NTNU
Det skapende universitet

Struktur og sikkerhet av nettverk ved integrerte operasjoner

Henrik Børstad Eriksen

Master i teknisk kybernetikk
Innlevert: juni 2013
Hovedveileder: Tor Engebret Onshus, ITK
Medveileder: Stig Ole Johnsen, SINTEF

Norges teknisk-naturvitenskapelige universitet
Institutt for teknisk kybernetikk

Masteroppgave, Teknisk Kybernetikk

Struktur og sikkerhet av nettverk ved integrerte operasjoner

Henrik Børstad Eriksen
henrik.eriksen@gmail.com

Veileder:
Professor Tor E. Onshus, NTNU

Ekstern veileder:
Seniorforsker Stig Ole Johnsen, SINTEF

Trondheim, Juni 2013

NTNU
Det skapende universitet

Institutt for Teknisk Kybernetikk
Fakultet for Informasjonsteknologi, Matematikk og Elektroteknikk

Oppgavetekst

Struktur og sikkerhet av nettverk ved integrerte operasjoner

På en plattform i Nordsjøen er det mange datasystemer som henger sammen og som er kritiske for operasjonen.

I forbindelse med e-drift, integrerte operasjoner og fjernoperasjon eller overvåking, er en opptatt av at koblingen til land og til andre organisasjoner ikke skal gi problemer verken når det gjelder sikkerhet (safety) eller på grunn av utilsiktet påvirkning eller spredning av informasjon til omgivelsene (security).

Oppgaven består i å kartlegge de nettstrukturer som finnes og de tiltak som er gjort på forskjellige installasjoner samt å sette seg inn i de prinsipper som kan anvendes for å sikre slike nettverk fra omgivelsene.

Oppgaven gitt: Januar, 2013
Hovedveileder: Tor Onshus, Institutt for Teknisk Kybernetikk, NTNU

Sammendrag og konklusjon

Et viktig steg videre for driften av oljeplattformer er integrerte operasjoner (IO) og IO vil kunne føre til kostnadsbesparelser på opp mot 300 mrd. for operatørene. Samtidig flyttes mennesker bort fra plattformene og redusere risikoen ved operasjonene. Gevinstene med IO kommer ikke gratis, og man står blant annet ovenfor store utfordringer i forbindelse med omstillingen til denne nye driftsfilosofien hvor langt flere av operasjonene sentraliseres i onshore operasjonssentraler.

For å kunne realisere IO må beslutningsdata som tidligere i stor grad har vært lokalisert i lukkede systemer offshore flyttes over på et nettverk delt med systemer på land slik at dataene også kan nåes fra land. På grunn av de høye kostnadene forbundet med å bytte ut eksisterende system har de nødvendige endringene for å realisere dette i stor grad blitt gjort iterativt, hvor ny funksjonalitet blir lagt til de eksisterende systemene. Denne måten å utvikle systemene på kan føre til at systemer som er designet for å operere i miljøer med naturlige barrierer eller proprietære kommunikasjonsprotokoller mot omverden mister disse egenskapene.

I kombinasjon med at flere systemer åpnes mer opp mot omverden, baserer en økende andel systemer seg på kommersielle operativsystemer som Microsoft Windows. Mengden sikkerhetshull i disse operativsystemene har ført til at flere aktører for angrep har fått øynene opp for disse systemene som potensielle angrepsmål. Den aktøren som utgjør kanskje største trusselen mot systemene er mye muligens flere land sine kraftige satsinger på cyberforsvar. Tyngden bak angrep fra denne type aktører ble vist med Stuxnet [9], som var det første kjente viruset rettet spesielt mot industrielle systemer. Gjennom bruke Windows har systemene også gjort seg sårbare mot vanlige datavirus [32].

I arbeidet med nye SAS (Sikkerhet- og automasjonssystemer) fokuseres det på å skape sikre systemer ved å designe robuste enheter, hvor enhetene på nettverket snakker samme språk ved bruk av sikre kommunikasjonsprotokoller. Da svært mange SAS-systemer består av gamle komponenter er det inntil videre nødvendig å bygge sikkerhetsmekanismene inn i nettverkene for å oppnå tilstrekkelig sikkerhet. Det finnes tradisjonelt mange måter å beskytte nettverk og omkringliggende systemer på fra de farer som oppstår når man åpner opp nettverkene sine. Det blir derfor i denne masteroppgaven presentert flere kjente standarder, driftsfilosofier og sikkerhetsmekanismer som kan benyttes for å øke informasjonssikkerheten i de industrielle systemene slik som beskyttelse-i-dybden, kryptering, VPN (virtuelle privatnettverk), brannmurer, DMZ-er (demilitarisere soner), IDS-er (intrusion detection system) og testmetoder.

Rapporter fra tilsyn og spørreundersøkelser benyttet som kilder i denne oppga-

ven viser at det langt fra alltid er tilfellet at disse mekanismene blir brukt på den måten eller i den grad de burde. Blant annet fjernet kun 19 % av de spurte ubenyttede funksjoner eller tjenester fra systemene (hardening) for å øke sikkerheten, kun 77 % benyttet tilstrekkelig sikre brannmurer mellom SCADA-systemet (supervisory control and data acquisition) og Internett, og bare 43 % svarte at de jevnlig utførte oppdateringer av systemene [3]. Undersøkelsene i [12] avdekket at svært få installasjoner hadde gjennomført en detaljert risikoanalyse av SAS, noe som kan føre til at uoppdagede og uønskede avhengigheter forblir ukjente, og som kan gi problemer dersom det oppstår feil i systemet. Alle disse funnene bør kunne skape en enighet om at det eksisterer problemer innen feltet.

Denne masteroppgaven foreslår flere tiltak som kan gjennomføres for å forbedre sikkerheten og robustheten i systemene. Det viktigste av disse er det å oppnå et godt samarbeid mellom aktører som organisasjonens ledelse, prosessavdelingen, IKT-avdelingen og leverandør av SAS. Ved gode relasjoner mellom disse fagdisiplinene kan organisasjonen skaffe seg bedre rutiner for oppgradering av systemene, sikrere brannmurer (som følge av bedre regelsett og korrekt plassering) og fullstendige risikoanalyser som dekker flest mulig aspekter av SAS.

For å realisere sikrere strukturer i nettverket, anbefales det bruk av godt konfigurerte brannmurer mellom viktige nettverkssegmenter, bruk av DMZ-er som logiske barrierer der viktig trafikk må utveksles mellom to nettverkssegmenter eller forskjellige kommunikasjonsprotokoller, og bruk av spesialiserte IDS-er i segmenter hvor det går kritisk trafikk. Dersom organisasjonen skal knyttes til tredjeparts serviceleverandører, anbefales det å benytte VPN-kommunikasjon i kombinasjon med redundante kommunikasjonskanaler mellom partene for å både få en sikker kommunikasjon og å øke robustheten mot for eksempel nettstorm. Dette vil også kunne gjøre det mindre nødvendig for at organisasjonene å belage seg på «security by obscurity» enn det er i dag, hvor mye av sikkerheten ligger i at færrest mulig vet hvilke sikkerhetsmekanismer og strukturer som benyttes i de forskjellige systemene.

En god kultur for sikkerhet kommer ikke gratis og er et mål hele organisasjonen, inkludert ledelsen, er nødt til å jobbe mot og satse på for å kunne realisere. Funnene gjort gjennom de tilgjengelige tilsynsrapportene tyder det på at dette ikke er en oppgave organisasjonene har vært villig til å ta, resultatet blir da at det ikke blir tatt grep før et angrep faktisk skjer og man kan bli klar over mangelen på sikkerhet. En annen løsning vil kunne være at de regulatoriske myndighetene innfører et obligatorisk regelverk, for eksempel basert på OLF104, som dekker informasjonssikkerhet i IO. Inntil dette er innført anbefales det at flere felttilsyn blir gjennomført hos operatørene slik at sikkerhetsnivået, benyttede mekanismer og arbeidet rundt disse kan kartlegges bedre.

Abstract and conclusion

An important improvement for the operation of oil platforms is the incorporation of Integrated Operations (IO), which can result in cost savings for operations at up to 300 billion NOK. At the same time personnel become redundant at the platform and be relocated to land-based operation centers and thereby reducing the risk of the operation. The winnings related to IO do not come for free, and the organization is facing monumental challenges related to operational philosophy when operations are centralized in land-based operations centers.

To realize IO, action data, traditionally localized in closed systems offshore, has to be moved to shared networks in such a way that data can be reached onshore. Because of the large costs related to replacing existing systems, the necessary upgrades to realize these have been done in an iterative manner, where new functionality is added to the existing. This method of development can result in situations where systems which are designed to operate in environments with natural barriers or proprietary communication protocols loose these properties.

More systems and networks are opening up the surroundings and in combination with this, more systems are being based on commercial operating systems such as Microsoft Windows. The number of vulnerabilities in these operating systems has gotten the attention of many different actors within the area of expertise as potential targets for attacks. The most severe threats to the systems are posed by different countries cyber defense divisions. The force behind attacks from these actors became clear with Stuxnet [9], the first known malware directly targeting industrial systems. Through the use of Windows, organizations have also made themselves more vulnerable to threats from traditional malware.

During design of new SAS (Security & Automation Systems), there is focus on developing secure systems through the use of resilient units, where the units communicate together by using secure communication protocols. Since many of the old SAS' consist mainly of old units, it is necessary to implement measures to increase security within the structures of the network to obtain sufficient protection. Traditionally there have been many possible measures available for securing networks and connecting systems from the arising problems when networks are becoming increasingly connected. This master thesis will present different well known standards, philosophies of operation and security mechanisms which can be utilized to improve security such as defense-in-depth, encryption, VPN (Virtual Private Network), firewalls, DMZ (DeMilitarized Zones), IDS (Intrusion Detection Systems) and test procedures.

Results from inspections and surveys are used which reveal that these mechani-

sms are not always utilized as they should. As an example, only 19 % of the respondents had stripped the systems of unwanted or unused functionality (hardening), only 77 % used sufficiently secure firewalls between the SCADA-system (Supervisory Control And Data Acquisition) and Internet, and only 43 % replied that they periodically updated their systems [3]. The survey from [12] revealed that very few of the platforms had conducted detailed risk analysis of SAS, which could potentially result in that undetected and unwanted dependencies remains hidden, and could cause problems. These results should lead to a consensus that there exists several problems within the field that need to be addressed.

This thesis recommend several measures that could be carried out to improve the security and resilience of the systems. The most important measure is to create a well-functioning work environment with sufficient communication between the organization management, process-department, IT-department and SAS supplier. Good relations between these different branches could be exploited to create better routines for updates, safer firewalls (as a result of better configuration and placement) and overall risk analysis covering as many aspects of SAS as possible.

To create safer network structures it is recommended to use well-configured firewalls between the different segments and to use DMZs to create logical barriers either where important traffic has to be exchanged between segments or where different communication protocols are used. In addition specialized IDS' should be utilized in segments that transport critical traffic. If the organization want to connect to a third-party service provider, secure VPN-communication is recommended in combination with redundant communication channels to improve resilience. This may allow the organization to relay less of the concept of «Security by obscurity» where the key component of security lies in the fact that as few as possible know what mechanism that are used to secure the network and the structure of the network.

A good culture for security is hard to come by, and is something the whole organization has to work together to achieve. The findings from the different inspections and surveys show that this is a task that the organizations are not doing willingly and that in most cases an accident has to occur before action is taken and appropriate countermeasures are implemented. A better solution would be for the regulating authorities to impose a mandatory act or regulation, e.g. based on OLF104, which cover security in IO. Until this has taken place, the recommended action is to conduct frequent field inspections such that the level of security, safety, utilized mechanisms and the work surrounding these fields can be monitored.

Forord

Denne masteroppgaven er skrevet for Institutt for Teknisk Kybernetikk på Norges teknisk-naturvitenskaplige universitet (NTNU) våren 2013, som det avsluttende arbeidet på min mastergrad (M.Sc.) innen Teknisk Kybernetikk, Instrumenteringssystemer.

Under arbeidet med oppgaven fant jeg fort ut at det for hver sten jeg ville snu, så ville det dukke opp mange nye. Dette har ført til at jeg har brukt utallige timer på å lese relevante og irrelevante artikler og bøker, men dette har resultert i at jeg har lært mye om mange forskjellige temaer innen fagfeltet.

Mye av det jeg har lært og gjort gjennom dette semesteret befinner seg nå i den oppgaven du holder i hendene, og jeg håper at den vil kunne være til hjelp for å forstå de konsekvenser innføringen av integrerte operasjoner kan ha for sikkerheten, hvem som kan tenkes å angripe systemene, hvordan disse kan tenkes å angripe systemene, hvilke mekanismer som kan benyttes for å sikre systemer og nettverker og hvordan situasjonen i dag er med tanke på sikkerheten i systemene.

Jeg vil få sende en takk til følgende personer:

- Min hovedveileder, Professor Tor Onshus, for din hjelp og for muligheten du ga meg til å fordype meg i en oppgave og et fagfelt som jeg synes er både viktig og spennende.

- Til mine foreldre for at dere alltid har støttet, oppmuntret, og trodd på meg.

- Til min eksterne veileder, Seniorforsker Stig Ole Johnsen ved SINTEF, for at du hjalp meg i gang med arbeidet og leste gjennom oppgaven min.

- Til alle gutten på kontor D252 for at dere har gjort hverdagen til en fest.

- Til Mari Holmen for din fantastiske korrekturlesing

- Til alle som på en eller annen måte gjennom mine seks år i Trondheim har vært en del av min tilværelse. Uten dere ville aldri studietiden min vært så morsom, berikende, lærerik og givende som den har vært.

Dere har alle mine dypeste takk.

Trondheim, 7. juni 2013

Henrik Børstad Eriksen
henrik.eriksen@gmail.com

Liste over forkortelser og definisjoner

B&G	**B**rann- og **g**assdeteksjonssystemer
CRIOP	**C**risis **I**ntervention **i**n **O**ffshore **P**roduction
CSET	**C**yber **S**ecurity **E**valuation **T**ool
DMZ	**D**e**M**ilitarized **Z**one, Demilitarisert sone
DoS	**D**enial **o**f **S**ervice, Nettstorm
HMS	**H**else, **M**iljø og **S**ikkerhet
IDPS	**I**ntrusion **D**etection & **P**revention **S**ystem
IDS	**I**ntrusion **D**etection **S**ystem, Deteksjonssystemer for inntrenging
IKT	**I**nformasjons- og **k**ommunikasjons**t**eknologi
IO	**I**ntegrerte **O**perasjoner
ISBR	**I**nformation **S**ecurity **B**aseline **R**equirements
MMI	**M**enneske-**m**askin **i**nteraksjon
NAS	**N**ød**a**vstengnings**s**ystener
NIDPS	**N**etwork **I**ntrusion **D**etection & **P**revention **S**ystem
OLF	**O**ljeindustriens **L**ands**f**orbund. Nå: Norsk Olje & Gass
PAS	**P**rosess**a**vstengnings**s**ystemer
PCS	**P**rocess **C**ontrol **S**ystem, Prosesskontrollsystemer
PLS	**P**rogrammerbar **L**ogisk **S**styring
Ptil	**P**etroliums**til**synet
SAS	**S**ikkerhet- og **a**utomasjons**s**ystemer
SCADA	**S**upervisory **C**ontrol **A**nd **D**ata **A**cquisition, System for overordnet regulering og datainnsamling
SIL	**S**afety **I**ntegrity **L**evel
SIS	**S**ikkerhets**i**nstrumenterte **s**ystemer
TCP/IP	**T**ransmission **C**ontrol **P**rotocol/**I**nternet **P**rotocol
TOCSSiC	**T**eststand **O**n **C**ontrol **S**ystem **S**ecurity **i**n **C**ERN

Innhold

Sammendrag og konklusjon **iii**

Abstract and conclusion **v**

Liste over forkortelser og definisjoner **viii**

1 Introduksjon & motivasjon **1**

2 SCADA-systemer **3**
2.1 Innledning 3
2.2 Historie 3
2.2.1 Lokal regulering 3
2.2.2 Sentralisert regulering 4
2.2.3 Distribuert regulering 5
2.2.4 Innføring i SCADA-systemer 7
2.3 Utfordringer med utviklingen av systemene 10

3 Integrerte operasjoner **13**
3.1 Introduksjon 13
3.2 Endringer som følger av IO 15
3.3 Fordeler med IO 16
3.4 Utfordringer med IO 17

4 Sikkerhet og robusthet **19**
4.1 Sikkerhet 19
4.2 Informasjonssikkerhet 20
4.3 Robusthet 21

5 Trusler og sårbarheter **23**
5.1 Aktører 23
5.2 Noen typer angrep mot datasystemer 27
5.2.1 Ondsinnet kode 27
5.2.2 Nettstorm (DoS) 28
5.2.3 Snooping 30
5.2.4 Phishing 30
5.2.5 Spoofing 31
5.2.6 Inputvalidering 33
5.3 Sårbarheter 34

6 Ulykker og hendelser **37**

6.1 Zotob 37
6.2 Stuxnet 37
6.3 Duqu 38
6.4 Flame 40

7 Struktur og sikringsmekanismer 43
7.1 Forskrifter 43
7.2 Standarder og retningslinjer 44
7.3 Beskyttelse-i-dybden 51
7.4 Metoder for å øke robusthet 53
7.5 Brannmurer 55
7.5.1 Forskjellige typer brannmurer 56
7.5.2 Konfigurering av brannmurer 58
7.5.3 Implementering av brannmurer i nettverket 59
7.6 Demilitariserte soner (DMZ) 60
7.6.1 Diodeløsningen 62
7.7 Deteksjonssystem for inntrenging (IDS) 67
7.7.1 Innledning til IDS 67
7.7.2 IDPS 68
7.7.3 NIDPS 69
7.7.4 Honningkrukker 69
7.8 Kryptering 70
7.8.1 Prosedyrer for kryptering 71
7.8.2 Problemer med kryptering 71
7.9 Virtuelle privatnettverk (VPN) 72
7.9.1 Fordeler og problemer ved bruk av VPN 73
7.10 Hardening 74
7.11 Testing av komponenter og systemer 74
7.11.1 TOCSSiC 75
7.11.2 CSET 76
7.11.3 CRIOP 77

8 Dagens situasjon 79
8.1 Kilder benyttet for å kartlegge dagens situasjon 79
8.2 Avhengigheter mellom delsystemer 80
8.3 Dagens situasjon vedrørende angrep mot industrielle nettverk . . 82
8.4 Status for sikkerhetsmekanismene til systemene 84
8.5 Overordnet oversikt over dagens situasjon 86

9 Anbefalte tiltak 89
9.1 Tverrfaglig samarbeid 89

9.2 Tiltak for å øke mengden oppdateringer 89
9.3 Tiltak for å forbedre sikkerheten i brannmurer 90
9.4 Standarder og retningslinjer som et virkemiddel for å oppnå bedre sikkerhet 91
9.5 Bruk av IDS for å skape sikker kommunikasjon i nettverkssegmenter 91
9.6 Robusthet og risikoanalyse 91
9.7 Testing av systemet som helhet 92
9.8 Aktiv bruk av DMZ for å skape sikre barrierer 92
9.9 Tiltak for å kunne reagere aktivt på trusler 93
9.10 Tiltak for å kunne handle proaktivt 93
9.11 Tiltak som kan gjennomføres av regulerende myndigheter 94

10 Videre arbeid **95**

Referanser **99**

Vedlegg **103**

A Innhold på CD **103**

Figurer

2.1 Lokal reguleringsarkitektur hvor kontrollrommet befinner seg ved eller svært nærme prosessen. 4
2.2 Sentralisert reguleringsarkitektur hvor prosessavsnittene er koblet mot et sentralt kontrollrom ved hjelp av separate kabler. Selve prosessereguleringen foregår også sentralisert. 5
2.3 Distribuert reguleringsarkitektur hvor reguleringen foregår på lokalt nivå, og kommunikasjonen mellom kontrollrom og prosessavsnitt foregår over en digital nettverksforbindelse. 6
2.4 Overordnet skisse over komponenter i et SCADA-system hvor to PLS-er med tilhørende I/O er koblet til et industrielt nettverk, en SCADA-kontroller, en datalogger og en MMI-enhet. Nummerering er i samsvar med numrene gitt komponentene i kapittel 2.2.4 . . . 8
2.5 En visualisering av delsystemene i et SCADA-system samt hvordan disse systemene overordnet er koblet sammen [15]. 10
3.1 Figuren viser utviklingen i kompleksitet innen oppgaver i oljeutvinning. Innen feltet har en hatt utvikling fra enkeltstående, løse problemer og mot komplekse og tett koblede prosesser. Figuren er en reproduksjon av en figur fra [4]. 14
3.2 En skisse over integrerte operasjoner, hvor plattformen er koblet til et onshore kontrollrom over en høyhastighets nettverksforbindelse. Enkelte tjenester kan være utsatt til servicetilbydere, koblet til kontrollrommet gjennom en (antatt) sikker Internettforbindelse. 15
4.1 Illustrasjon over hvordan risikoen reduseres fra regionen for uakseptabelt høy risiko og ned til regionen med en risiko vi kan akseptere. Reduksjonen av risiko gjøres ved hjelp av sikkerhetstiltak, kontrollsystemer, operatørens inngrep samt forskjellige sikkerhetsfunksjoner. 20
5.1 Figuren viser et typisk oppsett for distribuert nettstorm. Til venstre er koordinatoren for angrepet, i midten de infiserte maskinene som benyttes for å generere nødvendig trafikkmengde og til høyre er målet for angrepet. 29
5.2 Figuren viser hvordan en angriper kan forfalske en pakke for å få den til å komme seg gjennom bedriftens brannmur [44]. 31
5.3 Figuren illustrerer et mann-i-midten angrep. Aktør 1 og aktør 2 har opprettet kommunikasjon, og tror at denne går direkte mellom hverandre. I virkeligheten har en angriper plassert seg mellom aktørene, og kan kontrollere all trafikk mellom dem. 32

5.4 Figuren viser hvordan strengen «robusthet» blir skrevet til variabel *A* og overskriver deler av minneområdet til variabel *B* da antall bokstaver i «robusthet» overskrider tiltenkt variabelstørrelse. . . . 33
6.1 Illustrasjon over hvordan Stuxnet logisk plasserer seg mellom kontrollsystemet på PC-ene, S7, og PLS-ene i prosessen. Øverste bilde viser original logisk flyt, mens den nederste viser den logiske flyten etter infeksjonen av Stuxnet. Bildet er hentet fra Wikipedia. 39
6.2 «Peer-to-peer»-kommunikasjon i Stuxnet og Duqu. Infisert PC i trygg sone uten tilgang til Internett kommuniserer med kontrollserveren ved hjelp av en annen infisert PC plassert i utrygg sone. «*I*» markerer infisert PC. 41
6.3 Illustrasjon av hvordan Flame laster seg selv inn i datamaskinen. Enten lastes mssecmgr.ocx seg inn i registeret, eller så kjøres rundll32.exe. Videre blir forskjellige deler av programvaren lastet inn ved hjelp av forskjellige legitime funksjonskall i Windows. 42
7.1 En skjematisk oversikt over beskyttelse-i-dybde og hvilke sikkerhetsmekanismer som kapsler inn bedriftens informasjon. Som en kan se fra figuren har menneskene i organisasjonen mye lettere tilgang til deler av denne informasjonen, noe som gjør sikkerhetsplanlegging, trening, kursing og regler viktig for å opprettholde sikkerheten. Figuren er hentet fra [44]. 52
7.2 Figuren viser Løkmodellen, en visuell fremstilling av beskyttelse-i-dybden presentert i [11]. I motsetning til sikkerhetssirkelen er Løkmodellen tilpasset industrielle nettverk. Mellom de forskjellige lagene plasseres brannmurer eller tilsvarende sikkerhetstiltak for sørge for tilstrekkelige barrierer og sikkerhet mellom delsystemene. 53
7.3 Figuren er en videreutviklet utgave av en figur fra [15] som viser hvordan risikoanalyser kontinuerlig bør gjennomføres i organisasjonen for å øke robustheten og dermed sikkerheten til systemene. 55
7.4 Figuren viser hvordan brannmurer både kan benyttes som barrierer mellom bedriftsnettverket og Internett og mellom bedrifts- og kontrollnettverket. 56
7.5 Overordnet oversikt over hvordan en DMZ kan benyttes til å koble sammen bedrifts- og prosessnettverk. 61
7.9 Figuren viser hvordan en kan implementer en diodeliknende løsning for å forhindre at kommunikasjon flyter fra DMZ til SIS ved fysisk å fjerne denne koblingen. 63
7.6 En skisse over prosessnettverk, bedriftsnettverk, mulighet for ekstern tilkobling samt en DMZ som begrenser nødvendig aksess fra bedrift- til prosessnettverk 64

