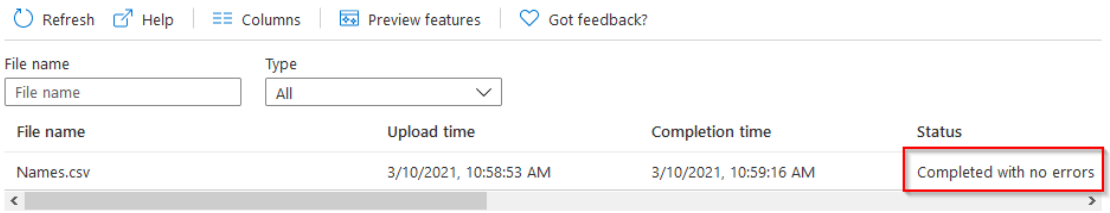
CSV-filen kan vi endre i Excel. Man må nok endre filen slik at man får kategoriene i hver sin kolonne. Deretter er det bare å være kreativ med navn og eposter. Ettersom dette kun er et testmiljø, trenger vi ikke å være strenge på standarder. Her får alle samme passord og man har kun fornavn i e-posten. Dette har vi gjort med tanke på å gjøre det lettere og jobbe med.

Når CSV filen er ferdig ser den slik ut:

1	version:v1.0						
2	Name [displ	User name [	Initial passw	Block sign in	First name [g	Last name [s	Job t
3	Jompa Torm	jompa@bach	abc123ABC	No			
4	Toril Bakke	Toril@bache	abc123ABC	No			
5	Chu Wang	Chu@bachel	abc123ABC	No			
6	Erik Hansen	Erik@bachel	abc123ABC	No			
7	Joakim Ski	Joakim@bac	abc123ABC	No			
8	Mia Forsberg	Mia@bachel	abc123ABC	No			
9	Roar Løkke	Roar@bache	abc123ABC	No			
10	Hilde Stor	Hilde@bach	abc123ABC	No			
11	Sandra Teit	Sandra@bac	abc123ABC	No			
12	Broder Jenss	Broder@bac	abc123ABC	No			
13	Yegor Stav	Yegor@bach	abc123ABC	No			
14	Randi Hest	Randi@bach	abc123ABC	No			

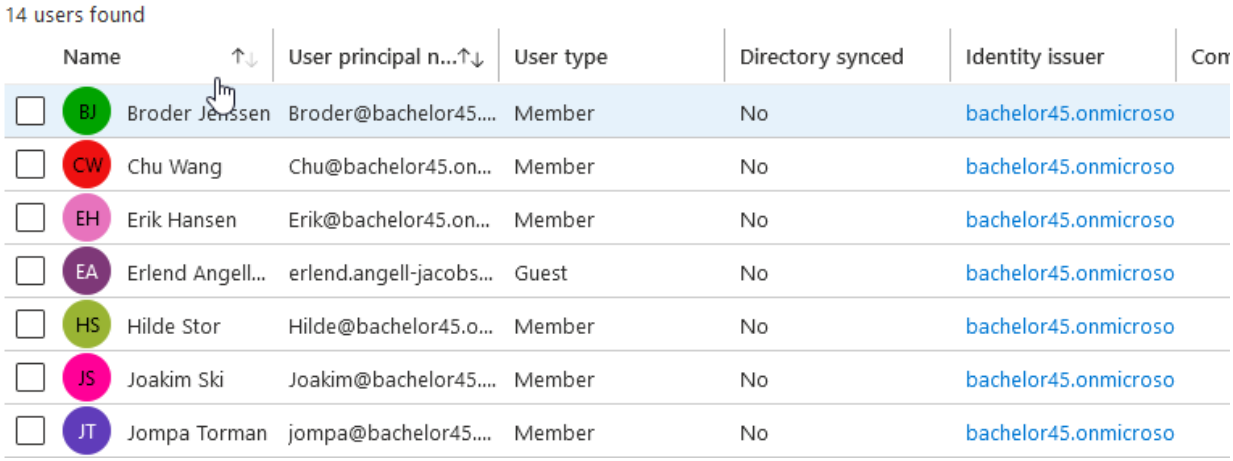
Figur 8 Oppsett av testbrukere - ferdig csv

Nå kan man laste opp filen og se at brukerne blir lagt til under **Bulk operation results** tabben.



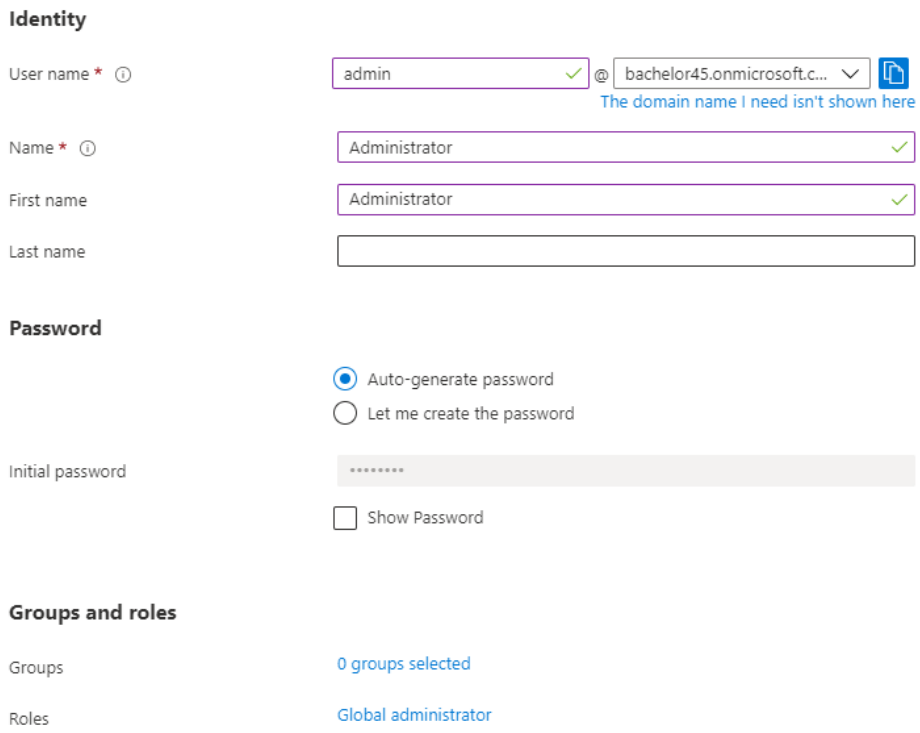
Figur 9 Oppsett av testbrukere - lastet opp csv

Man får også melding om at det gikk igjen i høyre hjørne. Når denne kommer opp, kan man gå inn å se på de nye brukerne under **Users** tabben.



Figur 10 Oppsett av testbrukere - oversikt over brukere

Vi har også behov for å opprette en domeneadministrator for «bachelor45» domenet. Da epostene våre hører til domenet «tisipfagskole.no» kan vi ikke bruke disse når vi skal melde maskiner inn i domenet vårt. Oppretter derfor brukeren «admin@bachelor45.onmicrosoft.com» og gir rettighetene «Global Administrator» som vist under:

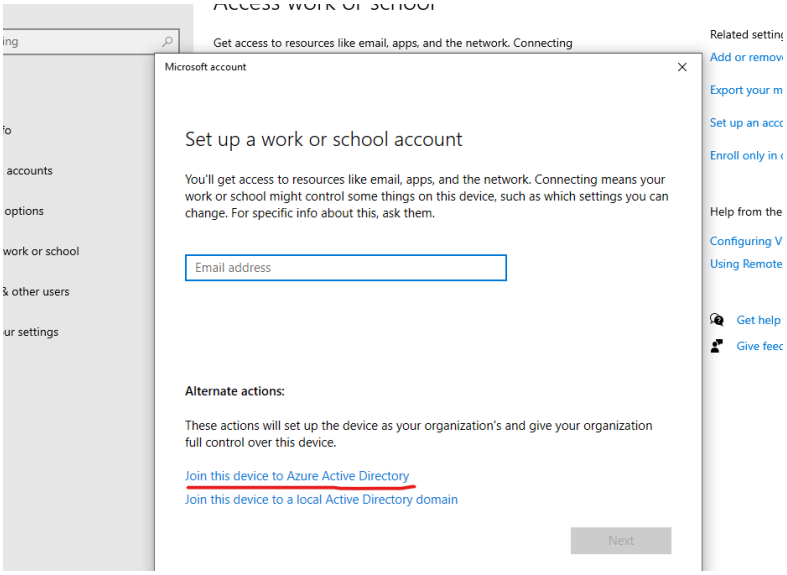


Figur 11 Oppsett av testbrukere - admin@bachelor45

Oppsett av test-maskin

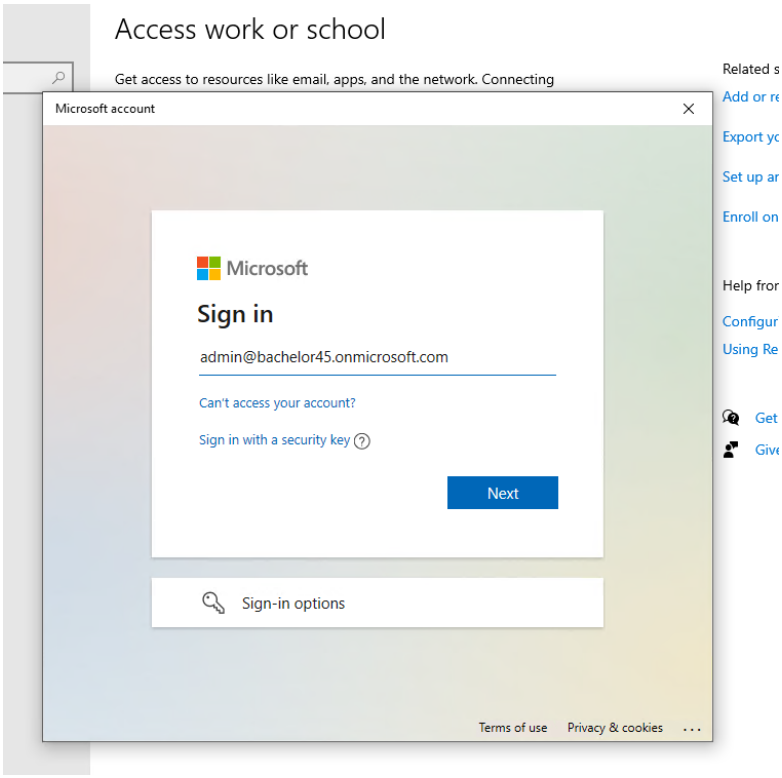
Varslene i Azure Sentinel skapes ikke kun av brukere, men også fra maskiner. Varslene vil som regel være koblet til en bruker, men får å skape et mere realistisk bildet på et miljø, er det naturlig å bruke noen maskiner.

Vi oppretter VM-er ved hjelp av Hypervisor på egen maskin. Etter å ha installert Windows 10 og oppdatert systemet, er maskinen klar for å meldes inn i Azure AD. Velger «Access work or school» under innstillinger og klikker på «Connect». Velger så «Join this device to Azure Active Directory»:



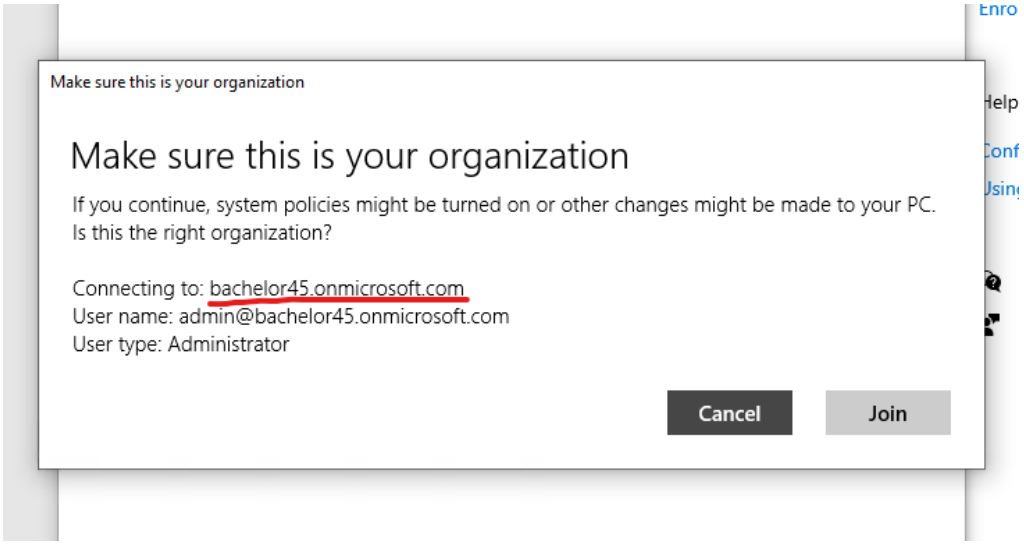
Figur 12 Oppsett av test-maskin - melde inn i Azure AD

Får da bedt om å logge inn. Bruker så domeneadministratoren vi har opprettet:



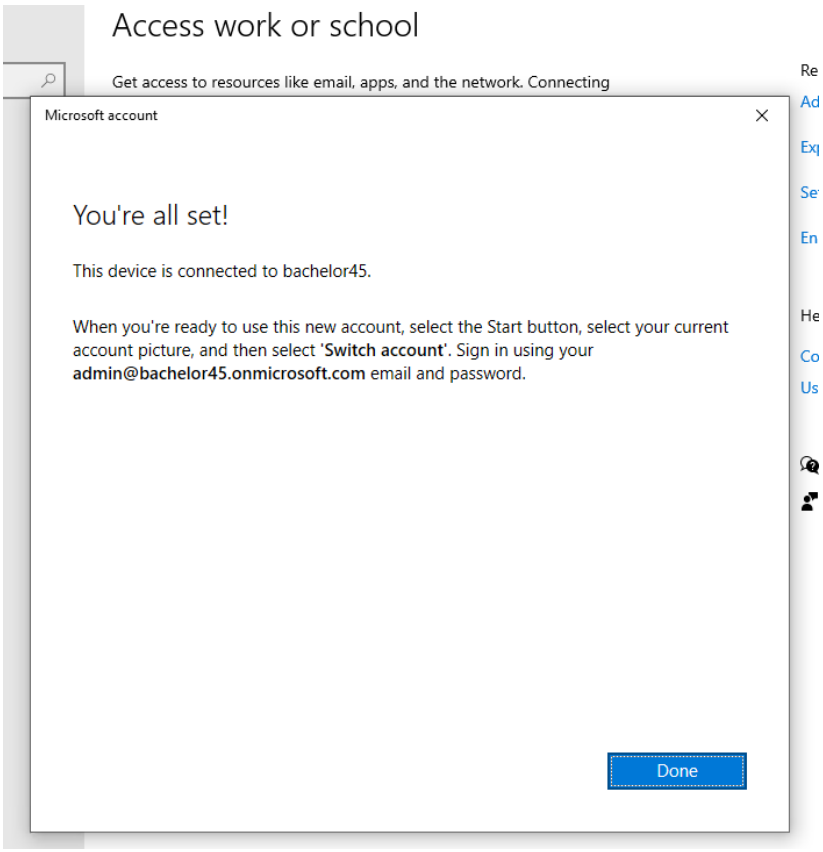
Figur 13 Oppsett av test-maskin - melde inn i Azure AD login

Passer så på at vi meldes inn i riktig domene:

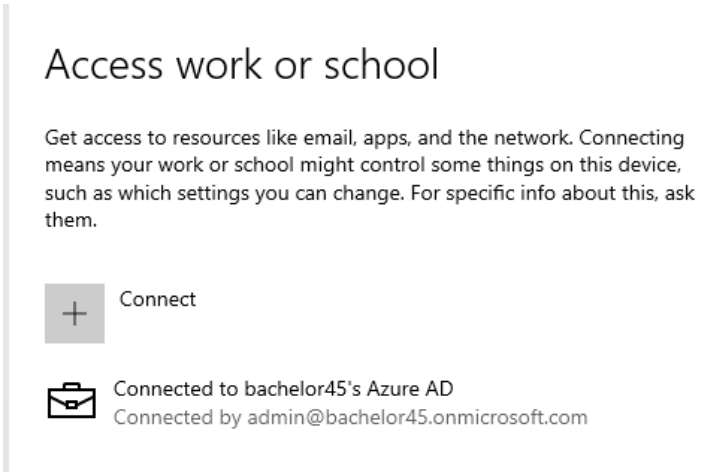


Figur 14 Oppsett av test-maskin - se over innmelding i Azure AD

Maskinen er nå meldt inn i domenet:



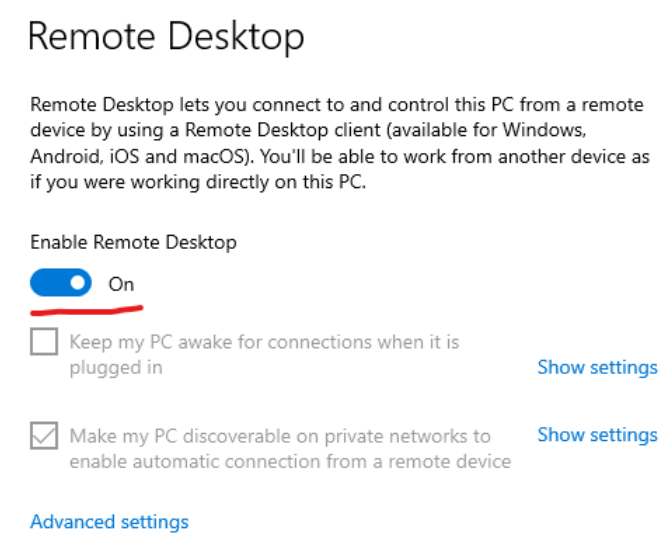
Figur 15 Oppsett av test-maskin - vellykket innmelding i Azure AD



Figur 16 Oppsett av test-maskin - vellykket inmelding i Azure AD

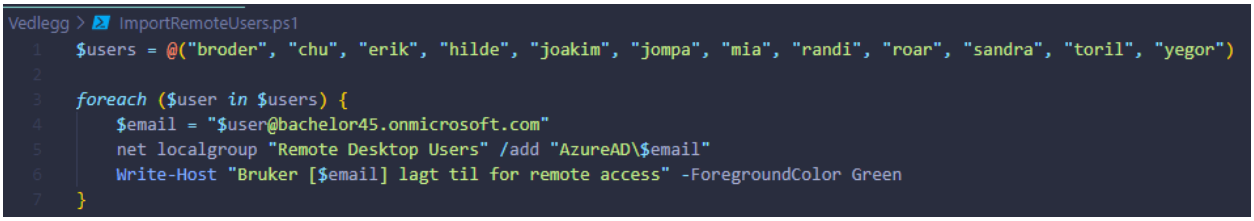
Restarter så maskinen og logger inn som Administrator brukeren (admin@bachelor45.onmicrosoft.com)

For å kunne logge inn som andre brukere, må vi huke av for «Enable Remote Desktop»:



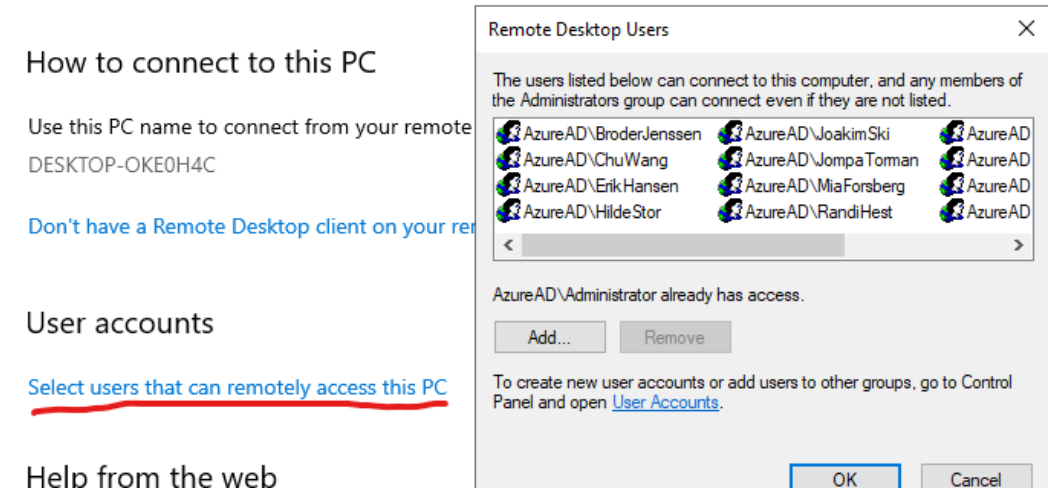
Figur 17 Oppsett av test-maskin - Remote Desktop

Deretter må vi legge til alle brukerne vi ønsker å kunne logge inn med. For dette, har vi opprettet et lite Powershell script:



Figur 18 Oppsett av test-maskin - ImportRemoteUsers.ps1

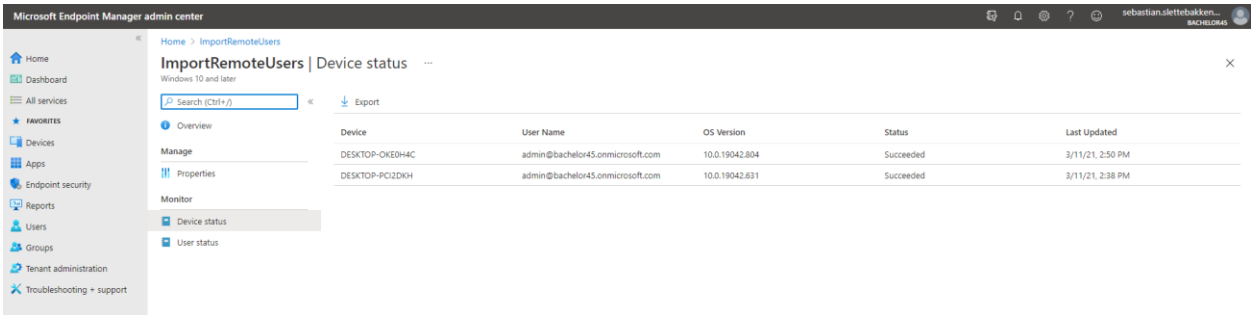
Scriptet legger brukerne til i den lokale gruppa «Remote Desktop Users». Vi kan sjekke at brukerne er lagt til:



Figur 19 Oppsett av test-maskin - Remote Desktop Users

Nå kan vi logge inn som disse brukerne for testing.

Vi legger inn scriptet i Intune (Microsoft Endpoint Manager) og kjører det på alle maskinene som blir lagt til i domenet:

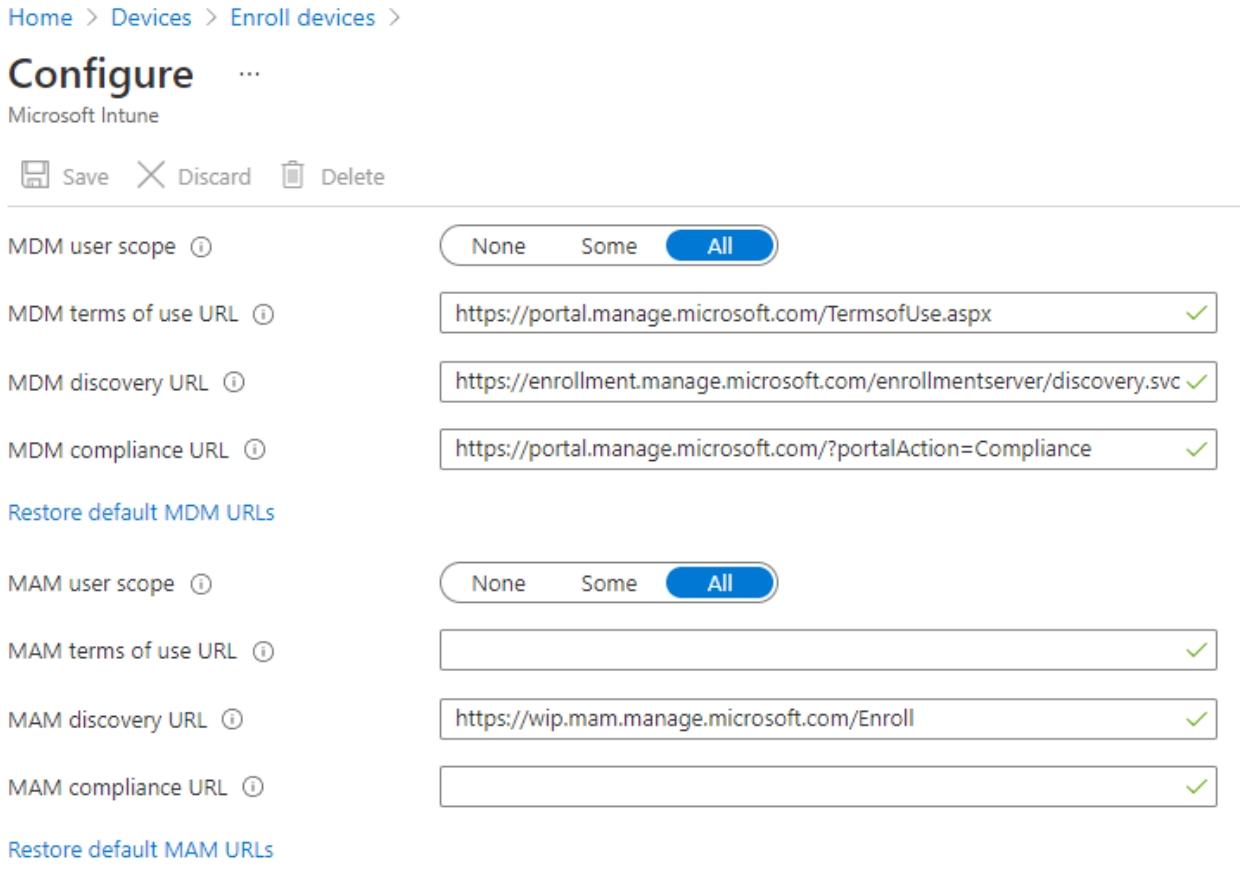


Figur 20 Oppsett av test-maskin - script lagt til i MEM for automatisk kjøring

Ser at det kjørte vellykket på de to maskinene vi har opprettet så langt.

Oppsett av Endpoint Manager og Defender for Endpoint

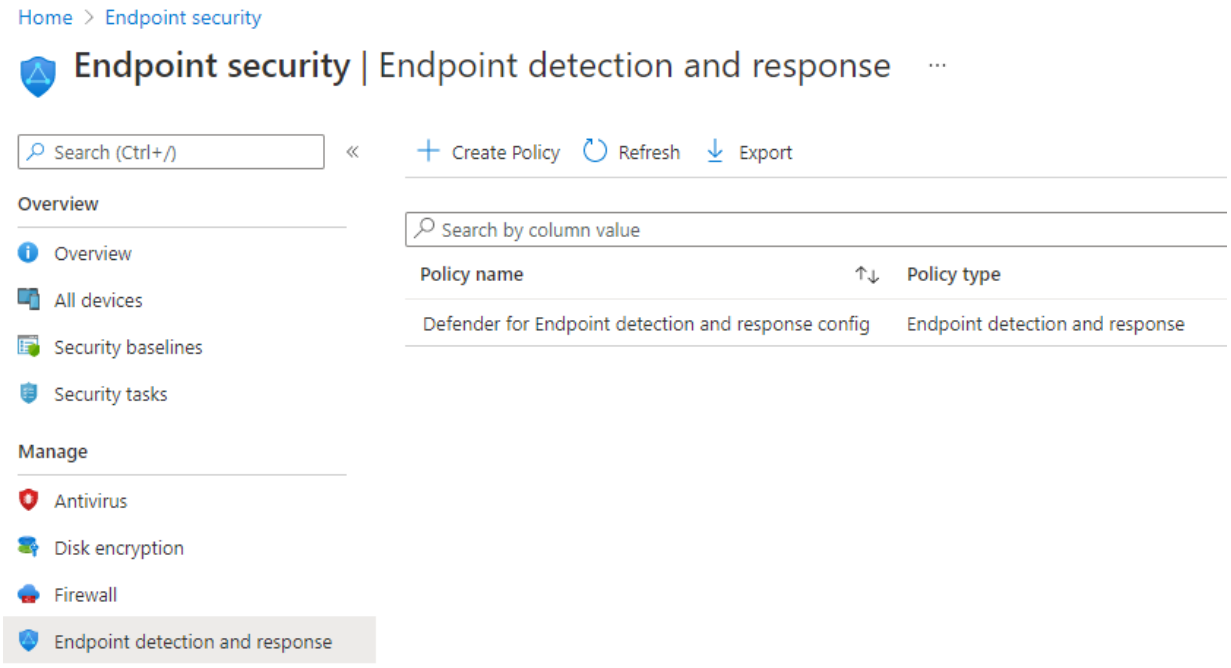
For å administrere enhetene i domenet, velger vi å sette opp automatisk utrulling av enheter ved hjelp av Microsoft Endpoint Manager (MEM/Intune). Setter opp innstillingene for automatisk utrulling slik:



Figur 21 MEM - utrulling av enheter

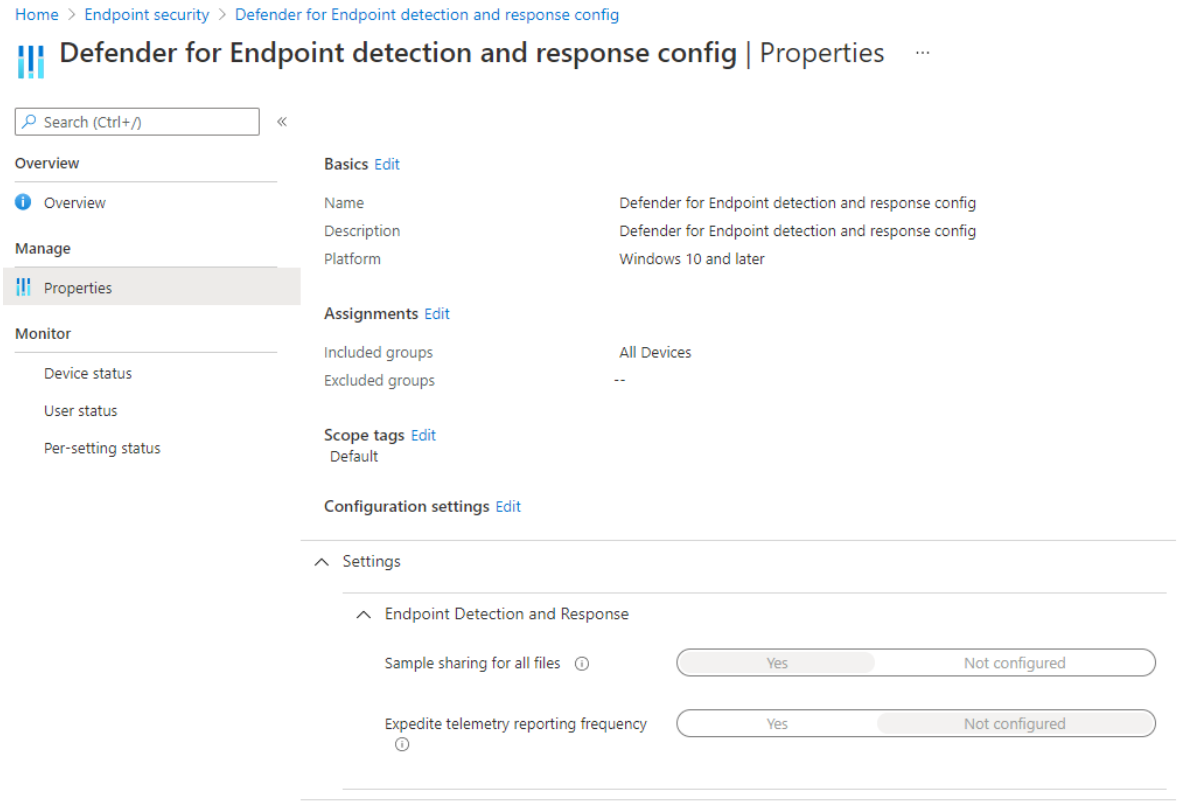
Da dette er et testmiljø, setter vi «user scope» til «All». Nye maskiner som meldes inn i Azure AD domenet vil nå dukke opp i MEM.