7.7 En skisse over prosessnettverk, bedriftsnettverk, mulighet for ekstern tilkobling samt en DMZ som begrenser nødvendig aksess fra bedrift- til prosessnettverk. Utstyrt med to brannmurer mellom prosess- og bedriftsnettverk, noe som øker sikkerheten. 65

7.8 Figuren viser en mulig segmentering av et industrielt nettverk ved hjelp av brannmurer mellom de forskjellige segmentene samt DMZ-er for å kunne begrense trafikken som trenger å gå gjennom barrierene. Figuren er i tråd med den oppdeling gitt i «Løkmodellen» fra kapittel 7.3 og fargene til de forskjellige delsystemene harmoniserer med de gitt i «Løkmodellen», figur 7.2. 66

7.10 Figuren viser hvordan en IDS kan kobles inn i et nettverk, samt hvilke delsystemer som benyttes for å realisere IDS-en. Monitoren leser av trafikken som går gjennom enheten, mens konfigurasjonsserveren lagrer og analyserer dataene som går gjennom monitoren. IDS-GUI brukes til å presentere data fra IDS-serveren som alarmer eller tilsvarende. 67

7.11 Figuren viser en av de tidlige krypteringsalgoritmene, Ceasarcifferingen [21]. Algoritmen baserer seg på at bokstavene i byttes ut med bokstaver som er flyttet et gitt antall plasser i alfabetet. Som kan sees i figuren, vil for eksempel bokstaven «D» bli byttet ut med bokstaven «A», bokstaven «E» med «B» og så videre. Bildet er hentet fra Wikipedia. 70

7.12 Figuren viser VPN-koblingen mellom to forskjellige lokalnettverk. IP-pakkene blir i dette tilfellet pakket om til IPsec-pakker ved bruk av rutere med VPN-funksjonalitet. På mottaker-siden blir IPsec-pakken dekryptert og konvertert tilbake til en vanlig IP-pakke før den blir videresendt til mottaker. 73

7.13 Skisse over maskinvareoppsettet av TOCSSiC for å finne svakheter i systemer. Brukeren kan konfigurere hvilke tester som skal kjøres på konfigureringsmaskinen, en maskin benyttes til analyse av trafikken på nettverket og en siste maskin til å kjøre de faktiske testene. To høyhastighets-switcher binder alt sammen. 75

7.14 Dialogvindu i nettverksskanneren Nmap. Til høyre sees oppsettet av skann, mens resultatet sees i det venstre vinduet. Her kan man blant annet se åpne porter, adresser, tjenester og operativsystem fra maskinene som er skannet. 77

8.1 Figuren viser alvorlighetsgradene av nettstormangrepene som hele 54 % av de spurte i [3] svarte at de hadde blitt utsatt for. 83

Tabeller

5.1 Oversikt over trusselaktører innenfor cyberangrep *[1/2]*, hentet fra [38]. 25
5.2 Oversikt over trusselaktører innenfor cyberangrep *[2/2]*, hentet fra [38]. 26
7.1 Standarder og retningslinjer innen sikkerhet for automasjonssystemer. [1/2] 49
7.2 Standarder og retningslinjer innen sikkerhet for automasjonssystemer. [2/2] 50
7.3 Eksempel på hvordan en tilstandsbevist pakkefiltrende brannmur kan holde oversikt over forskjellige tilkoblinger. 58

1 Introduksjon & motivasjon

Motivasjon
I 2009 brøt Stuxnet ut som det første viruset spesielt laget for å sette industrielle kontrollsystemer ut av spill, og viruset skapte blant annet store problemer for Irans atomprogram. Samtidig spredde viruset seg videre til tilsvarende systemer rundt i verden og skapte på sin vei store problemer og medieoppslag. Teknisk Ukeblad kunne 17. Mai 2013[1] melde om at angrep fra Midtøsten, rettet mot kritisk infrastruktur på den vestlige halvkule stadig øker, og dette er angrep som potensielt kan føre til skade på mennesker, miljø og profitt. Samtidig som sektoren stadig oftere opplever angrep, meldes det i [3] at 1/3 av de spurte systemadministratorer for disse systemene svarte at deres budsjetter for å utføre sikkerhetsarbeidet hadde blitt kuttet som resultat av finanskrisen [3].

Integrerte operasjoner (IO) er en ny driftsfilosofi innen oljesektoren hvor arbeidsoppgaver og beslutningstaking flyttes fra plattformene og inn på land, og har de siste årene virkelig begynt å gjøre sitt inntog. IO byr på nye muligheter for sektoren i form av høyere sikkerhet og større profitt, men samtidig som operatørene ser de store fordelene med IO, fører også driftsfilosofien med seg nye utfordringer som ikke er like lett å se rundt blant annet informasjonssikkerheten til de industrielle nettverkene og systemene som benyttes.

På grunn av disse sikkerhetsmessige utfordringene rundt IO vil denne masteroppgaven forsøke å kombinere og systematisere noe av den litteraturen som finnes innen fagfeltet for å kunne gi leseren en innføring (med en hovedvekt på de tekniske faktorene) i følgende temaer: Integrerte operasjoner, utviklingen av industrielle kontrollsystemer, hvilke trusler disse systemene står ovenfor og hvilke mekanismer eller strukturering av nettverkene som kan benyttes for å beskytte mot forskjellige angrep. For å gi et bilde av dagens situasjon vil oppgaven så gi leseren en oppsummering av de tilgjengelige tilsynsrapporter og spørreundersøkelser som er gjennomført for å kartlegge hvilke tiltak som blir gjort for å ivareta sikkerheten til disse systemene. Avslutningsvis vil det bli gitt flere forslag til tiltak som forfatteren av denne oppgaven mener at organisasjonen kan gjennomføre for å forbedre informasjonssikkerheten.

Mål
For å ha klare mål å jobbe mot ble det tidlig utformet noen punkter som ble benyttet under arbeidet med masteroppgaven.

1. Hva er integrerte operasjoner (IO) og hvilke muligheter og utfordringer fører denne driftsfilosofien med seg?

[1] http://www.tu.no/olje-gass/2013/05/17/prover-a-lamme-oljeproduksjon-med-cyberangrep

2. Hvilke angrep kan utføres mot industrielle systemer og nettverk, og hvilke mekanismer kan benyttes for å beskytte industrielle system mot disse?
3. Hva blir det gjort i dag for å opprettholde informasjonssikkerheten i slike system og hva kan gjøres bedre?

Organisering
Det vil nå bli gitt en kort gjennomgang av hvordan denne oppgaven er bygd opp og hva de forskjellige kapitlene inneholder.

Oppgaven starter i kapittel 2 med å gi en introduksjon av utviklingen industrielle reguleringssystemer har hatt fra helt enkle, lukkede systemer til dagens komplekse SCADA-systemer (*en: Supervisory Control And Data Acquisition*). Det vil så bli diskutert hvordan utviklingen innad i feltet kan skape problemer for dagens systemer. Dette kapittelet vil også belyse hvordan man tradisjonelt strukturerer SCADA-systemene inn i en rekke delsystemer. Oppgaven vil så i kapittel 3 gi leseren en introduksjon til ulike sider ved integrerte operasjoner, for så å definere begrepene sikkerhet, informasjonssikkerhet og robusthet i kapittel 4.

I kapittel 5 vil fokuset rettes mot de potensielle farene et industrielt datasystem står ovenfor, samt hvem som utøver disse truslene og hvilke sårbarheter systemene kan ha. I kapittel 6 vil et par av de store angrepene mot datasystemer man har hatt i senere tid bli presentert. Det vil så i kapittel 7, oppgavens mest omfattende kapittel, bli gitt en gjennomgang av forskjellige standarder, forskrifter og ikke minst sentrale driftsfilosofier og sikkerhetsmekanismer som organisasjonene kan benytte seg av for å gjøre systemene sine bedre sikret mot angrep eller oppnå en ønsket struktur på nettverket.

For å kartlegge hvilken situasjon man har på sikkerheten i dag vil det i kapittel 8 bli gitt et sammendrag og diskusjon rundt data tilgjengelig fra tilsyn og spørreundersøkelser. Det vil så i kapittel 9 presenteres hvilke tiltak som forfatteren av denne masteroppgaven foreslår kan gjennomføres for å ytterligere forbedre sikkerheten i slike systemer, før oppgaven rundes av i kapittel 10 med å gi en pekepinn for hvordan arbeidet i denne masteroppgaven kan videreføres av andre.

2 SCADA-systemer

Fremgang er menneskers evne til å gjøre det enkle komplisert.
— Thor Heyerdahl

2.1 Innledning

SCADA (*en: Supervisory Control And Data Acquisition*), system for overordnet regulering og datainnsamling, er samlebetegnelsen for systemene som binder sammen, overvåker og tar seg av den overordnede reguleringen i mange store industrielle nettverk [38]. SCADA-systemer blir hovedsakelig benyttet på anlegg hvor den fysiske avstanden mellom de forskjellige komponentene i anlegget er stor eller dersom det er nødvendig å samle sammen og lagre prosessdata [2].

Muligheten til å kunne overvåke store anlegg på en effektiv måte ved bruk av SCADA-systemer har gjort det mulig å håndtere svært komplekse prosesser som for eksempel prosessene man finner på en oljeplattform. Ved å integrere de forskjellige delsystemene, kan man ved hjelp av SCADA også oppnå økt sikkerhet og produksjon.

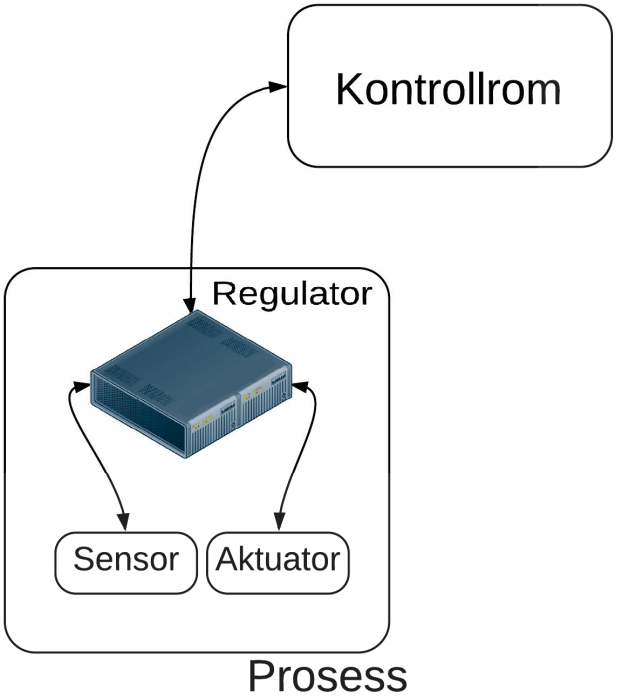
2.2 Historie

Utviklingen frem mot dagens SCADA-systemer har foregått over mange år, og historien om hvordan systemene har fått dagens form er viktig for å forstå hvorfor noen SCADA-systemer i dag ofte har sikkerhetsmessige svakheter.

2.2.1 Lokal regulering

Rundt 1930 begynte de første regulatorene å gjøre sitt inntog i prosessanlegg, og mer eller mindre automatiserte løsninger begynte å se dagens lys [29]. I den tidlige fasen med disse systemene befant alt utstyr som hadde en sammenheng med reguleringen av prosessavsnittene seg i fysisk nærhet av prosessen [5], en skisse av et prosessanlegg med lokal reguleringsarkitektur kan sees i figur 2.1. Denne fysiske nærheten mellom måleutstyr og prosessen gjorde at prosessingeniøren kom svært nært på den fysiske prosessen, og fikk en god forståelse for

prosessens dynamikk. Problemet med lokal regulering var blant annet at det kreves mye personell, og at disse, på grunn av fysisk nærhet til prosessen, var svært utsatt dersom noe utilsiktet og farlig skulle skje.

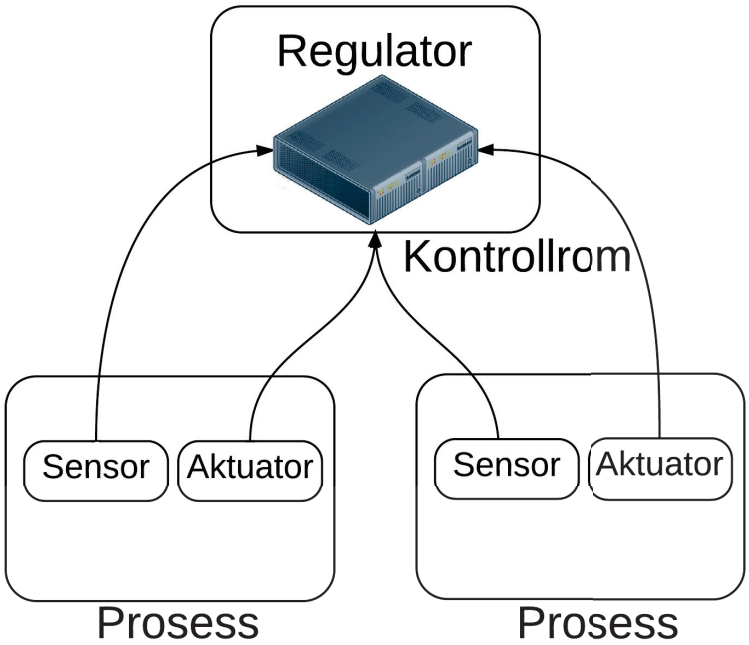


Figur 2.1: *Lokal reguleringsarkitektur hvor kontrollrommet befinner seg ved eller svært nærme prosessen.*

2.2.2 Sentralisert regulering

Etterhvert begynte det å dukke opp instrumenter med pneumatisk signalering på markedet, noe som gjorde at kontrollenheten og regulatorene kunne flyttes bort fra prosessen og sentraliseres i et kontrollrom [5], som vist i figur 2.2. Denne endringen i systemarkitekturen gjorde det mulig å ha færre operatører, og eksponeringstiden mot farlig prosessmiljø gikk derfor betydelig ned.

Problemet med sentralisert regulering var og er at operatørene kan få en dårligere fysisk forståelse av prosessen, samt at det er forbundet store utgifter med «punkt-til-punkt»-kobling benyttet i denne reguleringsarkitekturen når mengden prosessutstyr øker.



Figur 2.2: *Sentralisert reguleringsarkitektur hvor prosessavsnittene er koblet mot et sentralt kontrollrom ved hjelp av separate kabler. Selve prosessereguleringen foregår også sentralisert.*

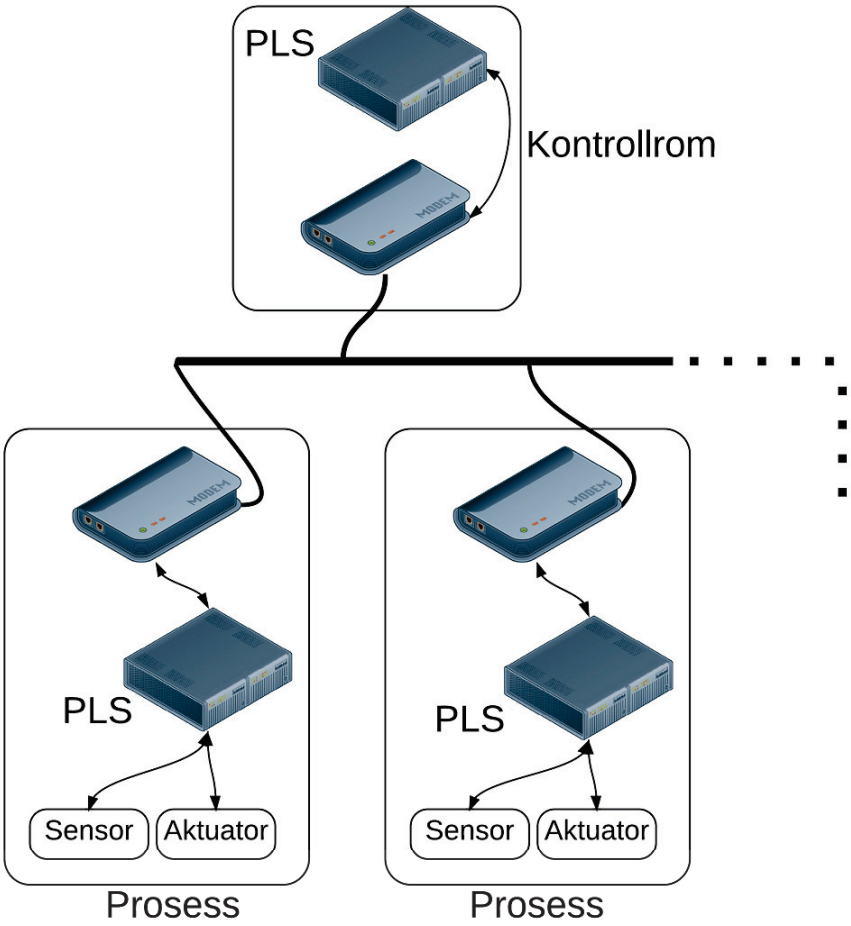
2.2.3 Distribuert regulering

Etterhvert som kontrollsystemer blir mer komplekse, øker antall kabler inn til kontrollrommet, og kostnadene ved installering og vedlikehold av kontrollsystemene øker deretter.

Distribuert reguleringsarkitektur kombinerer den lokale og den sentraliserte arkitekturen i en struktur bedre egnet for moderne reguleringssystemer. Distribuert reguleringsarkitektur plasserer regulatorene (for eksempel realisert ved bruk av PLS-er, programmerbar logisk styring) lokalt i prosessen, og disse bindes så sammen med kontrollrommet ved hjelp av en digital kommunikasjonskanal, en skisse av denne reguleringsarkitekturen kan sees i figur 2.3.

Den digitale kommunikasjonskanalen muliggjør bruk av «drop-down» buss-struktur på nettverket, og kabelmengden vil derfor vanligvis kunne reduseres betraktelig sammenlignet med sentralisert reguleringsarkitektur.

Siden reguleringen i den distribuerte reguleringsarkitekturen utføres lokalt, vil denne arkitekturen ikke være like avhengig av kommunikasjonen mellom kontrollrom og prosessen som den sentraliserte arkitekturen da de lokale regulatorene kan fortsette autonomt ved for eksempel kommunikasjonssvikt.



Figur 2.3: *Distribuert reguleringsarkitektur hvor reguleringen foregår på lokalt nivå, og kommunikasjonen mellom kontrollrom og prosessavsnitt foregår over en digital nettverksforbindelse.*

2.2.4 Innføring i SCADA-systemer

Som en naturlig utvikling av den distribuerte reguleringsarkitekturen fulgte SCADA-systemene, tilpasset komplekse anlegg spredd over store fysiske avstander med muligheter for logging og presentasjon av data. Et moderne SCADA-system består av flere sentrale komponenter, og de mest sentrale komponentene vil nå presenteres. Tallene på komponentene er i samsvar med de gitt i figur 2.4.

1. **SCADA kontrollserver:** Styrer datainnsamlingen fra enhetene i nettverket. Da det ofte er spesifisert hvilken ytelse systemet skal ha, setter dette krav til både programvaren og maskinvaren i denne serveren.

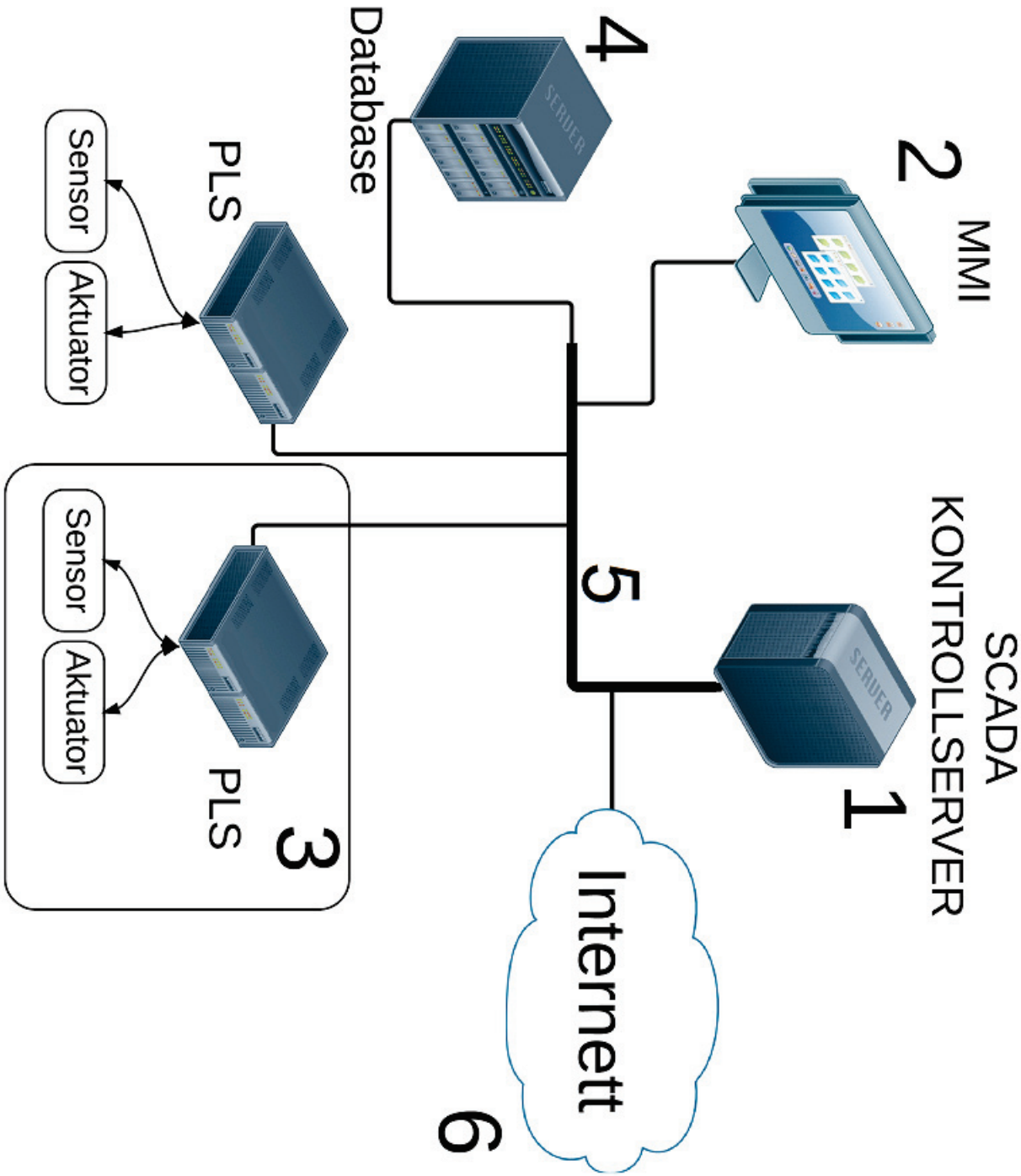
2. **Menneske-maskin interaksjon:** MMI-enheter (menneske-maskin-enheter) er en viktig del av SCADA-systemet da det er disse enhetene av systemet som er synlige for operatøren. MMI-enheter skal presentere prosessverdier, alarmer og trender til operatøren, samt å kunne benyttes for å endre settpunkter eller liknende i prosessen.

3. **Eksterne enheter:** De eksterne enhetene står gjerne i periferien av SCADA-systemene og kan være sensorer, aktuatorer eller PLS-er. De eksterne enhetene blir enten avlest av kontrollserveren eller så kan enhetene selv sende data periodisk til kontrollserveren.

4. **Loggserver:** Overvåker forhåndsbestemte verdier i prosessen og lagrer disse til senere bruk. Loggede data kan for eksempel brukes i trending av variabler eller for å detektere feil.

5. **Nettverket:** Nettverkets oppgave er å binde sammen den sentrale enheten og eksterne enheter. Det finnes svært mange forskjellige nettverk som kan benyttes i industrielle applikasjoner, og nettverket må derfor velges etter de behovene man har. Det er derfor viktig å kartlegge hvilke krav systemet har med tanke på faktorer som motstandsdyktighet for elektrisk støy, hastighetskrav og sikkerhet.

6. **Ekstern kommunikasjon:** I moderne SCADA-systemer har det blitt mer og mer vanlig å muliggjøre for ekstern kommunikasjon til og fra SCADA-systemet. Den eksterne kommunikasjonen åpner for nye muligheter, men åpner også opp for mange potensielle problemer, noe som vil bli diskutert senere.



Figur 2.4: *Overordnet skisse over komponenter i et SCADA-system hvor to PLS-er med tilhørende I/O er koblet til et industrielt nettverk, en SCADA-kontroller, en datalogger og en MMI-enhet. Nummerering er i samsvar med numrene gitt komponentene i kapittel 2.2.4*

Delsystemer i SCADA

Et SCADA-system deles gjerne opp i flere delsystemer med hver sine oppgaver, og dette delkapittelet vil gjennomgå disse delsystemene og deres forkortelser som blir benyttet både i industri og litteratur.

Av systemene for vanlig drift, benyttes forkortelsen PCS (*en: **P**rocess **C**ontrol **S**ystem*) for systemene forbundet med reguleringen av prosessen, og systemene for nedstengning av PCS under vanlig drift kalles PAS (***p**rosess**a**vstengnings**s**ystem*). Sammen med de sikkerhetsrelaterte delsystemene går disse to delsystemene under betegnelsen SAS (***s**ikkerhet- og **a**utomasjons**s**ystemer*).

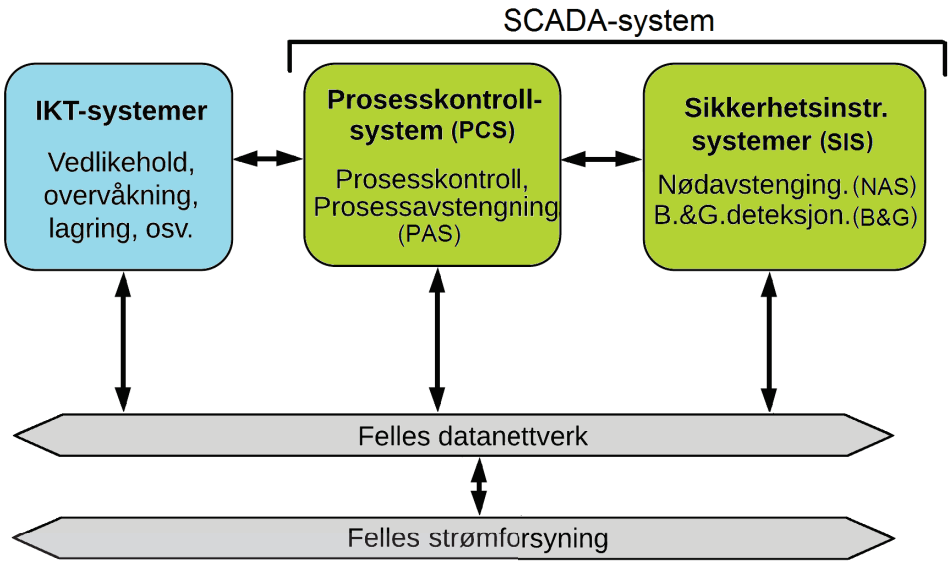
For å holde de generelle systemene adskilt fra systemene ansvarlige for sikkerhet, refereres disse sikkerhetssystemene til som SIS (***s**ikkerhets**i**nstrumenterte **s**ystemer*). Under SIS finnes systemene for nødavstenging av prosessen (NAS) som benyttes dersom PAS ikke fungerer. Under SIS finnes også systemer for brann- og gassdeteksjon (B&G).

De forskjellige delsystemene vil da altså bli som følger:

- **SAS:** Sikkerhet- og automasjonssystemer. Paraplybetegnelse for alle deler av systemet som enten er en del av automasjonssystemet eller sikkerhetssystemene.
- **PAS:** Prosessavstengingssystem. Systemet som skal sørge for avstengning av prosessen under vanlig drift.
- **SIS:** Sikkerhetsinstrumenterte systemer. Fellesbetegnelse for alle systemer som er ansvarlige for å opprettholde sikkerheten til prosessen eller bryte inn dersom noe går galt.
- **NAS:** Nødavstengningsssystem. Ansvarlig for å kunne stenge ned prosessen på en sikker måte uavhengig av PAS.
- **B&G:** Brann- og gassdetekssjonssystemer. Ansvarlig for deteksjon og håndtering av situasjoner vedrørende brann og gass.

Denne oppdelingen av SCADA-systemet er ikke gjort kun som en konseptuell modell, men er en oppdeling i overensstemmelse med hvordan det er ønskelig og/eller pålagt at systemet skal deles opp. Krav til atskilthet og uavhengighet mellom flere av disse systemene finnes blant annet i Ptils forskrifter, og vil bli diskutert nærmere i kapittel 7.1.

Det er i [15] visualisert hvordan disse systemene henger sammen, og denne figuren kan sees i fornorsket utgave i figur 2.5.



Figur 2.5: *En visualisering av delsystemene i et SCADA-system samt hvordan disse systemene overordnet er koblet sammen [15].*

2.3 Utfordringer med utviklingen av systemene

Avanserte prosessanlegg utgjør en betydelig levetidskostnad for de fleste organisasjoner. Dette fører gjerne til at reguleringssystemene og derunder SCADA-systemene gjerne gjennomgår flere iterative oppgraderinger for å kunne holdes moderne nok til å blant annet takle den økende datamengden eller andre krav slike systemer må håndtere. Denne formen for iterativ oppgradering av reguleringssystemer har ført til flere sikkerhetsmessige utfordringer som ikke var forutsatt da systemene ble designet.

En av oppgraderingene som har blitt gjort i mange systemer er skifte fra proprietære kommunikasjonsprotokoller til åpne kommunikasjonsprotokoller. Fordelen med å gå over på en åpen protokoll er at disse gjerne støtter høye overføringshastigheter, for eksempel opp til 1GB/s for TCP/IP (*en: **T**ransmission **C**ontrol **P**rotocol/**I**nternet **P**rotocol*) ved bruk av standardisert maskinvare. Bruken av slike åpne protokoller gjør at man kan benytte kommersielt tilgjengelig maskinvare, som kan senke investeringskostnadene betydelig.

Problemet med å benytte åpne protokoller er at langt flere har forsøkt å finne svakheter i disse protokollene, noe som kan gjøre SCADA-systemer med åpne

protokoller mer utsatt for tradisjonelle dataangrep.

Som nevnt tidligere i delkapitlet har systemene lang forventet levetid og dette kan by på mange problemer etterhvert som tiden går blant annet kan man forvente at det vil dukke opp nye trusler, angrep eller andre skadelige situasjoner etterhvert som årene går. Det er derfor viktig at både systemenes program- og maskinvare lar seg oppgradere samt at leverandøren kan levere oppgraderinger slik at systemene kan sikres mot nye trusler.

3 Integrerte operasjoner

Integrerte operasjoner vil si å fjerne skillet mellom disipliner, arbeidsgrupper og selskaper ved å benytte seg av sanntidsdata og ny teknologi
— Statoil

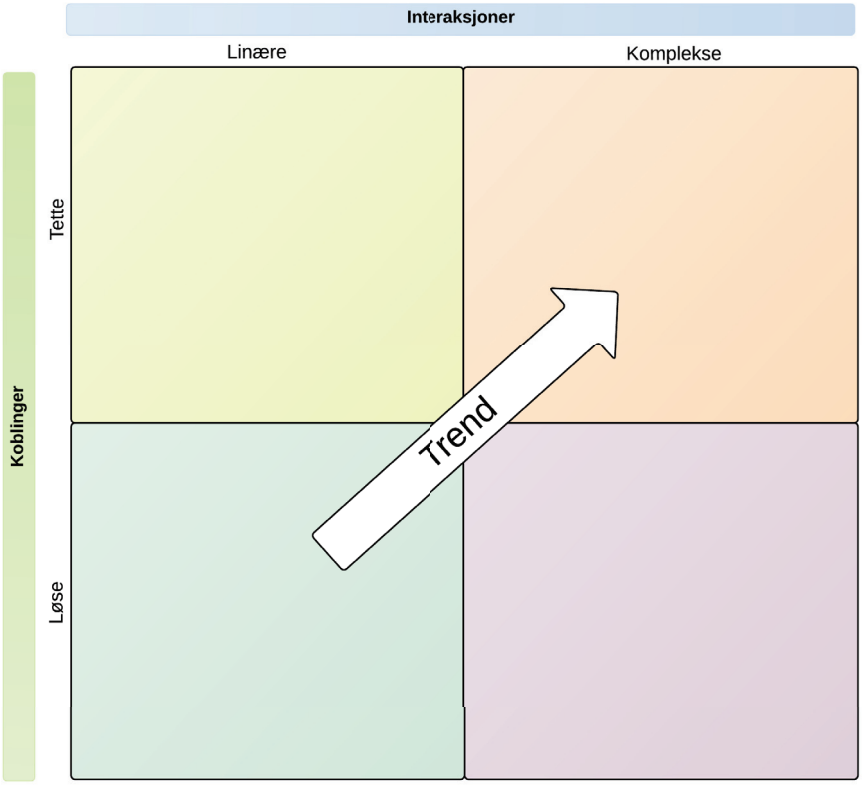
3.1 Introduksjon

Integrerte operasjoner ble i stortingsmelding 38 fra 2003-2004 [8] definert som følger: «*E-drift eller integrerte operasjoner (IO) innebærer bruk av informasjonsteknologi til å endre arbeidsprosesser for å oppnå bedre beslutninger, til å fjernstyre utstyr og prosesser og til å flytte funksjoner og personell til land*». Som nevnt i sitatet, så benytter IO den utviklingen som har skjedd innen nettverkskapasitet og nettverkshastighet i SCADA-systemer og høyhastighets nettverksforbindelser (for eksempel fiberoptiske nettverk) mellom on- og offshoreinstallasjoner for å realisere en driftsfilosofi som imøtekommer den økende kompleksiteten innen oljeoperasjoner. Grunnen til den økende kompleksiteten er at delsystemene blir mer avhengige samt at interaksjonene mellom prosessene blir mer komplekse, som vist i figur 3.1

Ved bruk av høyhastighets nettverksforbindelser mellom on- og offshore kan betydelige mengder prosessdata sendes i tilnærmet sanntid fra prosessen til operasjonssentraler onshore. IO kan benytte denne dataen til å endre driftsfilosofien som tradisjonelt har regjert innen fagfeltet, hvor eksperter har vært utplassert på de forskjellige oljeplattformene, og over mot en driftsfilosofi der ekspertpersonell onshore står for en langt større del av beslutningstakningen enn hva de tradisjonelt har gjort.

Med innføringen av IO kan man etterhvert tenke seg mer eller mindre ubemannede plattformer hvor det som er av personell i hovedsak utfører nødvendige inspeksjoner og reparasjoner, mens planlegging, overvåking, analyse og beslutningstakingen rundt operasjoner foregår onshore, enten i regi av operatør eller en av operatørens servicetilbydere, som vist i figur 3.2.

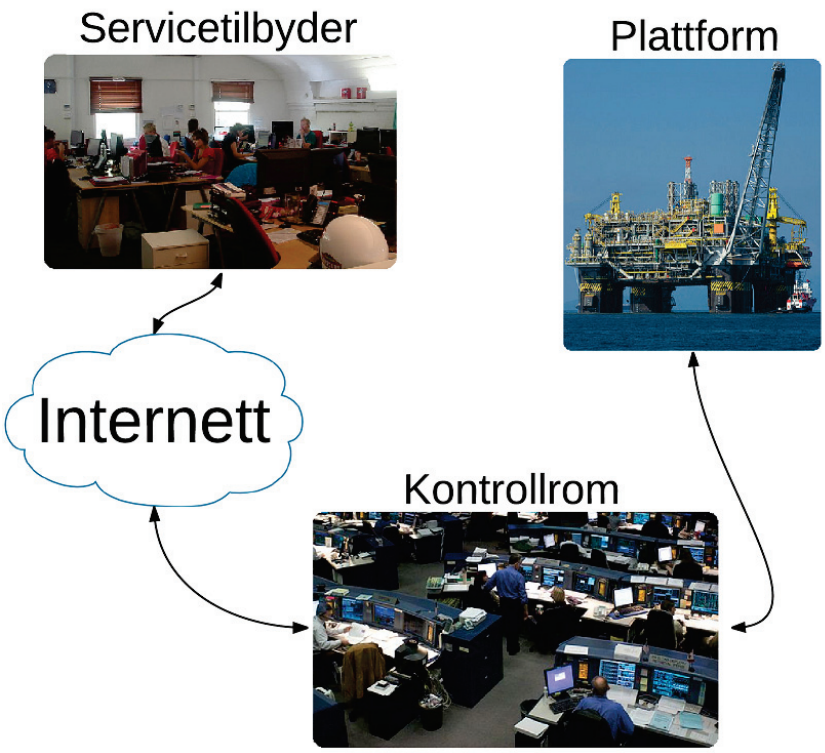
For oljeselskapene vil innføringen av IO skape en sterke kobling mellom onshore og offshore, og vil som nevnt kunne føre til økt profitt og forbedret sikkerhet. Norsk Olje & Gass har anslått opp mot 8-10 % økning i produksjon, 4-5 % økning i utvinningsgrad og opp til 30 % reduksjon i driftskostnadene ved bruk av



Figur 3.1: *Figuren viser utviklingen i kompleksitet innen oppgaver i oljeutvinning. Innen feltet har en hatt utvikling fra enkeltstående, løse problemer og mot komplekse og tett koblede prosesser. Figuren er en reproduksjon av en figur fra [4].*

IO [8]. Norsk Olje & Gass har estimert gevinstmulighetene til å utgjøre rundt 300 milliarder fordelt på følgende poster [36]:

- Besparelser som følge av reduserte kostnader ved boring: 21 milliarder NOK.
- Besparelser som følge av reduserte driftskostnader: 45 milliarder NOK.
- Gevinst som følge av økt utvinningsgrad: 174 milliarder NOK.
- Gevinst som følge av raskere produksjon: 50 milliarder NOK.



Figur 3.2: *En skisse over integrerte operasjoner, hvor plattformen er koblet til et onshore kontrollrom over en høyhastighets nettverksforbindelse. Enkelte tjenester kan være utsatt til servicetilbydere, koblet til kontrollrommet gjennom en (antatt) sikker Internettforbindelse.*

3.2 Endringer som følger av IO

Dette kapittelet vil behandle noen av endringene IO fører med seg i driften av operasjoner, da disse endringene ligger til grunn for å forstå flere av fordelene og utfordringene som følger med innføringen av IO.

Utstrakt innføring av IO vil uten tvil endre det innarbeidede hierarkiet og arbeidsmønstrene som til nå har vært vanlig innenfor offshore. I [25] presenteres fire følgende hovedendringer innen driften som vil følge innføringen av integrerte operasjoner:

- **Seriell → Parallell**
  Tradisjonelt har oljeplattformer vært utformet som samlebåndsprosesser tilsvarende det som har vært vanlig å finne i tradisjonell industri. Da det kan oppstå mye dødtid ved bruk av slike samlebåndsprosesser, vil det ved innføringen av IO være mulig å bruke avanserte verktøy for å lettere planlegge

og allokere ressurser mest mulig effektivt, noe som kan føre til en reduksjon av driftskostnadene for operatørene.

- **Enkeltdisiplinære arbeidsgrupper → Multidisiplinære arbeidsgrupper**
  Der oljeindustrien inntil nyere tid har benyttet eksperter offshore som har måtte samarbeide sammen med andre i utfordrende forhold offshore, vil IO i større grad benytte multidisiplinære arbeidsgrupper bestående av mange fagfelt for å oppnå en mer effektiv og sikker produksjon [28]. En multidisiplinær arbeidsgruppe vil kunne tolke ulike deler av prosessdataene individuelt, og fatte beslutninger basert på gruppens samlede kunnskap. At nye multidisiplinære arbeidsgrupper opprettes kan også føre til at ansatte fra forskjellige avdelinger møtes, noe som kan resultere i nye impulser og ideer [42].

- **Avhengig av fysisk tilstedeværelse → Uavhengig av fysisk tilstedeværelse**
  Den daglige driften av oljeplattformer har som nevnt inntil nylig i stor grad bestått av høyt kvalifisert personell plassert ut på de forskjellige plattformene. Dette ble tradisjonelt gjort fordi ekspertene behøvde tilgang til nødvendig prosessdata for å kunne bistå raskt dersom dette trengtes. IO gjør det mulig å sentralisere ekspertise onshore ved bruk av de store mengdene prosessdata som kommer fra plattformen i tilnærmet sanntid.

- **Beslutninger basert på erfaring → Beslutninger basert på sanntidsdata**
  Der beslutninger tidligere ble fattet av mindre grupper på grunnlag av operatørens kjennskap og nærhet til prosessen, gjør IO det mulig å kunne fatte bedre og sikrere beslutninger. Dette kan realiseres ved at personell onshore ved hjelp av IO har en enda bedre oversikt over alle prosessvariabler.

I tillegg til disse endringene som hovedsakelig er på systemnivå, vil det også nødvendigvis måtte gjøres endringer på komponent- og delsystemnivå [42]. Disse endringene vil for eksempel være å klargjøre utstyr for fjerndrift eller instrumentere nye prosessverdier som trengs.

3.3 Fordeler med IO

IO gir store muligheter for å rasjonalisere et ikke-optimalt driftsmønster, noe som kan føre til at norsk industri får en økonomisk gevinst i form av økt profitt og reduserte driftskostnader (se kapittel 3.1) som videre kan bidra til en bedret konkurranseevne [28]. Da IO også vil gjøre det mulig for operatørene å sette ut enkelte arbeidsoppgaver til tredjeparts kontraktører (serviceleverandører), for eksempel

til bedrifter som opererer i land med lavere drifts- og lønnskostnader enn i Norge, vil det være muligheter til å kunne øke gevinstene ytterligere.

Opprettelsen av multidisiplinære beslutningsgrupper onshore vil kunne føre til at en kan samle mye av den kunnskapen og erfaringen som i dag finnes i personell spredd rundt omkring på operatørens plattformer. Dette kan føre til mer gjennomtenkte beslutninger som i større grad er driftsoptimale og opprettholder sikkerheten bedre enn hva som tidligere har vært praksis.

I [19] er det også lagt vekt på de sikkerhetsmessige fordelene integrerte operasjoner kan føre med seg når en flytter en stor andel av personell fra off- til onshore. Færre mennesker på plattformene vil direkte resultere i for eksempel færre helikopterturer, redusert eksponering av fare for personell og lavere lønnskostnader for operatørene. Dette vil direkte kunne redusere noe av risikoen forbundet med operasjonene.

Utviklingen og integreringen av IO kan vise seg å være svært viktig for å opprettholde lønnsomheten og interessen for norsk sokkel og gode systemer for IO vil kunne være en attraktiv eksportartikkel for norske teknologiselskaper i fremtiden [19]. Dersom man tidlig får implementert gode løsninger, vil dette gi kompetanse som kan hjelpe norske bedrifter å få istand samarbeid med utenlandske operatører. Gode implementeringer av IO vil også klargjøre for operasjoner i arktiske strøk, hvor ekstreme temperaturer gjør mulighetene for manuell drift og inspeksjon enten er svært krevende eller umulig [19].

3.4 Utfordringer med IO

Selv om de potensielle gevinstene med innføring av IO er store, er det også flere utfordringer forbundet med IO, og dette delkapittelet vil forsøke å oppsummere de mest sentrale av disse.

En nødvendighet for utstrakt innføring av IO er en høyere grad av automatisering i delsystemer enn hva man har i dag. Dette vil i praksis bety at flere systemer må kunne kjøre mer eller mindre autonomt og operatører onshore må overvåke eller sitte som backup dersom det skulle skje noe galt. Ulempene med å gjøre operatøren til en passiv deltager (observatør) i systemdriften er behandlet i [31]. En av de største farene med denne utviklingen er at operatører kan miste den fysiske forståelsen med systemet, miste fokus og ikke få med seg om farlige situasjoner oppstår dersom arbeidsoppgavene er for ensformige [31]. IO vil også føre til at mye av den praktiske erfaringen til dagens offshorepersonell etterhvert kan forsvinne da flere av disse stillingene flyttes onshore eller blir overflødige [19]. Ved

å utvikle gode simulatorer og treningsprosedyrer vil man derimot kunne ha muligheten til å holde på mye av denne erfaringen og benytte den i trening av nytt personell slik at kunnskapen ikke går tapt.

Det ble i kapittel 3.2 gitt en kort innføring i det multidisiplinære samarbeidet som er en viktig del av driftsfilosofien i IO. Det å få multidisiplinære arbeidsgrupper til å fungere sammen er derimot ingen selvfølge, og i [19] nevnes flere potensielle problemer som kan oppstå innad i organisasjonen på grunn av dette. En av farene er at det kan oppstå situasjoner hvor ansvarsforholdet i arbeidsgruppene blir uklare, noe som kan føre til at det er uklart hvem som sitter med det egentlige ansvaret i farlige situasjoner.

På grunn av de store organisatoriske endringene som er nødvendige for å realisere IO, har man valgt å fase inn IO iterativt slik at overgangen skal bli mykere. Dette er i seg selv bra da organisasjonen kan omstille seg sakte, men sikkert, men det er også i [19] argumentert for at en slik kontinuerlig endring som går over mange år kan resultere i at en «sliter ut» organisasjonen. Dette kan skje ved at ansatte blir lei de stadige endringene i for eksempel roller og prosedyrer som følger av denne gradvise innfasing.

Det er også flere utfordringer knyttet til informasjonssikkerhet som det bør legges vekt på ved innføring av IO. Blant annet vil personell onshore være svært avhengige av de prosessverdiene som kommer fra offshoreinstallasjoner, er det veldig viktig at kommunikasjonslinken mellom on- og offshore er pålitelig. Dersom det skulle skje noe utilsiktet med den primære linken, bør det finnes prosedyrer og tekniske løsninger som legger til rette for enkelt å kunne overføre trafikken til redundante eller alternative kommunikasjonskanaler.

Avdelingene for automasjon og IKT (***I**nformasjons- og **K**ommunikasjons**t**eknologi*) har tradisjonelt vært adskilt og ikke hatt som tradisjon å jobbe tett sammen [19]. Disse to disiplinene vil ved innføring av IO bli koblet tettere sammen da automasjonssystemene vil sende sin data over IT-nettet. Tett samarbeid mellom disse disiplinene vil da bli viktig for å skape sikre løsninger for IO. Det er også svært viktig at det kartlegges hvilke mål som skal benyttes for å vurdere kvaliteten på løsninger, da disse faggruppene tradisjonelt har hatt forskjellige syn på dette. Flere av disse ulikhetene blir tatt opp i [38], og viktige forskjeller er blant annet at IKT-systemer ikke har det samme fokus på konsistent ytelse, konsistent respons og at det gjerne ikke er tatt hensyn den menneskelige sikkerheten på grunn av svikt i systemer i tradisjonelle IT-systemer.