For å onboarde enhetene til Defender for Endpoint, må vi konfigurere «Endpoint detection and response» i MEM. Oppretter følgende policy der:



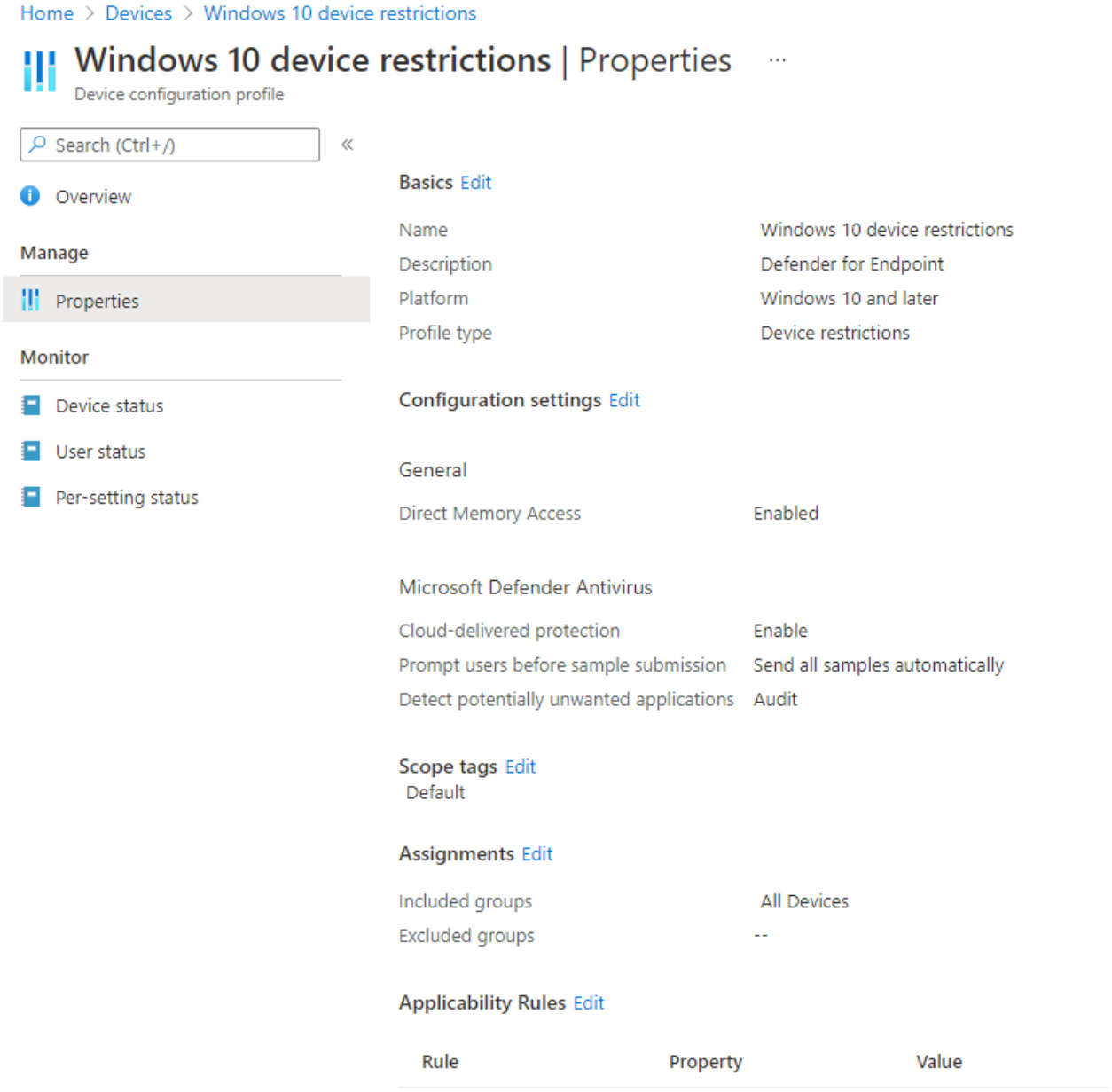
Figur 22 MEM - Endpoint detection and response

Innstillingene for denne policyen ser slik ut:



Figur 23 MEM - Endpoint detection and reponse config

Vi måtte også opprette følgende «Device configuration profile»:



Figur 24 MEM - Device configuration profile for Windows 10

Deretter er det bare å vente til maskinene dukker opp i Defender for Endpoint (securitycenter.microsoft.com). Det tar ca. 15min:

Figur 25 MDATP - enheter dukker opp

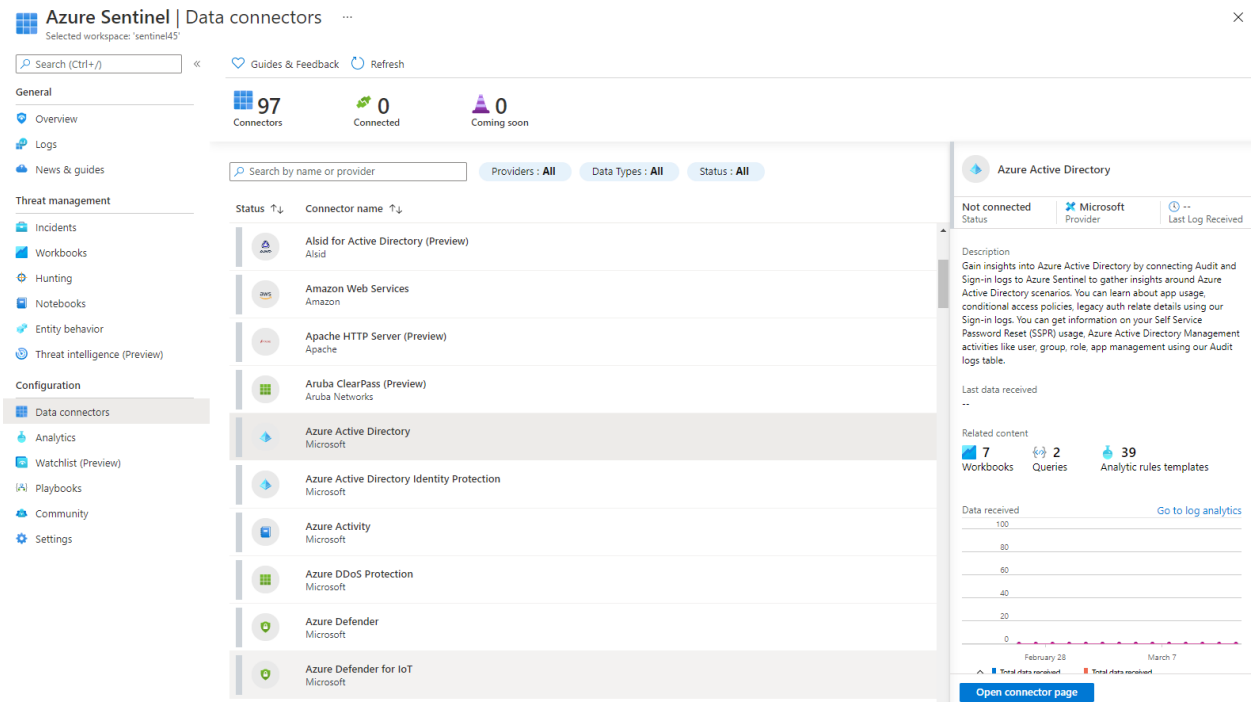
Oppsett Azure Sentinel

Det er viktig at man setter opp rett Data connectors, slik a Azure Sentinel får tilgang på rett data. For viss ikke vil man ikke motta data fra de resterende systemene, og man vil ikke kunne bruke Azure Sentinel. Det er denne dataen som vi senere skal lage regler for og automatisere.

Data connectors

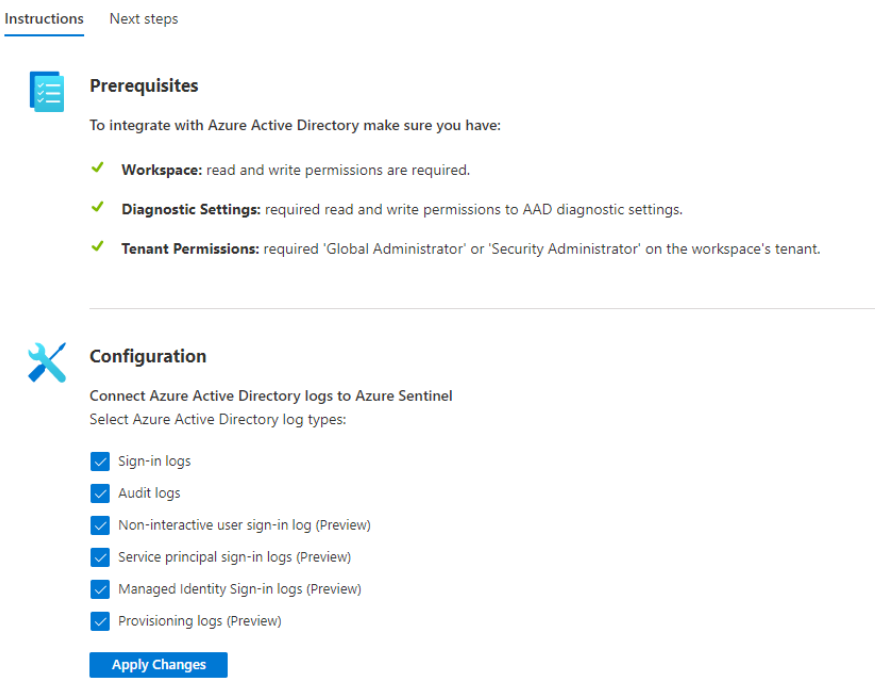
Noe av det første man må gjøre etter å ha aktivert Azure Sentinel, er å koble til data connectors. Dette integrerer andre tjenester som tilbys i Azure direkte med Sentinel. Vi legger til Azure Active Directory slik:

Velger Azure Active Directory under «Data connectors» og klikker på «**Open connector page**»:



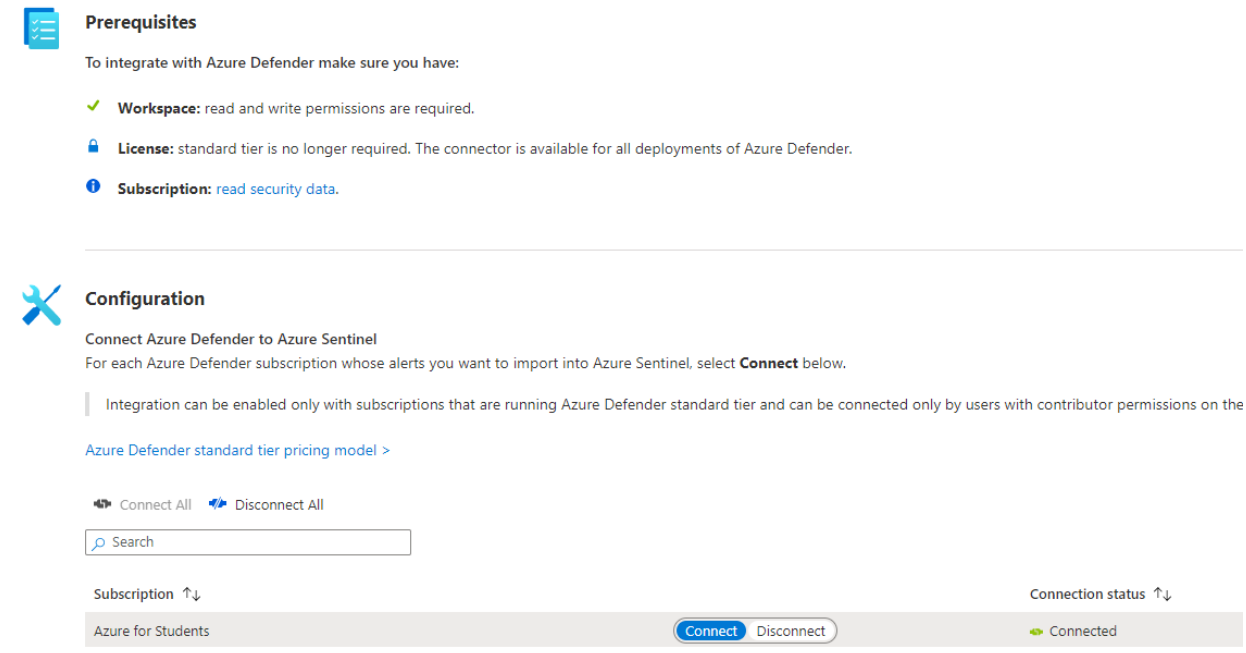
Figur 26 Sentinel Data connectors

Vi velger å huke av for alle loggene:



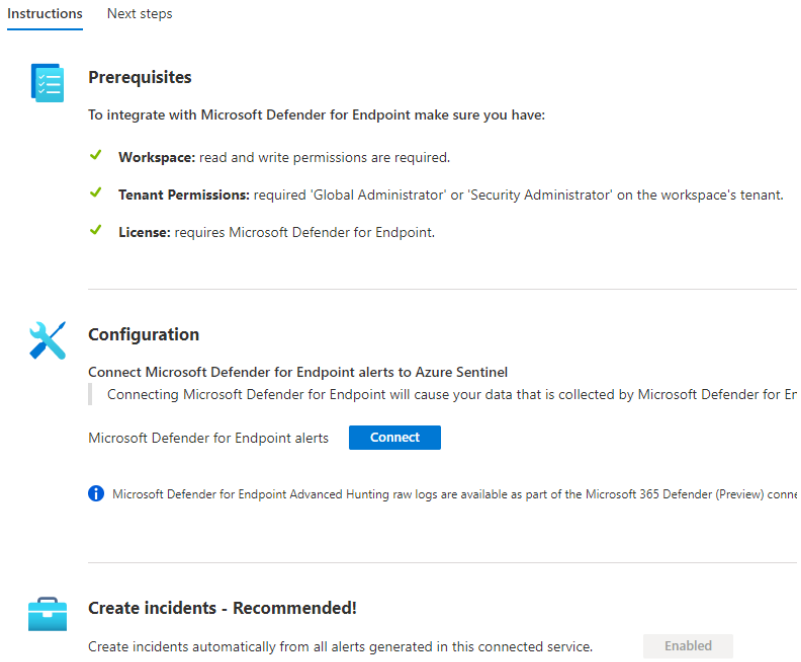
Figur 27 Azure AD data connector

Da er Azure AD lagt til i Azure Sentinel. Vi legger også til Azure Defender sin data connector:



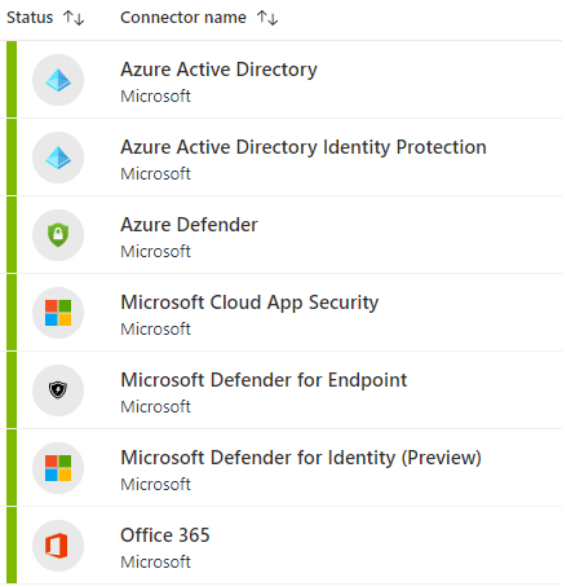
Figur 28 Azure Defender data connector

Etter å ha ordnet en Microsoft Defender for Endpoint trial, kan vi nå også legge til denne Data connectoren:



Figur 29 MDATP data connector

De vi har lagt til nå, er som følger (Office 365 blir ikke brukt i oppgaven):



Figur 30 Azure Sentinel - Data connectors lagt til

Trusselbeskrivelse

En trussel er summen av en angripers intensjon, kapabilitet og mulighetsrom. Det er sjelden mulighet til å påvirke intensjonen eller kapabilitetene. Angripere finner stadig på nye måter for å få tilgang på. Dermed er det opp til personer som jobber med datasikkerhet å jobbe frem et godt forsvar. Akkurat hvilket system du bruker, og hvilke strategier du legger er opp til den enkelte. For å sikre integriteten, konfidensialiteten og tilgjengeligheten til data, så jobber vi for å redusere mulighetsrommet. Det vil si å tette sårbarheter og etablere tiltak som reduserer muligheten for å utføre et vellykket angrep. Noen av tiltakene vi ser på vil også gå ut på å begrense skadeomfanget om et angrep skulle lykkes.

Når vi gjennomfører testingen senere i rapporten vil vi ta utgangspunkt i disse truslene når vi frembringer varsler. Varslene vil vise at de for eksempel er et mulig brute force angrep. Deretter vil vi automatisere ulike tiltak for den varselen.

Brute force angrep

Denne typen angrep går ut på at hackeren prøver å tippe ditt passord og brukernavn. Det finnes flere forskjellige metoder for dette. De vanligste er:

- Den enkleste er simple brute force. Denne går ut på å kun tippe. Denne har ingen logiske tanker bak seg.
- Hybrid brute force går ut på å først tenke logisk over hva passordet kan være. For også teste ulike variasjoner av passordet.
- Ordbok angrep sier seg selv. Her bruke man en ordbok for å tippe passord. En del personer bruker ord for å gjøre det lettere å huske passord. Dermed for en hacker er en ordbok en lur plass og begynne. (Cyclonis, 2021)
- Rainbow table attacks er når man bruker bestemte tabeller for å reversere hasher. Den kan brukes for å gjette passord mere nøyaktig.
- Reverse brute force attack fungerer på samme måte som ordbok metoden, men her bruker man en liste over passord som er ofte brukt. Her er det brukernavnene som blir byttet og ikke passordene. For man går nemlig for at brukerne bruker passord som er ofte brukt.
- Credential stuffing går ut på at man har fått tak i et passord og brukernavn som funker. Her bytter man heller nettsted for å se om brukeren bruker samme innlogging andre plasser. Denne kan være vanskelig å finne for sikkerhets ansvarlige. For de sikkerhets ansvarlige står det kun en mislykket eller vellykket innlogging.

Det er mange mulige måter å sikre seg mot slike hendelser for enkelt personer. Men også for bedrifter, Lockout Policy, progressive forsinkelser, multifactor authentication osv. (Impreva, 2021)

Malware angrep

Malware er et samlebegrep for en rekke skadelig programvare. Det omfavner virus, ramsomware og spyware. Det er mange forskjellige måter man kan få malware. Man kan for eksempel få det gjennom phishing angrep, eller av at man besøkte en nettside osv. Malware kan også spres til mobiltelefoner gjennom noe så enkelt som sosiale medier. Mye av dette kan fanges opp av en god antivirus-løsning.

Indikatorer på angrep

Det finnes forskjellige indikatorer som kan tyde på at et system har blitt angrepet. Disse er kjent som «Indicators of compromise» (IOC), og er viktig å ha en oversikt over. Eksempler på slike IOC-er er:

- Uvanlig trafikk inn og ut av nettverket
- Ukjente filer, programmer og/eller prosesser som kjøres i systemet
- Mistenkelig oppførsel/aktivitet på administrator eller andre privilegerte brukere
- Uvanlig aktivitet som trafikk fra ny IP-adresse i et annet land
- Nettverkstrafikk på nettverksporter som vanligvis ikke brukes (dette kan forhindres i brannmuren ved å stenge alle porter man ikke bruker)

Automatisert håndtering av sikkerhetshendelser

Automatisering er teknikken å få systemer til å fungere uten, eller med liten grad av menneskelig medvirkning. (SNL, 2021) Målet med automatisering er å raskere håndtere sikkerhetshendelser og returnere en tjeneste til normal-tilstand. Automatisering øker en analytikers evne til å håndtere flere hendelser og økte data-mengder på en raskere og mer presis måte. I mange sikkerhetshendelser er det flere steg i prosessen for innhenting av data som er lik i hver analyse. Her vil automatisering gi en økt verdi i form av raskere håndtering, men også sikre at prosessen blir utført likt hver gang. Det blir vanskelig å sammenligne analytikere og datamaskiner på noe annet en treffprosent, da begge har sine fordeler og ulemper. For eksempel vil en datamaskin gjøre tiltak mye kjappere enn en analytiker, mens en analytiker vil i større grad tenke gjennom hendelsen og ta med aspekter som en datamaskin ikke kan.

Automatisering er en god måte å effektivisere gjentakende arbeidet til analytikere på. Men det er noen kriterier automatiseringen må fylle for å være nyttig.

- Den må ikke skape nye sikkerhetshull. For eksempel om man automatiserer å stenge ut brukere etter for mange mislykkede innlogginger. Da kan man tenke at man avverger situasjonen og et mulig brute-force angrep. Men hva om en angriper gjør dette med alle andre brukere og admin. Da vil hele systemet stenges, og tilgjengeligheten forsvinner.
- Automatisering må både redusere risikoen for feilhåndtering, men også sørge for å håndtere en ekte sikkerhetshendelse. Avgjørelsen om å ta for mange eller for få må vurderes opp mot viktigheten av dataene som beskyttes.
- Automatisering av oppgaver må ha god sporbarhet. Dette muliggjør analyse og måling av automatiseringen, for å lære av feil, øke treffsikkerhet og måle verdien av automatiseringen over tid.
- Det er også ønskelig at automatiseringen er med på å minke tid som er brukt på en sikkerhetshendelse. Det bør ikke bruke mer tid enn å gjøre den manuelt.

En balanse mellom automatisering og menneskelig analyse må på plass for å sikre integritet, konfidensialitet og tilgjengelighet til data i et skiftende trusselbilde.

Effektivisering

Ett av målene med automatisering er å øke effektiviteten til en analytiker. Effektivisering er et viktig stikkord om man skal drive med datasikkerhet. Sikkerhetsverktøy, som Azure Sentinel, genererer mange alarmer, men mange av alarmene skal ikke håndteres av mennesker før de blir observert i sammenheng med andre alarmer eller informasjon. Dermed for å få gått igjennom så mange sikkerhetshendelser som mulig er det viktig at man effektiviserer enkelte aspekter. Ved en alarm er det flere ting som kan effektiviseres i etterkant av alarmen:

- Enrichment - samle informasjon og skaffe oversikt
- Vurdere informasjonen som er samlet

- Handle etter hvilke opplysninger du har fått

Det er viktig at man kommer raskt til en avgjørelse om alarmen er falsk-positiv eller ikke. Akkurat hvilke tiltak som vil effektivisere vil som regle variere litt fra hendelse til hendelse. Men prinsippet er det samme. Ting skal gå raskere. Et typisk tiltak for å effektivisere er å sikre at analytiker får rask oversikt over hendelsen. Når hendelsen er vurdert, må analytiker ha mulighet til å handle raskt.

Enrichment

Enrichment, eller berikelse av data, er viktig for å gi et bedre bilde av hva dataene forteller. Automatisering av berikelse vil sikre at alle data blir beriket på samme måte hver gang, samt at det vil spare tid. Et av stegene som kan automatiseres, er varsling når noe skjer. Dette kan gjøres ved å automatisk finne frem til riktig logger og relevant informasjon om en hendelse og presentere dette på en enkel og oversiktlig måte for analytikeren. Vi ønsker dermed å kunne legge ved ekstra informasjon ved hendelsen for å lettere kunne ta bedre avgjørelser.

Automatisering i Azure Sentinel

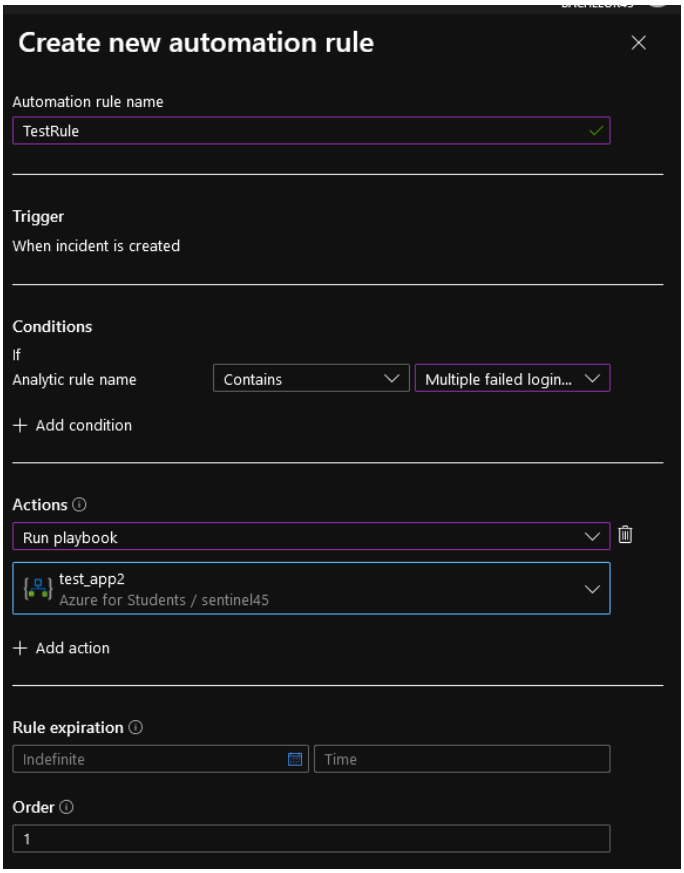
Azure Sentinel er et SIEM (Security Information and Event Management) system. Målet med et SIEM-system er å samle sikkerhets-data på ett sted for å korrelere, berike og alarmere på mistenkelig aktivitet som beskrives av dataene. Dette gjøres gjennom å samle inn logger fra ulike IT systemer og samle de på én plass. Dermed kan Sentinel motta og samle alle varslinger og alarmer på ett sted. Sentinel har også andre funksjoner som vil gjøre det lettere for en analytiker å behandle varslinger. For eksempel, er Investigation er et redskap man bruker for å få rask oversikt over en hendelse. Informasjonen blir presentert i for av et tankekart, hvor man kan klikke for mer informasjon. Når etterforskingen er ferdig, vil analytikeren ta en avgjørelse om alarmen er falsk-positiv eller ikke. Automatiseringen i Sentinel foregår som regel i for av Playbooks. Playbooks er en samling av Logic Apps. Logic Apps er apper med forhåndsdefinerte valg og muligheter. En måte å bruke disse på, er for eksempel gjennom å lage en Playbook som poster en melding på Teams. Da ville man gjennom Logic Apps bestemme hva som skal til for at den trigges. Når det er bestemt, kan man velge å poste en melding i Teams.

Om man gjør dette har man effektivisert litt, men om man skal automatisere hele håndteringen av hendelsen, må man gjøre et mer innvirkende grep på slutten på Logic Appen. For eksempel å isolere endepunktet eller blokkere en IP-adresse. Dette er grep som innvirker i arbeidshverdagen til de ansatte i bedriften. Dermed er det viktig at automatiseringen ikke blokkere ansatte som er falskpositive. Automatiseringen bør ha høy treffsikkerhet. Om ikke blir det ekstra arbeid på analytikeren, som må bruke tid på å rydde opp i situasjonen.

I Azure Sentinel fins det forskjellige metoder for å automatisere, men metodene benytter Playbooks i stor grad. Det fins ulike måter å automatisk kjøre en Playbook på. Man kan også trigge Playbooks manuelt ved sikkerhetshendelser.

Automation Rules

Automation Rules er de første metoden man kan automatisere. Den finner vi under Automation taben i Azure Sentinel. Når man oppretter en Automation Rule finner man først på et navn. Deretter ser man at triggeren til Automation Rule er «When an incident is created». Om man for eksempel har en Analytic Rule og den lager en incident. Så kan man sette ulike krav til at automatiseringen kan gjøres. Man legger som regel til Analytic Rules som skal automatiseres. Velger så hva handlingen er vanligvis er dette en Playbook.



Figur 31 Automation rule

Så kan man spørre seg hvorfor man ikke bare lager Analytic Rules. For gjennom Automation Rules kan man tilegne flere Analytic Rules. Man også gjøre andre ting som å endre alvorlighetsgrad, hvem som skal ta seg av alarmen, legge til emneknagger og endre status.

Analytic Rules

Analytics Rules er regler som skaper varslinger. Man kan lage sine egne varslinger eller velge ut ifra maler. Det er over 250 maler man kan velge ut av. Og om ingen av de passer kan man lage egne eller rediger malene slik at de blir skreddersydd. Under Analytic Rules kan man også velge videre steg for automatisering. Her kan man velge hvilke Playbooks som skal kjøres når Analytic Rulen trigges.

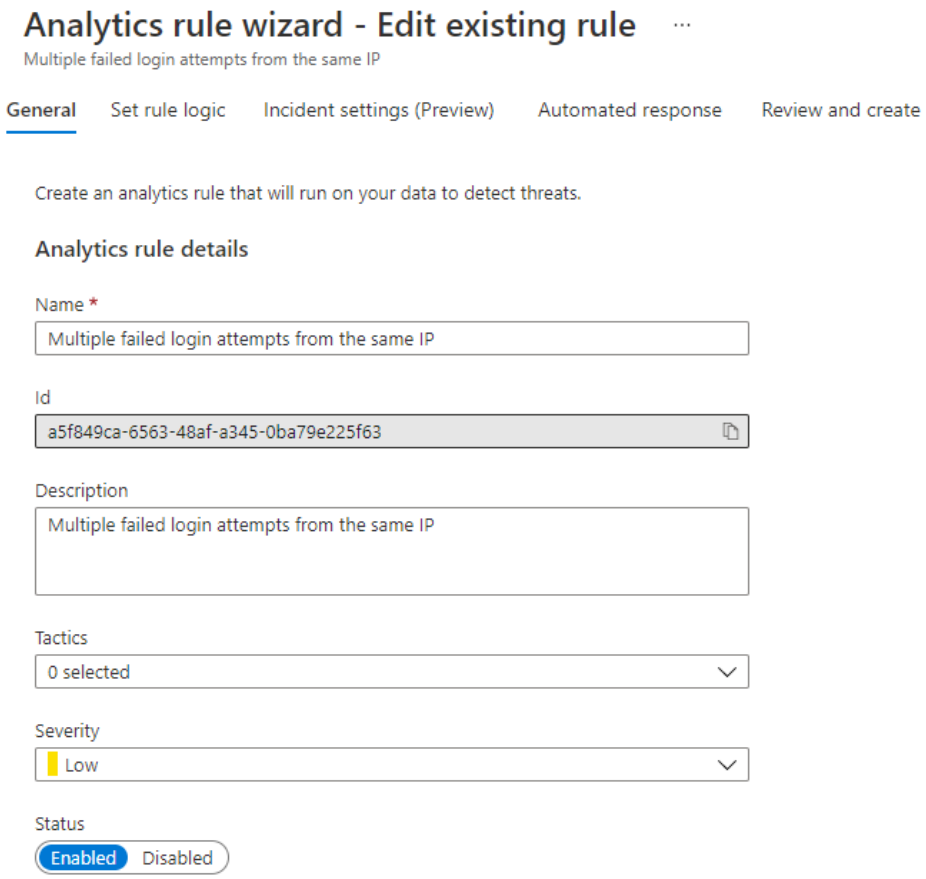
Tabell over automatisering

Navn	Type	Hensikt
Flere feil innloggingsforsøk fra samme IP-adresser	Analytic Rule som oppretter Incident	Fange opp IP-adresser som har flere feil innloggingsforsøk.
Varslinger i Teams	Playbook	Legger ut informasjon om varslinger på Teams
VirusTotal skanning av IP-adresse	Playbook	Hente IP-adresser som entitet fra en hendelse og skanne den. Legger så til kommentar til hendelsen om IP-en.
Legge inn IP-adresser fra hendelse i Watchlist	Playbook	Legge inn IP-adresser knyttet til en hendelse i en Watchlist. Kan både være for å blokkere eller slippe gjennom.
Isolere enhet i MDATP ved hjelp av Sentinel	Playbook	Isolere enheter som er knyttet til en hendelse.
Blokkere bruker/gi tilgang	Playbook	Blokkere brukere som er knyttet til en hendelse.
Tilbakestille passord til Azure AD bruker	Playbook	Tilbakestille passordet til brukere knyttet til en hendelse ved hjelp av Microsoft Graph API. Midlertidig passord blir sendt til Teams-kanal.
Blokkere IP i MDATP	Playbook	Blokkerer IP-adresser knyttet til en hendelse i MDATP ved hjelp av Microsoft Graph Security.
Oppdage pålogging fra IP i Watchlist-RiskIP	Analytic Rule som oppretter Incident	Varsle dersom noen har forsøkt å logge seg på fra en IP-adresse som ligger i Watchlist for Risky IP-adresser.