4 Sikkerhet og robusthet

Sikkerhet: Beskyttelse av informasjon, kritiske komponenter og systemer som bruker, lagrer eller videreformidler informasjon
— Michael E. Whitman

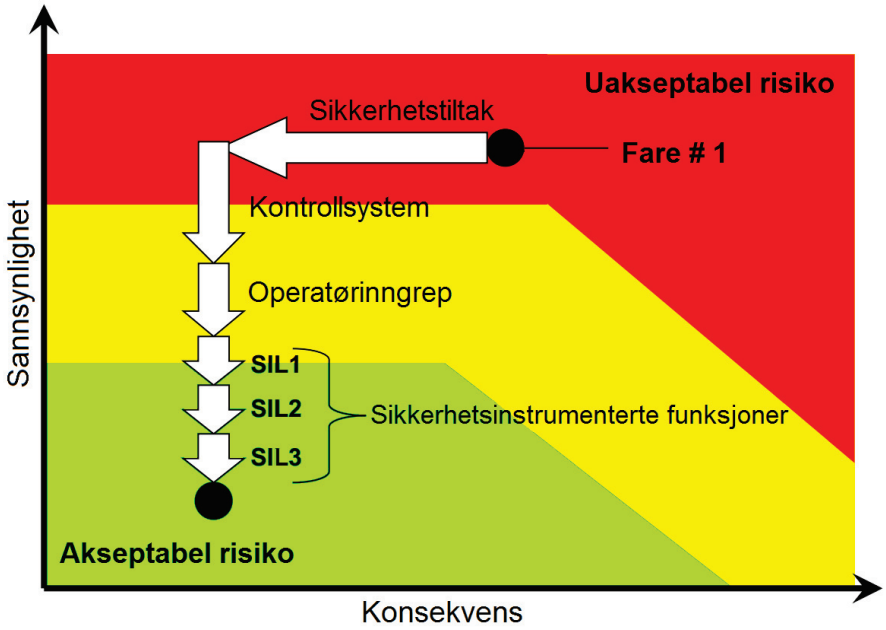
Sikkerhet og robusthet er begge uttrykk med sentrale roller innen industrielle reguleringssystemer, hvor en feiltilstand grunnet for dårlig sikkerhet eller robusthet kan føre til skader på mennesker, maskiner eller miljø.

De aller fleste har et forhold til begrepene sikkerhet og robusthet, dette kapittelet har som formål å formalisere disse begrepene.

Da man på norsk har samlet de to engelske begrepene «security» og «safety» i et ord, kan det lett oppstå feiltolkninger dersom det ikke kommer klart frem fra konteksten hvilket av disse to det er snakk om. «Sikkerhet» vil i denne oppgaven generelt betegne både «safety» og «security» med unntak av steder med eventuell tvetydighet, hvor «informasjonssikkerhet» vil bli benyttet for å betegne «security».

4.1 Sikkerhet

Sikkerhet (*en:safety*) blir av det amerikanske forsvarsdepartementet beskrevet som følger [15]: «Uavhengighet fra de tilstander som kan føre til død, skade, yrkesmessig sykdom, skade på utstyr og eiendom eller miljø.» Denne beskrivelsen indikerer at sikkerhet er noe det er viktig for en organisasjon å opprettholde, men dette er ikke kostnadsfritt. Organisasjonen er nødt til å finne en balanse mellom høy sikkerhet, som ofte vil resultere i kostbare feilnedstenginger, og en drift hvor en senker sikkerhetskravene litt for å unngå unødvendige kostbare nedstenginger [31]. For at organisasjoner skal kunne finne denne balansegangen er det viktig at det blir gjort en vurdering over hvilke risikoer (konsekvens · frekvens) organisasjonen finner akseptable. Denne vurderingen vil også danne grunnlag for hvor mye organisasjonen er villig til å satse på opprettholdelse av god informasjonssikkerhet. Forskjellige tiltak som kan gjennomføres av organisasjonen for å senke risikoen ved en operasjon fra uakseptabelt til akseptabelt nivå kan sees i figur 4.1.



Figur 4.1: *Illustrasjon over hvordan risikoen reduseres fra regionen for uakseptabelt høy risiko og ned til regionen med en risiko vi kan akseptere. Reduksjonen av risiko gjøres ved hjelp av sikkerhetstiltak, kontrollsystemer, operatørens inngrep samt forskjellige sikkerhetsfunksjoner.*

4.2 Informasjonssikkerhet

Informasjonssikkerhet (*en: security*) er i [44] definert som følger: «Beskyttelse av informasjon, kritiske komponenter og systemer som bruker, lagrer eller videreformidler informasjon».

Vet at systemer som tidligere har vært fysisk lukket fra hverandre blir sammenkoblet, og går over til å benytte åpne kommunikasjonsprotokoller slik som beskrevet i kapittel 3.4, har både muligheten og interessen for å kompromittere systemene økt. Det er derfor viktig at det blir igangsatt tilstrekkelige tiltak for å sikre informasjonen, da et brudd på informasjonssikkerheten i verste fall kan føre til svekkelse av den fysiske sikkerheten i systemene. Det vil blant annet i kapittel 5.2 bli presentert en rekke forskjellige angrep som kan utføres mot systemene, og som potensielt kan gjøre skade på systemene.

4.3 Robusthet

Som nevnt i kapittel 3.2 så har systemene benyttet på oljeplattformer blitt mer sammenvevde og komplekse, og trenden er at dette mest sannsynlig vil fortsette.

Robusthet (*en: resilience*) blir i [17] definert som følger: «*Et systems iboende egenskap til å endre sin virkemåte avhengig av ytre endringer eller forstyrrelser, slik at systemet kan fortsette sine operasjoner selv etter store uhell eller kontinuerlig høy last på systemet*». Robusthet vil derfor kunne være et designmål, og kunne benyttes til å forbedre sikkerheten ved komplekse problemer. Problemene som kan oppstå i slike prosesser er mange, og systemene som benyttes så komplekse at fullstendige risikoanalyser av alle scenarioer enten er svært krevende eller umulig.

Robuste systemer er designet for ikke å la seg affektere av feil og unormal drift som det ikke har blitt tatt høyde for under design, og gjør derfor systemene motstandsdyktige mot et større sett av angrep. Et scenario hvor robuste enheter kan utgjøre en forskjell er dersom et industrielt nettverk blir utsatt for en nettstorm (se kapittel 5.2.2). Dersom systemet ikke er designet for å være robust kan nettstormen gjøre at komponenten blir så opptatt av innkommende trafikk at den ikke får gjort oppgaven sin. En robust designet utgave av komponenten vil derimot i en slik situasjon kunne håndtere problemet ved å for eksempel stenge ned koblingen og dermed forstyrrelsene fra nettverket og fortsette med sine oppgaver inntil lasten på nettverket igjen er normal.

Det finnes mange metoder for å øke robustheten til et system, og det er stor interesse innen forsking og fra SAS-leverandører for å utvikle mer robuste systemer [40]. Det vil i kapittel 7.4 bli gitt en kort innføring i noen metoder organisasjonen kan bruke for å forbedre sine systemers robusthet.

5 Trusler og sårbarheter

Hvis du kjenner fienden og deg selv, skal du ikke frykte hundre slag.
— Kunsten å krige, Sun Tzu

For å kunne forsvare seg selv på best mulig måte, er det essensielt å kjenne sine fiender. Det er derfor viktig å sette seg inn i hvilke aktører som vil være tjent med et angrep mot sine industrielle datasystemer, og ikke minst hvilke metoder og taktikker disse aktørene kan benytte seg av for å gjennomføre angrepet.

I dette kapittelet vil det bli gitt en gjennomgang av aktørene bak angrep mot industrielle nettverk, noen av metodene disse kan benytte for å forårsake skade og utnytte noen sårbarheter som finnes i mange av disse systemene.

5.1 Aktører

Aktørene som utgjør en trussel mot industrielle datasystemer er mange, og de kan enten være eksterne eller interne. Skadene disse aktørene skaper kan enten være tilsiktede og utilsiktede. Det vil nå bli gitt en oversikt over forskjellige typer angripere.

Ekstern aktør som handler tilsiktet kan være alt fra profesjonelle hackere til ungdommer på gutterommet som har funnet ondsinnet programkode på Internett. En profesjonell hacker vil gjerne ha kunnskap innenfor felt som nettverksprotokoller, operativsystemer og programmeringsspråk, og kan benytte seg av denne kunnskapen til å skreddersy programvare for å bryte seg inn i mange forskjellige systemer [44]. Utviklingen i de senere årene har gjort det svært enkelt å dele ondsinnet programvare på Internett, og en av trendene har vært at profesjonelle hackere deler kildekode mot penger eller for respekt innad i miljøet. Denne utviklingen har gjort det mulig for hackere med mindre kunnskap å starte angrep, noe som åpner for angrep fra mange flere aktører.

Eksterne og utilsiktede aktører kan være privatpersoner eller bedrifter med infiserte datamaskiner som blir brukt til å utføre angrep (for eksempel nettstorm, se kapittel 5.2.2) mot andre uten deres samtykke eller kunnskap.

Med **interne og tilsiktede aktører** menes gjerne personer med innsideinformasjon om datasystemene og systemenes sikkerhetsmekanismer. Disse aktørene har gjerne fysisk tilgang til systemene, noe som gjør at vanlige barrierer som er implementert for å stoppe slike angrep kan forbigås. Det er også nærliggende å anta at

de interne aktørene har brukerkontoer eller tilsvarende aksess til systemer koblet mot nettverket, noe som kan gi direkte adgang til kritiske systemer uten å vekke mistanke.

Et eksempel på en **intern, men utilsiktet trusselaktør** kan være en ansatt som kobler til en minnepinne som, uten at vedkommende vet det, er infisert med virus inn i et delsystemet, noe som kan føre til en spredning av ondsinnet programkode på bedriftens nettverk.

Det er i [38] gitt en gjennomgang av de forskjellige trusselaktørene mot industrielle kontrollnettverk, og disse aktørene er gjengitt i tabell 5.1 og tabell 5.2. Som disse tabellene viser, så er de potensielle aktørene mot industrielle kontrollnettverk mange, mengden kunnskap noen av de har stor og tilgang til finansielle midler og politiske ressurser enkelte av aktørene har er svært store.

Trusselaktør	Beskrivelse
Angripere	Denne gruppen omfatter de som finner sikkerhetshull og bryter seg inn i systemer for å kunne høste respekt innad i miljøet. Det har tidligere krevd mye kunnskap å kunne sette i stand et slikt angrep, men oversikter over sikkerhetshull og verktøy for å utnytte disse hullene de siste årene blitt lett tilgjengelig på Internett. Dette har senket terskelen for å iverksette ondsinnede angrep, og et eksempel på en utbredt skadevare programmert av svært unge angripere er Zotob, som er nærmere beskrevet i kapittel 6.1.
Botnettoperatører	Botnett baser seg på at PC-er blir infisert med programvare som muliggjør fjernstyring av de infiserte maskinene. Datakraften til disse infiserte maskinene kan så bli videresolgt til en tredjepart og utnyttet til for eksempel svindel (phising), nettstormangrep (se kapittel 5.2.2) eller utsending av søppelpost.
Kriminelle grupper	Kriminelle grupper velger sine mål etter mulighet for økonomisk gevinst, og benytter metoder som søppelpost, svindleprogramvare og spionprogramvare for å utføre identitetstyveri eller annen form for svindel.
Utenlandske etterretningsgrupper	Etterretningsgrupper med finansiell støtte fra for eksempel forsvarsdepartementet. Gruppene kan benyttes til å samle etterretningsinformasjon, planlegge og utføre cyberangrep mot strategiske mål. Mange land har i dag et velutviklet cyberforsvar. Land som USA, Israel, Kina og etterhvert Iran står frem som store aktører innen dette feltet. Slagkraften kan blant annet sees i Stuxnet-angrepet, som er behandlet nærmere i kapittel 6.2.

Tabell 5.1: *Oversikt over trusselaktører innenfor cyberangrep [1/2], hentet fra [38].*

Trusselaktør	Beskrivelse
Innsideaktører	Personer med kjennskap eller tilgang til datasystemene kan benytte dette til sin fordel ved angrep av systemet. Dersom man har fysisk tilgang til systemene vil vanlige barrierer mot angrep kunne forbigås. Innsideaktører innebærer også tredjeparts kontraktører som bringer ondsinnet programvare inn i datasystem.
Phisere	Phisere får tak i identitetsinformasjon ved hjelp av en rekke forskjellige svindlemetoder, og benytter denne informasjonen til økonomisk gevinst enten direkte eller ved å videreselge informasjonen.
Spammere	Spammere tilbyr distribuering av søppelepost, gjerne med falsk informasjon for salg av produkter, distribuering av ondsinnet programvare eller for å angripe spesifikke mål. Spammere kan gjerne benytte seg av botnett for effektivt å spre spam.
Skadevareprogrammerere	Produserer skadevare for å skade et bestemt eller så mange mål som mulig ved et gitt sikkerhetshull. Flere eksempler på slik skadevare vil bli gitt i kapittel 6.
Terrorister	Terrorister kan ha som mål å ødelegge eller sette kritisk infrastruktur ut av spill for å skape så store skader som mulig, enten på mennesker, miljø eller materiell. Terrorister kan også benytte phising for å finansiere sin virksomhet.
Industrispioner	Industrispionasje kan benyttes mot konkurrenter for å få innblikk i deres industrielle hemmeligheter.

Tabell 5.2: *Oversikt over trusselaktører innenfor cyberangrep [2/2], hentet fra [38].*

5.2 Noen typer angrep mot datasystemer

Planlagte angrep blir i [6] delt inn i to kategorier: aktive og passive angrep. Aktive angrep er angrep hvor angriperen gjør en overlagt handling for å komme seg inn i systemet. Blant disse angrepene finner vi blant annet nettstorm (kapittel 5.2.2) og tradisjonelle innbrudd i datasystemer. Passive angrep er angrep hvor målet er å samle sammen så mye informasjon som mulig om et datanettverk og komponenter tilkoblet dette. Disse angrepene benyttes gjerne i planleggingsfasen før et større angrep, og er vanskeligere å oppdage enn aktive angrep da de i stor grad bare overvåke trafikken til og fra nettverket. Passive angrep som sniffing av passord, trafikk og informasjon vil bli behandlet i kapittel 5.2.3.

I tillegg til planlagte angrep kan eventuelle feiltilstander i systemene føre til ikke-planlagte trusler dersom feilen berører en sårbarhet i systemet (kapittel 5.3). Disse ikke-planlagte truslene vil ikke dekkes i denne masteroppgaven.

Dette delkapittelet vil se nærmere på noen av de vanligste typene planlagte angrep mot industrielle systemer og forklare eller referere til hvordan organisasjonen kan beskytte seg mot disse.

5.2.1 Ondsinnet kode

Ondsinnet programvare et et begrep alle som jobber innen IT bør ha kjennskap til og kunnskap om. Ondsinnet programvare er en samlebetegnelse på programvare skrevet for å skade eller ødelegge systemer, og inkluderer blant annet trojanske hester (bakdører), virus, skript og ormer.

Utviklingen av ondsinnet programvare er svært dynamisk og nye hull oppdages og utnyttes hele tiden. Siden programvare øker i kompleksitet, øker også mulighetene for potensielle sikkerhetshull dersom programmererne ikke har skrevet kode av god kvalitet. Som nevnt i kapittel 5.1 har terskelen for å produsere ondsinnet kode blitt lavere enn tidligere, både på grunn av økt generell datakompetanse og på grunn av kunnskapsspredningen og kodedelingen som foregår innad i miljøet. Dette har ført til en oppblomstring av slik skadevare.

Ondsinnet programvare er som nevnt en samlebetegnelse, og to av de sentrale typene er:

- **Virus:** Utnytter sikkerhetshull for å infisere og spre seg videre til andre maskiner ved hjelp av en vert, for eksempel en USB-minnepinne med viruset. Formålet med virus kan være å skape ødeleggelser, restarte utstyr, slette data og mye mer. Dersom viruset når sentrale maskiner i nettverket, kan

dette skape store problemer for bedriften og store mengder data kan gå tapt.

- **Ormer:** Ormer er mye av det samme som virus, men til forskjell fra virus er ormene selvspredende. Ormene kan spre seg i svært stor fart, og farten av spredningen vil øke eksponentielt med antall infiserte maskiner.

  Trojanske hester er en form for orm som installerer seg på PC-en for å så gjemme seg i systemet mens den åpner en bakdør inn i systemet som er kjent for angriperen. Denne bakdøren kan da benyttes for å komme seg inn på maskiner ved senere angrep.

  Moderne ormer kan være svært komplekse og de mest avanserte kalles polymorfiske, noe som vil si at ormene endrer angrepsstrategi og hvordan de gjemmer seg basert på systemene de møter.

Det finnes flere måter å delvis beskytte seg mot ondsinnet programvare, den mest kjente og brukte er antivirusprogrammer. Problemet med disse programmene er at de krever oppdateringer av virusdatabasen med jevne mellomrom for å kunne gi optimal beskyttelse. En annen svakhet med antivirusprogrammer er at de er reaktive, noe som vil si at leverandøren av programvaren først er nødt til å oppdage nye virus, analysere viruset, lage sjekker, tette igjen for viruset og til slutt distribuere denne oppdateringen til sluttbrukerne.

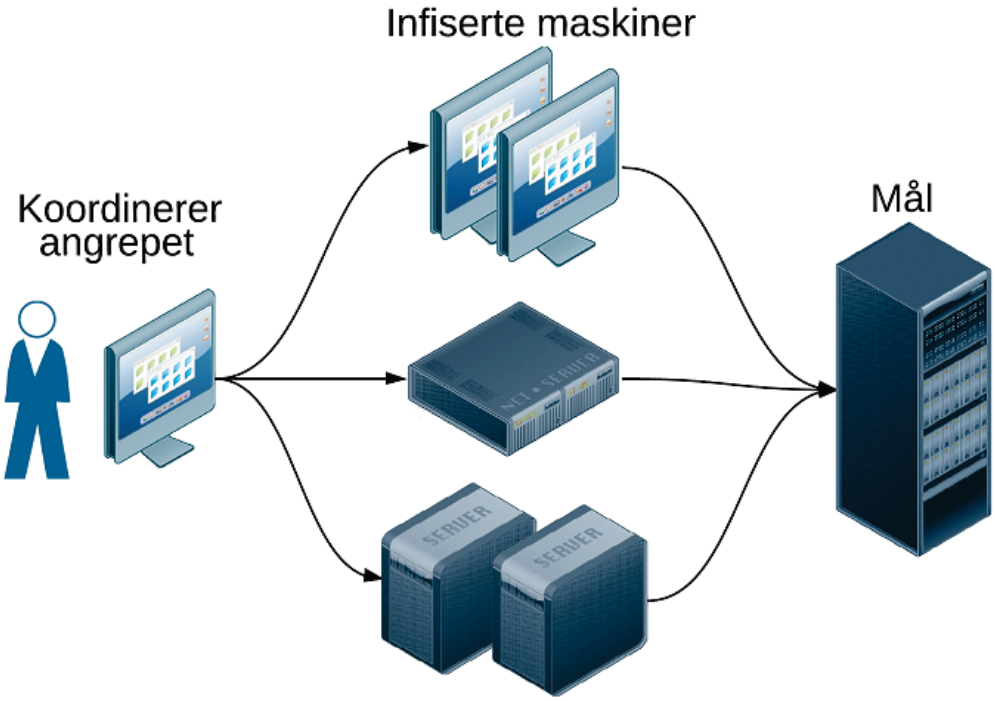
5.2.2 Nettstorm (DoS)

Nettstorm (*en: DoS, Denial of Service*) vil si at en angriper sender så mye data som mulig i form av for eksempel forespørsler om TCP-tilkoblinger, til en bestemt mottaker. Dersom mengden data som sendes er stor nok, vil mottakerens nettverksutstyr bli så opptatt med å betjene de falske forespørslene at de reelle forespørslene ikke blir betjent (tjenestenekt) [44].

Det finnes i hovedsak to forskjellige typer nettstorm:

- **Ordinær nettstorm**: Målet angripes kun fra én maskin. For at dette skal fungere er en avhengig av å ha en kraftig Internett-tilkobling i forhold til mottakeren slik at det kan sendes en mengde trafikk som er større enn det mottakeren kan takle.
- **Distribuert nettstorm**: Målet angripes fra flere forskjellige maskiner fra forskjellige Internett-tilkoblinger som koordineres av en sentral maskin. Ved distribuerte vil den aggregerte trafikken lettere forårsake tjenestenekt hos mottakeren. De forskjellige maskinene kan for eksempel være PC-er som er

infisert og benyttes til dette formålet ved hjelp av fjernstyring uten at eieren vet om det (botnet). Den distribuerte datakraften kan også komme fra en koordinert gruppe mennesker som har programvare installert designet for dette formålet. Eksempler på slik lett tilgjengelig programvare er *«Low Orbit Ion Cannon»*[2] og videreutviklingen av denne, *«High Orbit Ion Cannon»* som blant annet har blitt benyttet aktivt av gruppen «Anonymous»[3] for å angripe webservere. Et oppsett for distribuert nettstorm er vist i figur 5.1.



Figur 5.1: *Figuren viser et typisk oppsett for distribuert nettstorm. Til venstre er koordinatoren for angrepet, i midten de infiserte maskinene som benyttes for å generere nødvendig trafikkmengde og til høyre er målet for angrepet.*

Nettstorm-angrep rammer de ytre sikkerhetsbarrierene som skal beskytte nettverket mot angrep fra det åpne nettet. Selv om angrepet gjerne stopper i de ytterste lagene, vil angrepene kunne skape problemer for tredjeparts servicetilbydere som kan være avhengige av en tilkobling mot organisasjonen for å kunne utføre sine

[2] `http://sourceforge.net/projects/loic/`

[3] Anonymous er en løs sammenkoblet gruppe mennesker som har bedrevet «hackeraktivisme» siden tidlig på 2000-tallet. «Hackeraktivisme» går ut på å benytte for eksempel nettstorm for å fremme misnøye mot regjeringer, religiøse grupper eller organisasjoner. Gruppen har stått bak flere store angrep, og har bred støtte i enkelte kretser. Den løse sammensetning og strukturen i gruppen gjør det vanskelig å avdekke nøkkelpersonene bak organisasjonen.

tjenester. Dette gjør at det er svært viktig at en ved bruk av IO har robuste og redundante tilkoblingsmuligheter dersom systemene skulle bli utsatt for nettstorm.

Nettstorm kan også oppstå internt i prosessnettverket, for eksempel på grunn av feilmodus i en enhet eller på grunn av at én infisert maskin blir satt til å sende falsk trafikk på nettet.

Det finnes noen få måter å beskytte seg mot nettstorm, men denne beskyttelsen er gjerne en dyr ekstra-service fra tjenestetilbydere [3]. Alternativet er at organisasjonen har gode alternative kommunikasjonskanaler til de viktigste tilkoblingene, samt at det blir testet for intern robusthet mot slike angrep. Testverktøy som kan benyttes er beskrevet nærmere i kapittel 7.11.1.

5.2.3 Snooping

Snooping er en type passivt angrep som går ut på å kartlegge nettverket, nettverkets struktur og generell datatrafikk ved å for eksempel lytte på kommunikasjonen. Snooping kan benyttes av angripere før et angrep for å avdekke nettverkets eventuelle svakheter.

Siden snooping i all hovedsak går ut på å lytte til trafikk, så er det vanskelig å oppdage, og det finnes lett tilgjengelig verktøy for å utføre forskjellige typer snooping, noe som resulterer i at det kreves både lite tid og kunnskap for å gjøre utføre snoopingen.

Selv om snooping gjerne blir brukt av angripere, er det også et svært effektivt verktøy for nettverksadministratorer for å kartlegge eventuelle svakheter i nettverket som åpne nettverksporter eller sikkerhetshull. Dette lar nettverksadministratoren se hvordan nettverket oppfattes fra en angripers ståsted, og kan være svært verdifullt for å avdekke svakheter i sitt eget nett.

5.2.4 Phishing

Begrepet phishing kommer fra en omskriving av det engelske ordet «fishing», og handler om å fiske informasjon. Dette kan for eksempel brukes for å lure til seg forskjellig informasjon som brukernavn og passord. Metodene for å få offeret til å sende fra seg informasjonen er mange og eksempler kan være e-poster med falske hurtiglinker eller e-poster som utgir seg for å være fra en legitim sender.

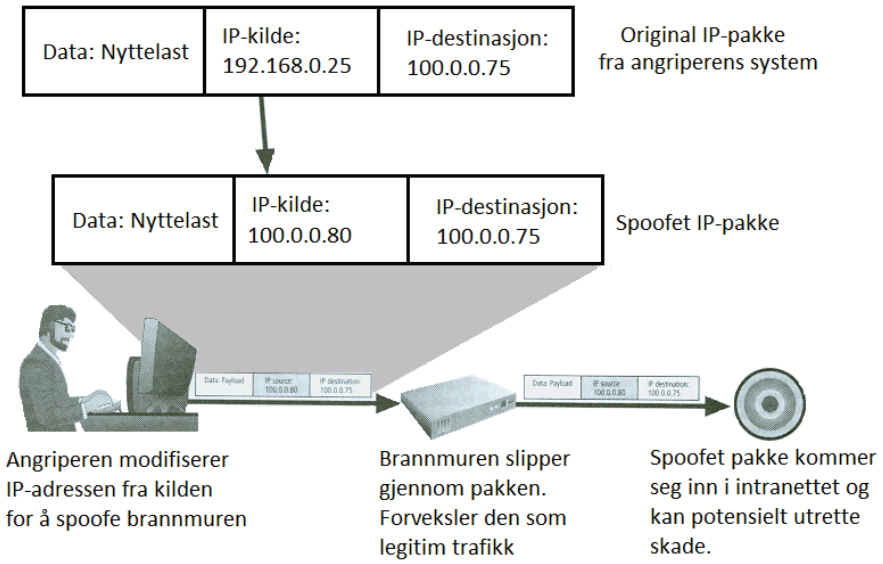
Et eksempel på hvordan dette kan gjøres, kan være en e-post som tilsynelatende er sendt fra en overordnet i en stor organisasjon. E-posten inneholder en lenke

til bedriftens nye e-læringsprogram, hvor man skal logge inn med brukernavnet og passordet man bruker i bedriften. E-læringsportalen er egentlig en falsk side som lagrer passord og brukernavn, og sender brukeren til en side som viser en tilsynelatende legitim feilmelding. Det er da stor sannsynlighet for at brukeren ikke vil bite seg merke i dette, og at en tredjepart nå vil ha et gyldig brukernavn og passord til systemet.

Det beste forsvaret mot phishing er god trening av de ansatte i IT-sikkerhet slik at brukerne blir klar over metoder som dette samt å jevnlig å få de ansatte bytte passord [38].

5.2.5 Spoofing

Spoofing er å forfalske datapakker slik at de tilsynelatende ser ut til å komme fra noen andre. Ved å blant annet endre, slette, omdirigere eller forfalske datapakker gir dette angriperen muligheter til å manipulere systemer på måter han/hun ellers ikke ville hatt tilgang til å gjøre [44]. En illustrasjon fra [44] på hvordan spoofing kan utføres sees i figur 5.2.

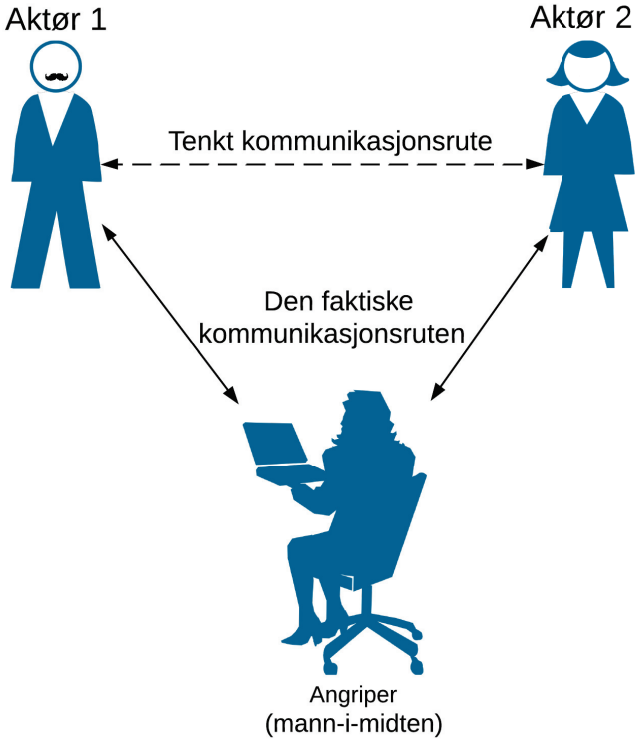


Figur 5.2: *Figuren viser hvordan en angriper kan forfalske en pakke for å få den til å komme seg gjennom bedriftens brannmur [44].*

Det er svært lett å spoofe trafikk på konvensjonelle TCP/IP-nettverk, hvor verktøyene for å monitorere og endre trafikk er lett tilgjengelig og enkle å bruke.

Heldigvis har en del nyere og mer avanserte rutere og brannmurer funksjonalitet som hindrer spoofing, ved å detektere om pakkens innhold har blitt endret ved bruk av sjekksummer eller liknende mekanismer [44].

En mer avansert form for spoofing er såkalte «mann-i-midten»-angrep, hvor angriperen plaserer seg mellom sender og mottaker som vist i figur 5.3. Dersom angriperen klarer å posisjonere, samt sikre seg at all trafikk mellom sender og mottaker går gjennom seg selv, så vil ikke sikkerhetsmekanismer som for eksempel kryptering hjelpe. Den eneste beskyttelsen mot slike angrep er at både mottaker og sender har svært strenge sikkerhetsrutiner som sørger for at angriperen ikke vil kunne komme i en slik posisjon at denne typen angrep blir mulig å gjennomføre [6].



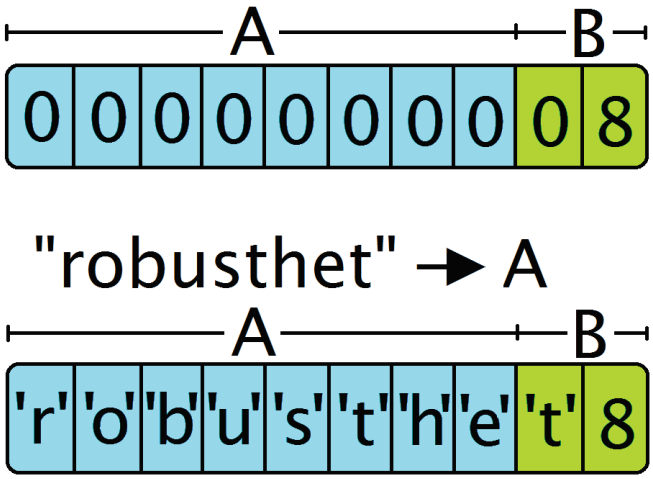
Figur 5.3: *Figuren illustrerer et mann-i-midten angrep. Aktør 1 og aktør 2 har opprettet kommunikasjon, og tror at denne går direkte mellom hverandre. I virkeligheten har en angriper plassert seg mellom aktørene, og kan kontrollere all trafikk mellom dem.*

5.2.6 Inputvalidering

Bufferoverflyt (*en: buffer overflow*) og kommandoinjeksjon er begge mulige utfall av dårlig eller ikke-eksisterende inputvalidering i systemer. Bufferoverflyt er et resultatet av programmeringsfeil, hvor det kan oppstå situasjoner som programmereren ikke har tenkt over eller behandlet i programkoden.

Et eksempel på en slik feil kan være et navnfelt hvor noen kan finne på å skrive inn et mange hundre bokstaver langt navn. Dersom programmereren ikke har tatt hensyn til dette, kan det overflødige av input i forhold til allokert minneplass overskrive nærliggende minne. Et eksempel på dette er vist i figur 5.4, hvor man prøver å skrive inn strengen «robusthet» i variabel *A*. Siden «robusthet» er for langt, blir den siste bokstaven, 't', feilaktig lagret i nabominnet som egentlig brukes av variabel *B*. Angripere kan benytte dette til å manipulere systemer ved å få tilgang til variabler de ellers ikke ville fått og med dette skape feiltilstander eller i verste fall laste inn kjørbar programkode.

Bufferoverflyt kan også oppstå ved at det sendes ufullstendige eller manipulerte datapakker inn til systemet, og at disse blir tolket feil av kontrollsystemets programvare siden formen på pakkene ikke er som forventet.



Figur 5.4: *Figuren viser hvordan strengen «robusthet» blir skrevet til variabel A og overskriver deler av minneområdet til variabel B da antall bokstaver i «robusthet» overskrider tiltenkt variabelstørrelse.*

Kommandoutførelse

Kommandosutførelse kan oppstå dersom angriperen får mulighet til å plante kjørbar programkode i systemets minne, og ved hjelp av for eksempel et semikolon eller en parentes klarer å kjørte koden. Denne type sikkerhetshull gjør at en angriper kan mate systemet med en lang kommando før det tilslutt blir sendt inn et semikolon eller parentes og koden utføres. Denne typen angrep har blant annet vært et problem i databaser, hvor SQL-spørringer i flere tilfeller har latt seg manipulere slik at man kan sende inn sine egne spørringer mot databasen. Da databaser blir benyttet blant annet for å realisere DMZ-er (se kapittel 7.6 for mer), er det svært viktig å være bevisst på dette slik at programvaren bygges opp på en slik måte at angriperen ikke får mulighet til dette.

Metoder for å beskytte seg mot overflyt

For å løse problemet med bufferoverflyt eller kommandoeksekvering, bør all input til programmet sjekkes nøye med tanke på størrelser, lengder og at dette ikke overskrider variablenes faste allokerte minnestørrelser. Det kan også med fordel benyttes «white-lists» i programvaren, som er lister med lovlige tegn som all inputen blir sjekket opp mot. Dersom inputen ikke er i samsvar med innholdet i denne listen, så vil inputen ikke bli matet videre inn i systemet.

Rutiner for å sikre seg mot bufferoverflyt og kommandoeksekvering vil være gode tiltak for å forbedre robustheten til systemet. Systemet vil ikke lenger oppføre seg ikke-deterministisk ved ukjent input, men ha et kjent sett med akseptert input som gir en deterministisk respons i systemet.

5.3 Sårbarheter

Dette delkapitlet vil sette fokus på noen av sårbarhetene man har i dagens systemer og nettverk og vil bygges rundt en liste over sårbarheter gitt i [16].

Det blir i [16] av blant annet NERC[4] og GE[5] listet opp ti sårbarheter som anses som de viktigste.

1. **Dårlige retningslinjer og kultur for sikring av prosess- og IKT-systemer.** Mange organisasjoner har dårlige eller ikke-eksisterende retningslinjer for

[4] North American Electric Reliability Corporation
[5] General Electric

sikker bruk av prosess- og IKT-systemene, og dette kan føre til sårbarheter ved at systemer blir brukt på feil måte.

2. **Dårlig utforming av system, manglende beskyttelse-i-dybden.**
   Dersom systemene ikke blir designet etter de standarder og retningslinjer som finnes innenfor fagområdet, vil man kunne få sårbarheter som resultat av for eksempel avhengigheter eller svak beskyttelse-i-dybden (kapittel 7.3).

3. **Manglende «patching» og vedlikehold.**
   Dersom systemene ikke blir oppgradert med siste tilgjengelige oppgraderinger, vil det kunne oppstå sårbarheter som kan bli utnyttet enten tilsiktet eller utilsiktet. Tiltak for skape å bedre rutiner for oppdateringer vil bli presentert i kapittel 9.2.

4. **Usikret bruk av trådløs datakommunikasjon.**
   Trådløs kommunikasjon blir stadig mer utbredt og kan føre til sårbarheter dersom implementasjonen av dette ikke blir gjort på en sikker måte.

5. **Dårlig kvalitet på kommunikasjonsprotokoller.**
   Tidligere da systemene var beskyttet fra omverden i lukkede nettverk var ikke sikkerhet i kommunikasjonsprotokollene essensielt, men når mange av nettverkene åpnes opp mot omverden og innkapsler de usikre protokollene i TCP/IP-pakker kan det oppstå sårbarheter dersom disse pakkene sendes over delte nettverk.

6. **Manglende bruk av rapportering og analyseverktøy.**
   Dersom brukerne av systemet ikke rapporterer feil, eller hendelser eller om disse ikke blir analysert, kan kjente feil forbli urettet og senere føre til farlige situasjoner.

7. **Svakheter i bruk av passord og autentisering.**
   Systemer som benytter seg av standardinnstillinger for passord eller har dårlige rutiner for passord vil kunne være sårbare for utnyttelse. Noen tiltak for å forbedre dette vil bli tatt opp i kapittel 9.9.

8. **Dårlig testing eller verifikasjon av systemer i utvikling og innkjøp.**
   Dersom systemene ikke er tilstrekkelig testet på komponentnivå og helhetlig nivå kan det finnes sårbarheter i systemene. Noen av rutinene som kan benyttes for testing av systemene vil bli presentert i kapittel 7.11.

9. **Manglende autentisering av kommandoer/trafikk.**
   Dersom det ikke finnes noen måte å autentisere om trafikken som transporteres over nettverket ikke er ondsinnet vil dette kunne føre til sårbarheter i

systemene. Tiltak som kan forbedre dette vil bli presentert i kapittel 9.8.

10. **Svak robusthet i systemene.**
    Dersom systemene ikke er designet med iboende robusthet (kapittel 4.3) vil systemet kunne være sårbare for uforutsette hendelser som det ikke er tatt høyde for sikkerhetsmessig. Metoder for å øke robustheten i systemene vil bli presentert i kapittel 7.4.

6 Ulykker og hendelser

Jeg vil si at dette er den mest sofistikerte trusselen vi noen gang har sett.
— Orla Cox, Senioranalytiker hos Symantec om Win32.Flamer

Selv om industrielle kontrollsystemer er designet for å opprettholde sikkerheten selv under uforutsette hendelser, så viser historien av selv de mest veldesignede systemer kan feile.

I dette kapitlet vil et lite utvalg av uheldige hendelser knyttet til kontrollsystemer bli presentert for å belyse hvor sårbare slike systemer kan være.

6.1 Zotob

Skadevaren Zotob dukket opp for første gang i 2005, og var rettet mot et sikkerhetshull; Microsoft Windows 2000, et operativsystem mye brukt i kontrollsystemer. Skadevaren ble utviklet av to gutter tidlig i 20-årene [10], og sendte en signalrakett til sikkerhetsmiljøet om hvor lett det hadde blitt å utvikle slik skadevare dersom sikkerhetshullet utnyttet var stort nok. Sikkerhetshullet som ble utnyttet i Zotob var en feil i Windows sin automatiske driverinstallasjon («plug-and-play»), noe som gjorde det mulig å eksternt ta seg inn på infiserte PC-er for å kontrollere de [32].

Zotob spredde seg raskt på Internett og nettverk, og selv om skadevaren ikke var skrevet spesielt for kontrollsystemer, ble også kontrollsystemer rammet da flere leverandører benyttet Windows som underliggende operativsystem i sine enheter. Blant annet stoppet 13 av DaimlerChrystlers bilfabrikker i rundt en time på grunn av Zotob, noe som førte til at rundt 50 000 arbeidere ble satt på vent eller sendt hjem mens utstyr måtte restartes [32].

6.2 Stuxnet

Stuxnet ble for første gang observert i juni 2009 [9], og antas å være resultatet av et cyberkrigsammarbeid mellom USA og Israel med det mål å vanskeliggjøre utviklingen av Irans atomprogram [33].

Stuxnet var det første viruset observert som spesielt rettet seg mot industrielle kontrollnettverk. Allerede i slutten av september hadde Stuxnet spredd seg fra

hovedmålet, atomkraftverket Natanz i Iran, og til rundt 100 000 PC-er rundt om i verden. Ved å utnytte hele fire 0-dags sikkerhetshull[6] ble Stuxnet av sikkerhetsprogramvareleverandøren Symantec beskrevet som en av verdens mest komplekse skadevarer da den opprinnelig ble analysert [9].

Grunnet oppbyggingen til Stuxnet blir det antatt at skadevaren ble introdusert til kontrollsystemet gjennom tilkoblingen av en minnepinne, noe som understreker viktigheten med å ha kontroll på tilkobling av fysisk ikke-godkjent utstyr i kontrollnettet, noe som diskuteres nærmere i kapittel 7.10. Etter at minnepinnen er koblet i en PC med Windows benytter Stuxnet en gyldig driversignering fra Microsoft for å tilegne seg nødvendige rettigheter til å laste seg selv inn i tilsynelatende uskyldige prosesser som allerede kjører på PC-en. Installasjonsprosedyren av Stuxnet søker også etter kjente prosesser fra forskjellige antivirusprogramvarer og endrer installasjonsrutinene etter hvilken antivirusprogrammvare som eventuelt kjører på systemet.

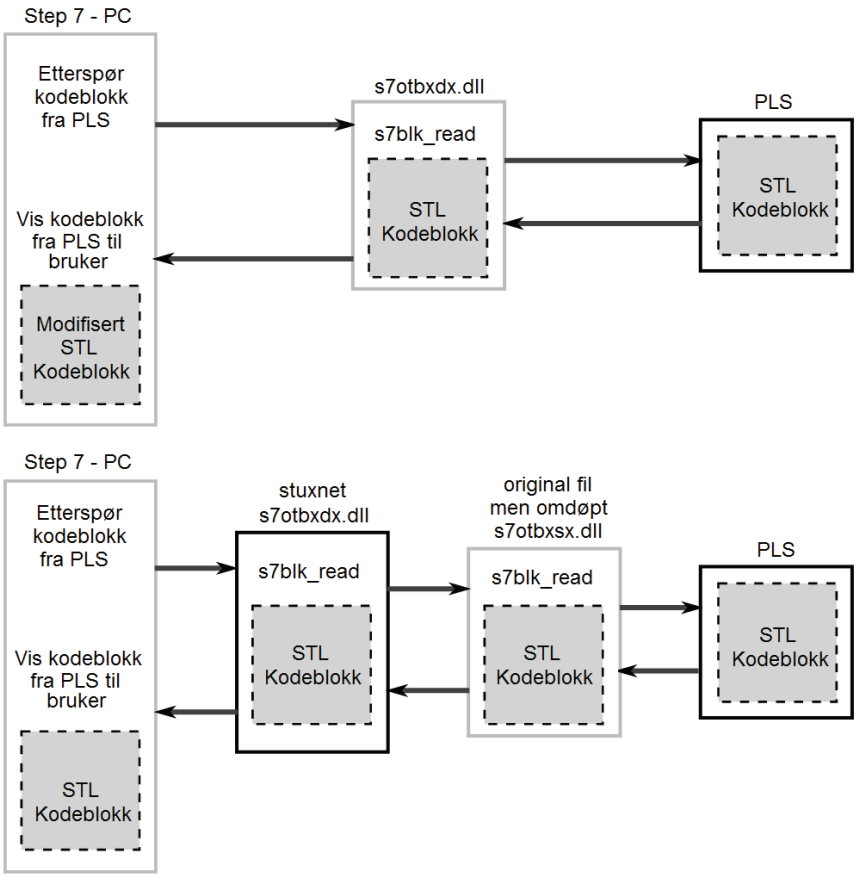
Etter den initielle infeksjonen spres Stuxnet videre på industrinettverket, enten gjennom delte mapper i Windows, gjennom å infisere delte Siemens S7-prosjektfiler eller gjennom Siemens WinCC SCADA-system [9]. Dersom Stuxnet når en PC tilkoblet en S7-PLS, så kan Stuxnet overvåke og endre kjørende program i PLS-en ubemerket [9], og skadeomfanget kan bli enormt dersom kritiske prosesser rammes. Hvordan Stuxnet påvirker den logiske flyten mellom PC-en med S7 og PLS-en ved å sette inn modifiserte funksjonsblokker kan sees i figur 6.1.

Det antas at skadene på Natanz var at opp mot 1000 sentrifuger benyttet i anrikningsprosessen midlertidig eller fullstendig ble satt ut av spill [1].

6.3 Duqu

Duqu er en etterkommer av Stuxnet, og et av de første registrerte tilfellene av denne skadevaren kom i september 2011. Duqu inneholder mange tilsvarende eller identiske metoder og funksjoner fra Stuxnet, og i [26] blir det antatt at omkring 50 % av kildekoden er identisk med Stuxnet. På grunn av den store likheten i disse to skadevarenes oppbygning, er det mange som mener at de som har laget Duqu enten er en del av eller har et sterkt bånd til utviklingsmiljøet bak Stuxnet.

[6] 0-dags sikkerhetshull er sikkerhetshull som ikke er kjent før første angrep. Disse hullene gir derfor hackerene et forsprang da sikkerhetsselskapene og programvareleverandører ikke før etter første angrep får mulighet til oppgradere programvaren



Figur 6.1: *Illustrasjon over hvordan Stuxnet logisk plasserer seg mellom kontrollsystemet på PC-ene, S7, og PLS-ene i prosessen. Øverste bilde viser original logisk flyt, mens den nederste viser den logiske flyten etter infeksjonen av Stuxnet. Bildet er hentet fra Wikipedia.*

Til forskjell fra Stuxnet, så er Duqu ikke rettet direkte mot kontrollnettverk og PLS-er, men mot å logge aktiviteter og tastetrykk på infiserte maskiner. Denne informasjonen vil da senere kunne benyttes av de bak angrepet for å kartlegge strukturen på nettverket, og for å sikre seg passord til forskjellige maskiner og programvarer.

Den initielle infeksjonen av Duqu kan for eksempel skje gjennom et modifisert Word-dokument sendt som et vedlegg og åpnet på en PC på innsiden av bedriftens sikkerhetsbarrierer. Etter dette sprer skadevaren seg til andre PC-er over nettverket ved bruk av metoder tilsvarende de benyttet av Stuxnet. I tillegg benyttes også informasjonen samlet gjennom monitorering av aktiviteter for å komme seg inn på flere PC-er.

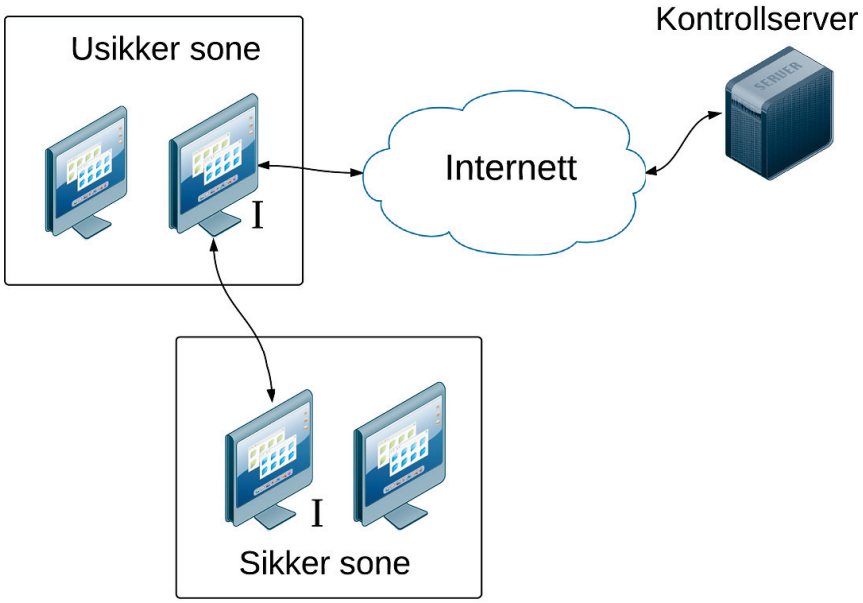
Duqu benytter i likhet med Stuxnet et «peer-to-peer» nett for å opprettholde kommunikasjon med en kontrollserver utenfor bedriften. Dette gjør at infiserte PC-er plassert i en trygg sone, uten direkte kommunikasjon med Internett, kan kommunisere gjennom en infisert PC plassert i en utrygg sone som har lov å kommunisere med Internett [39]. Denne formen for kommunikasjon gjør også at all trafikk fra bedriften og til kontrollserver vil samles fra én PC, og gjør det vanskeligere å detektere mistenkelig trafikk fra skadevaren. En enkel fremstilling av hvordan «peer-to-peer»-kommunikasjonen fungerer i Stuxnet og Duqu kan sees i figur 6.2.