Tabell 7 Oversikt over Playbooks og Analytic rules

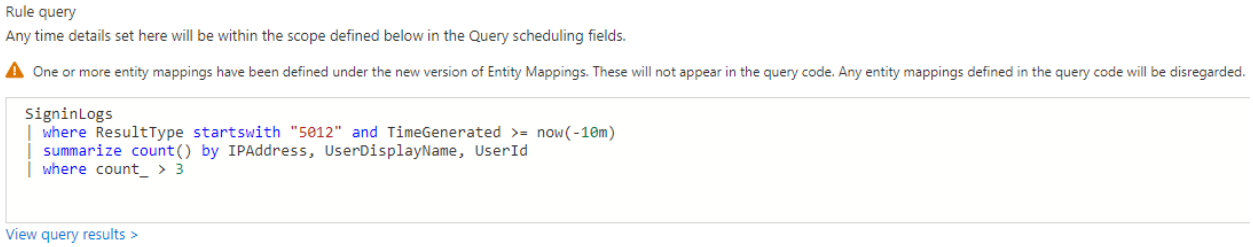
Flere feil innloggingsforsøk fra samme IP-adresse (Analytic Rule)

Starter med å opprette en **Analytic Rule** som vil opprette en **Incident** etter den finner hendelser i loggene. Denne regelen vil lete etter mislykkede innloggingsforsøk fra samme IP-adresse. Setter innstillingene slik under **General**:



Figur 32 Flere feil innloggingsforsøk fra samnme IP-adresse - Generelle innstillinger

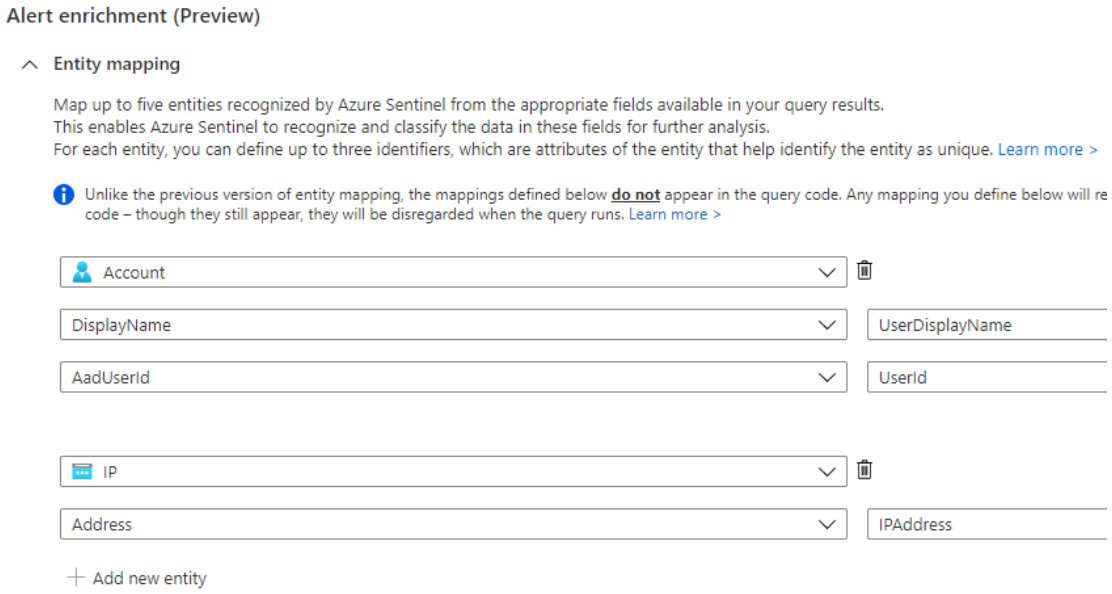
Spørringen vi ønsker å kjøre ser slik ut:



Figur 33 Flere feil innloggingsforsøk fra samnme IP-adresse - KQL spørring

Den ser gjennom tabellen/loggen for påloginger og finner de som feilet (med for eksempel ResultType lik 50126) de siste ti minuttene. Så teller den antallet og returnerer en incident dersom det er over 3

mislykkede innlogginger. For enrichment, legger vi inn brukeren som har prøvd å logge på og IP-adressen som ble brukt:

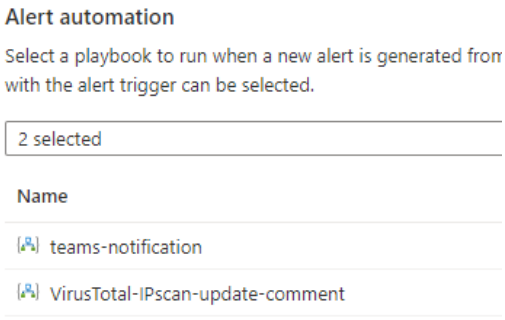


Figur 34 Flere feil innloggingsforsøk fra samnme IP-adresse - Enrichment

Spørringen skal kjøres hvert tiende minutt og skal se gjennom de siste ti minuttene med logger:

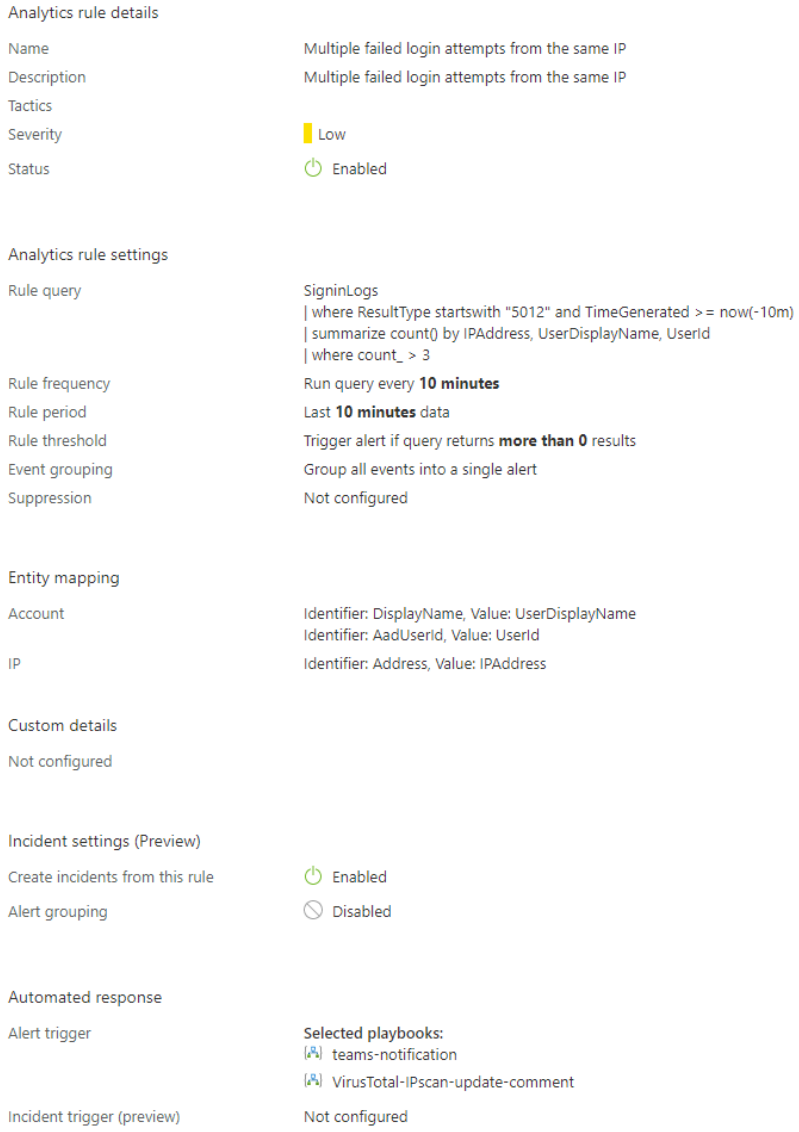
Figur 35 Flere feil innloggingsforsøk fra samnme IP-adresse - Query scheduling

Blar videre til **Automated Response** og huker av de Playbookene vi ønsker at regelen skal kjøre dersom den oppretter en hendelse. Vi har en som oppretter varsel i en Teams-kanal og en som skanner IP-adressen:



Figur 36 Flere feil innloggingsforsøk fra samnme IP-adresse - Automatisering

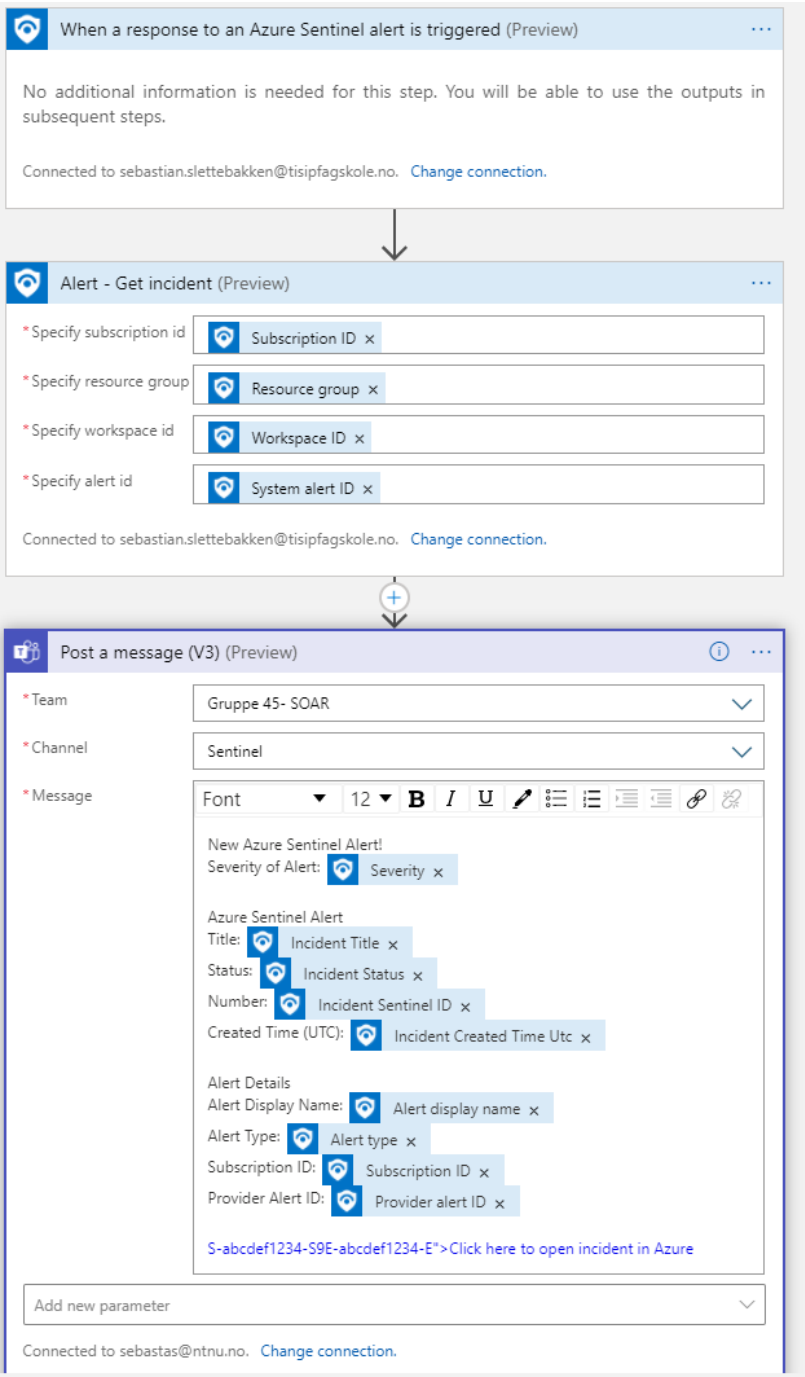
Sjekker at regelen ser grei ut og lagrer:



Figur 37 Flere feil innloggingsforsøk fra samnme IP-adresse - Oversikt

Varsling i Teams-kanalen (Playbook)

For å lettere kunne varsles når en hendelse blir opprettet i Sentinel, ønsker vi å kunne sende en varsling med et kort sammendrag av en ny sikkerhetshendelse i en egen Teams-kanal. Oppretter så den følgende Playbooken:



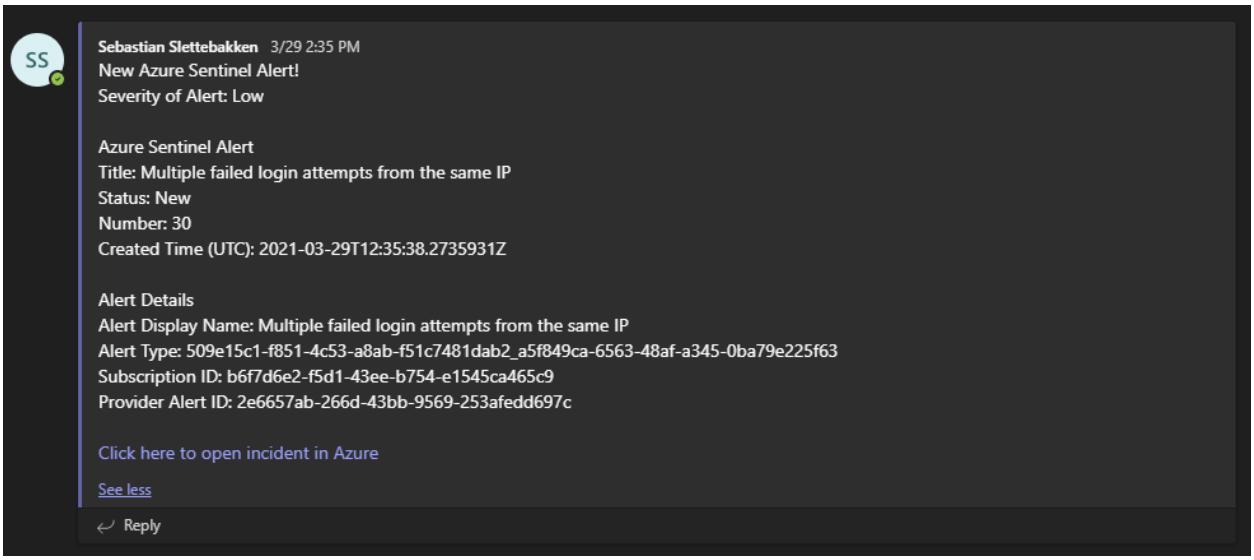
Figur 38 Varlsing i Teams-kanal - Logic App

For riktig formatering av hyperlenken som sendes, må vi legge inn HTML kode manuelt med «» i **Logic App code view** som vist under:

```
ı<br>\n<a href=\"@{body('Alert_-_Get_incident')?['properties']?['incidentUrl']}\">Click here to open
```

Figur 39 Varlsing i Teams-kanal - riktig HTML for link til hendelsen

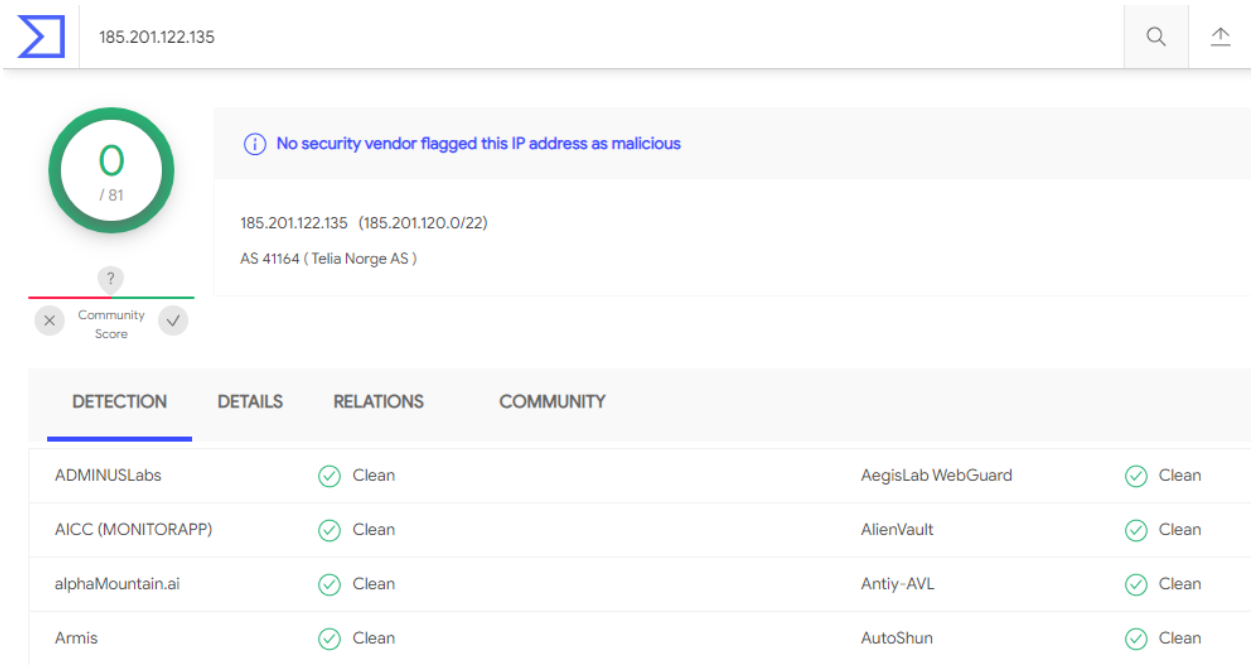
Resultatet når man kjører denne Playbooken på en hendelse kan se slik ut:



Figur 40 Varlsing i Teams-kanal - eksempel på kjøring

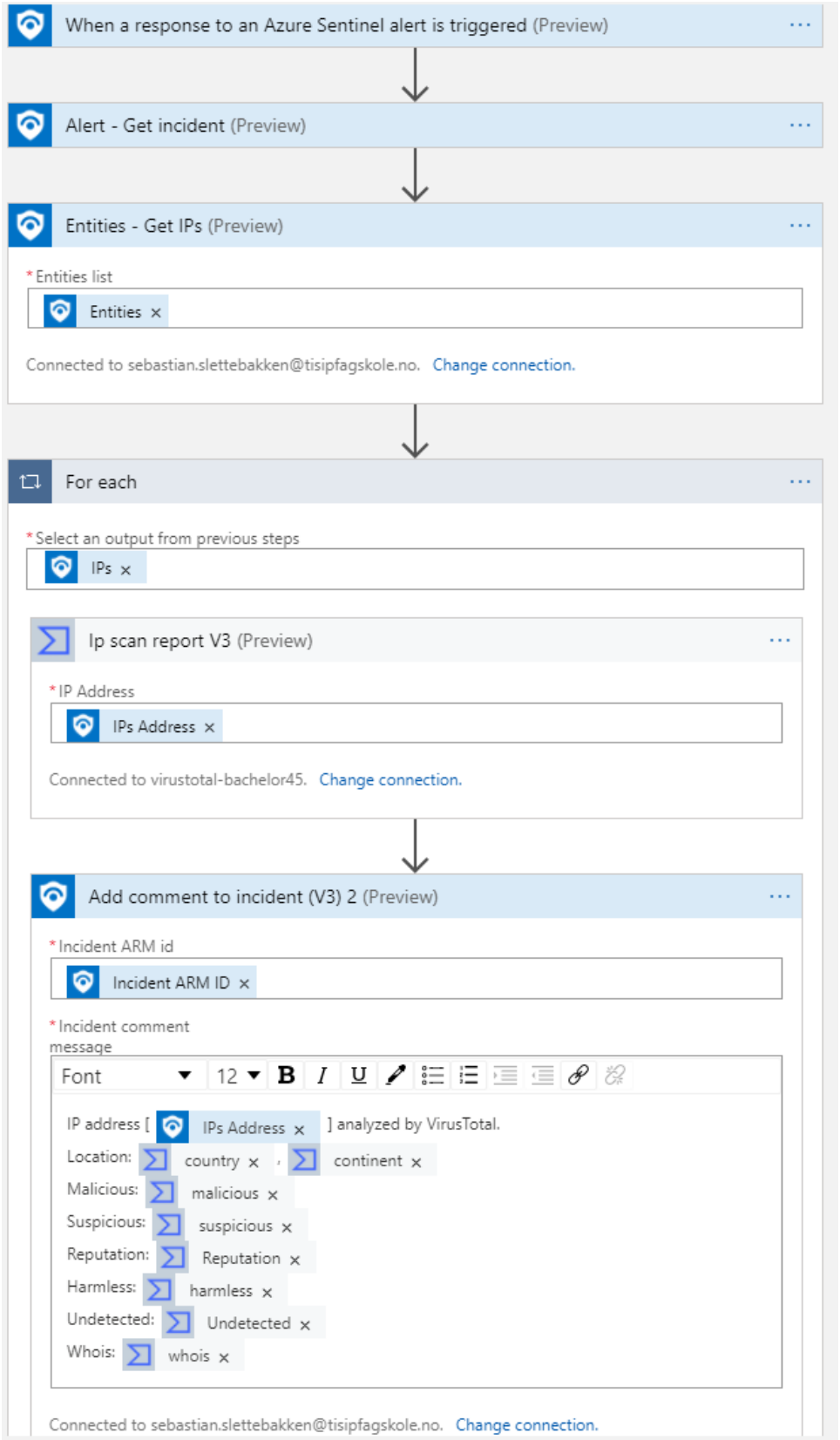
VirusTotal IP skanning (Playbook)

Det kan være greit å finne ut litt ekstra info om en ukjent IP-adresse for analytikere som må vurdere om deler av et system kan være kompromittert. Man kan da kjøre skanne IP-adressen hos VirusTotal og få info om noen har markert IP-en som farlig eller ikke. Under ser vi hvordan dette kunne blitt gjort manuelt, ved å fylle ut IP-adressen man ønsker å skanne og får opp resultatet:



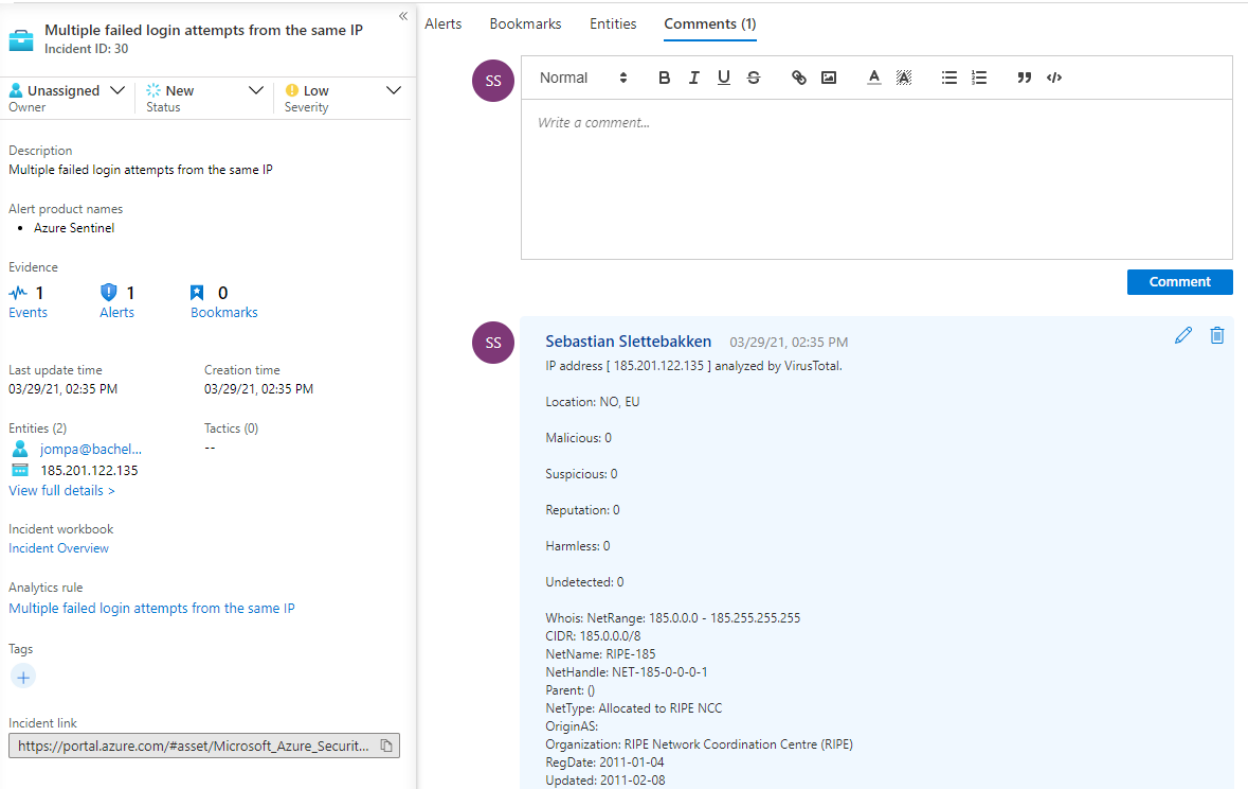
Figur 41 VirusTotal IP scan eksempel

Det er ingen som har oppdaget noe skummel trafikk på denne adressen (fra Telia Norge AS). Vi ønsker å automatisere denne prosessen. Oppretter derfor en gratis bruker på VirusTotal og får tak i API-nøkkelen vår derfra. Denne legger vi inn i Logic App som en API connection og vi får Playbooken som følger på neste side:



Figur 42 VirusTotal IP scan - Logic App

Vi henter altså hendelsen, og IP-adressen som entitet for hendelsen. Dersom dert er flere IP-er, vil alle disse også bli kjørt i loopen. Så blir IP-adressen skannet ved hjelp av **IP scan report v3** fra TotalVirus (med vår API connection). Oppdaterer så hendelsen med info vi får fra VirusTotal. Eksempel på en slik kjøring kan se slik ut:

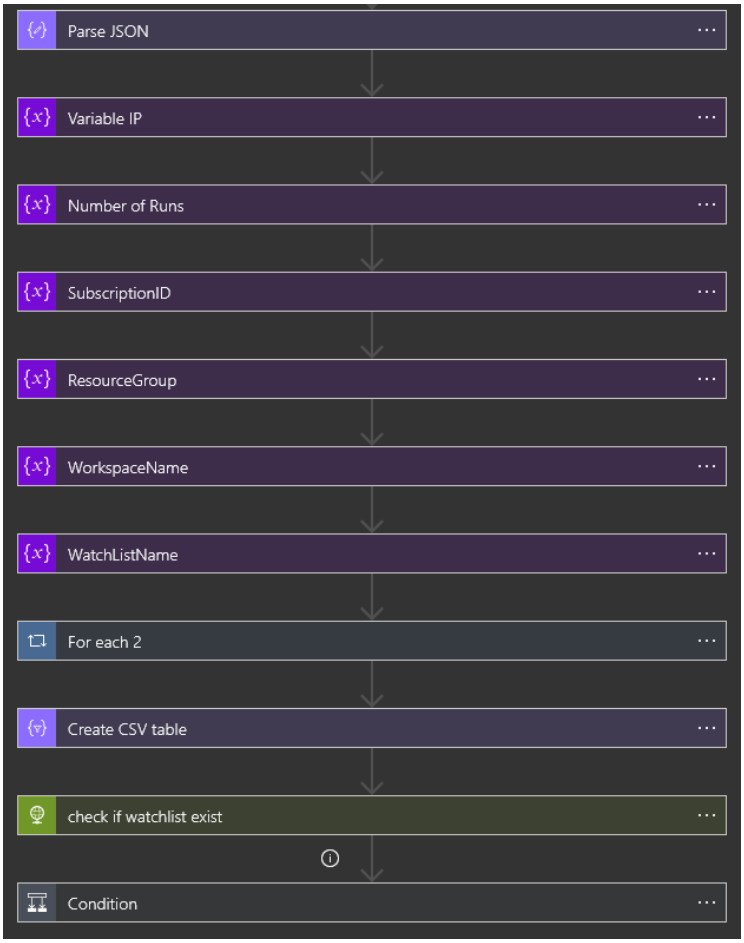


Figur 43 VirusTotal IP scan - eksempel på kjøring med kommentar

Legge inn IP-adresser i Watchlist

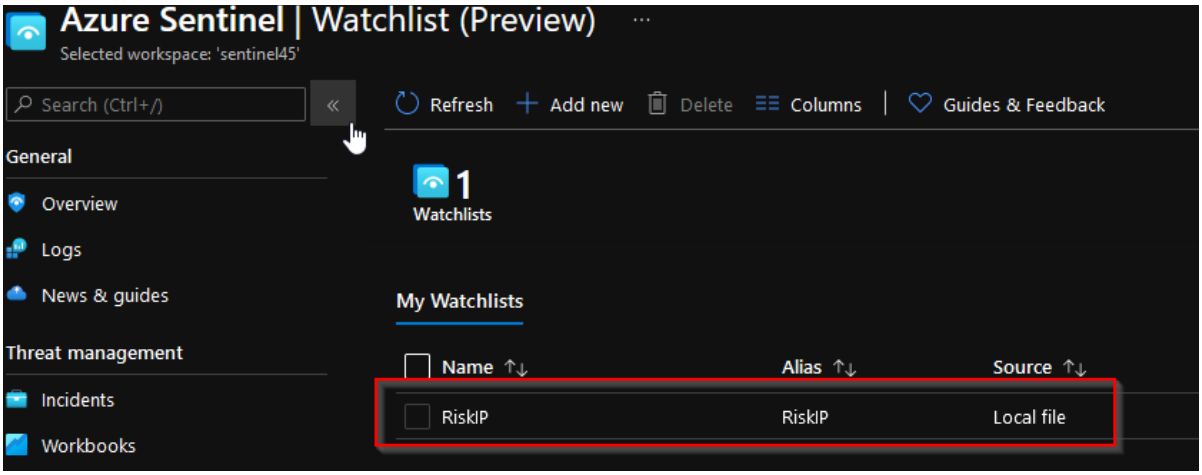
Watchlist er en måte så samle og lagre data på i Azure Sentinel. Dataen kan brukes på mange forskjellige måter. Du vil senere kunne bruke den dataen du legger inn i Watchlist, til å for eksempel i søk, etterforsking, trusselsøk og i Playbooks. Hvordan man bruker dataen vil være opp til den enkelte. Om man bruker Watchlist på noen som helst måte vil det være nyttig med en Playbook som henter IP-adressen fra en hendelse og legger den inn i en Watchlist. Man kan legge inn annen data enn kun IP-adresser. Man kan for eksempel bruke brukere, grupper og enheter.

Denne Playbooken er hentet fra Azure Community og endret litt (Shasha, 2021). Vi endret det slik at det i større grad skal kunne automatiseres ved en senere anledning. Vi fikset på variabler slik at de passet systemet. Når man arbeider med Watchlist finner man fort ut at det er ikke bare å legge inn data. All data som importeres i Watchlist må være i CSV-format. Playbooken vi har er slik at den oppretter en CSV-fil for hver gang den kjører. Deretter importerer den inn dataen i Watchlist. Playbooken sjekker også om Watchlist eksisterer fra før av, om ikke opprettes det en ny en. Slik ser Playbooken i Logic App designer:



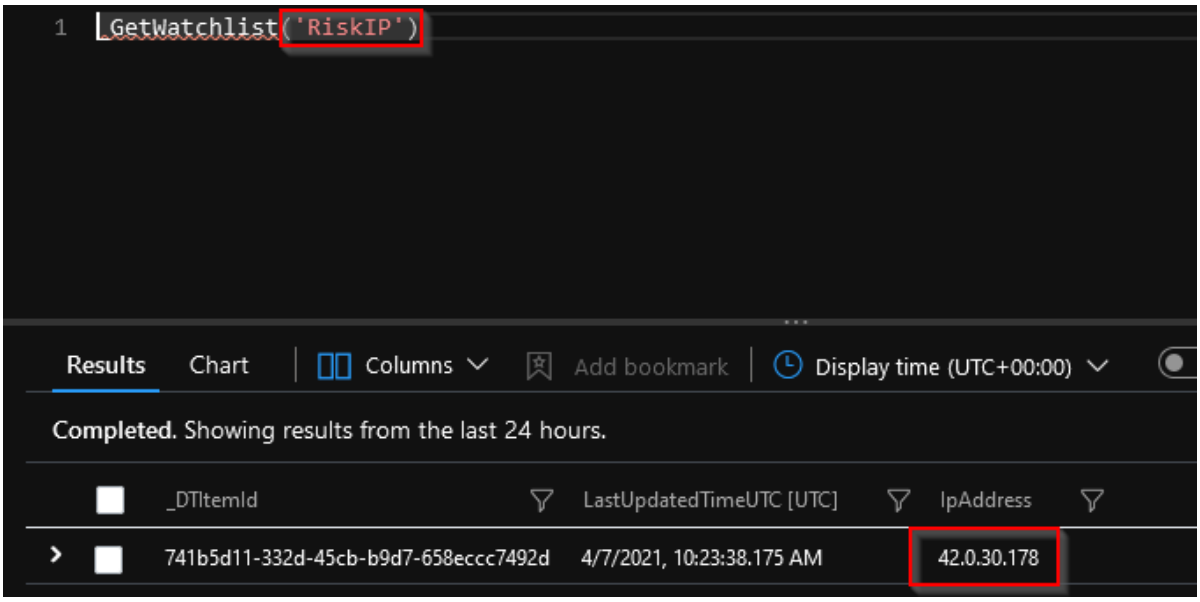
Figur 44 Legge til IP i Watchlist - Logic App

Først henter den IP-adressen fra en hendelse. Deretter legger den inn en del variabler. Så legger man til IP-adressen i en CSV-fil. Nå sjekker Playbooken om Watchlist eksisterer. Til slutt importerer den dataen inn i Watchlist. Under er et bilde over Watchlisten som ble laget. I dette tilfellet er det en IP-adresse for en som prøvde å legge på for mange ganger fra Malaysia:



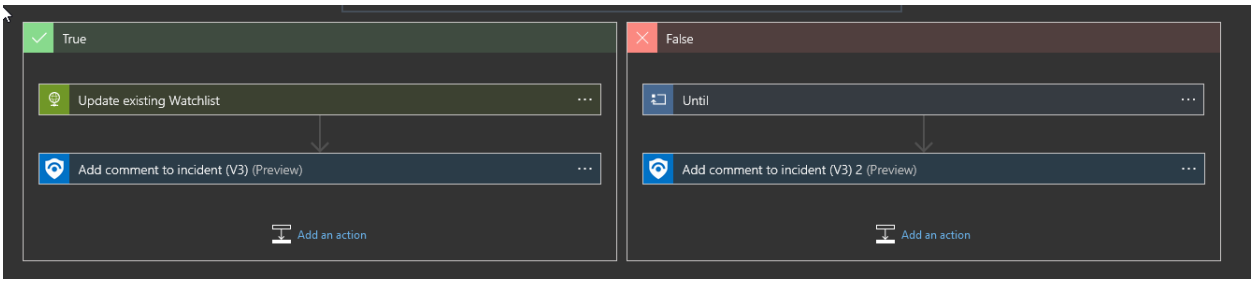
Figur 45 Legge til IP i Watchlist - vise at det funker

Her er en eksempelspørring for å vise at IP-adressen ligger inne i Watchlisten:



Figur 46 Legge til IP i Watchlist - vise at det funker

Som sagt gjør den ikke noe annet enn å legge inn data i en Watchlist. Man må bruke dataen i etterkant for at det skal være noen grunn i å bruke Watchlist. Med små forandringer i denne Playbooken kan man også ta ut brukere, grupper og enheter fra sikkerhetshendelser og legge de i Watchlist. Helt til slutt legger vi ved en Action som legger til en kommentar på hendelsen.