6.4 Flame

Flame, Skywiper eller Win32.Flame ble for første gang analysert i 2012, men har sannsynligvis infisert maskiner ubemerket fra rundt 2010 (eller muligens tidligere da første kompilerte utgaver er fra 2007) [22]. Flame blir i [22] beskrevet som verdens mest avanserte ondsinnede programvare, og størrelsesmessig er Flame på 20 Mb, vesentlig større enn for eksempel Stuxnet som kun er på 500 Kb[7].

Funksjonelt er Flame, i likhet med Duqu, en svært avansert informasjonsstjeler som kan infisere maskiner samt propagere på nettverk. Mye av grunnen til Flames store størrelse sammenlignet med for eksempel Stuxnet og Duqu er at Flame består av svært mange forskjellige moduler som kan gjennomføre forskjellige typer informasjonsinnhenting. Sentralt blant disse modulene er: logging av tastaturtrykk, mulighet for å ta skjermbilder, lytte på eventuelle mikrofoner tilkoblet datamaskinen, lese og skrive til USB-enheter eller Bluetooth. Hvilke moduler som skal være aktive kan stilles inn ved hjelp av eksterne kommandosentraler, og disse kommandosentralene kan også etter behov brukes til å laste inn nye moduler

[7] http://www.reuters.com/article/2012/06/11/us-media-tech-summit-flame-idUSBRE85A0TN20120611

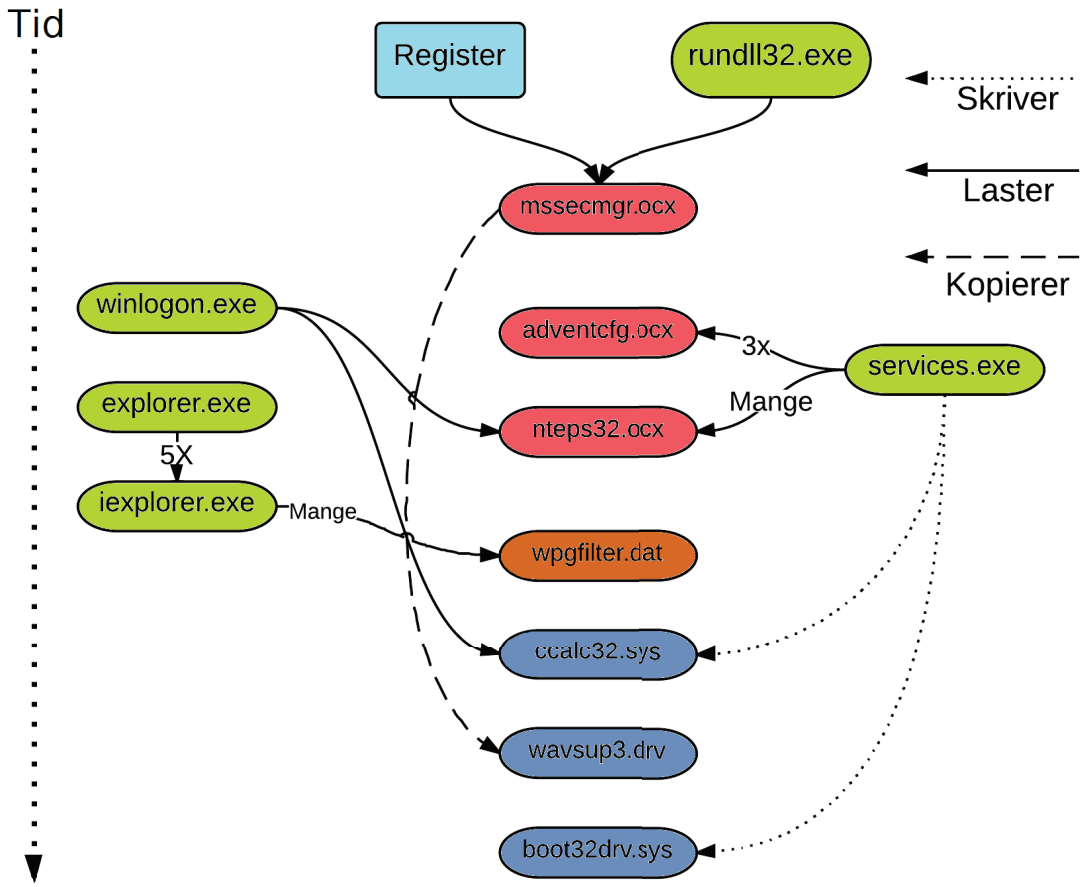


Figur 6.2: *«Peer-to-peer»-kommunikasjon i Stuxnet og Duqu. Infisert PC i trygg sone uten tilgang til Internett kommuniserer med kontrollserveren ved hjelp av en annen infisert PC plassert i utrygg sone. «I» markerer infisert PC.*

til programvaren [22]. Da kommandosentralene gjemmer seg bak rundt 50 domenenavn, alle med skiftene IP-adresser, gjør det svært vanskelig å spore de opp.

I arkitektur er Flame en svært kompleks skadevare med fem forskjellige typer krypteringer, mange forskjellige innlastbare moduler, samt 346 forskjellige taktikker som kan benyttes ved infeksjon av en PC avhengig av hvilke sikkerhetsmekanismer som finnes på maskinen [22]. Deler av den avanserte infeksjonsprossedyren kan sees i figur 6.3.

Det er antatt at selv med en gruppe høyt kvalifiserte hackere ville det tatt flere år å implementere noe så avansert som Flame, noe som indikerer at de som står bak skadevaren har stor finansiell støtte. At programvaren er så spisset mot informasjonsinnhenting/cyberspionasje, underbygger teorien om at Flame kan være utviklet som en del av et eller flere lands cyberforsvar.



Figur 6.3: *Illustrasjon av hvordan Flame laster seg selv inn i datamaskinen. Enten lastes mssecmgr.ocx seg inn i registeret, eller så kjøres rundll32.exe. Videre blir forskjellige deler av programvaren lastet inn ved hjelp av forskjellige legitime funksjonskall i Windows.*

7 Struktur og sikringsmekanismer

En kjede er aldri sterkere enn sitt svakeste ledd.
— Ukjent

For å få til en sikker implementasjon av integrerte operasjoner slik at systemene er beskyttet mot potensielle angrep, så er det viktig at organisasjonen implementer sikkerhetsmekanismer som sørger for at man kan skape tilstrekkelig sikkerhet i nettverkene.

Det vil i dette kapitlet først bli gitt en gjennomgang av de mest sentrale forskriftene som operatørene er nødt til å forholde seg til, samt hvilke eksisterende standarder og retningslinjer som kan benyttes i sikkerhetsarbeidet. Videre vil det bli gitt en gjennomgang av flere av de mest sentrale maskinvarebaserte, programvarebaserte, konseptuelle og strukturelle sikkerhetsmekanismene som kan benyttes for å sikre industrielle nettverk.

7.1 Forskrifter

For å skape og opprettholde et ønskelig HMS-nivå (**h**else, **m**iljø og **s**ikkerhet) har Petroliumstilsynet (Ptil) utformet flere forskrifter. Forskriftene gjelder i henhold til rammeforskriften § 2 for alle som driver med petroleumsvirksomhet på norsk sokkel (definert i rammeforskriftenes § 6).

For petroliumsvirksomhet til havs gjelder også Ptils innretningsforskrift, og denne forskriften er sentral for innføringen av sikker IO da innretningsforskriften §§ 32 - 34 definerer at B&G, NAS og PAS skal kunne utføre sine tiltenkte funksjoner uavhengig av andre systemer. Siden integrerte operasjoner som tidligere nevnt i kapittel 3.4 i stor grad knytter sammen disse systemene i et felles nettverk, er det svært viktig at uavhengigheten mellom disse kritiske systemene blir ivaretatt. Noen mekanismer å ivareta denne uavhengigheten vil bli diskutert senere i dette kapitlet.

Utover innretningsforskriften gjelder også styringsforskriften for operatører på norsk sokkel. Styringsforskriftenes paragrafer har til hensikt å redusere risikoen forbundet med oljeutvinning, og styringsforskriftens § 4 lyder som følger: «*Ved reduksjon av risiko som nevnt i rammeforskriften § 11, skal den ansvarlige velge tekniske, operasjonelle og organisatoriske løsninger som reduserer sannsynligheten for at det oppstår skade, feil og fare- og ulykkessituasjoner*». En viktig faktor for å

kunne redusere sannsynligheten for en ulykke er å innføre tilstrekkelige barrierer mellom de forskjellige delsystemene.

Det pliktes videre å analysere risikoen til systemet, hvilke barrierer som finnes, hvilke barrierer som behøves og hvordan disse barrierene bør driftes i henhold til styringsforskriftens § 5. Barrierer kan i disse nettverkene realiseres ved bruk av for eksempel brannmurer og DMZ-er som vil bli forklart og diskutert i henholdsvis kapittel 7.5 og 7.6.

7.2 Standarder og retningslinjer

Det finnes en rekke standarder og retningslinjer relevante både for vanlige IT-nettverk og for industrielle nettverk. Dette kapittelet vil forsøke å gi en oversikt over de mest sentrale av disse standardene og retningslinjene, og de mest sentrale av disse er gjengitt i tabell 7.1 og 7.2. Mange av disse standardene og retningslinjene burde være sentrale i implementeringen av sikre nettverksløsninger. I kombinasjon med retningslinjene vil de tradisjonelle sikkerhetsstandardene være viktige for å sørge for at implementasjonen av integrerte operasjoner blir så sikker som mulig.

Det vil nå bli gitt en litt grundigere gjennomgang av et par de mest sentrale standardene og retningslinjene fra tabell 7.1 og 7.2.

IEC61508

IEC61508 [14] er en svært sentral standard for systemer hvor det stilles krav til sikkerhet. Standarden definerer blant annet risikobegrepet, graderer konsekvens og frekvens, samt metoder for å kalkulere komponenters eller systemers SIL-nivå (*en:**S**afety **I**ntegrity **L**evel*). Ved bruk av denne standarden deles sikkerheten av systemer inn i fire SIL-nivåer som går fra SIL1 til SIL4, hvor SIL4 er klassifiseringen av de sikreste systemene.

Det finnes flere metoder å beregne SIL-nivået på og for utregninger henvises det til [30].

IEC61511

Tett knyttet til IEC61508 er IEC61511, som tar for seg de mer praktiske temaene rundt design av sikkerhetskritiske systemer. Standarden er ment for å veilede de-

signeren om hvordan ønsket SIL-nivå kan oppnås helt fra systemene er på tegnebordet og til igangsetting. IEC61511 fokuserer spesielt på SIS og kravene rundt dens delsystemer.

OLF104

OLF104 er utarbeidet av Norsk Olje & Gass (tidligere oljeindustriens landsforbund, OLF) for å gi operatører krav og retningslinjer for bruk av informasjonsteknologi i kontrollnettverk og er dermed svært sentral for innføringen og videreutviklingen av integrerte operasjoner. Retningslinjene bygger på IEC27001 og IEC27002 som beskriver sikkerhetsmekanismer i IKT-nettverk, og OLF104 oppsummerer dette i 16 retningslinjer eller ISBR-er (*en: information security baseline requirements*), hentet fra [27]:

1. **Sikkerhetsreglementet for prosesskontroll-, sikkerhet- og supporteringssystemer skal være dokumentert.**
   Dokumentet skal forklare bedriftens sikkerhetsreglementer, samt definere roller og ansvar for dette feltet i organisasjonen.

2. **Det skal bli gjennomført risikovurderinger for prosesskontroll-, sikkerhet- og supporteringssystemer og nettverk.**
   Under risikovurderingen vil systemenes risiko i form av konsekvens og frekvens bli forsøkt kartlagt. Retningslinjene legger ingen føringer på hvilken metodikk eller hvor detaljerte risikovurderinger som bør gjennomføres.

3. **Prosesskontroll-, sikkerhet- og supporteringssystemer skal ha en bestemt systemeier, ansvarlig for å opprettholde tilstrekkelig sikkerhet i systemet.**
   Systemeierens oppgave er å sørge for at systemet har et tilstrekkelig sikkerhetsnivå. Dette kan innebære alt fra oppdateringer, autorisering av brukere og data samt kontakt med leverandører av program- og maskinvare.

4. **Nettverksinfrastrukturen skal dele opp nettverket i segmenter, og all kommunikasjon skal bli monitoret.**
   Segmentering av nettverket gjør at forskjellige deler av systemet kan jobbe i individuelle delsystemer. Dette muliggjør blant annet høyere sikkerhet og at delsystemer lettere kan opprettholde eventuelle sanntidskrav.

5. **Brukerene av prosesskontroll-, sikkerhet- og supporteringssystemer skal få opplæring innen bedriftens krav til informasjonssikkerhet og akseptabel bruk av disse systemene.**
   Feil bruk av IKT-utstyr kan lett føre til sikkerhetsbrister, og god trening av

ansatte og andre som er tilkoblet bedriftens nettverk kan enkelt sørge for at risikoen for slike brudd senkes.

6. **Prosesskontroll-, sikkerhet- og supporteringssystemer skal kun benyttes til sitt bestemte bruk.**
   Utstyr skal konfigureres spesielt til sitt bruk, og unødvendig funksjonalitet skal fjernes for å redusere de potensielle tilkoblingsmulighetene og sikkerhetshullene til systemet.

7. **Det skal dokumenteres at det finnes tilstrekkelig nødevakueringsplaner for prosesskontroll-, sikkerhet- og supporteringssystemer samt at disse skal testes.**
   Disse dokumentene skal definere hvordan man kan få prosessen og tilstøtende IKT-systemer i drift igjen etter stans. Dokumentene skal også inneholde kontaktinformasjon til nøkkelpersoner, prosedyrer for igangsetting, plasseringen av viktig utstyr og annen tilsvarende informasjon.

8. **Kravene til informasjonssikkerhet skal integreres i bedriftens eksisterende design-, prosedyre- og igangssettingsprosesser.**
   Kravene bør distribueres til eventuelle leverandører og det bør kunne dokumenteres at disse kravene imøtekommes.

9. **Kritiske prosesskontroll-, sikkerhet- og supporteringssystemer skal ha definerte og dokumenterte service- og supportnivåer.**

10. **Infrastrukturendringer skal ikke skje uten arbeidstillatelse fra organisasjonen. Endringer av strukturen må skje i henhold til bedriftens regler.**
    Bedriften bør utarbeide regler og rutiner for endring i infrastrukturen, og en liste over punkter som bør være med i en slik arbeidstillatelse kan finnes i [27].

11. **Fullstendig og oppgradert oversikt over nettverkstopologien med alle komponenter og tilkoblingsmuligheter skal være tilgjengelig.**
    Nettverk øker i kompleksitet, og fullstendig detaljert oversikt over komponenter og tilkoblingsmuligheter er derfor essensielt for å holde oversikt. Oversikten bør inneholde navn, logiske adresser, fysisk plassering, konfigurasjoner, åpne porter og hvilken protokoller som blir sendt til og fra enheten.

12. **IKT-systemer skal holdes oppdatert når tilkoblet prosesskontroll-, sikkerhetsystemer eller supporterende nettverk.**
    Et uoppdatert system kan gjøre kontrollnettverket åpent for angrep, er det viktig at alt utstyr koblet til disse nettverkene blir oppdatert når oppdate-

ringer blir gjort tilgjengelige fra leverandøren. For kritisk utstyr må organisasjonen utvikle rutiner for å sikkert kunne oppdatere systemet.

13. **Prosesskontroll-, sikkerhet- og supporteringssystemer skal ha oppdatert og aktiv beskyttelse mot ondsinnet programvare.**
    Oppdatert og aktiv beskyttelse mot ondsinnet programvare kan sørge for at kjent ondsinnet kode kan bli oppdaget og stoppet før den kommer inn i systemet.

14. **Alle tilgangsforespørsler skal blokkeres om tilgang ikke er gitt eksplisitt.**
    Som en sikkerhetsanbefalling bør alle tilkoblingsforsøk blokkeres i brannmurer, og nye typer tilkoblinger bør kun godkjennes ved søknad og innvilgning av denne søknaden.

15. **Operasjons- og vedlikeholdsprosedyrer skal dokumenteres og holdes oppdatert.**
    Oppdaterte operasjons- og vedlikeholdsprosedyrer gjør at systemet lettere kan driftes og holdes i god stand.

16. **Rapporteringsprosedyrer for nesten-ulykker og ulykker skal dokumenteres og implementeres i organisasjonen.**
    Ved bruk av gode systemer for å registrere og rapportere nesten-ulykker, ulykker og angrep slik at man kan man utbedre feil. Slike systemer kan for eksempel benyttes til å utarbeide bedre rutiner eller systemer.

OLF104 inkluderer også litt mer utfyllende beskrivelse av de forskjellige retningslinjene, og referer til relevante kapitler i standarder som IEC27001 og IEC27002. OLF104 er ikke myndighetspålagte retningslinjer, og operatørene står derfor fritt til å velge om de vil benytte seg av retningslinjene. Et eksempel på en integrering av OLF104 finner vi hos Statoil, hvor retningslinjene er flettet inn i de eksisterende tekniske kravene og arbeidsprosessene til organisasjonen [37].

SeSa

SeSa (*SecuritySafety*) er en metodisk prosedyre for å vurdere om en kobling fra Internett inn mot SIS vil ha påvirkning på SIL-nivået. Rapporten gir også en god oversikt over mulige angrep et slikt nettverk kan bli utsatt for, samt tips til hvordan sikkerhetskomponentene i nettverket kan struktureres og konfigureres for å stå imot slike angrep.

ISA/IEC-62443

ISA/IEC-62443 (tidligere ANSI/ISA-99) blir både av EU og Statoil utpekt som kanskje den viktigste standarden for å implementere sikre SCADA-systemer. Standarden dekker aspekter som design, sammenkobling, topologi og prosedyrer for drift av SCADA-systemer. ISA/IEC-64443 er delt opp i de fire deler:

1. **Generell del:**
   Tar for seg begreper og definisjoner relevant for standarden, og beskriver hvordan sikkerhetsarbeidet bør utføres gjennom hele levetiden til SCADA-systemet.
2. **Regler og prosedyrer:**
   Denne delen av standarden beskriver hvordan organisasjonen (systemeieren) bør utvikle og vedlikeholde gode sikkerhetsprosedyrer.
3. **Systemintegrering:**
   Denne delen av standarden er beregnet på de som designer SCADA-systemer og hjelper designeren å lage sikre løsninger. Det blir også gitt retningslinjer for sikker design av delsystemer, samt hvordan delsystemer kan integreres på en sikker måte.
4. **Komponentleverandør:**
   Denne delen er laget for de som produserer PLS-er eller tilsvarende komponenter til SCADA-systemet og lister blant annet opp de tekniske løsningene som anbefales for å realisere sikre komponenter.

Standard	Beskrivelse
IEC61508	*Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems*. • Risiko. • Risikoanalyse. • SIL (Safety Integrity Level). • Formler for kalkulering av SIL.
IEC61511	*Functional safety - Safety instrumented systems for the process industry sector*. • Funksjonell sikkerhet. • Hvordan designe SIS for å få et akseptabelt SIL-nivå.
OLF104	*Norwegian Oil and Gas Information Security Baseline Requirements, ISBR*. • Retningslinjer for hvordan en kan implementere «best practice». • Bygger på IEC 27001 og IEC 27002. • Tillegg til organisasjonens rutiner.
IEC27001	*Information technology - Security techniques - Information security management systems - requirements.* • Information system management system (ISMS). • Hvordan designe, implementere og drifte systemer sikkert.
IEC27002	*Information technology - Security techniques - Code of practice for information security management.* • Beste praksis for å implementere sikre systemer.
SeSa	*The SeSa Method for Assessing Secure Remote Access to Safety Instrumented Systems.* • Metode for å systematisk analysere om ekstern tilkobling svekker SIL. • Forslag til struktur og løsninger for økt sikkerhet.

Tabell 7.1: *Standarder og retningslinjer innen sikkerhet for automasjonssystemer. [1/2]*

Standard	Beskrivelse
IEC18028-5	*Information technology - Security techniques - IT network security - Part 5: Securing communications across networks using virtual private networks* • Sikring ved bruk av VPN. • Planlegging og implementasjon av VPN.
IEC18028-4	*Information technology - Security techniques - IT network security - Part 4: Securing remote access.* • Ekstern tilgang. • Teknikker for å sikre tilkoblingen fra ekstern lokasjon.
IEC18028-3	*Information technology - Security techniques - IT network security - Part 3: Securing communications between networks using security gateways.* • Teknikker for å gjøre aksesspunkter sikre. • Pakkefiltrering. • Tilstandsbasert inspeksjon av pakker. • Applikasjonsproxy. • NAT. • Innholdsanalyse.
ISA/IEC-62443	*Network and system security for industrial-process measurement and control.* • Passord. • Autentisering. • Filter/blokkerings-metoder. • Kryptering. • VPN. • Monitoreringsverktøy. • Automatiseringssoftware og deres sikkerhetshull.
NISCC Guide	*Good Practice Guide - Process Control and SCADA Security.* • Retningslinjer for god praksis innen SCADA-sikkerhet. • Risiko. • Sikkerhetsarkitektur. • Tredjepart-risiko.

Tabell 7.2: *Standarder og retningslinjer innen sikkerhet for automasjonssystemer. [2/2]*

7.3 Beskyttelse-i-dybden

Beskyttelse-i-dybden (*en:defence-in-depth*) er et mye brukt begrep innen informasjonssikkerhet, og er nødvendig for å sikre tilstrekkelig sikkerhet i nettverk. Denne beskyttelsesfilosofien gjør det mulig å beskytte seg mot dagens komplekse trusselbilde på en annen måte enn hva som ville vært mulig med bare én sikkerhetsmekanisme.

Beskyttelse-i-dybden kan også hjelpe til å sørge for at det skapes tilstrekkelig med barrierer mellom de forskjellige delsystemene, noe som vil bli diskutert nærmere i kapittel 7.3.

For å kunne stå bedre rustet mot trusler, er det viktig å bygge opp beskyttelsen av nettverket lagvis og gjerne ved bruk av flere forskjellige sikkerhetsmekanismer. En slik lagvis oppbygning gjør at et angrep som mot formodning slipper gjennom en sikkerhetsbarriere forhåpentligvis vil bli fanget opp av neste sikkerhetsmekanisme. Et slikt oppsett vil også kunne øke robustheten til systemet da oppsettet vil kunne beskytte nettverket mot et bredere spekter av angrep enn hva som ville vært tilfelle ved bruk av single løsninger.

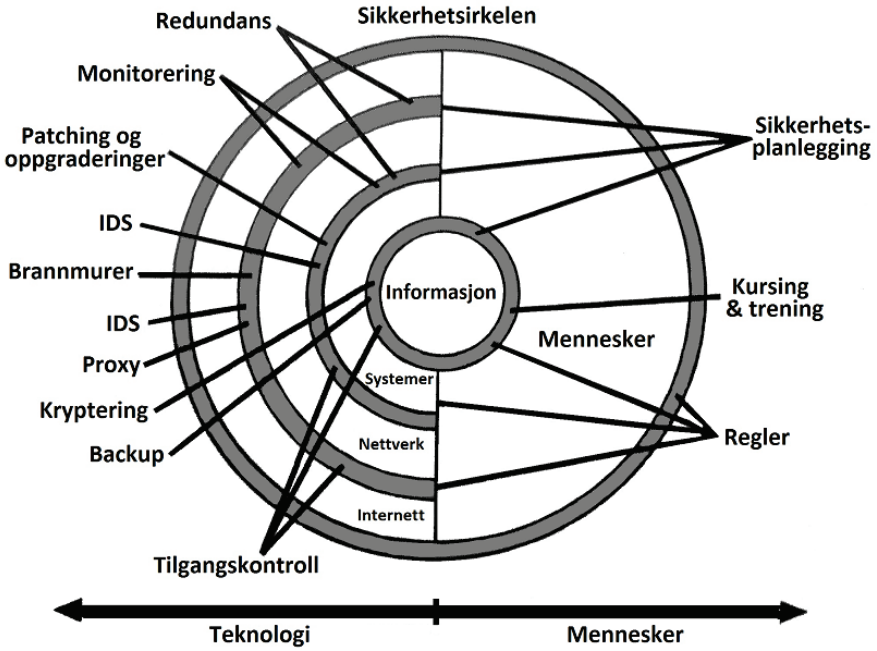
I tillegg til de klassiske sikringsmekanismene som brannmurer (kapittel 7.5), DMZ (kapittel 7.6) eller inntrengingssystemer (kapittel 7.7) kommer tiltak som berører de menneskelige og organisatoriske faktorene i organisasjonen som er viktige for at bedriften skal oppnå god beskyttelse-i-dybden [7]. Tiltak for å forbedre beskyttelsen innen disse feltene er mange, men typiske effektive grep kan være å sørge for at organisasjonen har gode rutiner og regler for hvordan man skal behandle datasystemene på en sikker måte. Dette kan blant annet gjøres ved å holde nettverksadministratorer, andre nøkkelpersoner rundt driften av nettverkene og de vanlige brukerne tilstrekkelig oppdatert på og trent i å håndtere truslene som finnes.

For at organisasjonen skal få en oversikt over de segmenteringene som er gjort eller burde gjøres i nettverket kan det være til hjelp å benytte seg av visuelle modeller. Noen av disse vil nå bli gjennomgått.

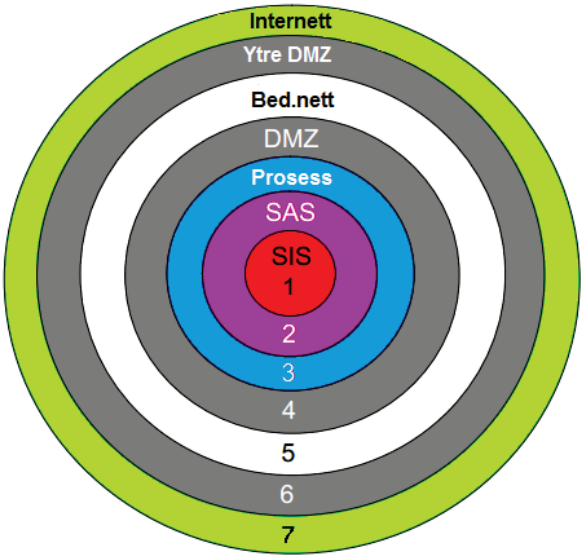
Det blir i [44] benyttet en visuell modell for å illustrere de forskjellige lagene og sikkerhetssystemene som kan brukes i nettverk kalt «sikkerhetssirkelen». Denne kan sees i figur 7.1. Som det kan sees i figuren, ligger bedriftens informasjon (data) på den teknologiske siden kapslet inn av flere sikkerhetsmekanismer lagvis. Dette gjør at potensielle angripere må forsere mange hinder for å nå denne informasjonen. Sikkerhetssirkelen visualiserer også en annen viktig faktor for sikkerheten, den menneskelige. Som kan sees fra figur 7.1 er organisasjonens informasjon beskyttet langt svakere mot menneskene innad i organisasjonen, dette

gjør at de interne aktørene utilsiktet kan utgjøre en stor trussel for organisasjonens informasjon dersom disse får ført med seg virus eller lignende inn i nettverket, som nevnt i kapittel 5.1.

Der sikkerhetssirkelen er gyldig for alle typer nettverk, blir det i [11] forslått en tilsvarende modell, «Løkmodellen», tilpasset industrielle nettverk. Som illustrert i figur 7.2 bør de forskjellige delsystemene innkapsle hverandre, på den måten kan man oppnå beskyttelse-i-dybden for de innerste og mest kritiske systemene. Sikkerhetsmekanismer som brannmurer eller tilsvarende vil typisk kunne finnes mellom lagene slik at man oppnår tilstrekkelig beskyttelse og segmentering mellom de tilstøtende delsystemene. Hvordan Løkmodellen kan realiseres i et fysisk system vil bli vist i kapittel 7.6.



Figur 7.1: *En skjematisk oversikt over beskyttelse-i-dybde og hvilke sikkerhetsmekanismer som kapsler inn bedriftens informasjon. Som en kan se fra figuren har menneskene i organisasjonen mye lettere tilgang til deler av denne informasjonen, noe som gjør sikkerhetsplanlegging, trening, kursing og regler viktig for å opprettholde sikkerheten. Figuren er hentet fra [44].*



Figur 7.2: *Figuren viser Løkmodellen, en visuell fremstilling av beskyttelse-i-dybden presentert i [11]. I motsetning til sikkerhetssirkelen er Løkmodellen tilpasset industrielle nettverk. Mellom de forskjellige lagene plasseres brannmurer eller tilsvarende sikkerhetstiltak for sørge for tilstrekkelige barrierer og sikkerhet mellom delsystemene.*

7.4 Metoder for å øke robusthet

Det ble i kapittel 4.3 gitt en kort definisjon på og innføring i robusthetsbegrepet, en systemdesignmetode som er svært viktig for å realisere sikre løsninger for IO. Flere av punktene i OLF104 refererer til prinsipper forbundet med robusthet.

Det vil i dette delkapitlet bli gitt en kort gjennomgang av hvordan organisasjoner kan benytte robusthet for å forbedre sikkerheten i systemene sine. Johnsen har i [20] listet opp fire punkter som kan være nyttige for organisasjoner som ønsker å øke sine systemers robusthet:

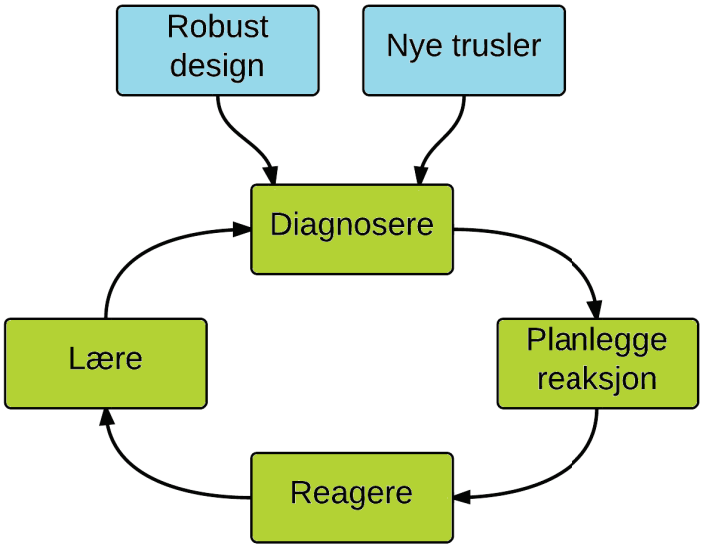
- **Forventning:** Å vite hva som kan oppstå av hendelser eller forskjellige angrep.
- **Årvåkenhet:** Å vite hva en skal se etter av forskjellige situasjoner, og aktivt se etter disse.
- **Respons:** Å vite hva som skal gjøre dersom farlige situasjoner eller angrep oppstår.
- **Lære:** Å ha systemer for å rapportere hendelser, samt studere disse rapportene nøye.

For å oppnå robusthet er det viktig at organisasjonen gjennomfører en grundig risikoanalyse. En slik risikoanalyse gjennomføres for å avdekke potensielle problemer eller sammenhenger som kan utgjøre en risiko for systemet. Dette er blant annet anbefalt i #2 i OLF104. På grunn av den økte kompleksiteten IO fører med seg kan disse undersøkelsene kreve både mye tid og ressurser å utføre. Grundige analyser er likevel viktig for å gi en god oversikt over flest mulig risikoer, samtidig som at analysen kan hjelpe til å gi en oversikt over systemet som helhet, ikke bare på delnivå.

Når risikoene blir avdekket, er det viktig at det planlegges forskjellige løsninger eller prosedyrer for hvordan man skal håndtere de. Dette gjør at problemer kan møtes med gjennomtenkte løsninger dersom de dukker opp, og dette kan eliminere mange menneskelige feil som blir gjort i stressende situasjoner.

Dersom en skulle være uheldig å oppleve en svikt på sikkerhetssystemene, er det viktig at organisasjonen lærer av hendelsen, og utbedrer problemet slik at det ikke skjer igjen. Feilhendelsene bør i henhold til OLF104 #16 systematiseres og rapporteres slik at organisasjonen kan ta vare på og lære av disse opplysningene. Disse feilene kan også avdekke risiko og avhengigheter en kanskje ikke hadde tenkt over, noe som bør utløse en ny runde med risikoanalyse. For å opprettholde sikkerheten i systemene er det viktig at organisasjonen jobber med dette arbeidet kontinuerlig.

Det er i [15] gitt en figur som illustrerer hvordan arbeidet med robusthet, beskrevet i det foregående avsnittet, bør gjennomføres, og denne figuren kan sees i en videreutviklet utgave i figur 7.3.



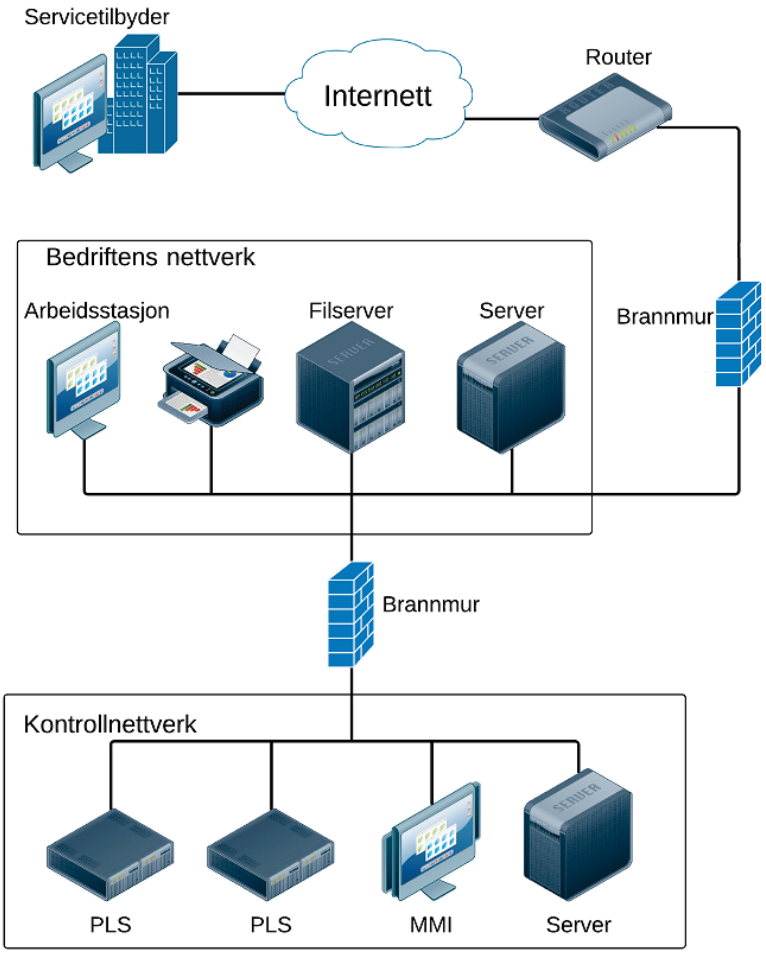
Figur 7.3: *Figuren er en videreutviklet utgave av en figur fra [15] som viser hvordan risikoanalyser kontinuerlig bør gjennomføres i organisasjonen for å øke robustheten og dermed sikkerheten til systemene.*

7.5 Brannmurer

Brannmurer er en av de viktigste sikkerhetsrelaterte komponentene i nettverk [44], og opprettholder sikkerheten ved å benytte brukerbestemte lister eller konfigurasjoner for å bestemme om datatrafikk skal få slippe gjennom brannmuren eller stoppes.

Brannmurer blir gjerne plassert i periferien av nettverkene, for eksempel mot Internett, men blir i tillegg i stor grad brukt innad i nettverk for å realisere beskyttelse-i-dybden (kapittel 7.3). Dette gjøres blant annet ved å segmentere nettverket i flere delnettverk i henhold til OLF104 #4, noe som også kan begrense omfanget av eventuelle interne trusler som kan oppstå dersom systemer blir kompromittert eller det oppstår feilsituasjoner [34]. En figur av hvordan brannmurer kan brukes både i periferien og i sentrale deler av nettverket kan sees i figur 7.4.

I dette kapitlet vil først forskjellige typer brannmurer bli presentert før viktigheten av konfigurering og aktiv drift av brannmurene i nettverket blir diskutert.



Figur 7.4: *Figuren viser hvordan brannmurer både kan benyttes som barrierer mellom bedriftsnettverket og Internett og mellom bedrifts- og kontrollnettverket.*

7.5.1 Forskjellige typer brannmurer

Brannmurer kan være basert på maskinvare, programvare eller det mest vanlige, en kombinasjon av disse to. Brannmurene kan utføre flere oppgaver i et nettverk:

- **Pakkefiltrering:** Inspisere datapakkene som kommer inn til brannmuren for så å bestemme om disse datapakkene skal få lov til å slippe gjennom.
- **Applikasjonsproxy:** Noen applikasjoners datatrafikk får slippe gjennom, mens trafikken til andre applikasjoner enten blir blokkert eller omdirigert.
- **Logging:** Logge trafikken som går gjennom eller blir blokkert av brannmuren. Den loggede datatrafikken kan blant annet benyttes til å avdekke

forsøk på angrep.

- **Muliggjøre VPN:** Enkelte brannmurer kan benyttes for å realisere sikre «punkt-til-punkt»-tilkoblinger ved hjelp av VPN. Dette vil bli nærmere diskutert i kapittel 7.9.

Det finnes mange typer forskjellige brannmurer og de utfører oppgavene sine basert på forskjellige prinsipper. Hvilke brannmurer som skal benyttes bør avgjøres etter en vurdering av hvilke krav man har til brannmuren. For eksempel kan det postuleres kriterier som: maksimal gjennomstrømning (*en: throughput*), muligheten for overvåking av trafikken gjennom brannmuren eller hvilket nettverkslag brannmuren skal undersøke [34].

De to mest sentrale typene av brannmurer benyttet i nettverk vil nå bli gjennomgått.

Pakkefiltrerende brannmurer

Datapakker som blir sendt gjennom et nettverk består gjerne av to deler: pakkehode og pakkedata. I hodet finnes typisk informasjon om hvem sender og mottaker er, mens datafeltet inneholder pakkens faktiske data. Tilnærmet alle brannmurer [44] sjekker hodet av pakken og bestemmer så, ved hjelp av en forhåndskonfigurert liste, om datapakken skal få slippe gjennom brannmuren.

Denne enkleste formen for filtrering av datapakker kalles *tilstandsløse pakkefiltrering*. Brannmurene som benytter denne metoden ignorerer typen kobling mellom sender og mottaker,- en pakke er en pakke, uansett hvilket tidspunkt den ankommer.

Brannmurer som er *tilstandbevisste og pakkefiltrerende* lagrer tilstanden til de forskjellige koblingene som går gjennom brannmuren i en intern liste. Listen inneholder typisk en oversikt over attributter som tilstanden på tilkoblingen (etablerende, etablert eller terminerende) og sekvensnummer på pakkene som ventes.

Et typisk eksempel på en slik liste er hentet fra [34] og kan sees i tabell 7.3. Denne egenskapen gjør at brannmuren har en oversikt over når interne maskiner venter på respons på et kall gjort mot destinasjoner utenfor organisasjonens nettverk. Når responsen på et kall kommer vet brannmuren at et svar er forventet og datapakken kan derfor slippe gjennom [44].

Senders adresse	Senders portnr.	Dest. adresse	Dest. portnr.	Tilstand
192.168.1.100	1030	192.0.2.71	80	Etablerende
192.168.1.102	1031	10.12.18.74	80	Etablert
192.168.1.101	1033	10.66.32.122	25	Etablert
192.168.1.106	1035	10.231.32.12	79	Terminerende

Tabell 7.3: *Eksempel på hvordan en tilstandsbevist pakkefiltrende brannmur kan holde oversikt over forskjellige tilkoblinger.*

Applikasjonsfiltrerende brannmurer

En annen måte å filtrere trafikk på er å bruke brannmurer som filtrerer på applikasjonslaget. Disse brannmurene går dypere ned i trafikken, og ser på datafeltet til datapakken fremfor hodet slik de pakkefiltrende brannmurer gjør [34]. Dette gjør at man i enda større grad har mulighet til å kontrollere hva som slipper inn og ut av nettverket. Disse brannmurene kan for eksempel benyttes til å blokkere bestemte typer datafiler som kommer inn i bedriftens nettverk, for eksempel via e-postvedlegg [34].

Ulemper med applikasjonsfiltrerende brannmurer er at de fleste brannmurene enda ikke er så avanserte at de støtter mer enn et par applikasjoner av gangen [44], noe som fører til at de må rekonfigureres dersom andre applikasjoner skal benyttes. Tidligere brukte også disse brannmurene mye tid på prosessering av hver pakke, noe som reduserte ytelsen til nettverket, men dette har blitt et mindre problem da dagens maskinvare er vesentlig raskere.

En aktuell løsning på ytelsesproblemet kan være å kombinere en pakkefiltrerende og en applikasjonsfiltrerende brannmur i serie. Den pakkefiltrerende brannmuren kan da ta seg av grovfiltreringen, mens den applikasjonsfiltrerende sjekker pakken nøyere. Et slikt oppsett vil også kunne forbedre beskyttelse-i-dybden til nettverket da pakkene må gjennom av to forskjellige filter.

7.5.2 Konfigurering av brannmurer

Brannmurer er som nevnt innledningsvis en av de viktigste sikkerhetsmekanismene i nettverk, men dette forutsetter at de er korrekt konfigurert. Det sentrale spørsmålet i denne sammenhengen er: Hvilken trafikk skal slippe gjennom brannmuren, og hvilken skal bli stoppet? Det er i [38] gitt en god oppsummering over hvordan en brannmur skal konfigureres, og poengene derifra vil bli gjengitt i

dette delkapittelet:

- Hovedregelen for konfigurasjon av brannmurer er å sperre for absolutt all trafikk bortsett fra den som er absolutt nødvendig. Denne reglen harmoniserer blant annet med OLF104 #14.
- Det er viktig å få istand et godt samarbeid mellom IT-avdelingen, som gjerne står for driften av brannmurene, og prosessingeniørene som vet hvilken trafikk som er nødvendig for prosessen [38]. Det bør også utarbeides rutiner for hvordan det skal søkes om å få godkjent nye typer trafikk.
- Alle regler for hvilke pakker som får slippe gjennom brannmuren må inneholde IP-adressen og portnummeret koblet til datapakken.
- Trafikk skal ikke gå direkte fra kontroll- til bedriftsnettverket eller vice versa, men gjennom en DMZ (kapittel 7.6).
- Kommunikasjonsprotokollen benyttet mellom kontrollnettverket og DMZ skal blokkeres mellom DMZ og bedriftsnettverket, og protokollen som benyttes mellom DMZ og bedriftsnettverket skal blokkeres mellom DMZ og kontrollnettverket. Dette gjør det vanskelig for trafikk å feilaktig flyte gjennom DMZ-en.
- All trafikk som går ut fra nettverket skal være merket med destinasjonens IP-adresse og portnummer for å kunne slippe gjennom brannmuren.
- Ingen av enhetene som er koblet til kontrollnettverket skal ha tilgang til Internett, verken direkte eller gjennom en brannmur.
- Styring og konfigurering av brannmurer må kun skje fra et begrenset antall maskiner. Dette er blant annet for å forhindre at noen forfalsker konfigurasjonene eller sletter/endrer logger.
- Det skal utvikles rutiner for taktisk gjennomgang av brannmurenes logger slik at eventuelle angrep, forsøk på angrep eller mistenkelig trafikk kan detekteres.

7.5.3 Implementering av brannmurer i nettverket

For å få optimal beskyttelse er det ikke nødvendigvis hensiktsmessig å plassere så mange brannmurer man kan i organisasjonens nettverk, og det blir i [34] foreslått en fremgangsmåte for hvordan arbeidet rundt implementeringen av brannmurer bør gjennomføres:

1. **Planlegge**: Organisasjonen må som nevnt tidligere kartlegge sine behov og krav til brannmurene. Dette bør blant annet gjøres slik at brannmurene dimensjoneres etter hvor mye trafikk som kan forventes å gå gjennom de.

2. **Konfigurere**: Brannmurene bør konfigureres nøye i henhold til for eksempel foregående kapittel eller tilsvarende retningslinjer for konfigurering.

3. **Teste**: Før brannmurene settes i drift i nettverket bør de testes, både med tanke på nettstorm (kapittel 5.2.2) og for å kunne avdekke andre svakheter. For eksempel kan testverktøy som TOCSSiC (kapittel 7.11.1) brukes. En fullstendig liste over hva som kan inkluderes i en slik test kan finnes i [34].

4. **Iverksette**: Når det er bekreftet at brannmurens oppførsel under test er i henhold til de krav og spesifikasjoner fra punkt 1 og 2, kan brannmuren implementeres i tråd med de retningslinjer organisasjonen har for installering av nytt utstyr i nettverket.

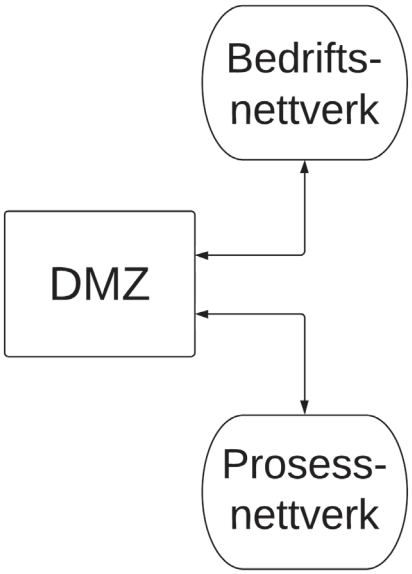
   Det er samtidig viktig at organisasjonen oppdaterer samtlige oversikter over nettverkstopologien som den nye installeringen berører. Dette harmoniserer med OLF104 #11, og er viktig for at organisasjonen alltid skal kunne vite hvilke enheter som finnes på nettverket.

5. **Drifte**: Etter at brannmuren er satt i drift er ikke arbeidet ferdig. Det er viktig at logger og konfigurasjoner i brannmurene blir gjennomgått jevnlig slik at eventuelle feil eller angrep kan avdekkes så raskt som mulig [34].

Selv om brannmurer er viktig i sikring og strukturering av nettverk, er heller ikke disse komponentene feilfrie. En stor begrensing med brannmurer er at de må lese trafikken for at de skal kunne ta en beslutning om pakken skal passere gjennom. Dette gjør det blant annet vanskeligere for brannmurene å takle krypterte eller ufullstendige datapakker som ikke er «korrekt» formatert.

7.6 Demilitariserte soner (DMZ)

DMZ (*en: DeMilitarized Zone*) er forkortelsen som benyttes på begrepet demilitariserte soner i nettverk. DMZ-er opprettes ved bruk av blant annet brannmurer som skaper logiske barrierer mellom for eksempel bedrifts- og kontrollnettverket eller mellom bedriftsnettverket og Internett slik som vist i figur 7.5.



Figur 7.5: *Overordnet oversikt over hvordan en DMZ kan benyttes til å koble sammen bedrifts- og prosessnettverk.*

Denne logiske barrieren vil, hvis den er konstruert korrekt, forhindre at nettverkstrafikken flyter direkte fra det ene segmentet av nettverket og til det andre. DMZ er et godt eksempel på hvordan beskyttelse-i-dybden (se kapittel 7.3) kan forbedre nettverkssikkerheten da DMZ-er innfører en ekstra barriere mellom «A» og «B» både i form av et logisk skille og det faktum at det benyttes én eller flere brannmurer for å kunne lage DMZ-en.

Et praktisk eksempel på anvendelsen av DMZ kan gis av situasjonen når man ønsker å lese og skrive til en prosessvariabel fra bedriftsnettverket. Det kan være risikabelt å åpne for direkte manipulering av denne variabelen da dette kan skape muligheter for blant annet angripere å komme seg inn i prosessnettverket. Istedenfor direkte manipulering kan endringen bli sendt fra bedriftsnettverket til en DMZ. På den andre siden av DMZ-en kan en PLS jevnlig sjekke om det har kommet oppdateringer i prosessvariabelen hos DMZ-en. Når det kommer en oppdatering kan PLS-en lese og sette denne verdien i prosessen.