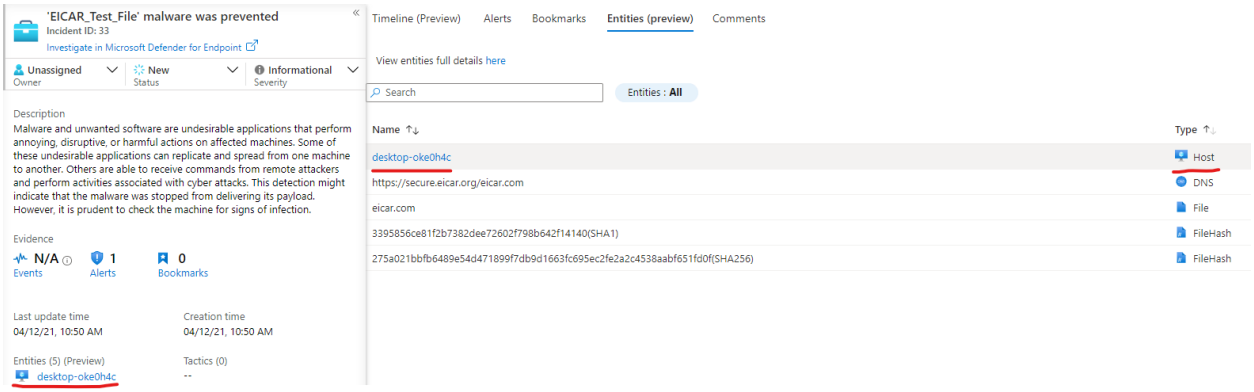
Figur 47 Legge til IP i Watchlist - legger til kommentar

Vi legger inn en kommentar uansett utfallet. Kommentarene spesifiserer om IP-adressen blir lagt til i en eksisterende Watchlist, eller om det opprettes en ny.

Isolere enhet i MDATP ved hjelp av Sentinel (Playbook)

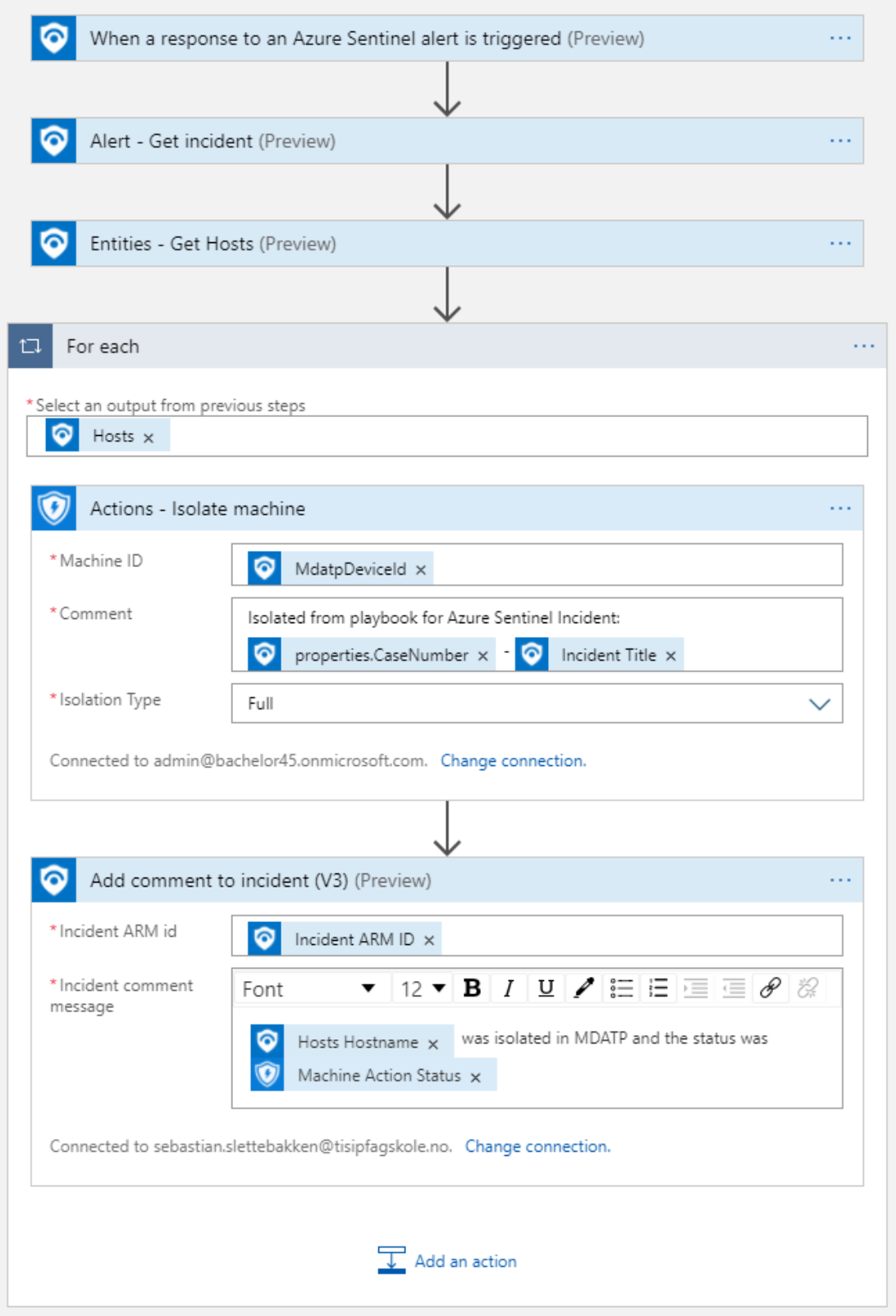
Det kan være nyttig å ha muligheten til å isolere en enhet dersom man merker at den har blitt kompromittert. For å forenkle denne prosessen, velger vi å opprette en Playbook som kan gjøre dette for oss. Playbooken må kjøres manuelt på en hendelse der man ønsker å isolere en enhet som har skapt en hendelse. For å isolere enhetene ved hjelp av Sentinel, har vi opprettet følgende Logic Appen som vist under.

Da den må kjøres manuelt på en hendelse, er vi ikke redde for at maskiner som er koblet til domenet blir automatisk isolert og man ikke får tilgang til de lenger. Logic Appen starter med å hente info om hendelsen den kjøres på. Den henter også info om hvilken enhet som skapte hendelsen, så lenge den entiteten ligger ved i hendelsen:



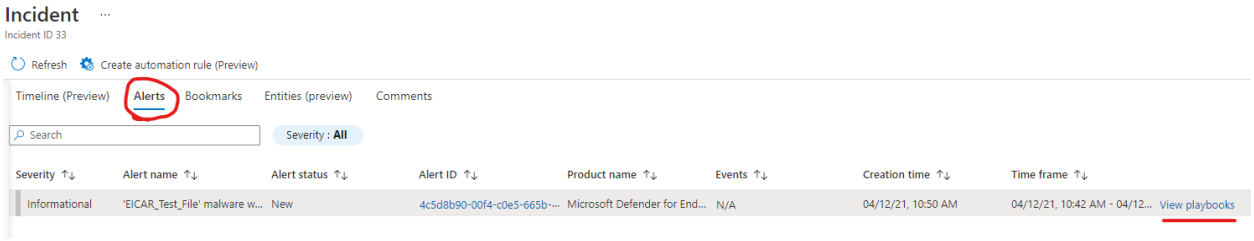
Figur 48 Isolere enhet i MDATP - Entiteter

Deretter isolerer den alle maskinene (entitetene) som er knyttet til hendelsen og legger til en kommentar.



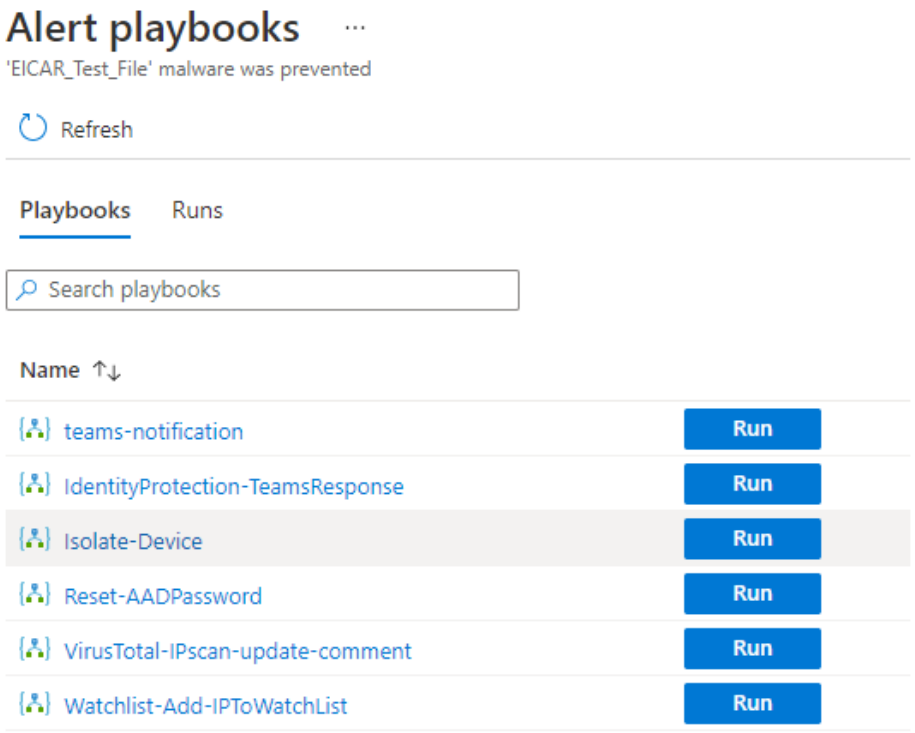
Figur 49 Isolere enhet i MDATP - Logic App

Ved testkjøring av Playbooken, går man inn på en hendelse der en enhet er listet som entitet, trykker på **Alerts** fanen og velger **View Playbooks** knyttet til den:



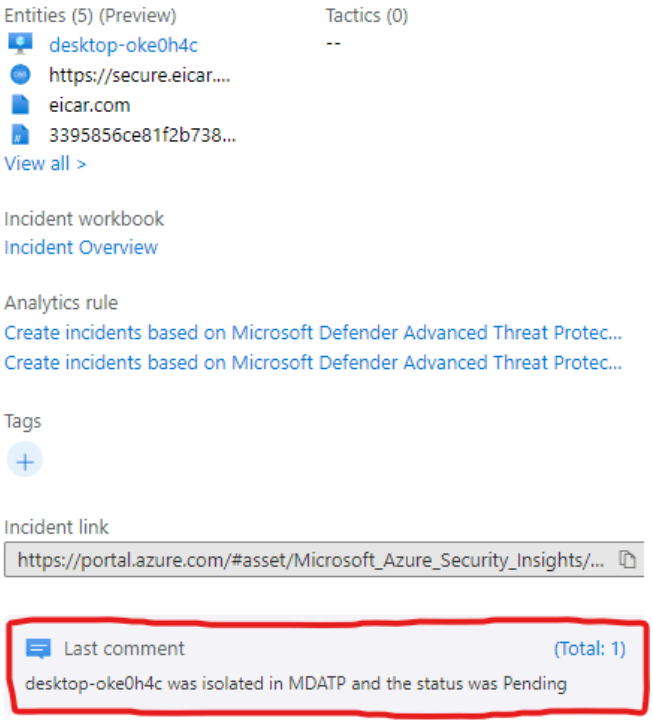
Figur 50 Isolere enhet i MDATP - hvordan kjøre Playbook på hendelse

Får så opp en liste med Playbooks, der man velger den man ønsker å kjøre:



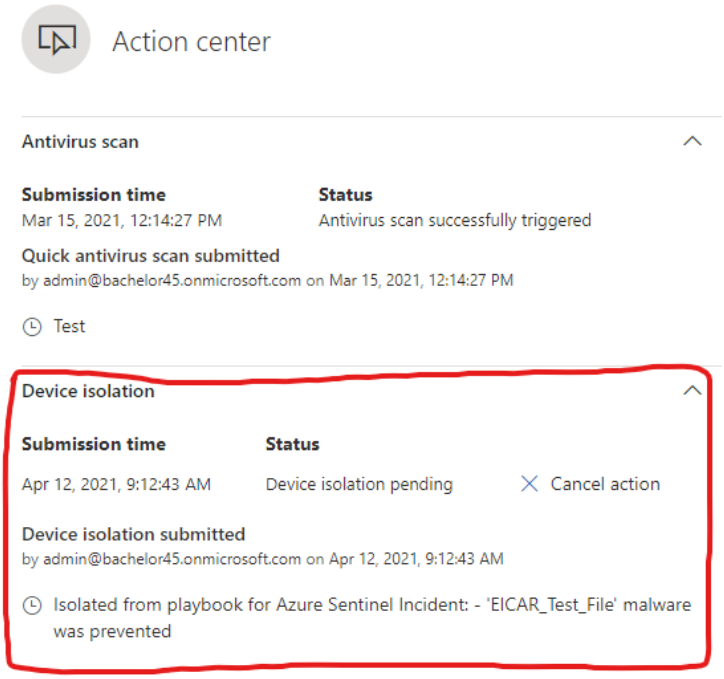
Figur 51 Isolere enhet i MDATP - eksempel kjøre Playbook på hendelse

Velger **Run** på Playbooken **Isolate-Device**. Ser at det ble lagt til en kommentar til hendeslen:



Figur 52 Isolere enhet i MDATP - eksempel kjøring med kommentar for sporbarhet

I MDATP, på enheten «desktop-oke0h4c», ser vi at enheten har blitt isolert:

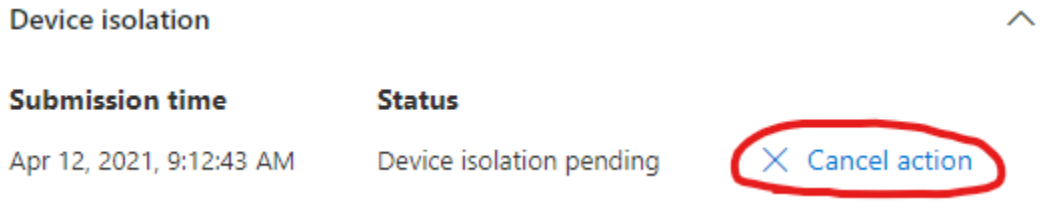


Figur 53 Isolere enhet i MDATP - MDATP enhet status

På enheten ser vi at vi får varsling:

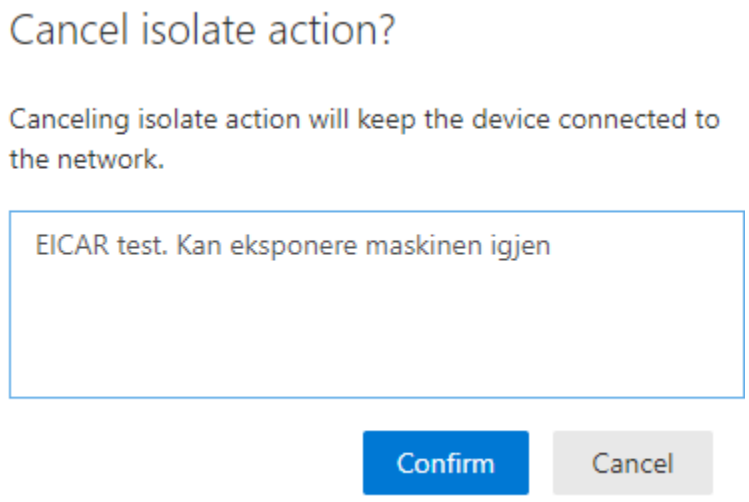
Figur 54 Isolere enhet i MDATP - test-maskin varsling

For å legge enheten tilbake (eksponere), velger vi **Cancel action**:



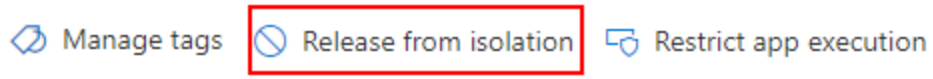
Figur 55 Isolere enhet i MDATP - kansellere isolering

Blir bedt om å skrive en liten kommentar:



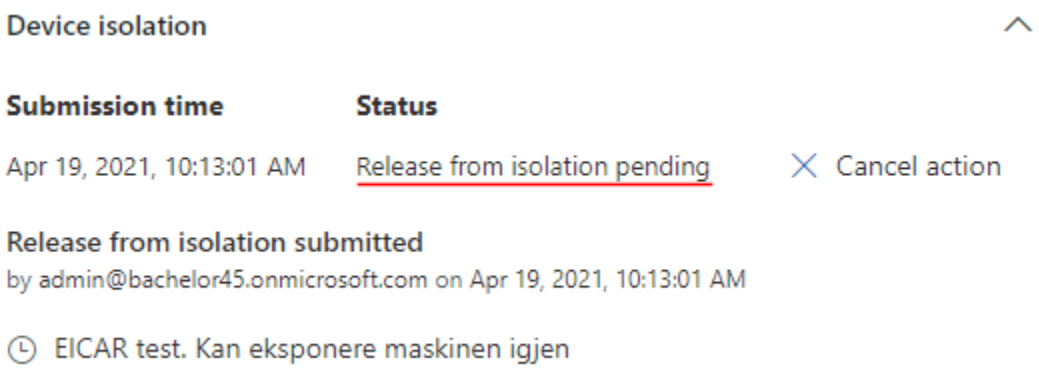
Figur 56 Isolere enhet i MDATP - kansellere isolering kommentar

Kan så velge å «Release from isolation»:



Figur 57 Isolere enhet i MDATP - frigjøre fra isolering

Får så en oppdatert status-melding:

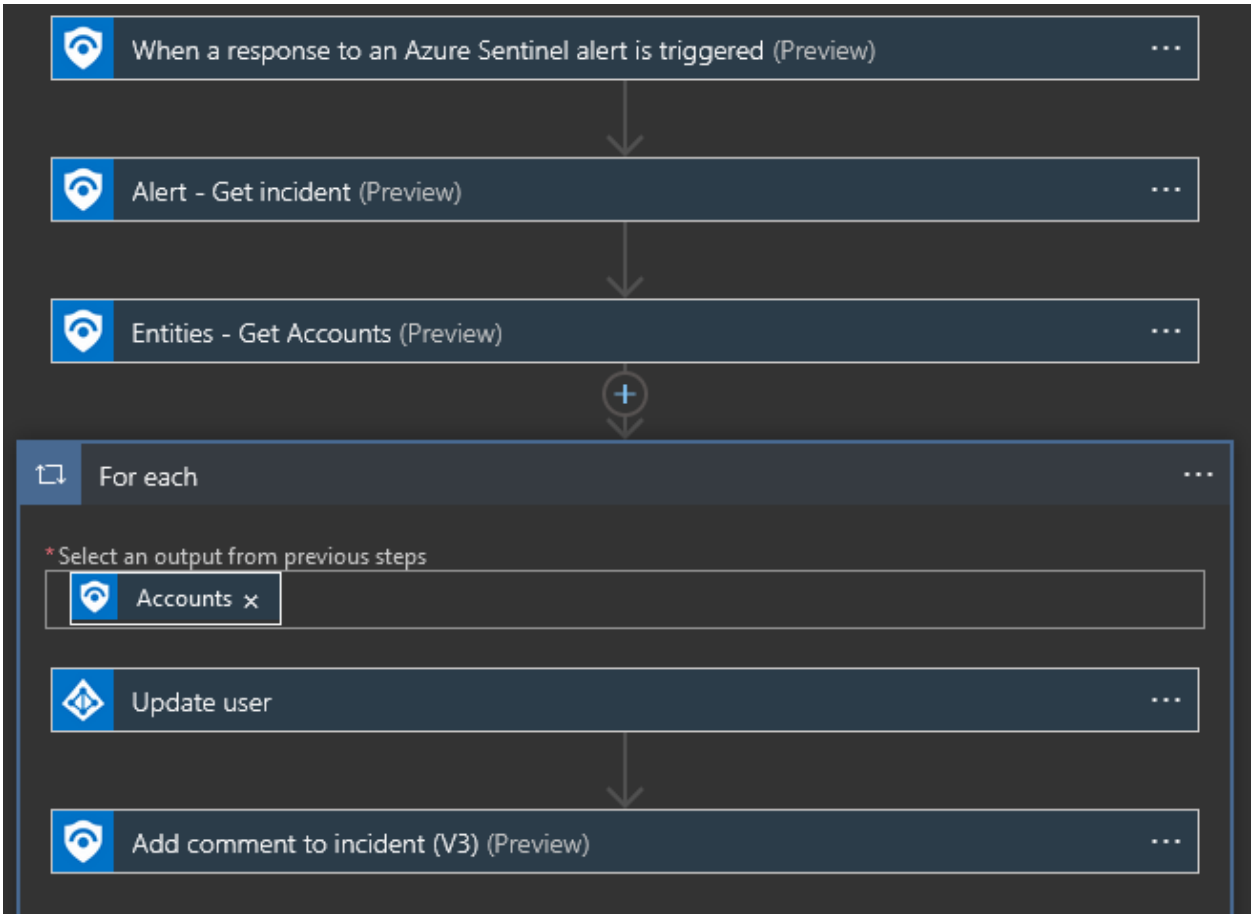


Figur 58 Isolere enhet i MDATP - frigjøre fra isolering status

Blokkere bruker

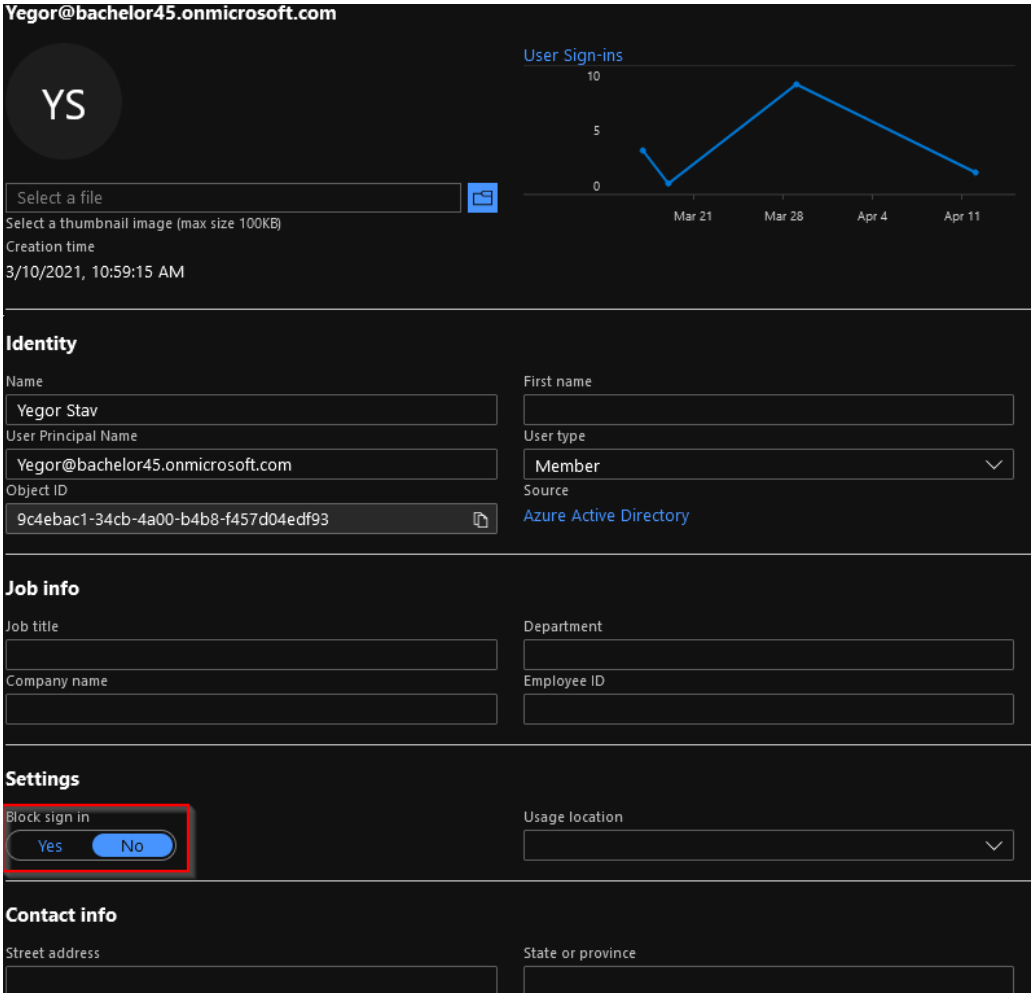
Noen ganger må man handle raskt om man finner ut at en bruker er for eksempel komprimert. En Playbook for å raskere blokkere en bruker. Playbooken kan også settes opp til å kjøre automatisk, men dette må man være forsiktig med. I Azure AD kan man bestemme om en bruker skal få lov til å logge inn eller ikke.

I Logic App designer er det en gruppe med actions knyttet til Azure AD. Den vi er på jakt etter heter Update User. Her får man valget om man skal Enable eller Disable bruker. Vi velger å Disable bruker og legger til en kommentar ved hendelsen.



Figur 59 Blokkere bruker - Logic App

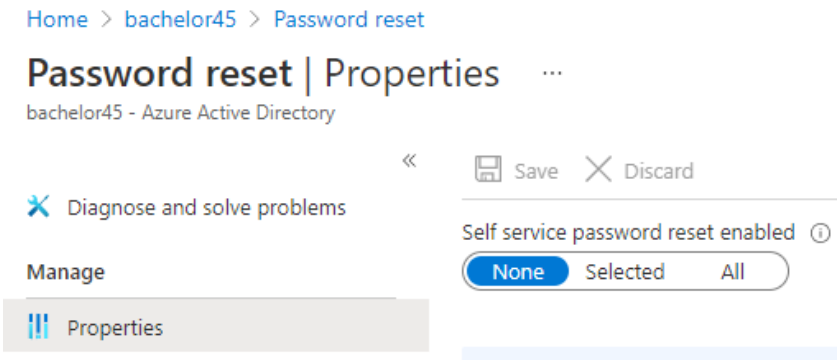
Nå som brukeren ikke lenger kan gjøre noe, kan man etterforske om hva som faktisk hendte. Om man kommer frem til at det er en falsk-posistiv kan man Enable brukeren. Dette gjør man enten gjennom å lage en ny Playbook som gjør det samme som denne bare motsatt. Eller så kan man gå inn i Azure AD og manuelt sette innstillingen. Som vist på bildet under:



Figur 60 Blokkere bruker - vise at det funker

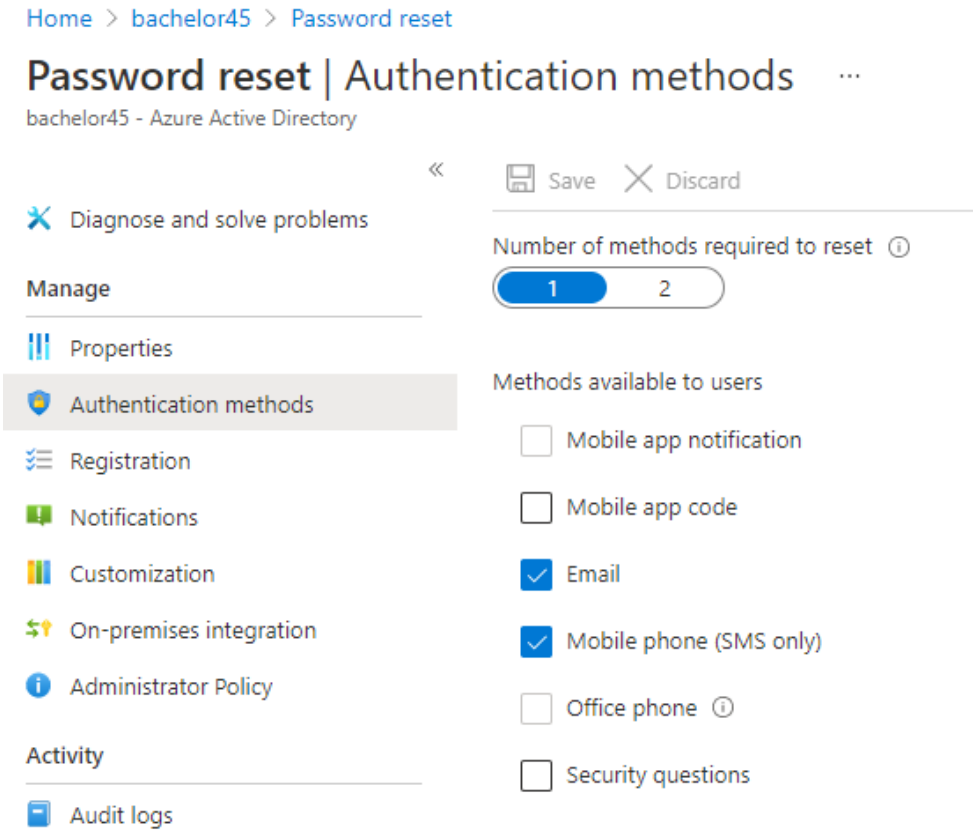
Tilbakestill passord til kompromittert bruker (Playbook)

Det er mulig å skru på «self service password reset» i Azure AD, som vist under:



Figur 61 Tilbakestill passord - Self service password reset

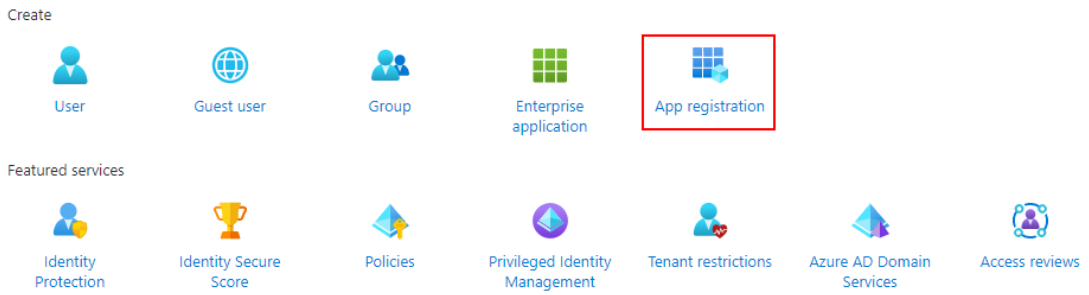
Men det krever en av følgende autentiseringsmetoder:



Figur 62 Tilbakestill passord - Self service password reset metoder

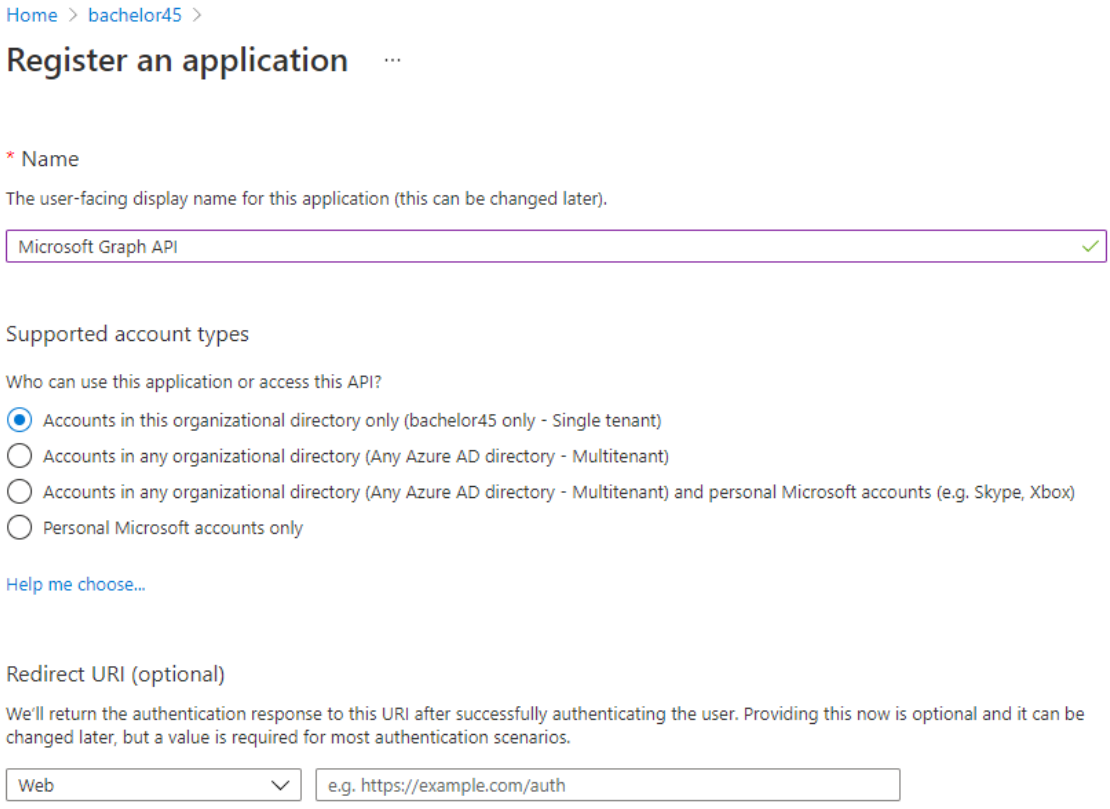
Da brukerne i Azure AD domenet (@bachelor45.onmicrosoft.com) vårt ikke har epost, og vi ikke vil knytte mobilnumrene våre til disse brukerne, velger vi å ikke skru på denne funksjonaliteten.

Denne Playbooken krever et litt mer avansert oppsett i forkant, da vi tar i bruk Microsoft Graph API for å tilbakestille passordet til en bruker i Azure AD. Starter med å gå inn på **Azure AD** og velger **App registration**:



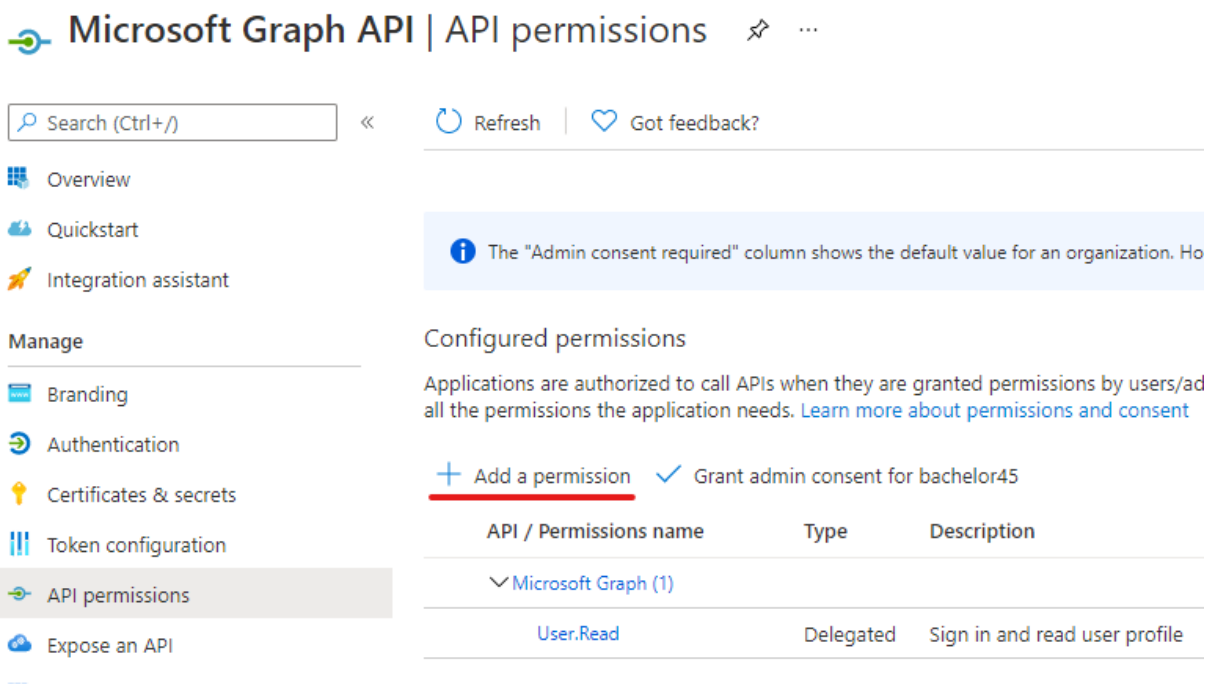
Figur 63 Tilbakestill passord - App registration

Setter følgende innstillinger:



Figur 64 Tilbakestill passord - Microsoft Graph API app registration

Legger så til API rettigheter ved å gå inn på **API permissions**:



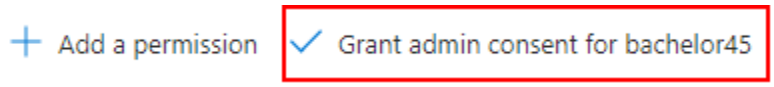
Figur 65 Tilbakestill passord - Microsoft Graph API legge til rettigheter

Der har vi lagt til følgende rettigheter:

API / Permissions name	Type	Description	Admin consent req...	Status
Microsoft Graph (6)				
Directory.AccessAsUser.All	Delegated	Access directory as the signed in user	Yes	Granted for bachelor45
Directory.ReadWrite.All	Application	Read and write directory data	Yes	Granted for bachelor45
User.ManageIdentities.All	Delegated	Manage user identities	Yes	Granted for bachelor45
User.ManageIdentities.All	Application	Manage all users' identities	Yes	Granted for bachelor45
User.Read	Delegated	Sign in and read user profile	No	Granted for bachelor45
UserAuthenticationMethod.Re	Delegated	Read and write all users' authentication methods.	Yes	Granted for bachelor45

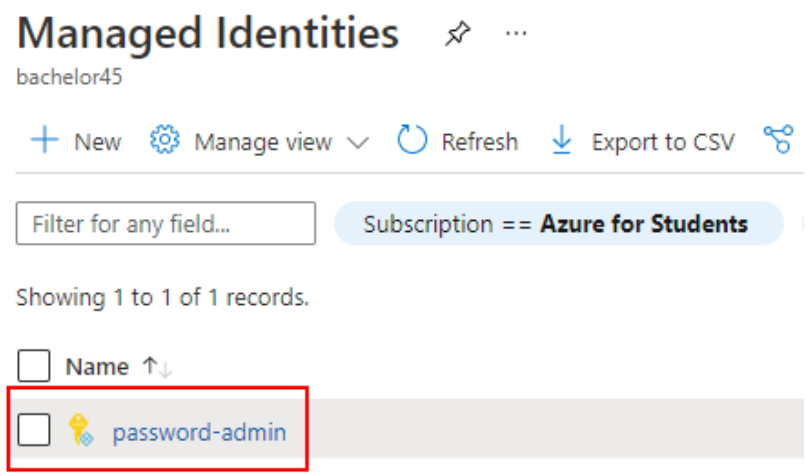
Figur 66 Tilbakestill passord - Microsoft Graph API rettigheter

Må huske å godkjenne «admin consent» ved å klikke på knappen:



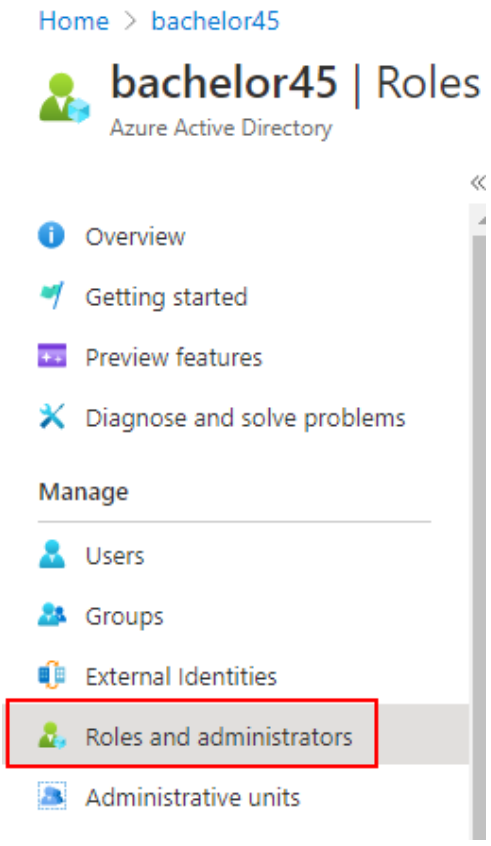
Figur 67 Tilbakestill passord - Microsoft Graph API rettigheter krever godkjenning fra admin

Vi må deretter opprette en **Azure Managed Identity** som skal brukes til å kjøre en http request i Playbooken. Går inn på **Azure Managed Identities** og oppretter en ny som vi kaller «password-admin»:



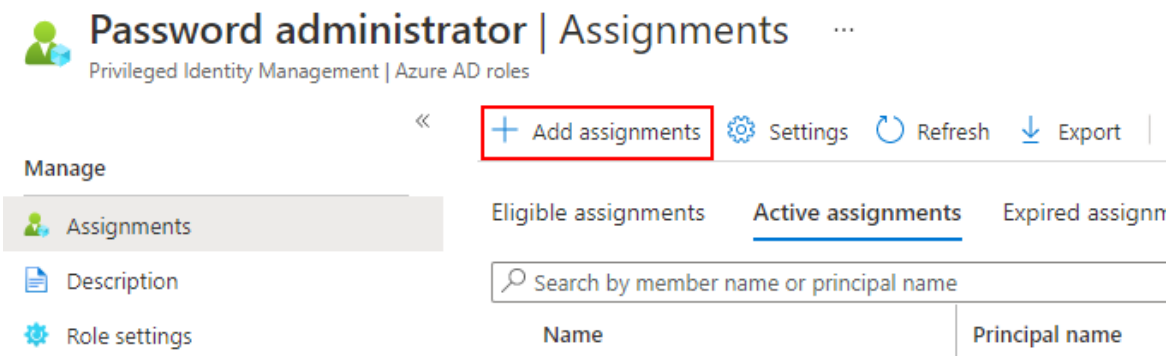
Figur 68 Tilbakestill passord - Azure Managed Identity opprettelse

Vi har gitt denne identiteten rollen som «Password administrator» ved å gå inn på **Azure AD** og velge **Roles and administrators**:

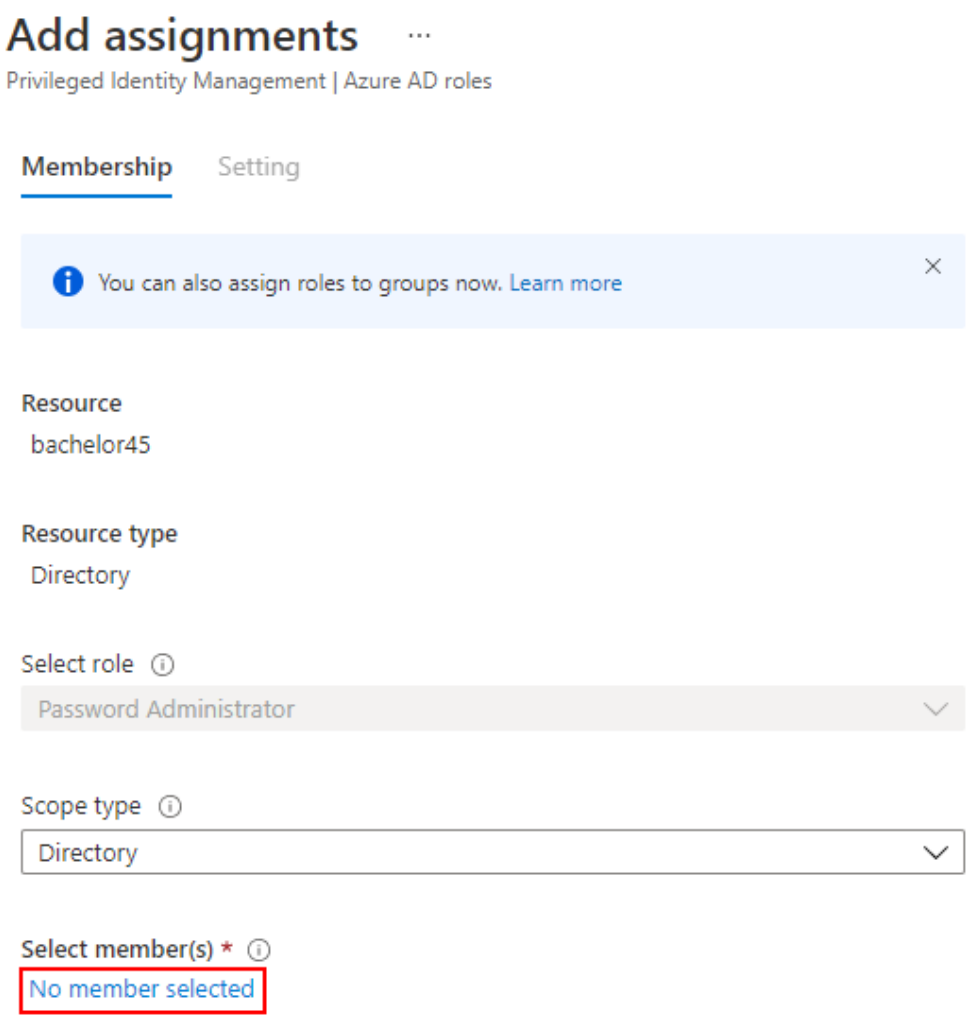


Figur 69 Tilbakestill passord - Azure AD roller for identitet

Finner frem til «Password administrator» og velger å legge til medlem:

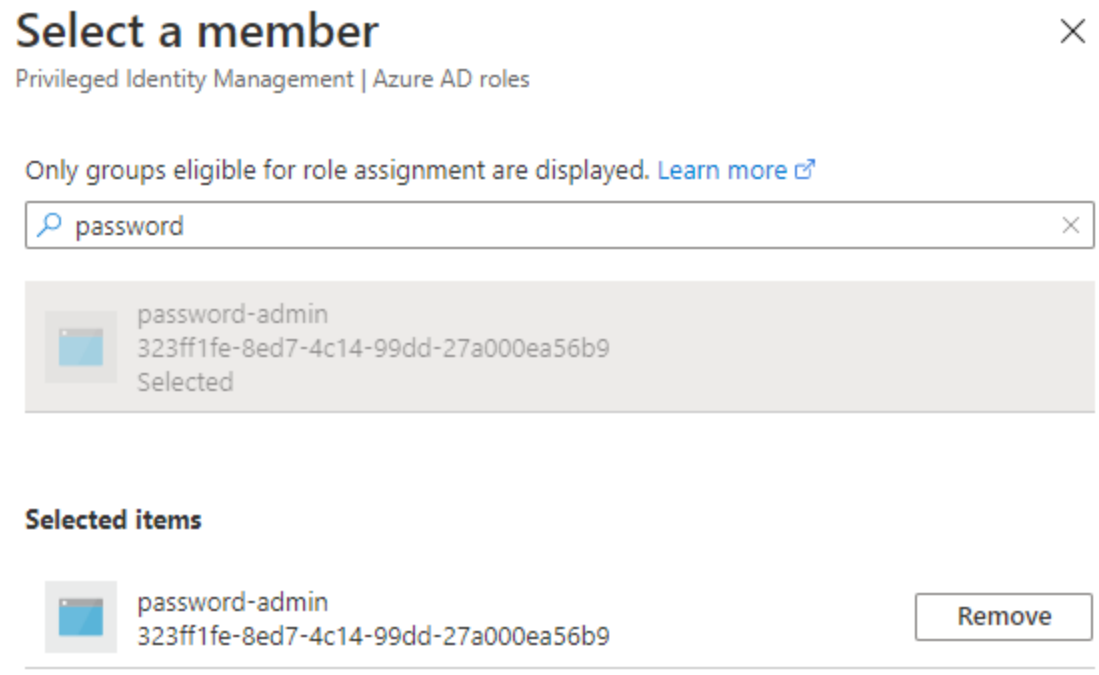


Figur 70 Tilbakestill passord - Azure AD passord admin



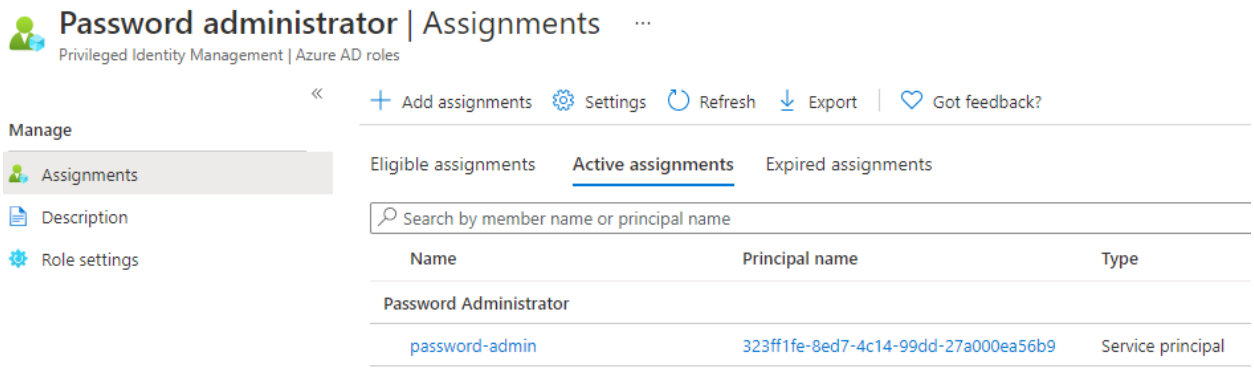
Figur 71 Tilbakestill passord - Azure AD finne identitet

Finner så identiteten vi har opprettet:



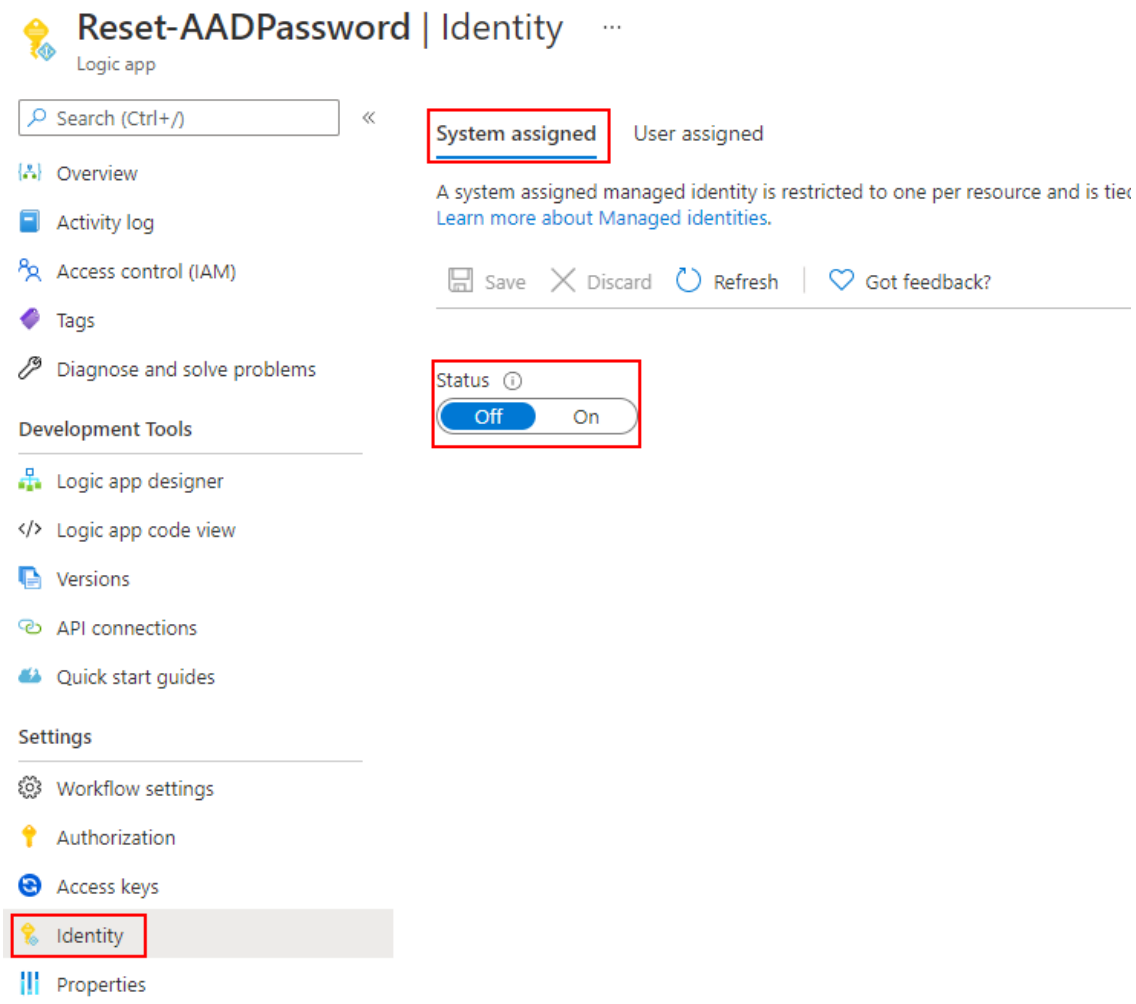
Figur 72 Tilbakestill passord - Azure AD passord admin til identitet

Og ser at identiteten har fått rettigheten:



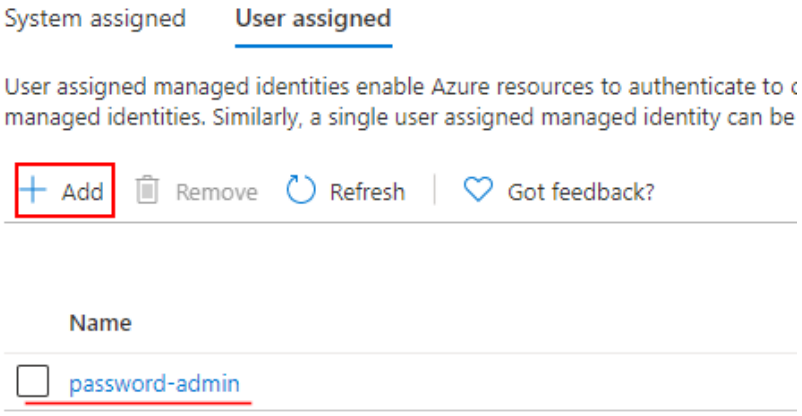
Figur 73 Tilbakestill passord - Azure AD passord admin oversikt

Nå kan vi opprette Playbooken. Før vi legger til logikken i Playbooken, må vi konfigurere identiteten den kjøres med. Velger å skru av «System assigned» som vist under:



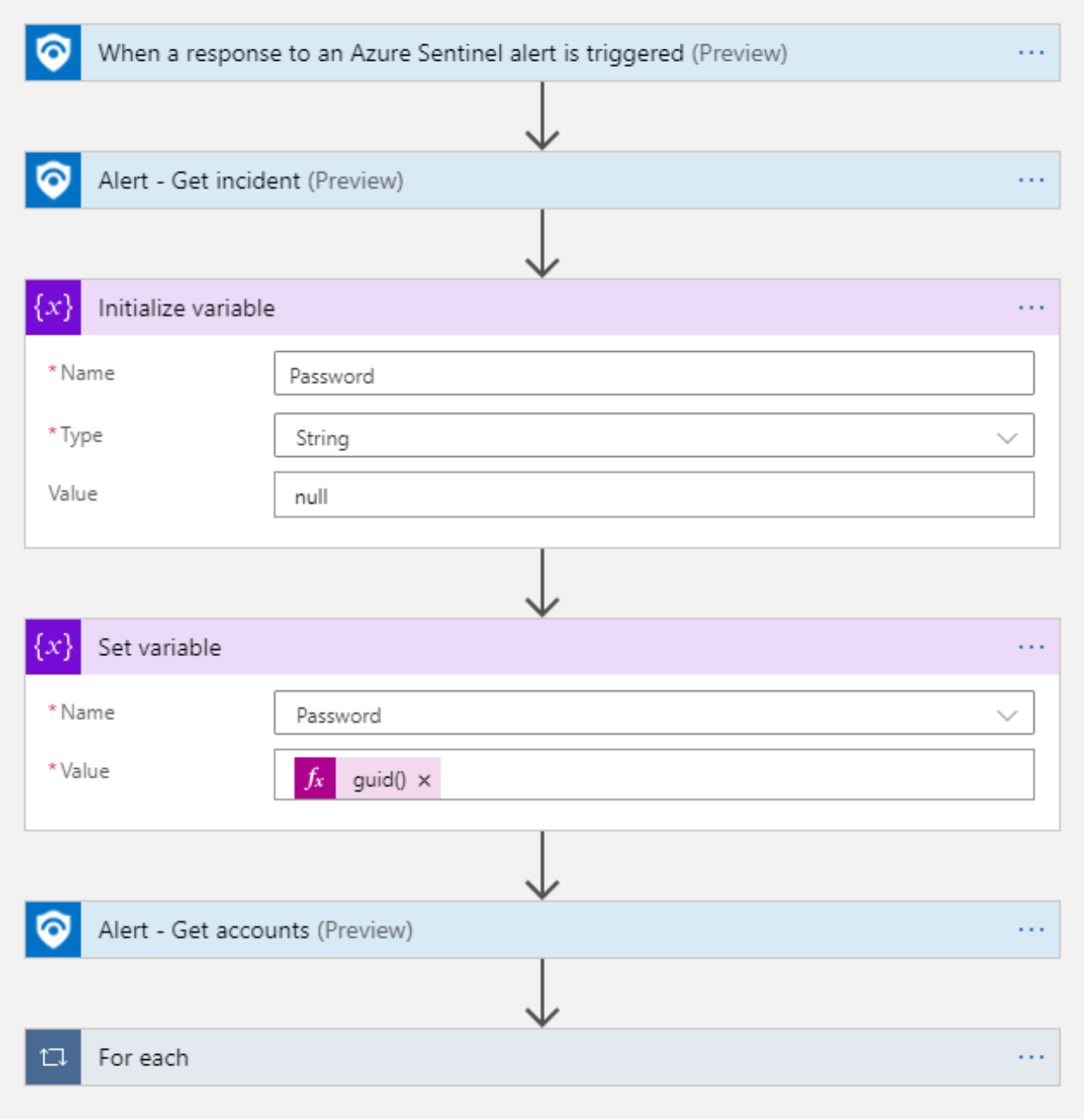
Figur 74 Tilbakestill passord - Logic App identitet

Vi kan så skru på «User assigned» identitet og velger den vi har opprettet i foregående steg:



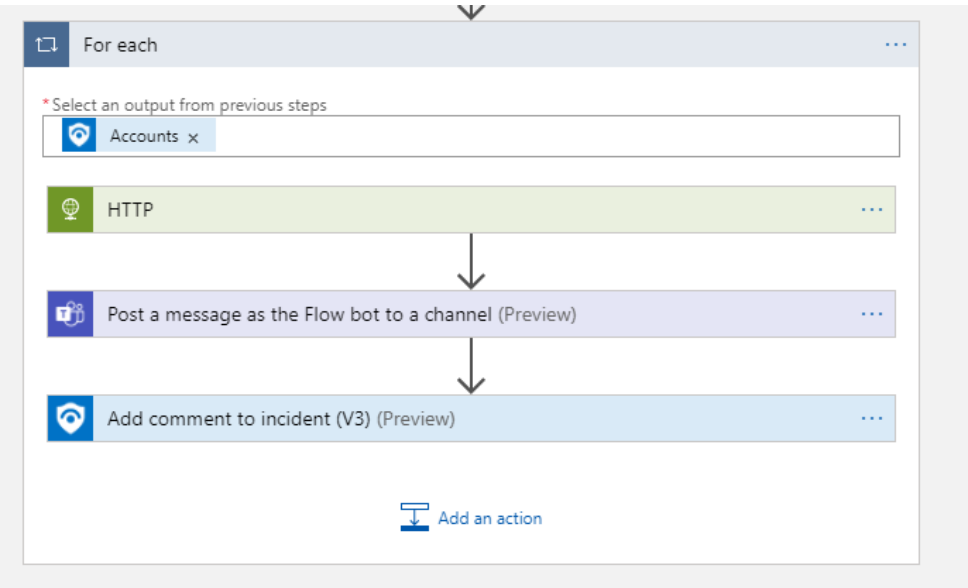
Figur 75 Tilbakestill passord - Logic App identitet legger til passord admin

Så kan vi se på logikken i Playbooken. Vi oppretter et tilfeldig passord som skal brukes for å tilbakestille passordet til brukeren neste gang h*n logger på. Playbooken ser slik ut:



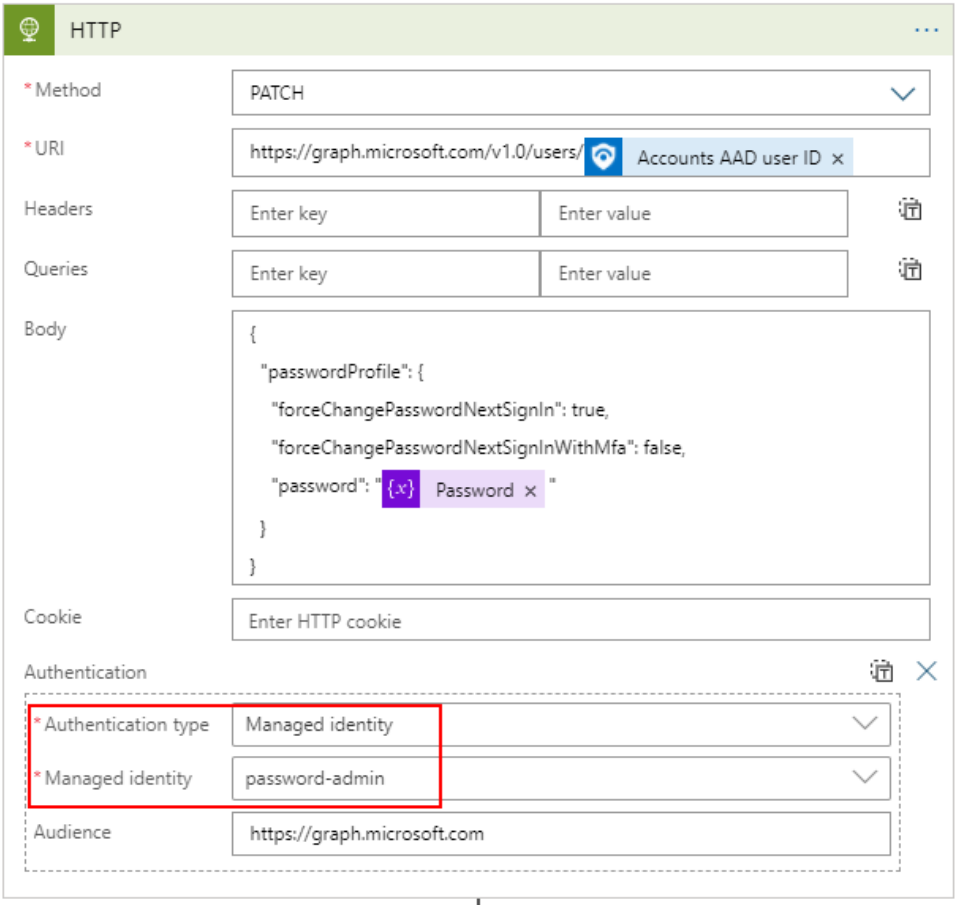
Figur 76 Tilbakestill passord - Logic App oversikt

Vi henter info om brukeren ved hjelp av **Alert – Get accounts (preview)**. Denne Playbooken kjøres manuelt på en hendelse der man ønsker å tilbakestille passordet på brukeren involvert. Dersom det er flere brukere involvert, vil alle få tilbakestilt passordet sitt ved at vi bruker en «for each» løkke. Den ser slik ut:



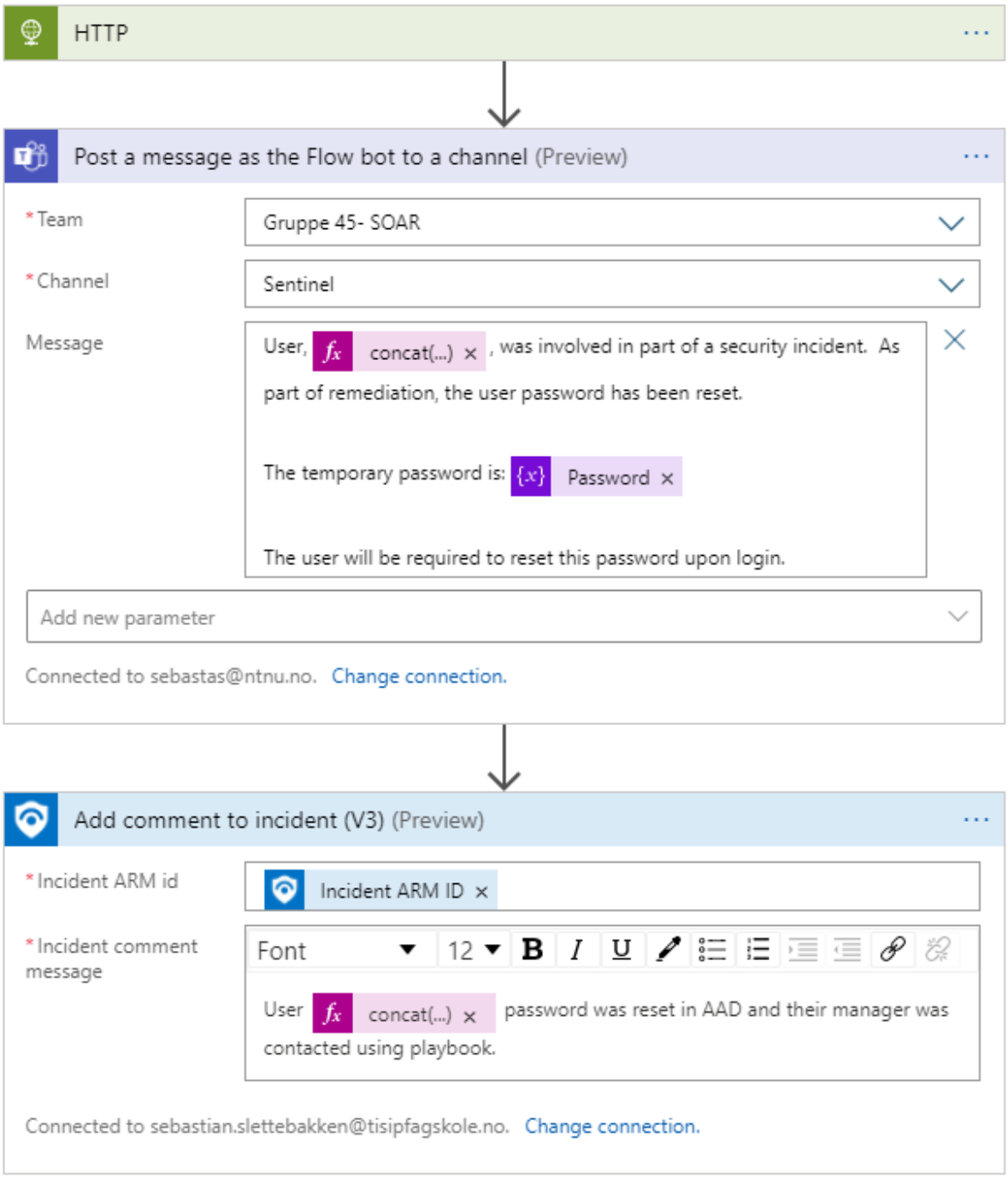
Figur 77 Tilbakestill passord - Logic App for hver bruker/entitet

For hver bruker (entitet), kaller den en http-forespørsel som ser slik ut:



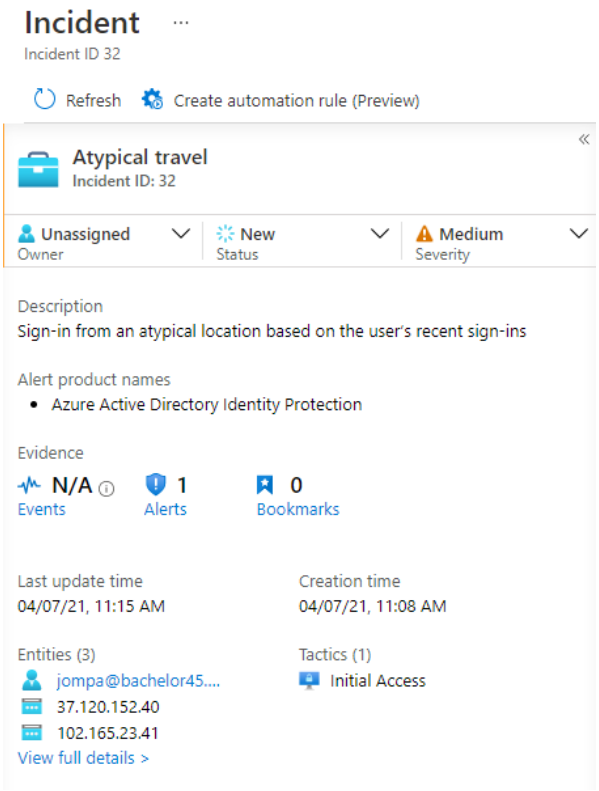
Figur 78 Tilbakestill passord - Logic App http-forespørsel til passord admin identitet

Her er det viktig å sette **Authentication type** til å være «Managed identity» og så sette **Managed identity** til å være den vi la til i forrige steg. I vårt tilfelle er dette identiteten «password-admin». Hvis man bruker en «system assigned» identitet vil man få en 403-feilmelding ved at man ikke har riktig tilgang til å kalle på Microsoft Graph API-et. Videre i løkken, informerer vi i Teams-kanalen med det nye, midlertidige passordet og legger til en kommentar i hendelsen:



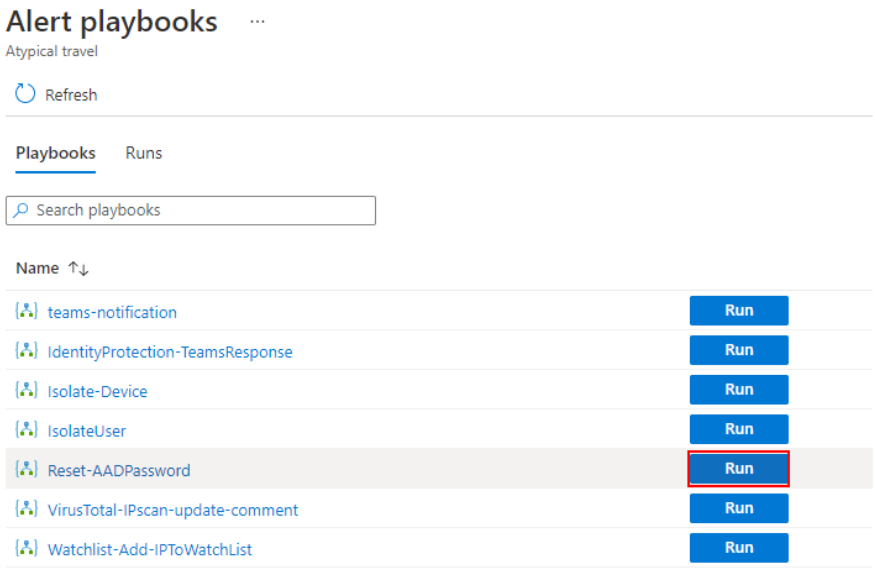
Figur 79 Tilbakestill passord - Logic App sporbarhet

Velger å teste Playbooken på en hendelse der vi mistenker at en bruker kan ha blitt kompromittert:



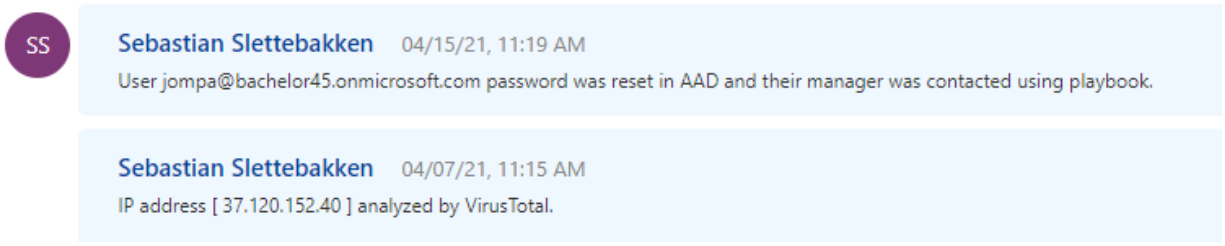
Figur 80 Tilbakestill passord - eksempel på kjøring

Velger å kjøre Playbooken (manuelt) for å tilbakestille passordet for den aktuelle brukeren:



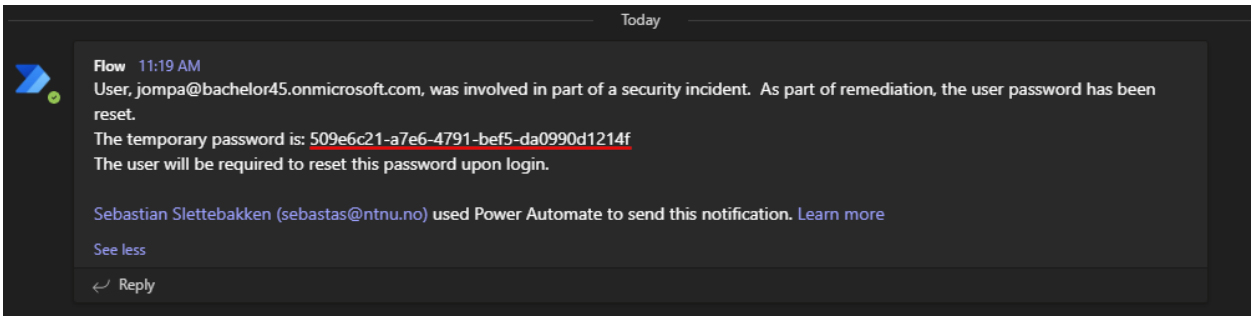
Figur 81 Tilbakestill passord - velge Playbook å kjøre

Kommentar blir lagt ved hendelsen:



Figur 82 Tilbakestill passord - kommentar blir lagt til ved kjøring

Ser at vi får opp en melding i Teams-kanalen med det midlertidige passordet:

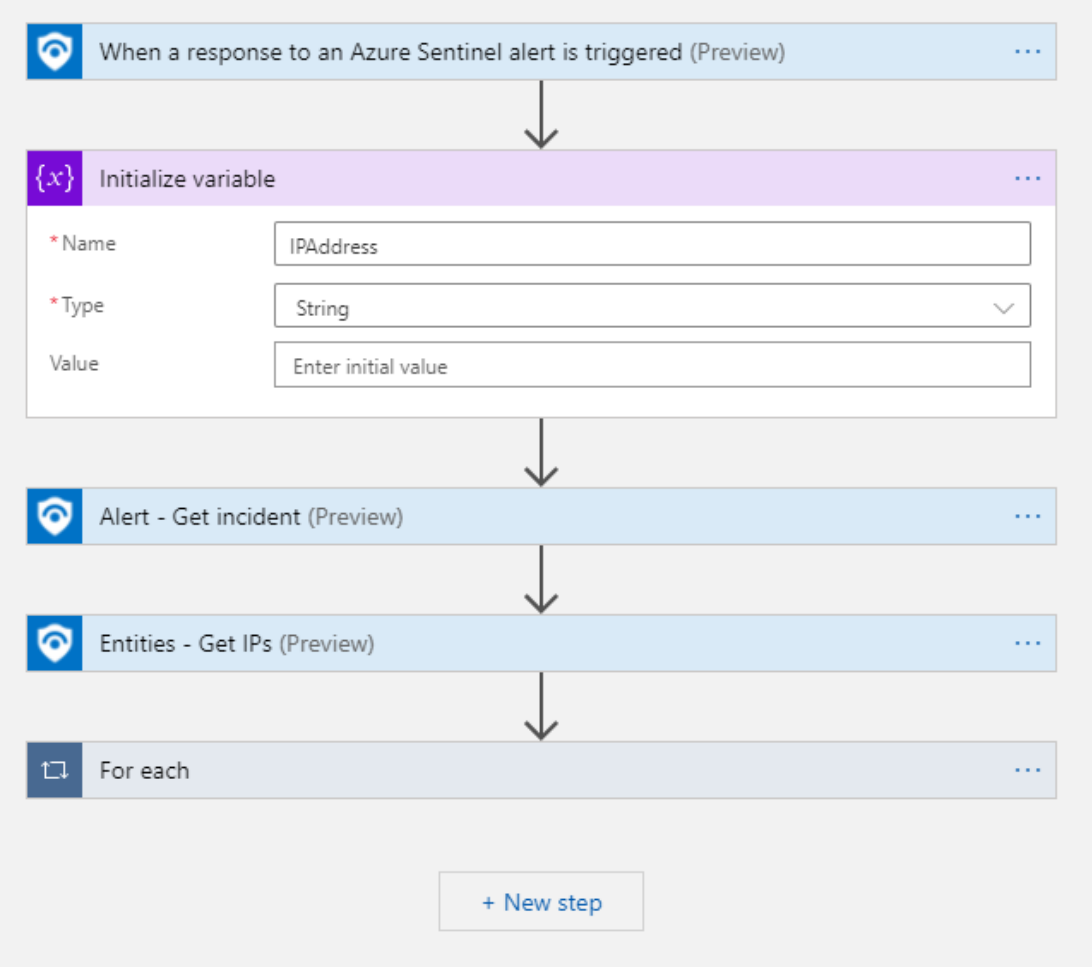


Figur 83 Tilbakestill passord - Teams-varsel

Administrator må formidle dette midlertidige passordet til den aktuelle brukeren på en forsvarlig måte. Da brukeren sitt passord har forandret seg, nytter det ikke å sende på epost. (Vi har dessuten ikke Office 365 lisens å dele ut til brukerne i test-miljøet vårt). Brukeren må bruke passordet listet over, for så å kunne opprette et nytt passord neste gang brukeren skal logge inn.

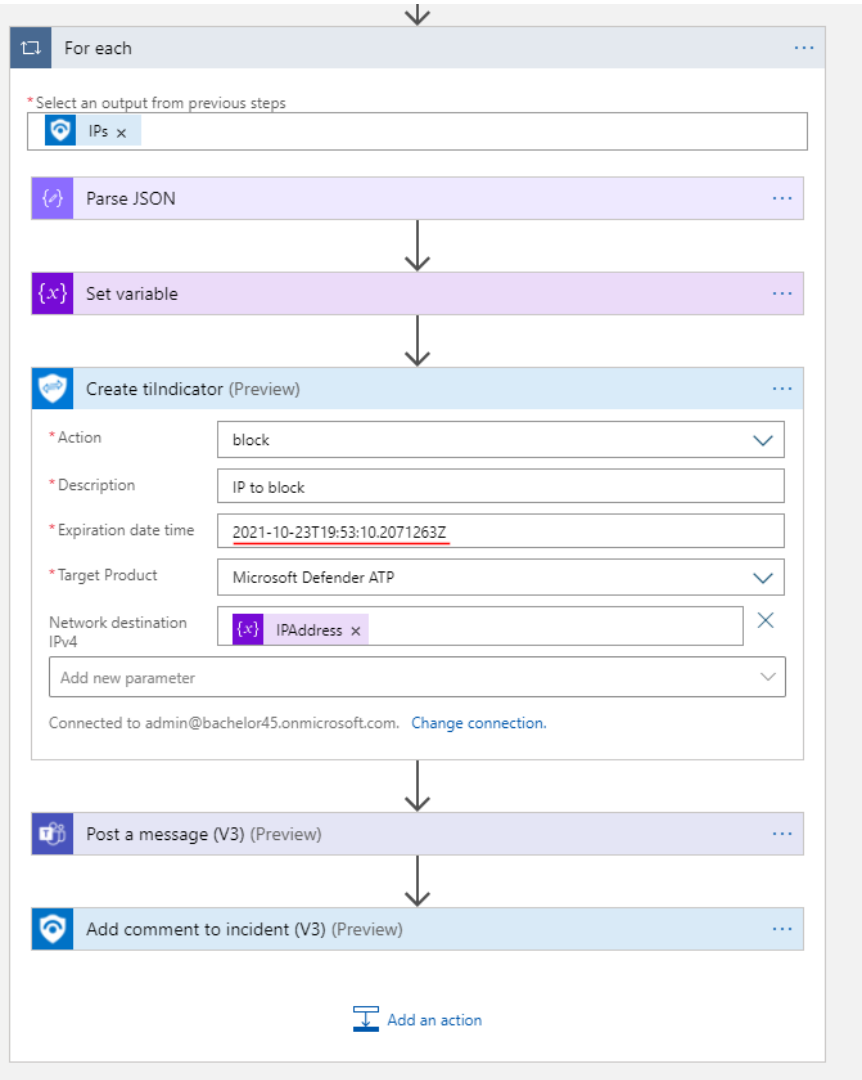
Blokkere IP i MDATP (Playbook)

Det kan være nyttig å blokkere IP-adresser i MDATP dersom man ser det er aktivitet på en ukjent adresse. Vi kan forenkle prosessen ved å opprette en Playbook som legger inn en regel automatisk ut ifra IP-adressen(e) knyttet til en hendelse i Sentinel. Playbooken tar i bruk Microsoft Graph Security og ser slik ut:



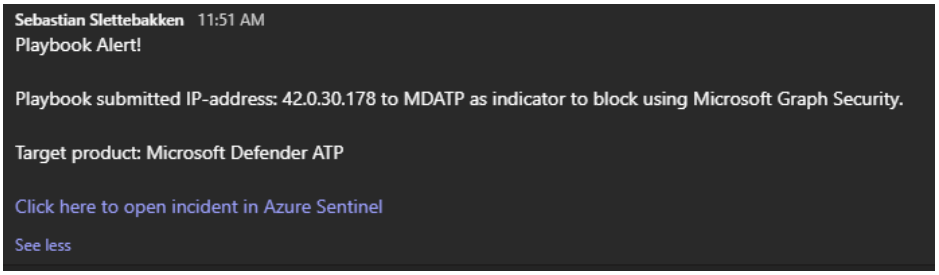
Figur 84 Blokkere IP i MDATP - Logic App oversikt

Vi har en for-løkke som legger til alle IP-adressene knyttet til hendelsen (entiteter). For-løkken formaterer IP-en og setter det til en variabel før den kaller på Microsoft Graph Security for å legge til en indikator. Det går også an å legge til parametere som alvorlighetsgrad (severity), da den foreløpig kun oppretter «Informational» alerts. Løkken ser slik ut:



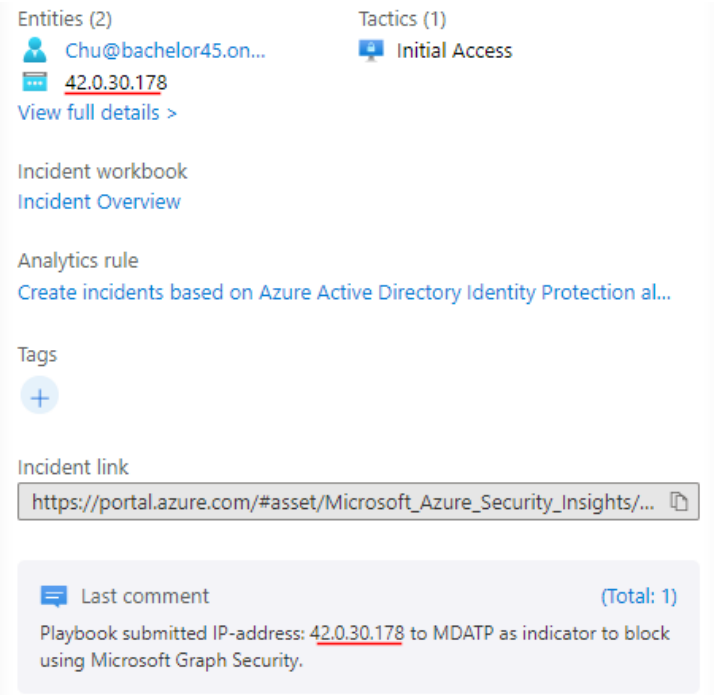
Figur 85 Blokkere IP i MDATP - for hver IP/entitet

For Microsoft Graph Security kreves det en utløpsdato. Denne har vi satt til å være 23. oktober 2021. Det må i så fall fornyes dersom man ønsker å blokkere IP-en lengre enn dette, eller endre verdien i Playbooken. I tillegg blir det sendt en melding in Teams-kanalen:



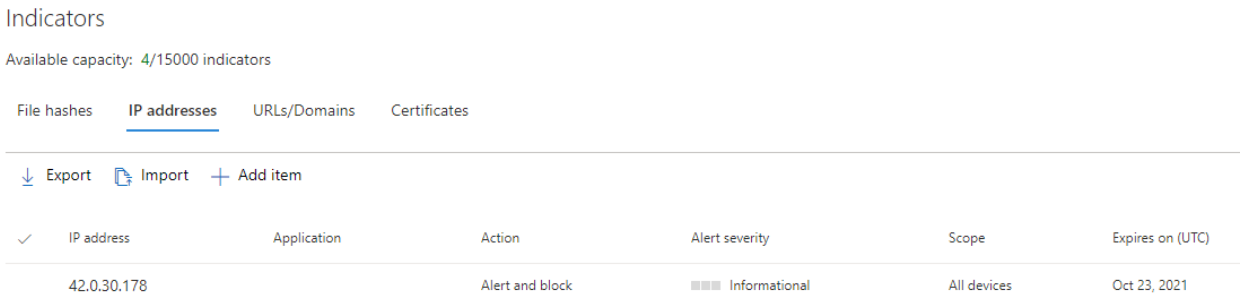
Figur 86 Blokkere IP i MDATP - Teams-varsel

Og det blir lagt til en kommentar til hendelsen:



Figur 87 Blokkere IP i MDATP - kommentar i Sentinel

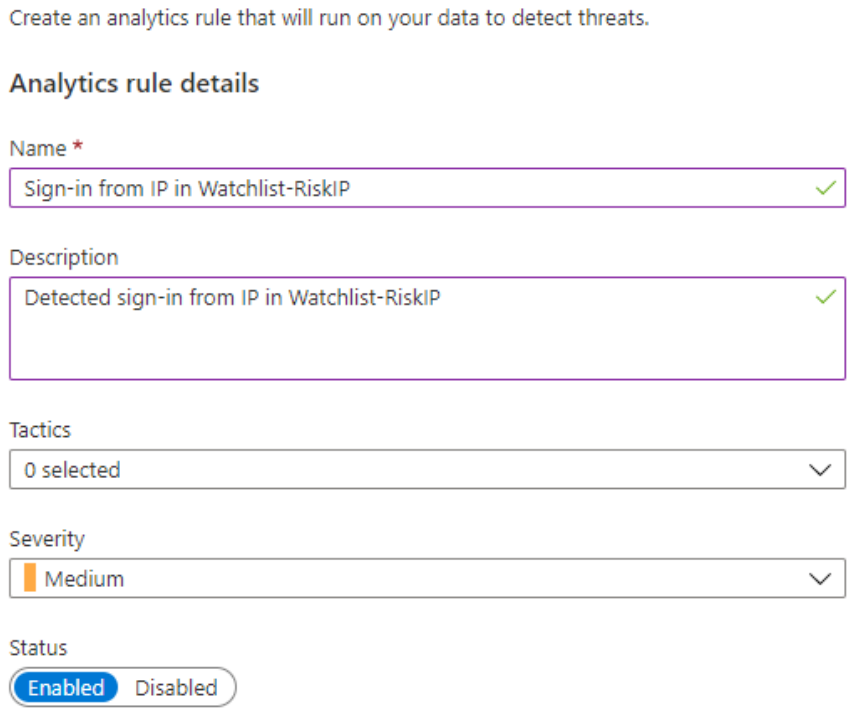
Vi kan også se at IP-en har blitt lagt til i MDATP. Klikker inn på **Settings** -> **Rules** -> **Indicators** og velger **IP addresses**. Der ser vi IP-adressen og utløpsdatoen:



Figur 88 Blokkere IP i MDATP - vise at det funker i MDATP

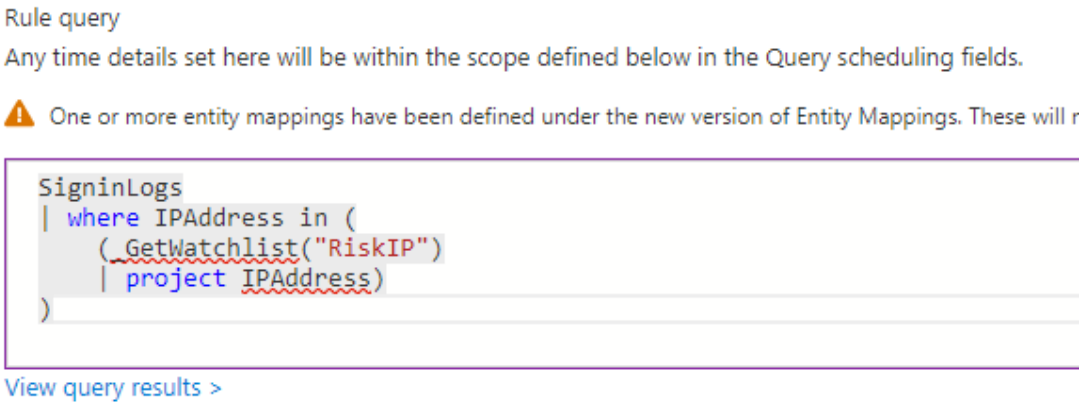
Oppdage pålogging fra IP i Watchlist-RiskIP (Analytic Rule)

Oppretter Analytic Rule med følgende innstillinger:



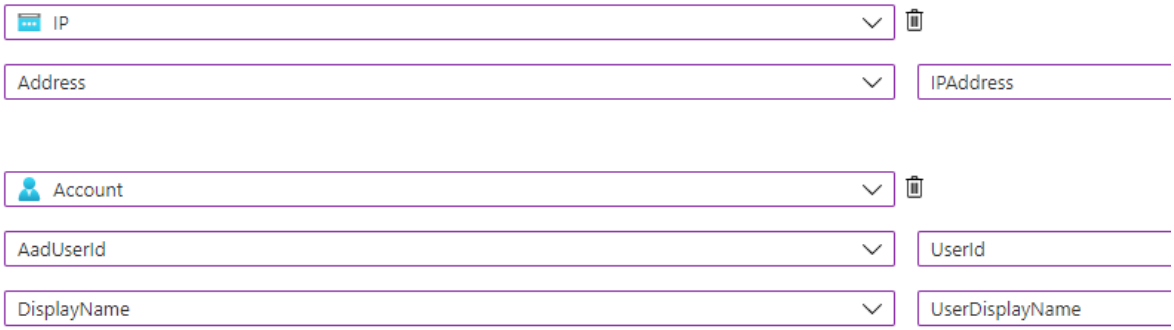
Figur 89 Oppdage pålogging fra IP i Watchlist - Analytic rule details

Setter følgende Rule query:



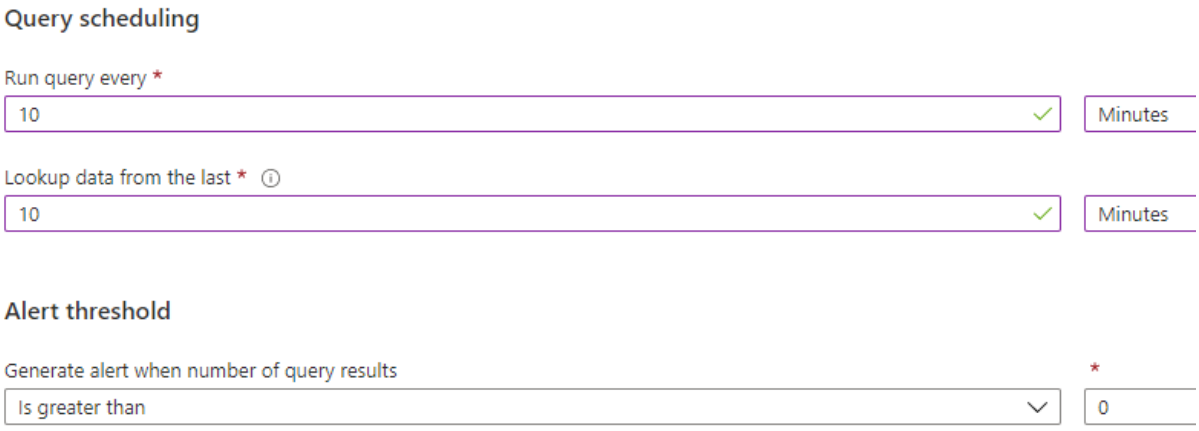
Figur 90 Oppdage pålogging fra IP i Watchlist - KQL query

For **Alert enrichment** kan vi mappe opp følgende entiteter:



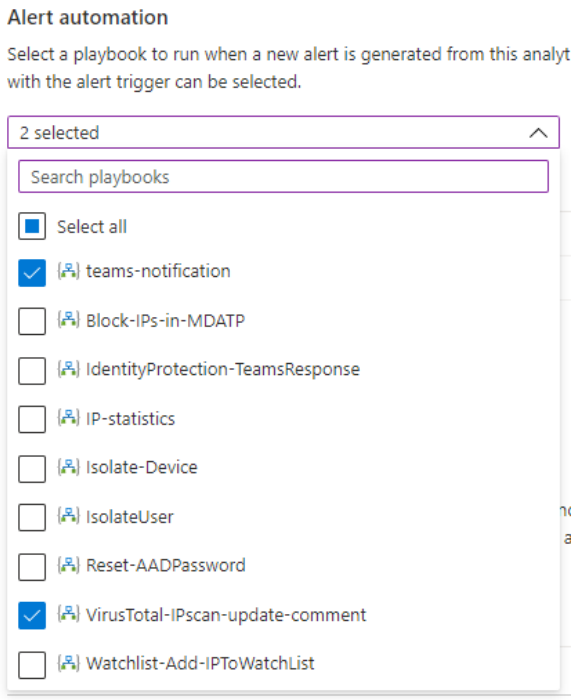
Figur 91 Oppdage pålogging fra IP i Watchlist - Alert enrichment

Setter **Query scheduling** til hvert tiende minutt:



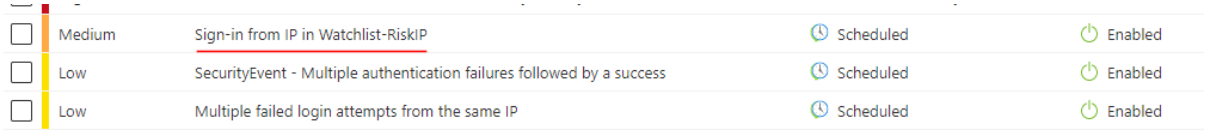
Figur 92 Oppdage pålogging fra IP i Watchlist - Query scheduling

For **Automated response** velger vi å huke av for varsling i Teams-kanalen og kjøre en VirusTotal-IPscan. Da IP-en allerede eksisterer i Watchlist, vet man at den er Risky, men vi velger kun å varsle om påloggingen. Respons til en slik hendelse kan være å blokke IP-adressen, brukeren og/eller isolere enheten dersom man kan det.



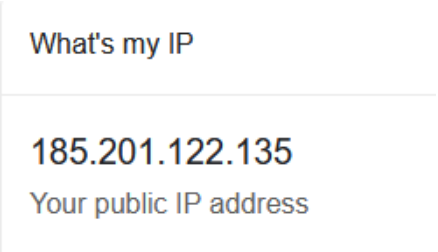
Figur 93 Oppdage pålogging fra IP i Watchlist - Alert automation

Ser at regelen ble lagt til



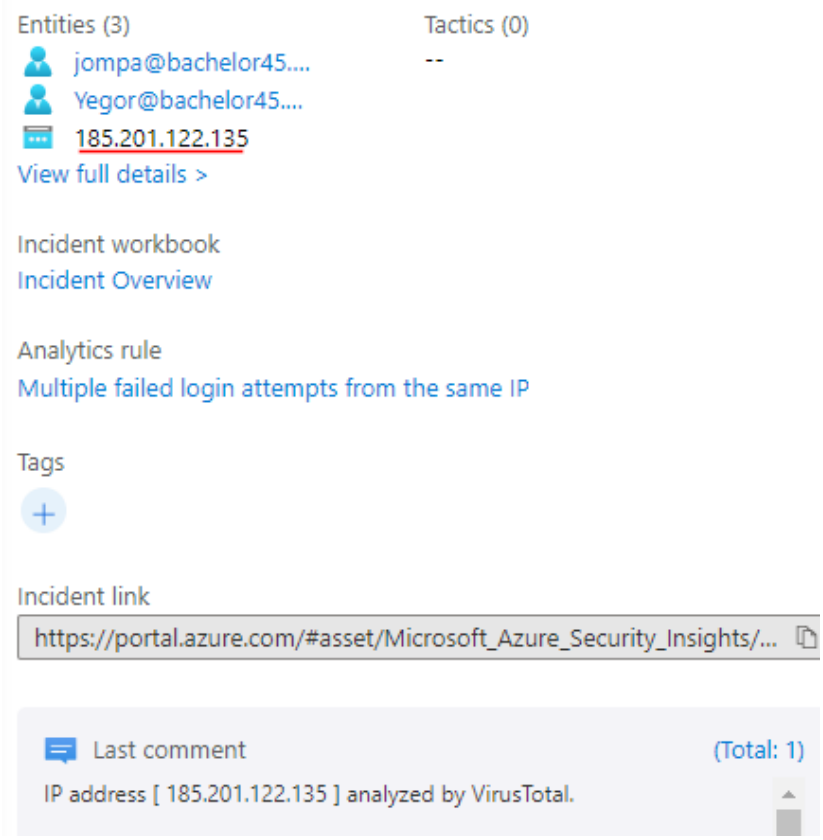
Figur 94 Oppdage pålogging fra IP i Watchlist - regel er lagt til

For å teste denne regelen, generer vi en ny hendelse ved hjelp av den andre Analytic Rulen: **Multiple failed login attempts from the same IP**. Kobler til et annet nettverk for å bruke en annen offentlig IP:



Figur 95 Oppdage pålogging fra IP i Watchlist - testing med følgende IP

Genererer så en hendelse i Sentinel:

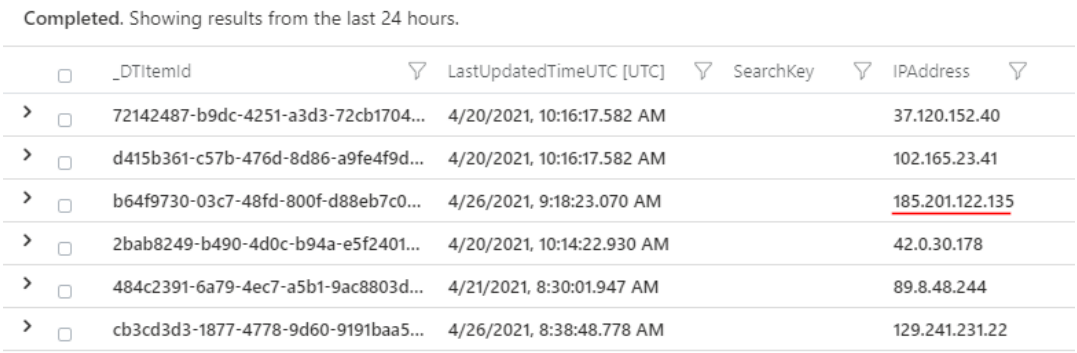


Figur 96 Oppdage pålogging fra IP i Watchlist - testing ser hendelsen med samme IP

Velger deretter å kjøre Playbooken: **Watchlist-Add-IPToWatchList**:

Figur 97 Oppdage pålogging fra IP i Watchlist - testing kjører så Playbook for å legge til i Watchlist

Vi kan også passe på at IP-adressen faktisk ligger i Watchlisten ved spørringen:

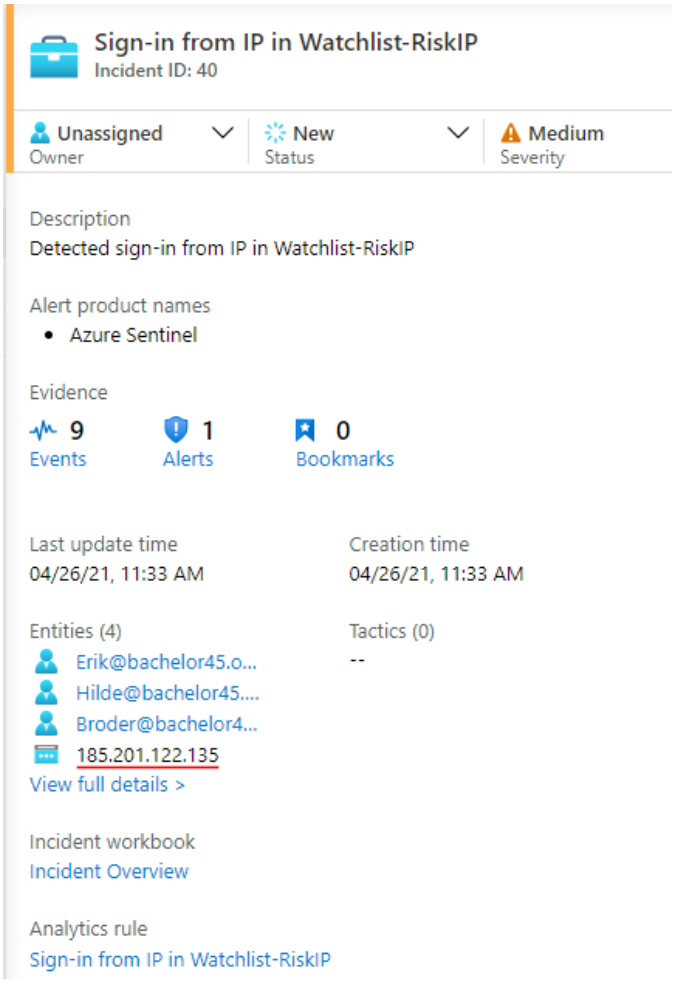


Completed. Showing results from the last 24 hours.

_DTItemId	LastUpdatedTimeUTC [UTC]	SearchKey	IPAddress
72142487-b9dc-4251-a3d3-72cb1704...	4/20/2021, 10:16:17.582 AM		37.120.152.40
d415b361-c57b-476d-8d86-a9fe4f9d...	4/20/2021, 10:16:17.582 AM		102.165.23.41
b64f9730-03c7-48fd-800f-d88eb7c0...	4/26/2021, 9:18:23.070 AM		185.201.122.135
2bab8249-b490-4d0c-b94a-e5f2401...	4/20/2021, 10:14:22.930 AM		42.0.30.178
484c2391-6a79-4ec7-a5b1-9ac8803d...	4/21/2021, 8:30:01.947 AM		89.8.48.244
cb3cd3d3-1877-4778-9d60-9191baa5...	4/26/2021, 8:38:48.778 AM		129.241.231.22

Figur 98 Oppdage pålogging fra IP i Watchlist - testing ser at IP ligger i Watchlist

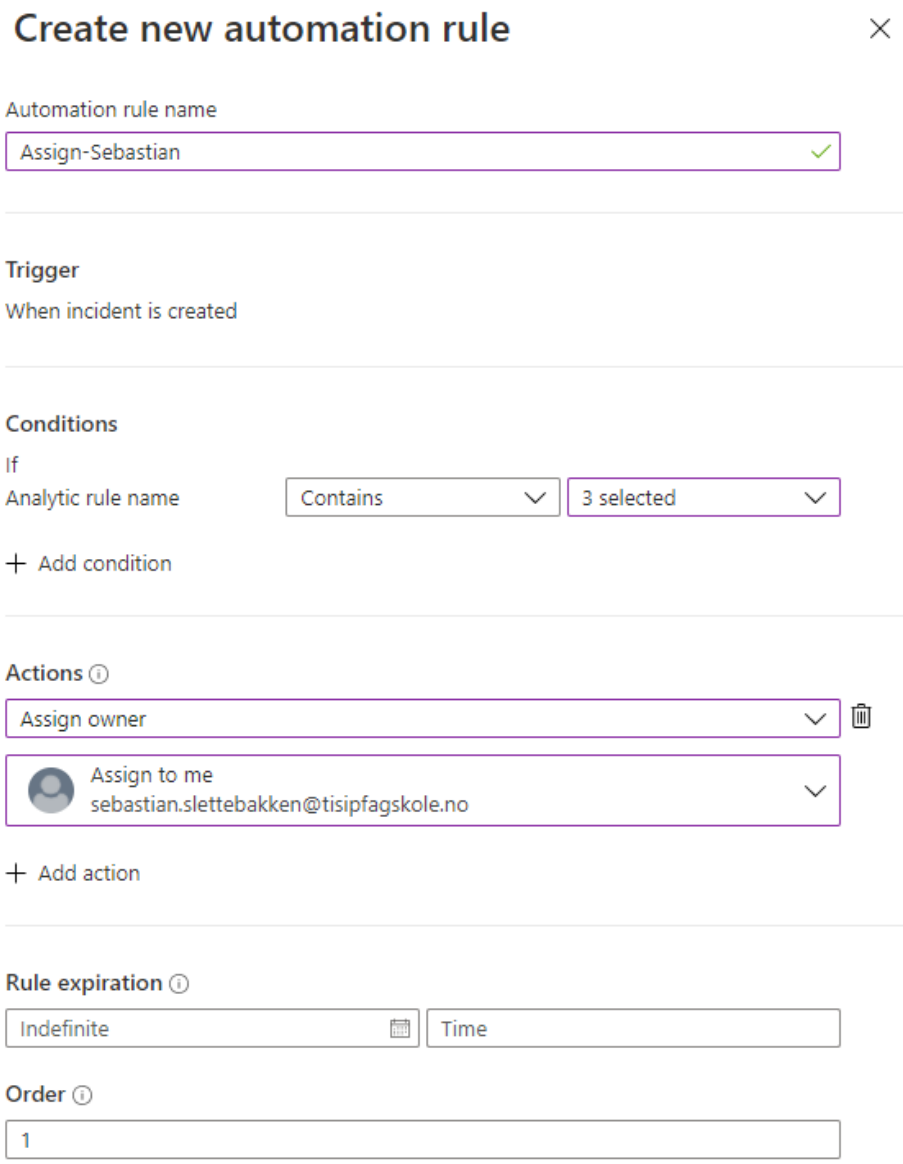
Ser at IP-adressen ligger der, som forventet. Vi kan nå prøve å teste regelen **Sign-in from IP in Watchlist-RiskIP** ved å prøve å logge inn igjen med samme IP-adresse. Ser at vi etter noen minutter får opp en hendelse:



Figur 99 Oppdage pålogging fra IP i Watchlist - eksempel på hendelse generert fra regelen

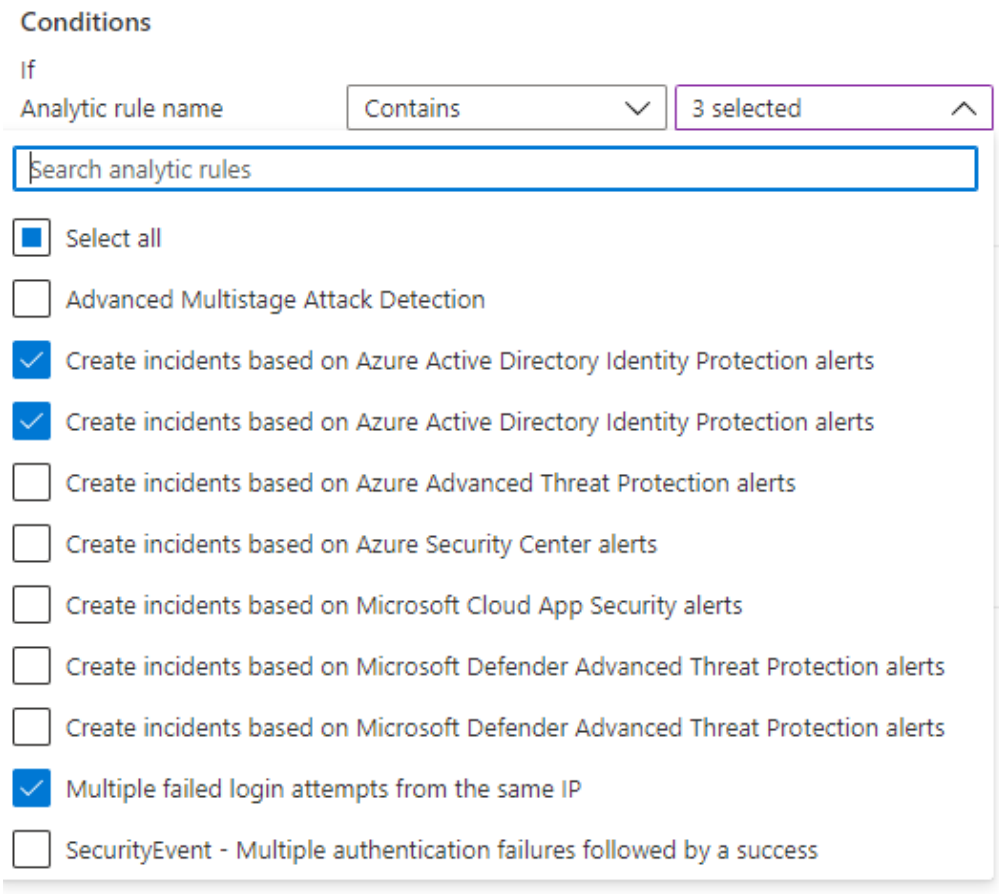
Orchestration

Vi har også mulighet for å delegere hendelser til spesifikke brukere automatisk. Dette kan gjøres med **Automation rules**. Vi kan opprette en regel som skal delegere hendelser relatert til spesifikke hendelser som vist under:



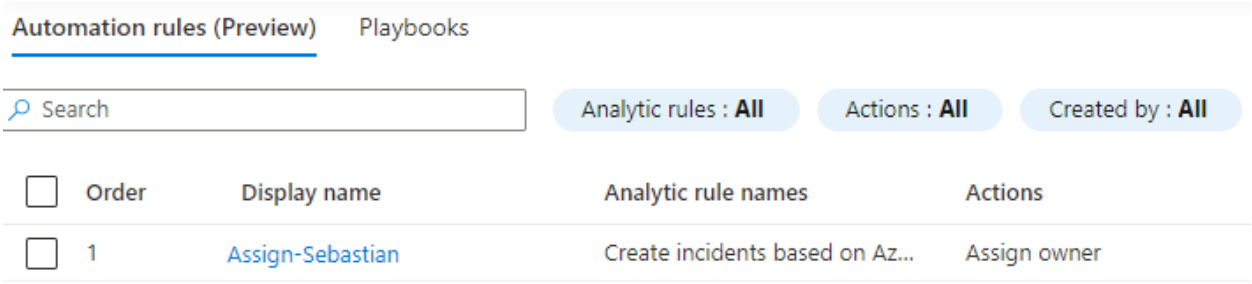
Figur 100 Orkestrering - Automation rule

Setter følgende betingelse:



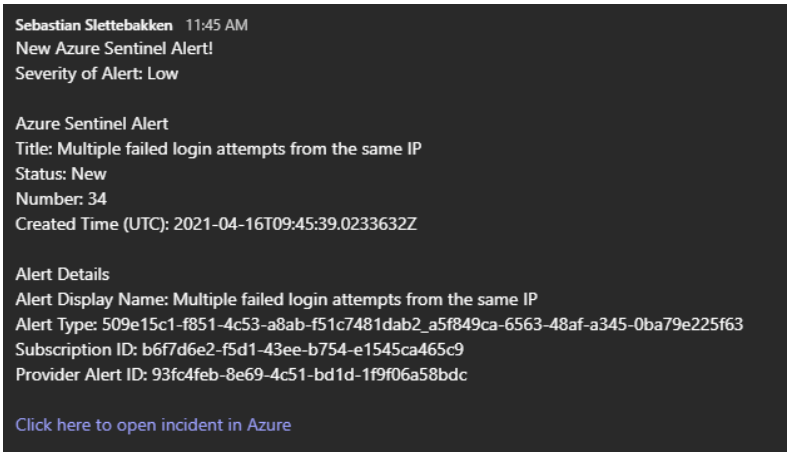
Figur 101 Orkestrering - Automation rule conditions

Ser at regelen ble lagt til og kan nå prøve å teste den ved å trigge en **Multiple failed login attempt from the same IP** hendelse.



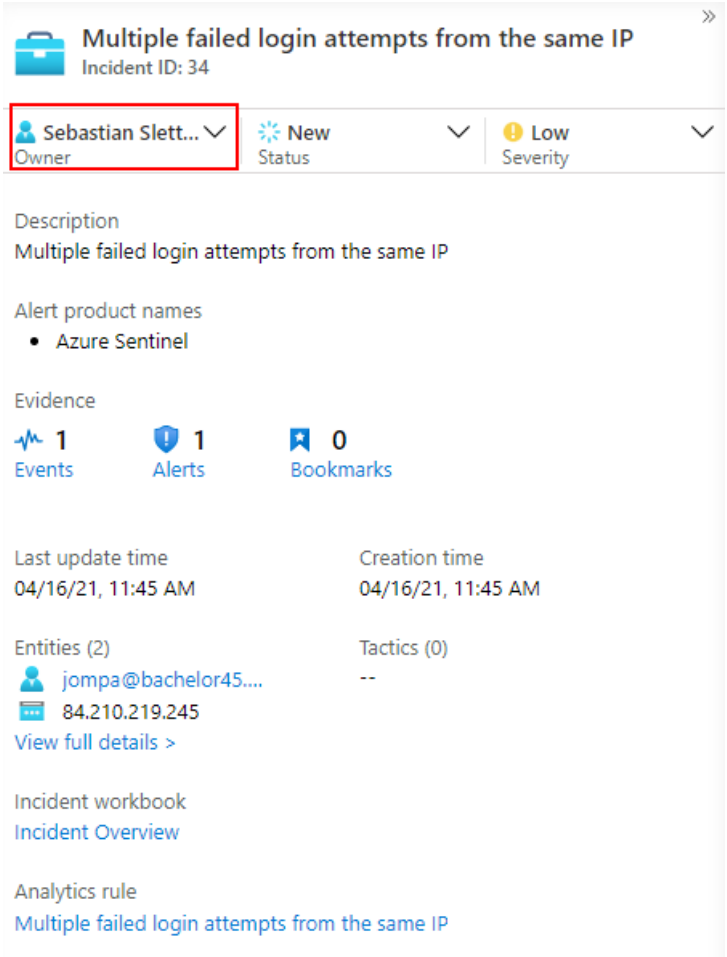
Figur 102 Orkestrering - Automation rule lagret

Ser at vi får varsel/melding i Teams-kanalen:



Figur 103 Orkestrering - Teams-varsel

Og ser at hendelsen automatisk ble delegert til Sebastian:



Figur 104 Orkestrering - automatisk delegert hendelse

Testing og måling

Frem til nå har vi bare vist *hvordan* man automatiserer. I dette kapitelet skal vi vurdere om det har en hensikt, og hvordan målene ved automatisering oppnås. Målet med automatisering er å raskere håndtere sikkerhetshendelser og returnere en tjeneste til normal-tilstand. Som nevnt tidligere i rapporten, er det viktig at automatiseringen ikke skaper nye sikkerhetshull og at den reduserer risikoen for feilhåndtering. Alt som foregår automatisk, må også kunne spores tilbake og presisjonen må være høy for å unngå feil. Vi må dermed passe på, ved å teste, at vi ivaretar nivået av tilgjengelighet, konfidensialitet og integritet. Vi vil måle tid spart og antall klikk, men også se på sporbarhet og presisjon.

Generelt om Playbookene

Effekten av automatisering kan blant annet måles i tid spart, antall klikk spart, økt sporbarhet og presisjon. Alle Playbookene våre er med på å skape verdi som tid spart, økt sporbarhet i det analytikere gjør og økt presisjon i form av at Playbooken kjøres likt hver gang og vi vet hva som skjer. Nedenfor følger en kort beskrivelse av Playbookene, eventuelle utfordringer med dem og hva slags verdi de skaper.

VirusTotal skanning av IP-adresse

IP-adresser som brukes i kommunikasjon er en viktig kilde til informasjon for å korrelere kommunikasjonen med andre aktiviteter som er observert på internett. Vi kan for eksempel skanne en IP-adresse på VirusTotal sine sider for å finne ut om andre har merket denne IP-en som ondsinnet. VirusTotal tilbyr også et API som gjør at man slipper å gå inn på nettsiden deres for også søke etter IP-adresser. Vi kan videre automatisere denne prosessen ved å ta i bruk Logic App for VirusTotal og fylle inn API-nøkkelen vi har. Den automatiserte prosessen vil poste en kommentar på hendelsen. På denne måten kan vi automatisk berike hendelser med informasjon om IP-adresser knyttet til den spesifikke hendelsen. Denne Playbooken kjøres automatisk ved hjelp av egne Analytic Rules, men må ellers kjøres manuelt på en hendelse. Verdien er også presisjon, ved at man får beriket med den samme informasjonen hver gang.

Legge inn IP-adresser fra hendelse i Watchlist

Denne er vanskeligere å beregne. Per dags dato er det ingen enkel måte å legge inn IP-adresser i Watchlist. For hver gang man skal legge til noe, må man opprette en ny Watchlist for så å legge inn de nye IP-adressene i en CSV-fil. Om man har CSV-filen ferdig satt opp vil de likevel ta 4-5 minutter å få lagt inn i Watchlist. Om man skal bruke Watchlist i stor grad er det ikke noe tvil om at man må automatisere. Automatiseringen vil gi mulighet for å oppdatere eksisterende Watchlist. Da vil man kunne holde seg til 1-4 Watchlist og man kan bruke Watchlistene effektivt i spørringer og regler ved senere tidspunkt. Om man skal jobbe effektivt med denne, bør man opprette flere Paybooks rettet mot Watchlist. Denne Playbooken legger til IP-adresser i Watchlist. Det kan også være naturlig å opprette en Playbook som kan

brukes til å fjerne den siste IP-adressen som er lagt til, dersom man ved uhell legger til feil IP. Som sagt tidligere er det store fordeler av at automatisere disse stegene om man skal bruke Watchlist.

Isolere enhet i MDATP ved hjelp av Sentinel

Dersom en enhet blir kompromittert, vil det være viktig å reagere fort for å unngå at angripere kan infiltrere flere maskiner i nettverket. For å unngå dette, kan man isolere en enhet. Dette innebærer å koble maskinen fra nettverket (andre maskiner og infrastruktur), men beholder fortsatt en kobling til Defender for Endpoint. Dette gjør at angripere ikke kan hente ut info om andre maskiner på nettverket, samtidig som man kan overvåke enheten og arbeide med å «redde» den. Denne prosessen gjøres manuelt i Microsoft Defender Security Center, som betyr at vi må bruke tid på å gå inn på riktig side. Sentinel har riktignok en snarvei for å åpne hendelser knyttet til MDATP i Microsoft Defender Security Center. For oss innebærer dette at vi må logge ut av Tisip-brukeren vår og logge inn med admin@bachelor45.onmicrosoft.com brukeren som har riktig tilgang til siden. Deretter er det noen klikk for å finne frem til riktig enhet, velge å isolere den og skrive inn en kommentar til hvorfor man vil isolere. Alt dette kan skje automatisk ved å kjøre en Playbook som isolerer enheten knyttet til hendelsen, legger til isolerings-kommentar og varsler om at enheten blir isolert. På denne måten har man tettet sikkerhetshullet en kompromittert enhet utgjør, og minket sannsynligheten for at malware og andre trusler kan ha spredd seg på det interne nettverket.

Blokkere bruker/gi bruker tilgang

Om man finner ut at en bruker er kompromittert må man handle kjapt og hindre brukeren i å gjøre for mye skade. Playbooken setter brukeren som «Disabled». Man kan også gjøre dette ved å gå inn i Azure Active Directory og manuelt finne brukeren, for så å blokkere brukeren. På samme måte som man blokkerer brukere ved mistanke om at den er kompromittert, kan man også finne at en ikke-kompromittert bruker er blokkert. Da vil man raskt kunne gi tilgang til brukeren igjen. Utfordringene med å a i bruk denne Playbooken er at man kan blokkere flere brukere. Dersom det er flere brukere involvert i en hendelse så vil alle brukerne blokkeres, selv om analytikeren kun ønsker å blokkere én.

Tilbakestille passord til Azure AD bruker

Denne Playbooken tar i bruk Microsoft Graph API og sender en http-forespørsel for å tilbakestille passordet til en Azure AD bruker. Det blir opprettet et midlertidig passord som brukeren må bruke når man skal logge inn igjen neste gang, og blir da bedt om å opprette et nytt passord. Det er analytikeren som sitter med dette passordet (i Teams-kanalen) og må få overlevert det til brukeren på en sikker måte. Hvis man skal gjøre dette manuelt vil det ta litt lengre tid, ettersom man må inn i Active Directory. Det er også mulighet for å sette opp «Self service password reset», men da vi ikke har tilgang på Office 365 og epost til brukerne våre, valgte vi å håndtere tilbakestilling av passord på denne måten. Utfordringen med denne Playbooken er egentlig det samme som for å blokkere brukere. Man kan ikke velge å bare tilbakestillle passordet til én bruker dersom det er flere involvert i hendelsen.

Blokkere IP-adresser i MDATP

Legger til en IP-adresse som IP indikator i MDATP. Dette er nyttig å gjøre automatisk ved at man slipper å måtte kopiere en adresse fra en hendelse i Sentinel, gå inn på MDATP, velge innstillinger -> regler -> indikatorer og så manuelt legge inn adressen. Her blir det satt en utløpsdato, men denne kan endres i Playbooken, eller man kan forlenge den i MDATP. Vi sparer altså tid og antall klikk og øker presisjonen og sporbarhet.

Tid spart

Når man automatiserer, er hensikten å spare tid eller arbeidskraft. Jo mindre tid man bruker, desto mer tid får man til å gjøre andre ting, som for eksempel trusselsøk eller håndtere andre, mer krevende sikkerhetshendelser. Vi vil teste på følgende måte:

1. Teste Playbooken 5 ganger for å finne gjennomsnittstiden
2. Finne gjennomsnittstiden brukt uten Playbooken (manuelt) 5 ganger
3. Beregne i snitt hvor mye man kan spare ved denne automatiseringen.

Navn på Playbook	Tid med automatisering	Tid uten automatisering	Tid spart per hendelse
Varslinger i Teams	3 sek	3 min*	Ca. 3 min
VirusTotal skanning av IP-adresse	4 sek	45 sek	Ca. 40 sek
Legge inn IP-adresser fra hendelse i watchlist	3 sek	5 min*	Ca. 5 min
Isolere enhet i MDATP ved hjelp av Sentinel	3 sek	35 sek **	Ca. 30 sek
Blokkere bruker	2 sek	25 sek	Ca. 20 sek
Gi tilgang til bruker	2 sek	25 sek	Ca. 20 sek
Tilbakestille passord til Azure AD bruker	2 sek	25 sek	Ca. 20 sek
Blokkere IP i MDATP	5 sek	45 sek**	Ca. 40 sek

Tabell 8 Oversikt over tid spart

*** = vanskelig å beregne ettersom det vil i stor grad variere fra hendelse til hendelse**

**** = Kan ikke gjøres i Azure-portalen. Må åpne opp Microsoft Defender Security Center siden og gjøre det derfra. Sekunder regnes fra dashbordet på den siden, og ikke fra Azure/Sentinel**

Dersom en hendelse har flere entiteter knyttet til seg, enten enheter, brukere eller IP-adresser, vil disse Playbookene være enda mer effektive, da de kjører på alle entitetene knyttet til hendelsen. Det vil si at man kan spare enda mer tid ved at man for eksempel blokkerer tre brukere på 2-3 sekunder og ikke 80-90 sekunder, som det ville tatt dersom man skulle blokkert tre brukere manuelt. Dette gjelder også for antall klikk nevnt under.

Antall klikk

En annen ting vi kan se på er antall klikk en analytiker må gjøre per hendelse. Tid brukt på hver sikkerhetshendelse vil variere i noen grad, men antall klikk vil ikke variere like mye. Dette vil gi et bedre bilde på at man forenkler arbeidshverdagen til analytikere. Det vil også øke presisjon, ved at man unngår å klikke feil (desto flere klikk, desto større sjanse for at man kan klikke feil).

I denne testen antar vi at en analytikere har alle sidene som kreves for å løse hendelsen oppe før hendelsen er begynt.

Navn på Playbook	Antall klikk med automatisering	Minst mulig klikk manuelt	Antall klikk spart ved å automatisere
Varslinger i Teams	3 klikk	8 klikk + tekst	5 klikk + tekst
VirusTotal skanning av IP-adresse	4 klikk	6 klikk + en kopiering	2 klikk + en kopiering
Legge inn IP-adresser fra hendelse i watchlist	3 klikk	18 klikk*	15 klikk
Isolere enhet i MDATP ved hjelp av Sentinel	3 klikk	8 klikk + tekst	5 klikk + tekst
Blokkere bruker	3 klikk	6 klikk	3 klikk
Gi tilgang til bruker	3 klikk	6 klikk	3 klikk
Tilbakestille passord til Azure AD bruker	3 klikk	5 klikk	2 klikk
Blokkere IP i MDATP	3 klikk	10 klikk + fylle ut IP	7 klikk

Tabell 9 Oversikt over antall klikk

*** = opprettelse av CSV-fil er tatt med**

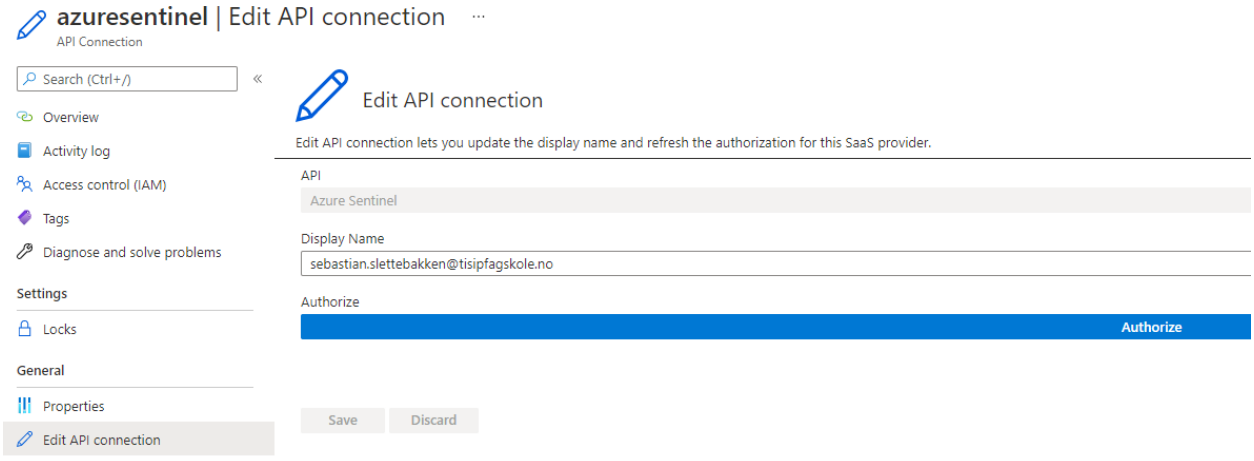
I tabellen over viser det at automatiseringen har sin virking både på tid og antall klikk. Automatiseringen sparer analytikere for en del jobb, og ved kjøring av Playbooks minker man sannsynligheten for feilklikk.

Presisjon

Ved å automatisere håndtering av hendelser, øker man presisjonen ved at ting blir gjort på samme måte hver gang. Så lenge man har testet automatiseringen og vet at det man automatiserer fungerer som det skal, vil man i tillegg til å spare tid, unngå at analytikere gjør feil i prosessen når de håndterer en hendelse. Sikkerhetsanalytikere kreves 24/7 for å håndtere hendelser, og de med nattevakt har kanskje større sjanse for å trykke feil eller glemme å huke av eller skrive en kommentar underveis i håndteringen. Ved at det er automatisert i form av en Playbook, trenger de kun å kjøre riktig Playbook til riktig type hendelse. Dette effektiviserer og forenkler arbeidet. Som nevnt er det viktig at automatiseringen er gjort riktig, da dersom Playbooken er upresis, vil man ikke få noe nytte ut av automatiseringen og kan derimot skape nye sikkerhetshull.

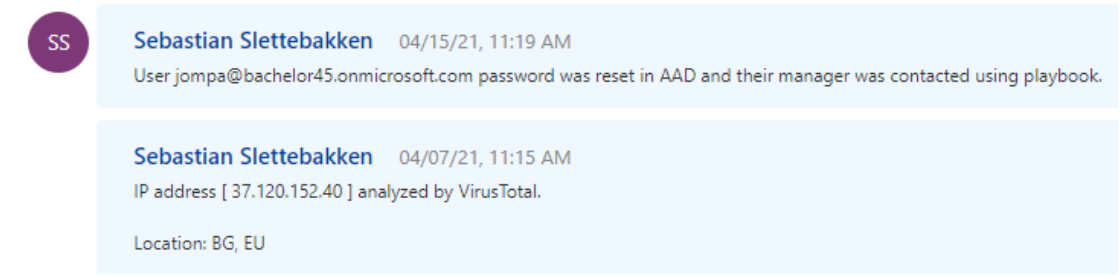
Sporbarhet

Sporbarhet er viktig i systemer generelt for å kunne sjekke hvem som har gjort hva og når. Dette blir enda viktigere når man skal automatisere prosesser. Dersom noe blir automatisert og gjort feil, vil man kunne gå tilbake og se hva som ble gjort, av hvem og til hvilken tid. Vi har valgt å løse sporbarhet ved å legge til kommentarer til hendelsene når vi kjører Playbooks. Noen av Playbookene sender også viktig informasjon i Teams-kanalen (som midlertidig passord for en bruker som får tilbakestilt passordet sitt). Azure Logic Apps tar i bruk API connections for utføre handlinger som å hente info om hendelser, legge til kommentarer til hendelser og poste melding i en Teams-kanal. API connection som er knyttet til Azure Sentinel i de fleste Playbookene ser slik ut:



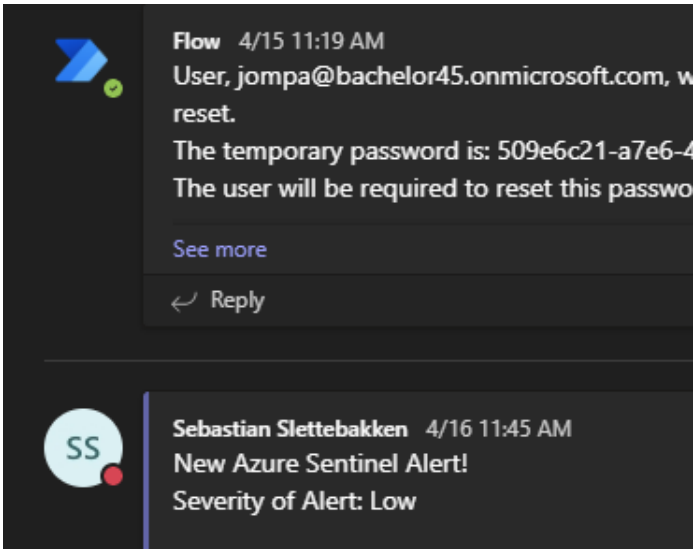
Figur 105 Sporbarhet - Azure Sentinel API connection

Den er knyttet til brukeren Sebastian Slettebakken. Det er den API connection som brukes for å legge til kommentarer til hendelser, så alle kommentarene vil bli publisert fra denne brukeren. Uavhengig av hvem som kjører Playbooken, vil det se slik ut:



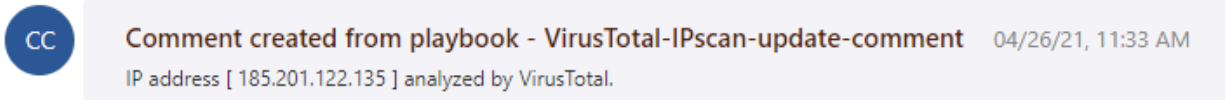
Figur 106 Sporbarhet - kommentarer til hendelser

Det er ingen måte å opprette en API connection som kan kommentere en hendelse ut ifra hvilken bruker som kjører Playbooken. Vi har dermed ikke klart å legge til sporbarhet for hvem som gjør hva (kjører Playbooks), men vi har altså muligheten for å se når det ble gjort og hva som ble gjort. Det samme gjelder for API connection for Teams. Den er knyttet til Sebastian sin NTNU-bruker for at vi skal få riktig tilgang til å poste meldinger i en egen kanal for Sentinel. Meldingene vil da bli sendt med Sebastian som avsender, og i noen tilfeller som en Flow-bot. Vi har heller ikke her mulighet til å se hvem som gjør hva, men vi får info om hva som blir gjort og når:



Figur 107 Sporbarhet - varslinger i Teams-kanal

Azure Sentinel blir stadig oppdatert med ny funksjonalitet, og vi ser etter hvert i prosjektets gang at nye kommentarer som blir lagt til hendelser fra Playbooks får en annen «avsender»:



Figur 108 Sporbarhet - oppdatering kommentar med annen "avsender"

Avslutning

I denne rapporten har vi gått gjennom hvordan vi har satt opp og testet ut løsningen vår i Azure Sentinel. Vi har vist praktisk oppsett med skjermutklipp av Logic Apps og Playbooks, i tillegg til teori rundt trusler og hvorfor vi ønsker å automatisere nettopp de funksjonene vi har automatisert. I forrige kapittel nevnte vi hva slags effekt vi har fått av å automatisere, og vi kan konkludere med at vi sparer analytikere for tid samtidig som vi har økt presisjon og sporbarhet på det som blir gjort i Sentinel. Vi har sett på ulike måter vi kan benytte Azure Sentinel til å øke treffsikkerheten og effektivisere arbeidet til analytikere gjennom automatisering av enkle prosedyrer, som var det oppgaven/problemstillingen vår gikk ut på. Automatiseringen har også åpnet noen nye dører, spesielt med tanke på Watchlist. Watchlist er tungvint å jobbe med om man ikke automatiserer. Når vi har automatisert har vi i stor grad fokusert på å lage snarveier for en analytiker. Analytikeren skal slippe å gå inn på tre forskjellige sider for å blokkere en bruker for eksempel. Med automatiseringen kan analytikerne blokkere alle brukere knyttet til en hendelse, med bare ett klikk. Vi ser i tabellen om antall klikk spart at en analytiker vil slippe mye klikking og skriving om man bruker Playbookene hyppig. Vi har også forsøkt med automatisering som skal bidra til Enrichment, og tok i bruk VirusTotal for å berike hendelser med informasjon om IP-adresser. Vi viser at det er relativt enkelt å sette opp, og om man går inn på Community oppbevaringsstedet (Azure, 2021), kan man finne flere ferdige maler for Playbooks som kan brukes for blant annet Enrichment. Noen av disse tar i bruk tredjepartsløsninger som tar betalt for tjenestene sine, for eksempel RiskIQ, HYAS og RecordedFuture. Vi har også sett på muligheten for å automatisere orkestrerings-prosessen ved å automatisk delegere sikkerhetshendelser til spesifikke personer. Gjennom testing har vi passet på å ivareta systemets integritet, konfidensialitet og tilgjengelighet.

Neste steg i oppgaven blir å skrive en sluttrapport. Den vil inneholde hvordan vi har opplevd prosjektets gang, hva vi har lært og utfordringer vi har møtt på underveis. Vi kommer også til å vurdere om vi har nådd målene i forhold til prosjektplanen. Et annet viktig moment i sluttrapporten går ut på videre arbeid med prosjektet og ting vi kunne gjort annerledes. Sluttrapporten vil være en naturlig avslutning, og skal gi en oppsummering og oversikt over prosjektet.

Referanser

Azure. (2021, April 28). *Azure-Sentinel Community Playbooks*. Retrieved from GitHub: https://github.com/Azure/Azure-Sentinel/tree/master/Playbooks

Cyclonis. (2021, Mars 16). *Brute-force*. Retrieved from Cyclonis: https://www.cyclonis.com/nb/hva-er-brute-force-angrep-hvordan-du-kan-forhindre-det/

Impreva. (2021, Mars 16). *Brute force attack*. Retrieved from Impreva: https://www.imperva.com/learn/application-security/brute-force-attack/

Maestral. (2021, Mars 16). *Introduction to Microsoft Azure Sentinel*. Retrieved from Maestral: https://maestralsolutions.com/introduction-to-microsoft-azure-sentinel/

Microsoft. (2021, April 15). *Connect data sources*. Retrieved from Microsoft docs: https://docs.microsoft.com/en-us/azure/sentinel/connect-data-sources

Microsoft. (2021, Mars 10). *Onboarding using Microsoft Endpoint Manager*. Retrieved from Microsoft Docs: https://docs.microsoft.com/en-us/windows/security/threat-protection/microsoft-defender-atp/onboarding-endpoint-manager

Microsoft. (2021, April 15). *Tutorial: Use playbooks with automation rules in Azure Sentinel*. Retrieved from Microsoft docs: https://docs.microsoft.com/en-us/azure/sentinel/tutorial-respond-threats-playbook

Microsoft. (2021, April 15). *What is Azure Logic Apps?* Retrieved from Microsoft docs: https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-overview

Shasha, Y. (2021, April 8). *Azure Sentinel-Playbooks*. Retrieved from Github: https://github.com/Azure/Azure-Sentinel/tree/master/Playbooks/Watchlist-Add-IPToWatchList

SNL. (2021, Mars 24). *Automatisering*. Retrieved from SNL: https://snl.no/automatisering

Gruppe 45

Bachelorprosjekt

SOAR i Azure Sentinel

Sluttrapport

Versjon 1.0

Forfattere: *Erlend Angell-Jacobsen, Sebastian Slettebakken*

Revisjonshistorie

Dato	Versjon	Beskrivelse	Forfatter
03/05/2021	1.0	Første utkast	Erlend Angell-Jacobsen, Sebastian Slettebakken

Innholdsfortegnelse

Tabell- og figurliste

Oppgavebeskrivelse

Oppgaven vi har tatt for oss er tidligere beskrevet og avgrenset i Forstudierapporten. Der nevnte vi at bakgrunnen for prosjektet er at vi skal skrive en bacheloroppgave for Sopra Steria. De ønsker at vi skal finne svar på «Hvordan gjøre Security Orchestration Automation and Response (SOAR) på tvers av enterprise-miljøer». Vi valgte å formulere problemstillingen:

Hvordan kan Azure Sentinel brukes til å øke treffsikkerheten og effektivisere arbeidet til en anlaytiker gjennom automatisering av enkle prosedyrer?

Vi valgte deretter å redusere omfanget av oppgaven til å fokusere på Azure Sentinel og muligheten for å automatisere der.

Arbeidets art

Arbeidets art er i hovedsak en driftsoppgave med fokus på å sette opp og konfigurere systemet (Azure Sentinel). Det er ingen utvikling av et nytt programsystem, da vi har tatt i bruk systemene Microsoft allerede har laget. Løsningen og resultatet vi kommer frem til, vil være tilgjengelig for Sopra Steria å ta i bruk dersom de ønsker det.

Oppdragsgivere og kontaktpersoner

Oppdragsgiver for oppgaven er Sopra Steria ved Pål Mathisen. Han er leder av SOC-en til Sopra Steria i Norge og vil bistå med assistanse og kunnskap om Azure Sentinel.

Prosjektet har også Stein Meisingseth fra NTNU som veileder.

Prosjektdeltakere er Erlend Angell-Jacobsen og Sebastian Slettebakken.

Kommunikasjon underveis har foregått ved hjelp av ukentlige møter på Teams og på epost. Det har vært tett oppfølging fra veiledere som har gjort at vi i prosjektgruppen har fått rask tilbakemelding på arbeidet vi har gjort, og svar på spørsmål vi har hatt.

Hvordan oppgaven ble løst

Denne oppgaven har en del dokumentasjon. Hvert dokument har sin hensikt. Vi har ikke kun skrevet dokumentasjon, men vi har også jobbet en del med skytjenester og annen programvare. Videre i dette kapittelet skal vi også se nærmere på arbeidsfordelingen i arbeidsgruppen.

Dokumentasjon

Nå som oppgaven er nære ferdigstilling kan vi se på hvordan arbeidet ble utført. I forstudierapporten satte vi noen standarder og metoder for gjennomføringen av prosjektet. Disse ble fulgt gjennom hele prosjektet. Passord testbrukerne ble holdt standardisert og prosjekt dokumentasjonen ble skrevet etter NTNUs maler. I dette prosjektet har vi produsert en del dokumentasjon. Under kommer det en tabell over dokumentasjonen og en beskrivelse til hver.

Dokumentasjon	Beskrivelse
Forstudierapport	Forstudier har som formål å avdekke styrker og svakheter, muligheter og risikoelementer. Den skal også kartlegge hvilke ressurser som trengs.
Designrapport	Hensikten med dette dokumentet er å beskrive designet til systemet/installasjonen som skal utvikles. Den er kun basert på beregnet på kunden/oppdragsgiver og er en konseptuell og tekstlig beskrivelse av systemet.
Driftsrapport	Hensikten med driftsrapporten er at den skal så godt som mulig beskriver hvordan selve prosjektet er gjort. Systemet skal testes og vurderes.
Sluttrapport (denne)	Prosjektet skal oppsummeres. Skal se på om mål har blitt nådd. Hva gikk bra og hva gikk dårlig? Videre arbeid osv.

Tabell 10 Oversikt over dokumentasjon

Når vi har produsert disse dokumentene bruker vi kilder. Vi brukte som regel Microsoft Docs, men vi har også brukt noen andre sider. Vi har vært forsiktig med hvilke sider vi brukte og passet på at de er troverdige.

Maskinvare og programvare

Maskinvare var ikke et fokusområde i prosjektet, men vi endte opp med å bruke noen maskiner uansett. Disse VM-ene ble i hovedsak brukt til å fremprovosere sikkerhetshendelser. Men vi har jobbet en del med tjenester i Azure. I Azure har vi brukt følgende systemer:

- Azure Sentinel
- Azure Active Directory
- Intune/Microsoft Endpoint Manager
- Microsoft Defender for Endpoint (MDATP)
- Logic Apps

Vi har også brukt Word til å skrive sammen i dokumentasjonen. Så har vi brukt Teams som en plass for lagring og kommunikasjonsplattform. Vi har brukt MS Project til å planlegge frem i tid.

Arbeidsfordeling

Prosjektet har en arbeidsgruppe bestående av to personer. Vi har hatt møter nesten hver dag etter prosjektet begynte. Vi møtes og diskuterer som om vi skulle møttes fysisk, men møtene har vært over Teams. Jobbet mye sammen, og diskutert frem løsninger. Arbeidet har vært tilnærmet 50/50. Vi har forsøkt å holde arbeidet likt fordelt. Begge i arbeidsgruppen møter opp til avtalt tid. Arbeidsgruppen prøver etter beste evne å fullføre dokumentasjon til tidsfrister. Bortsett fra en periode hvor vi ikke hadde tilgang på Azure Sentinel har vi fulgt planen i Project-filen. Vi har kommet i gang med sluttrapporten på riktig dato og sagt oss ferdige med driftsrapporten.

Gjennomføring av prosjektet

I dette kapittelet kommer vi til å vurdere hvordan gjennomføring av prosjektet gikk. Vi kommer til å se på momenter vi syns gikk bra, momenter vi ikke syns gikk så bra og hva som kunne blitt gjort annerledes. Vi kommer også til å vurdere måloppfyllelsen i forhold til prosjektplanen.

Hva som gikk bra

Gjennomføringen av prosjektet har alt i alt gått veldig bra. Vi mener gruppesamarbeidet fungerte bra, til tross for at vi ikke har samarbeidet noe tidligere i studiet. Vi har som nevnt over møttes så å si hver dag på Teams for å jobbe sammen. Hjemmekontor har fungert bra, og det har vært viktig for oss å møtes hver dag for å holde hverandre oppdatert på hva vi gjør og diskutere underveis.

Vi er også fornøyde med at tidsfrister ble overholdt, selv om vi kom i gang med driftsrapporten litt senere enn planlagt. Mer om dette i punktene under. Til slutt er vi også fornøyde med det vi har fått til å automatisere, altså resultatet. Selve oppsettet i Azure Sentinel gikk bedre enn forventet når vi først fikk tilgang, og vi er fornøyde med Playbookene vi har opprettet. Selv om de ikke er veldig avanserte, har vi fått til å opprette Playbooks vi mener er gode, presise og vil spare sikkerhetsanalytikere for tid og manuelt dagligdags arbeid.

Utfordringer

Allerede i uke 6 begynte vi å se på hvordan vi skulle få tilgang til Azure Sentinel og demo-miljøet. Vi brukte mye tid på å prøve å ordne Microsoft E5 lisens gjennom NTNU og etter hvert Crayon. Vi hadde møter med den andre bachelorgruppen, ved Adrian Gilberg og Erlend Saugstad, og deres veileder, Vegar Åsmul, for å prøve å få ordnet tilgang til et demo-miljø. Dette var ikke så lett, og når den andre gruppen bestemte seg for å gå vekk fra demo-miljøet fulgte vi etter hvert i samme retning. Det var først i uke 10 at vi fikk tilgang til Azure Sentinel ved hjelp at Jostein Lund opprettet brukere for oss i tisip-domenet. Fikk til å bruke student-subscription slik at vi fikk $100 å bruke i Azure. Dette var avgjørende, da prosjektdeltakerne ikke skulle legge inn egen kortinfo og betale for å gjennomføre oppgaven.

Vi hadde i utgangspunktet lyst til å bruke demo-miljøet, slik at vi lettere kunne simulere hendelser og aktivitet i domenet vårt. Da dette krevde at vi la inn kortinfo for å få Azure kreditt til å opprette en Sentinel workspace, valgte vi å gå bort ifra dette miljøet og heller måtte trigge hendelser manuelt og teste på en annen måte enn det vi først hadde regnet med at vi kom til å gjøre. Det å ordne tilgang og finne ut hva vi trengte av lisenser var en mer tungvint prosess i forhold til hva vi hadde sett for oss, og vi brukte mye tid på å få tilgang.

Hva kunne vært gjort annerledes

Vi kunne prøvd å få ordnet Office 365 også, for å kunne ta i bruk dette når vi automatiserte. Det er ikke lett å sette seg inn i Microsoft sine lisenser og hvordan vi skulle fått en trial-periode, og vi brukte allerede såpass mye tid på å ordne tilgang til Azure Sentinel at vi så det ikke som hensiktsmessig å bruke mer tid på prøve å ordne lisenser.

Tekniske problemer

Som nevnt over i punktet om hva som gikk dårlig, slet vi med å få ordnet tilgang til Azure Sentinel i noen uker. Dette førte til at vi kom senere i gang med driftsrapporten enn antatt. Vi opplevde også noen tekniske problemer i Azure Sentinel, med type tilgang som krevdes og Playbooks som ikke ville kjøre slik vi ønsket. Som regel var det bare å google for å finne ut hvordan man kunne løse feilmeldingen, eller så fant vi andre måter å løse det på.

Begrensninger

Den største begrensningen i prosjektet er, som nevnt tidligere, at vi ikke har tilgang på Office 365 i domenet vårt. Vi fikk dermed ikke tatt i bruk for eksempel epost i Playbookene våre. Vi ble dermed nødt til å løse tilbakestilling av passord på en annen måte, da funksjonalitet som «self service password reset» krever å sende epost til brukerne. Vi har også vært begrenset av økonomi ved at vi ikke har tatt i bruk betalte tredjepartsløsninger som RiskIQ når det gjelder å berike hendelsene våre ved hjelp av Playbooks. En annen begrensning vi kan ta i betraktning er muligheten for å møtes fysisk for møter innad i prosjektgruppen og med veileder fra NTNU. Vi skulle gjerne også møtt oppdragsgiveren vår fysisk, men grunnet pandemien har vi begrenset oss til å ha hjemmekontor og digitale møter.

Vurdere måloppfyllelse til prosjektplanen

Under har vi et utklipp av prosjektplanen (fra 05.05.2021). Dokumentet ligger også i Teams-kanalen til gruppa og vil bli lagt som vedlegg ved endelig innlevering av oppgaven.

GANTT CHART

	Task Mode	Task Name	Duration	Start	Finish	Prede	Resource Names
1		Planlegging	5 days	Wed 06.01.21	Tue 12.01.21		Erlend;Sebastian
2		**Forstudierapport**	**18 days?**	**Wed 13.01.21**	**Fri 05.02.21**	**1**	**Erlend;Sebastian**
3		Kap. 1 - Introduksjon	1 day	Tue 26.01.21	Tue 26.01.21		Erlend;Sebastian
4		Kap. 2 - Bakgrunn	4 days	Wed 13.01.21	Mon 18.01.21		Sebastian
5		Kap. 3 - Prosjektmål	4 days	Wed 13.01.21	Mon 18.01.21		Erlend
6		Kap. 4 - Interessenter	2 days	Thu 14.01.21	Fri 15.01.21		Erlend
7		Kap. 5 - Suksessfaktorer	2 days	Thu 14.01.21	Fri 15.01.21		Sebastian
8		Kap. 6 - Risikoanalyse	3 days	Tue 19.01.21	Thu 21.01.21		Erlend;Sebastian
9		Kap. 7 - Kost/nytte	5 days	Tue 19.01.21	Mon 25.01.21		Sebastian;Erlend
10		Kap. 8 - Retningslinjer	1 day	Mon 25.01.21	Mon 25.01.21		Erlend
11		Kap. 9 - Organisering	1 day	Mon 25.01.21	Mon 25.01.21		Erlend;Sebastian
12		Kap. 10 - Anbefaling	1 day	Tue 26.01.21	Tue 26.01.21		Erlend;Sebastian
13		Gjennomgang av dokumentet	5 days	Mon 01.02.21	Fri 05.02.21		Erlend;Sebastian
14		**Designrapport**	**10 days?**	**Mon 01.02.21**	**Fri 12.02.21**	**2**	**Erlend;Sebastian**
15		Innledning	1 day	Thu 11.02.21	Thu 11.02.21		Erlend;Sebastian
16		Dokumentets hensikt	2 days	Mon 01.02.21	Tue 02.02.21		Erlend;Sebastian
17		Avgrensning	1 day	Tue 02.02.21	Tue 02.02.21		Erlend;Sebastian
18		Referanser	1 day	Thu 11.02.21	Thu 11.02.21		Erlend;Sebastian
19		Oversikt over dokumente	1 day	Thu 11.02.21	Thu 11.02.21		Erlend;Sebastian
20		Kunden og behov	1 day	Wed 03.02.21	Wed 03.02.21		Erlend;Sebastian
21		Beskrivelse av teknisk løsı	2 days	Wed 03.02.21	Thu 04.02.21		Erlend;Sebastian
22		Detaljert løsningsbeskrive	5 days	Fri 05.02.21	Thu 11.02.21		Erlend;Sebastian
23		**Driftsrapport**	**45 days?**	**Mon 01.03.21**	**Fri 30.04.21**	**14**	**Erlend;Sebastian**
24		Praktisk oppsett	40 days	Mon 01.03.21	Fri 23.04.21		Erlend;Sebastian
25		Testing og måling	13 days	Mon 05.04.21	Wed 21.04.21		Erlend;Sebastian
26		Ferdigstille rapport	10 days	Mon 19.04.21	Fri 30.04.21		Erlend;Sebastian
27		Sluttrapport	4 days	Mon 03.05.21	Thu 06.05.21	23	Erlend;Sebastian
28		Egenvurdering	0 days	Fri 07.05.21	Fri 07.05.21	27	Erlend;Sebastian
29		Presentasjon	5 days	Mon 10.05.21	Fri 14.05.21	28	Erlend;Sebastian
30		Endelig frist	0 days	Thu 20.05.21	Thu 20.05.21		

Figur 109 Prosjektplan

Arbeidet med forstudierapporten gikk greit med tanke på tidsforbruk og frister. Vi satte frist for den til å være ferdig den 5. februar og rettet opp noen feil og sa oss ferdige med rapporten mandag den 1. februar. Vi lå da noen dager foran planen, noe vi var fornøyde med.

Vi begynte på designrapporten samme dag (1. februar) og utnyttet at vi lå foran planen. Arbeidet med rapporten gikk bra og vi fikk gode tilbakemeldinger underveis. Fristen vi satte for å være ferdig var 12.

februar, men vi sa oss ikke ferdige med rapporten før tirsdag 16. februar. Vi brukte altså litt lenger tid enn planlagt, selv med forspranget vi hadde.

Etter designrapporten var ferdig, var det på tide å se på driftsrapporten og faktisk sette opp det praktiske. For å kunne gjøre dette, trengte vi tilgang til Azure og Azure Sentinel. Det var først i uke 10 at vi fikk tilgang på det vi trengte, og all tiden etter designrapporten frem til uke 10 gikk til å prøve å få satt opp et miljø i Azure og finne informasjon knyttet til Azure Sentinel. Vi hadde regnet med å bruke litt tid på å få tilgang og til å sette opp miljøet, men dette tok lengre tid enn forventet. Vi klarte allikevel å holde oss innen fristen vi satte oss selv, og sa oss ferdige med driftsrapporten 30. april.

Sluttrapporten begynte vi på 3. mai som stemmer overens med planen vår. Vi satser på å bli ferdige med denne rapporten i løpet av uke 18. Vi har også planer om å skrive egenvurdering i løpet av uke 18, men mulig det blir forskjøvet til uke 19.

Uke 19 vil gå til å forberede presentasjon og eventuelt gjøre ferdig egenvurdering. I tillegg kommer vi til å måtte ferdigstille prosjekthåndbok med møtereferat, prosjektplan, timeliste og til slutt ferdigstille rapportene.

Vi har også ført timelister for prosjektet. De vil bli lagt ved i prosjekthåndboken, men utklipp (fra 05.05.2021) blir vist under.

Timeliste for bacheloroppgave - Erlend Angell-Jacobsen								
Ukenr.	mandag	tirsdag	onsdag	torsdag	fredag	lørdag	søndag	sum
1			1,0		1,5	1,0		3,50
2	2,0	1,0	3,0	2,0	2,5			10,50
3	2,0	4,0	3,5	2,5	3,0			15,00
4	3,0	4,0	4,0	3,0	3,0			17,00
5	2,0	3,0	3,0	3,0	2,5			13,50
6	2,0	2,0	2,0	2,0	3,0			11,00
7	1,5	1,0	3,0	2,0	3,0			10,50
8		3,0	2,0	1,0	1,0			7,00
9	1,0	1,0	1,0	1,0	1,0			5,00
10	2,0	2,0	4,0	4,0	4,0			16,00
11	4,0	2,5	3,5	2,5	3,5			16,00
12	3,0	4,0	3,0	4,0	5,0			19,00
13	6,0	5,0						11,00
14		5,0	5,0	2,0	4,5			16,50
15	4,0	0,5	2,5	6,0	4,0			17,00
16	5,0	4,0	5,0	2,0	3,5			19,50
17	3,0	3,0	6,0	5,0	3,0			20,00
18	3,0	5,0	6,0					14,00
19								0,00
						Total sum		242,00

Figur 110 Timeliste - Erlend

Timeliste for bacheloroppgave - Sebastian Slettebakken								
Ukenr.	mandag	tirsdag	onsdag	torsdag	fredag	lørdag	søndag	sum
1			1,0		1,5		1,0	3,50
2	2,0	1,0	3,5	2,0	2,5			11,00
3	2,0	2,0	6,0	2,5	3,0	2,0	1,0	18,50
4	1,0	5,0	5,0	3,0	3,0			17,00
5	2,0	2,0	3,0	3,0	2,5			12,50
6	2,0	2,0	2,0	2,0	2,0			10,00
7	1,5	1,0	3,0	2,0	3,0			10,50
8		3,0	2,0	1,0	1,0			7,00
9	0,5	1,0	1,0	1,0	1,0			4,50
10	2,0	2,0	5,0	5,0	4,0			18,00
11	4,0	2,5	3,5	2,5	3,5			16,00
12	3,0	4,0	3,0	4,0	5,0			19,00
13	7,0	6,0						13,00
14		6,5	5,0	2,0	4,5			18,00
15	4,0	0,5	2,5	6,0	4,0			17,00
16	5,0	4,0	6,0	0,0	3,5			18,50
17	3,0	3,0	7,0	6,0	3,0			22,00
18	3,0	5,0	6,0					14,00
19								0,00
						Total sum		250,00

Figur 111 Timeliste - Sebastian

Vi ser av timelistene at vi har brukt mindre tid en antatt på å løse oppgaven. Det har derimot ikke gått på bekostning av kvalitet, da vi allikevel har løst den slik at vi er fornøyde med sluttresultatet. Underveis har vi ført opp ukerapporter for arbeidet vi gjør. Disse vil også følge med i prosjekthåndboken, og et utklipp fra en slik ukerapport er vist her:

Uke 17 (26. april – 2. mai)

Mandag: Ferdig testing/dokumentasjon av analytic rule for å oppdage pålogging fra IP i watchlist.
Begynne på innledning og avslutning.

Tirsdag: Så på tilbakemelding fra Pål angående kapittelet Testing. Jobbet videre med innledning

Onsdag: Jobbet videre med avslutning, skrev sammendrag og abstract. La inn bildetekst til alle bilder og begynte å formatere dokumentet litt. Endret også på Testing ved å legge inn punk om generelt om playbook-ene.

Torsdag: Lese gjennom rapporten. Kommentere og redigere.

Fredag: Gå gjennom rapporten og rette opp. Fikse litt på struktur under Testing og måling. Ferdigstille avslutning.

Lørdag:

Søndag:

Figur 112 Ukerapport - eksempel

Vurdere måloppfyllelse til resultater

I dette prosjektet satte vi en del mål for gjennomføringen av prosjektet. Vi delte de opp i tre kategorier, resultatmål, effektmål og prosessmål. Vi har klart å oppfylle alle målene i noen grad. Under ser du en liste over resultat målene vi satte:

- *Azure Sentinel system med noen grad av automatisering*
- *Statistikk på hvilken grad et Azure Sentinel system vil bidra for analytikere*
- *Prosjektet skal være ferdig før 20.05.2021*

Vi har laget et Azure Sentinel system med noen grad av automatisering. Vi ente opp med å automatisere og effektivisere noe enklere hendelser en tenkt, men fremdeles automatisert i noen grad. I driftsrapporten presenterer vi statistikk på hvor my arbeidet til en analytiker effektiviseres. Dette viser vi med antall klikk og tid spart hver gang man benytter en Playbook. Vi er på god vei til å bli ferdig før 20.05.2021.

Effektmålene og prosessmålene er ofte ikke like konkrete som resultatmålene, og kan dermed være litt vanskeligere å måle. Effektmålene kommer i en liste under:

- *Redusere tiden en analytiker bruker på en sikkerhetshendelse*. Gjennom bruk av automatisering og fremstilling av informasjon
- Åpne muligheten for automatisering i større grad med bruk av Azure Sentinel
- *Redusere IT-kostnader*, ved at systemet skal i større grad baseres på automatisering
- *Økt IT-kompetanse*. Analytikere vil bli i større grad satt til viktigere hendelser og ikke gjentakende små sikkerhetshendelser

Som vist i driftsrapporten vil vi kutte tiden en analytiker bruker på enkelte hendelser. Vi åpner flere muligheter for automatisering ved å blant annet ta i bruk Watchlist. Det er vanskelig å si hvor mye man kutter kostnader, men en analytiker vil bruke litt mindre tid ved enkle hendelser.

Prosessmålene kommer i en liste under:

- Kompetansebygging på Azure Sentinel.
- Kompetansebygging på teamarbeid.
- Teammedlemmer blir bedre kjent.
- Erfaringer skapes gjennom arbeid med bachelor.

Disse målene synes arbeidsgruppen at har blitt oppfylt i stor grad. Begge har blitt kjent med Azure Sentinel, og med hverandre. Begge har fått nye erfaringer som de kommer til å ta med seg videre i livet.

Videre arbeid

I dette kapitlet skal vi se nærmere på hva som vil være en naturlig vei videre. Datasikkerhet er vanskelig å få helt sikkert. Man må utvikle systemer og metoder for å være mest berett for trusler som befinner seg på nett.

Får vå del ville det vært naturlig å integrere Office 365 inn i diverse automatiseringer. Da vil man få mulighet til å sende e-post både til brukere og admin. Brukere vil få muligheten til å benytte «self service password reset».

Enrichment er også en side av automatiseringen vi ikke fikk sett så mye på. De fleste tredjepartene med mulighet for berikelse av sikkerhetshendelser, koster penger. For eksempel RiskIQ og GreyNoise. Disse vil være enkle å implementere tvert man er villig til å betale for tjenesten. Dette vil være naturlig for å sikre kvalitet dataen som blir vist. Om man benytter tre forskjellige tjenester kan man kan man være mer sikker på at dataen stemmer. La oss si at VirusTotal fant ikke denne IP-adressen som en mulig trusel. Derimot både RiskIQ og GreyNoise finner at den har tidligere vært involvert i angrep. Om man kunne hadde tatt i bruk VirusTotal kunne man risikere å i verstefall slippe gjennom en hacker.

Senere kan man se på mer avanserte automatiseringer som vil gjøre større inngrep. For eksempel automatisk blokkere brukere om det er mange ting som tyder på at brukeren er kompromittert. For at en automatisert regel skal kunne stenge en bruker ute er bør det være flere indikatorer som tyder på at brukeren kompromittert. En slik indikator kan for eksempel være impossible travel. Deretter kan det også være lurt å se om brukeren har VPN. Når man føler seg trygg på at det skal mye til før automatiseringen slår ut på en falsk-positiv, kan man sette inn harde tiltak som å blokkere brukeren.

I tillegg til å fortsette utviklingen med mer avanserte automatiseringer, bør man også fortsette med de enkle snarveiene. Det kan godt være at det er automatiseringer som er enkle og fortsatt bidra en del. Disse er som regel enkle å implementere. En mulig løsning er å automatisere nesten alle tiltak slik at en analytiker vet at tiltakene ligger ved Playbooks.

SOAR i Azure Sentinel

NTNU

Kunnskap for en bedre verden