DMZ-er kan også benyttes til lagring av dataserier ved at prosessen jevnlig sender data til DMZ-en, noe som kan benyttes til for eksempel trending av prosessdata. At data samles i DMZ-en på denne måten gjør at man unngår periodiske kall fra bedriftsnettverket og til prosessnettverket.

Det finnes flere måter å implementere en DMZ på, og det blir i [38] gitt en gjen-

nomgang av flere konfigurasjoner. Den enkleste løsninger med én brannmur med to innganger og kan sees i figur 7.6. Dette er den billigste og enkleste løsningen for å realisere DMZ. En mer avansert løsning er å benytte to brannmurer, noe som kan sees i figur 7.7. I løsningen som benytter to brannmurer vil man kunne filtrere data bedre, og de to brannmurene kan konfigureres forskjellig etter hvilken type trafikk som går på det aktuelle segmentet.

En mulig implementasjon av Løkmodellen fra kapittel 7.3 som benytter to DMZ-er kan sees i figur 7.8. Det benyttes her brannmurer mellom de forskjellige delsystemene for å oppnå god segmentering av nettverket, og DMZ-er mellom nettverkssegmenter hvor man ønsker et ekstra lag sikkerhet. En indre DMZ ligger mellom prosess- og bedriftsnettverket, mens en ytre DMZ er plassert mellom bedriftsnettverket og Internett.

Ulempen med DMZ er at det kreves mye arbeid og kunnskap for å kunne konfigurere de sikkert og korrekt, samt at det må investeres i maskinvare for å kunne realisere DMZ-en. En annen ulempe med DMZ-er er at de kan redusere gjennomflytsevnen [45], noe det er viktig å ta i betrakting dersom trafikken som skal gå gjennom DMZ-en er tidskritisk.

Konklusjonen i litteraturen er at fordelene med DMZ-er er så store sett fra et sikkerhetsmessig ståsted at det anbefales å benytte seg av disse i nettverket selv om det kan medføre en del arbeid og kostnader for organisasjonen.

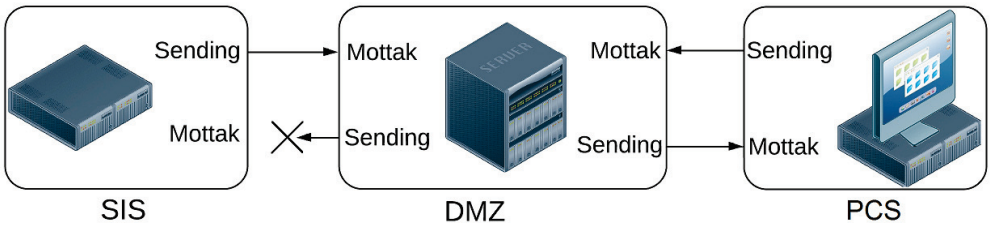
7.6.1 Diodeløsningen

Diodeløsningen blir beskrevet i [11], og er en analogi til den elektriske komponenten diode som bare kan lede strøm i en retning. Ideen bak diodeløsningen er å holde kommunikasjonskanaler som ikke trenger å være to-veis enveis.

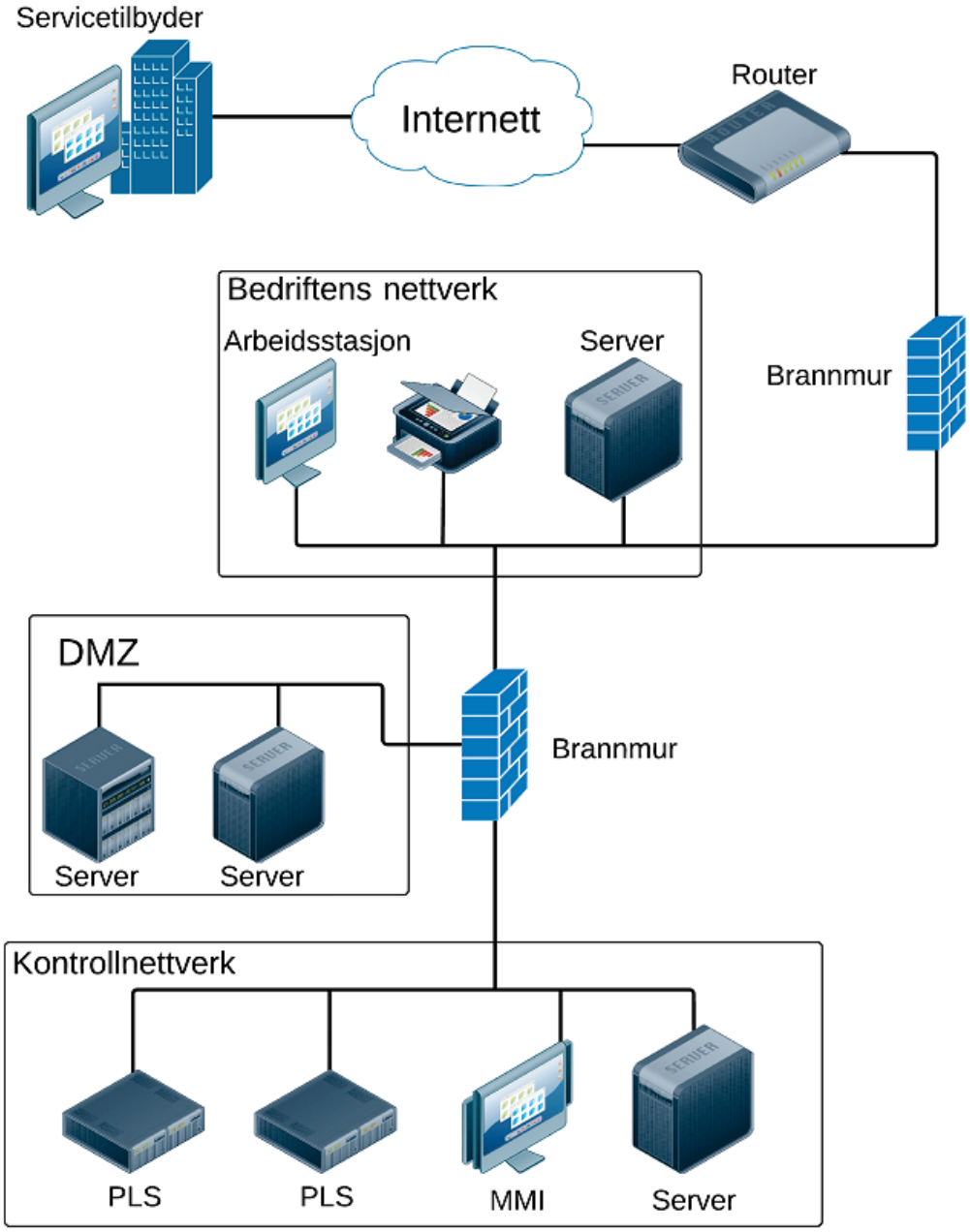
Et eksempel på slike kommunikasjonskanaler er kommunikasjonen mellom SIS og en DMZ hvor det skal overføres statusvariabler fra sikkerhetssystemene og til DMZ-en. Dersom denne kommunikasjonskanalen ikke blir satt opp korrekt kan kanalen under et angrep eller feiltilstand potensielt benyttes for å sende uønsket data fra DMZ-en og inn i sikkerhetssystemet.

Problemet med diodeløsningen er at flere moderne kommunikasjonsprotokoller, som for eksempel den utbredte TCP/IP-protokollen, avhenger av at mottakeren av datapakkene sender en bekreftelse på at pakken er mottatt. Det blir i [11] foreslått en løsning på dette problemet hvor det benyttes en akseptløs protokoll slik som for eksempel UDP, samt at det lages en fysisk løsning hvor kommunikasjonen kun kan gå en vei. Dette kan løses ved å lage et system hvor man fysisk

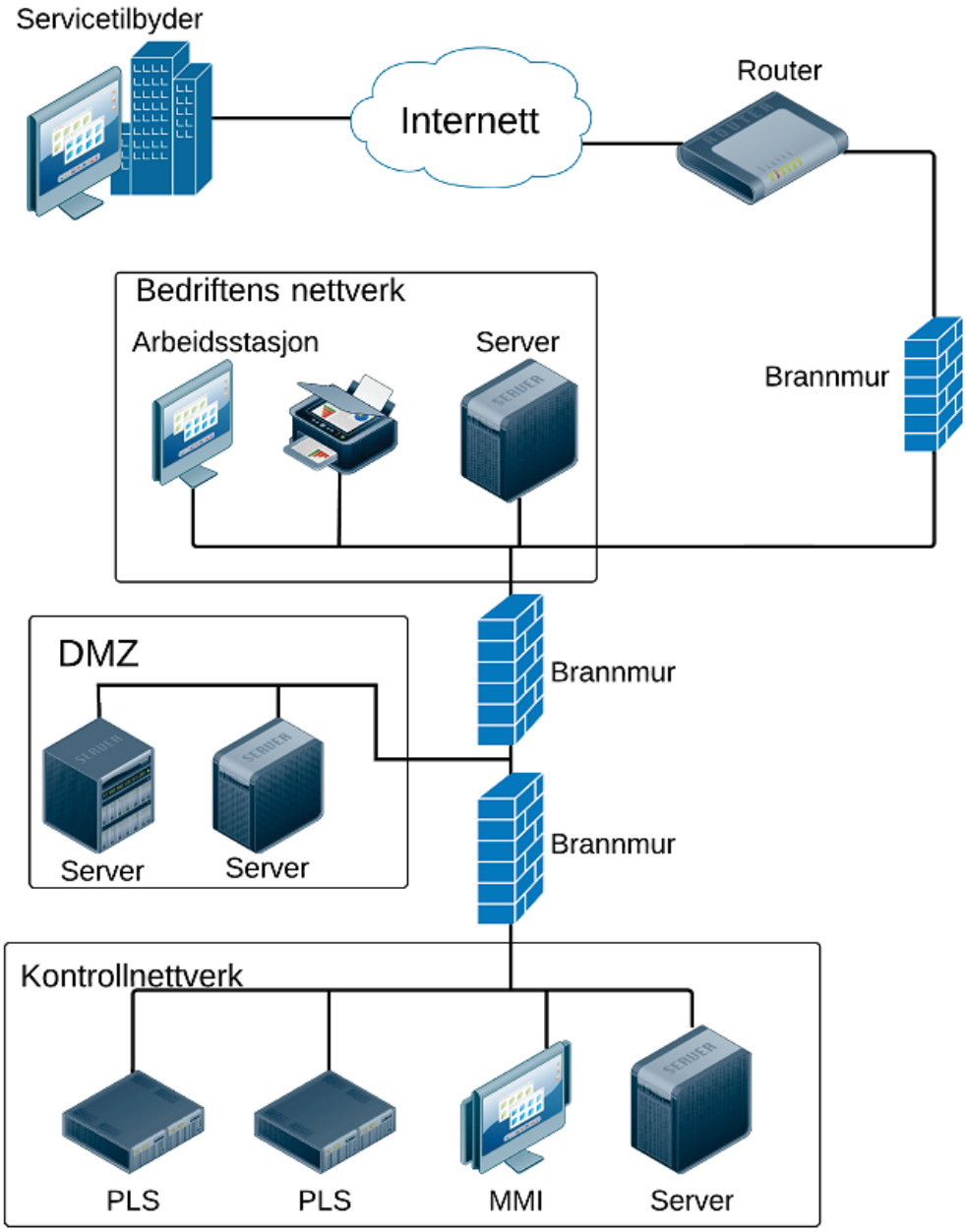
kobler av den innkommende koblingen til det indre systemet, som vist i figur 7.9.



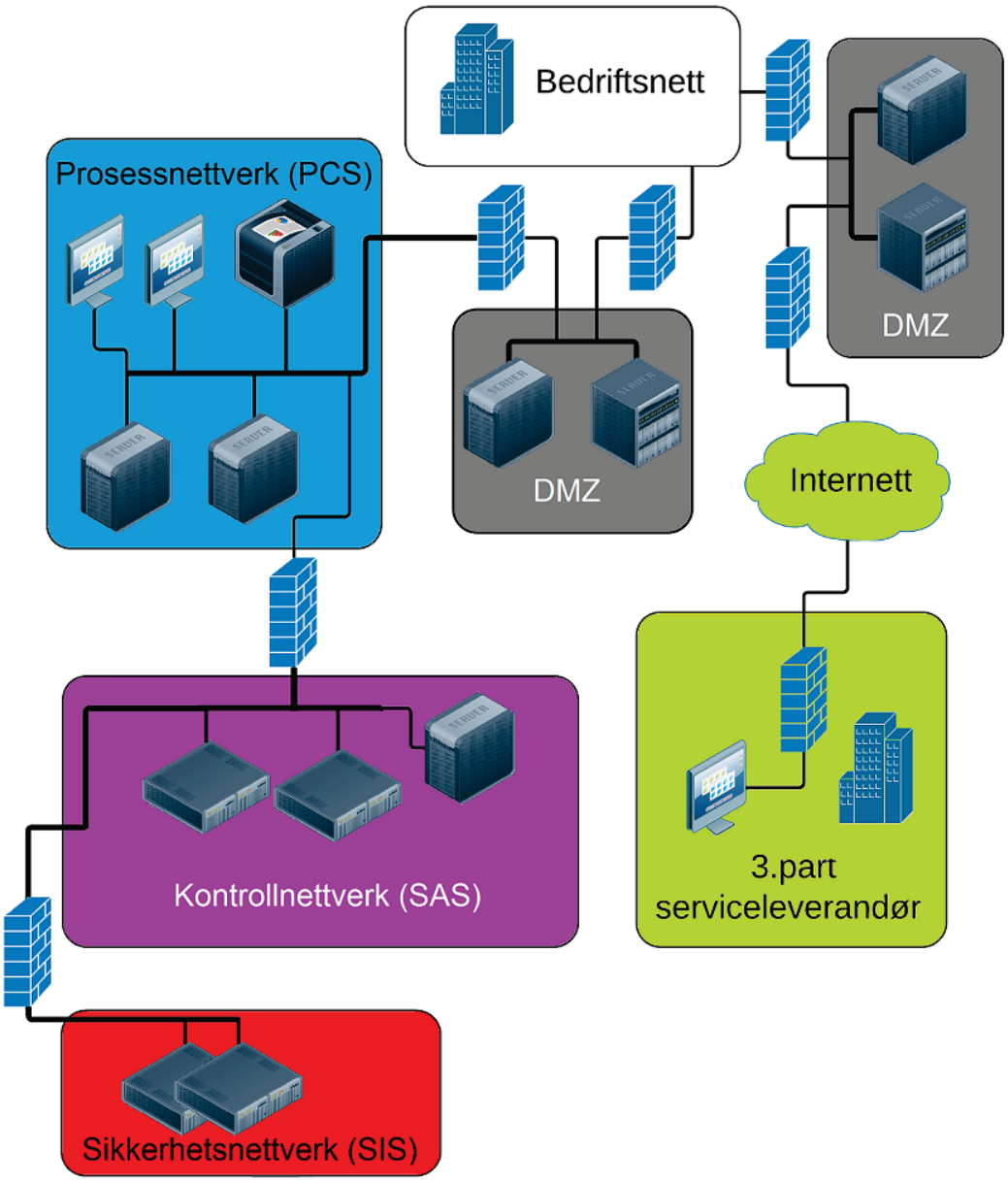
Figur 7.9: *Figuren viser hvordan en kan implementer en diodeliknende løsning for å forhindre at kommunikasjon flyter fra DMZ til SIS ved fysisk å fjerne denne koblingen.*



Figur 7.6: *En skisse over prosessnettverk, bedriftsnettverk, mulighet for ekstern tilkobling samt en DMZ som begrenser nødvendig aksess fra bedrift- til prosessnettverk*



Figur 7.7: *En skisse over prosessnettverk, bedriftsnettverk, mulighet for ekstern tilkobling samt en DMZ som begrenser nødvendig aksess fra bedrift- til prosessnettverk. Utstyrt med to brannmurer mellom prosess- og bedriftsnettverk, noe som øker sikkerheten.*



Figur 7.8: *Figuren viser en mulig segmentering av et industrielt nettverk ved hjelp av brannmurer mellom de forskjellige segmentene samt DMZ-er for å kunne begrense trafikken som trenger å gå gjennom barrierene. Figuren er i tråd med den oppdeling gitt i «Løkmodellen» fra kapittel 7.3 og fargene til de forskjellige delsystemene harmoniserer med de gitt i «Løkmodellen», figur 7.2.*

7.7 Deteksjonssystem for inntrenging (IDS)

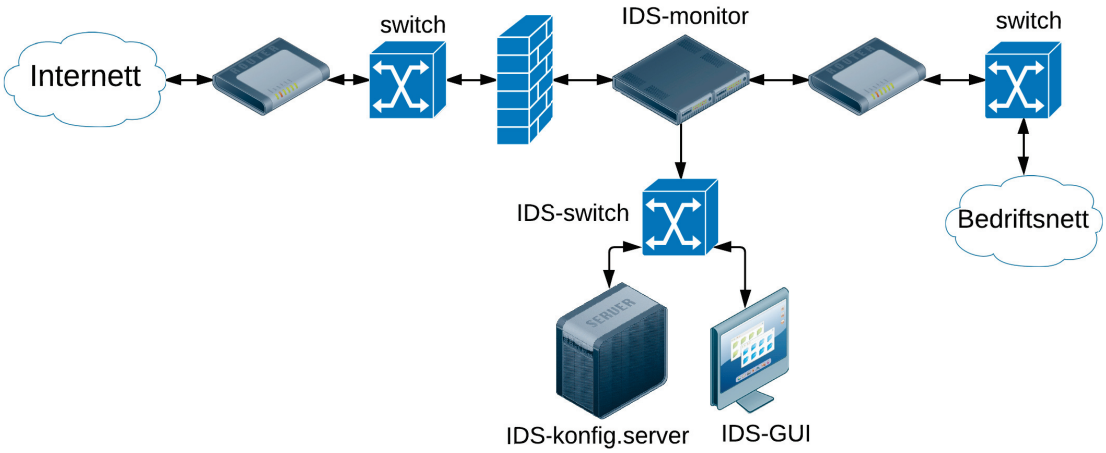
7.7.1 Innledning til IDS

Deteksjonssystemer for inntrenging, IDS (*en: Intrusion Detection System*), er systemer og prosedyrer for å detektere, og reagere på inntrenginger, og for å hjelpe nettverket tilbake til normaltilstand. Systemer for deteksjon av inntrenging har vært i bruk siden 90-tallet, og sammenlignes gjerne med boligalarmer, hvis oppgave både er å være et preventivt tiltak ved å skremme bort tyver samtidig som det er et reaksjonært system dersom det skulle skje et innbrudd. IDS-er bidrar også til å forbedre «beskyttelse-i-dybden» på nettverket som er diskutert nærmere i kapittel 7.3.

IDS består av flere delsystemer som for eksempel:

- Mekanismene for å overvåke trafikken på nettverket.
- Systemer som benyttes for logging og konfigurering av IDS-en.
- Systemer for grafisk fremstilling av logger eller alarmer.

En illustrasjon på hvordan en IDS kan kobles inn i et nettverk samt IDS-ens forskjellige delsystemer kan sees i figur 7.10.



Figur 7.10: *Figuren viser hvordan en IDS kan kobles inn i et nettverk, samt hvilke delsystemer som benyttes for å realisere IDS-en. Monitoren leser av trafikken som går gjennom enheten, mens konfigurasjonsserveren lagrer og analyserer dataene som går gjennom monitoren. IDS-GUI brukes til å presentere data fra IDS-serveren som alarmer eller tilsvarende.*

7.7.2 IDPS

IDS-er kan kombineres med rutiner for å unngå eventuelle angrep og går da under betegnelsen IDPS (*en: Intrusion Detection & Prevention System*). NIST[8] har i [44] laget en liste over fordeler ved å implementere IDPS-er i industrielle nettverk:

1. IDPS (og IDS) øker risikoen ved et innbrudd for angripere siden sannsynligheten for å bli oppdaget eller tatt øker.
2. Systemene kan detektere angrep som ikke ville blitt oppdaget med andre sikkerhetsmekanismer. Dette blir for eksempel gjort ved å se etter mønster i datatrafikken som indikerer angrep.
3. IDPS kan detektere nettverksaktivitet som er karakteristisk før angrep. Et eksempel på slik trafikk kan være at noen skanner nettverket for åpne porter.
4. IDPS kan bruke data logget fra tidligere angrep for å kartlegge truslene mot organisasjonen. Dette kan hjelpe organisasjonen å finne ut av hvor komplekst dette trusselbildet er, hvilken informasjon angriperne er ute etter og muligens avdekke hvem angriperne er.
5. IDPS kan benyttes som kvalitetskontroll på sikkerhetsløsningene til organisasjonen. Dette gjøres for eksempel ved å gjennomgå loggene fra IDS-en for å kartlegge om sikkerhetsmekanismer har blitt forbigått av angriperen.
6. IDPS kan lettere samle diagnosedata som senere kan benyttes til forbedringer av nettverket.

Dersom IDPS detekterer angrep kan den terminere tilkoblingen eller blokkere for innkommende data hos angrepets mål. Dersom dette ikke lar seg gjøre så kan IDPS-en rekonfigurere brannmurinnstillingene for å stenge angrepet ute. Det er vanlig praksis at IDPS-en i tillegg varsler nettverksadministratorene til systemet om at et mulig angrep kan være i gang.

Det finnes flere metoder IDS-er eller tilsvarende mekanismer kan benytte for å bestemme om det forekommer et angrep mot organisasjonen. En vanlig metode er å sammenligne trafikken mot trafikk forbundet med angrep. Andre alternativer er å sammenligne aktiviteten gjennom IDS-en med aktiviteten som er vanlig på nettverket. Et stort avvik fra den «normale» aktiviteten kan da være en indikasjon eller bare et faresignal på at noe unormalt foregår på nettverket, og dette kalles derfor *abnormalitetsbasert deteksjon*. Eksempler på størrelser som kan måles i

[8] National Institute of Standards and Technology `http://www.nist.gov/index.html`

abnormalitetsbasert deteksjon kan være datatrafikk, antall innloggingsforsøk fra en bestemt bruker eller ressursbruk av spesielle tjenester eller brukere.

En siste og tredje metode som kan benyttes til å detektere angrep er *tilstandsspesifikk kommunikasjonsprotokollanalyse* hvor leverandøren av systemet lager en profil over den type trafikk som skal sendes over nettverket med den aktuelle kommunikasjonsprotokollen. Trafikk som ikke passer inn i denne profilen vil så kunne brukes til å utløse en alarm i IDS-en [35].

7.7.3 NIDPS

Mens IDS-er kobler seg direkte på trafikkstrømmen kan noen typer IDS-er, NIDPS (*en: Network Intrusion Detection & Prevention System*), kobles inn på nettverket på tilsvarende måte som annet nettverksutstyr. NIDPS-ene gir en fordel i og med at de er passive deltagere i nettverket, og dermed lettere kan inkluderes i et eksisterende nettverk. NIPDS-er kan lettere overvåke bestemte segmenter og det er mindre sannsynlighet for at disse systemene blir oppdaget av angripere da NIDPS-er er vanskeligere å oppdage for nettverksskannere enn vanlige IDPS-er.

Problemet med NIDPS kontra IDS eller IDPS, er at NIPDS-er kan bli satt ut av spill dersom de er underdimensjonerte i forhold til mengden trafikk på nettverket. Dette vil kunne være et problem under nettstorm eller tilsvarende feiltilstander hvor nettverkstrafikken øker drastisk. En eventuell løsning kan være å bruke mekanismer som dynamisk fordeler trafikken over flere NIDSP-er [35], men dette øker både kompleksiteten og vedlikeholdskostnaden til systemene.

NIDPS-er kan ikke uten videre analysere krypterte pakker og kan ha problemer med ufullstendige pakker, noe som kan føre til at de ikke kan brukes i nettverkssegmenter hvor man benytter kryptering. Et generelt problem for IDS, IDPS og NIDPS er at de krever aktiv drifting for å være effektive, og følsomheten må justeres over en lengre innkjøringsperiode slik at en unngår for mange falske alarmer [35].

7.7.4 Honningkrukker

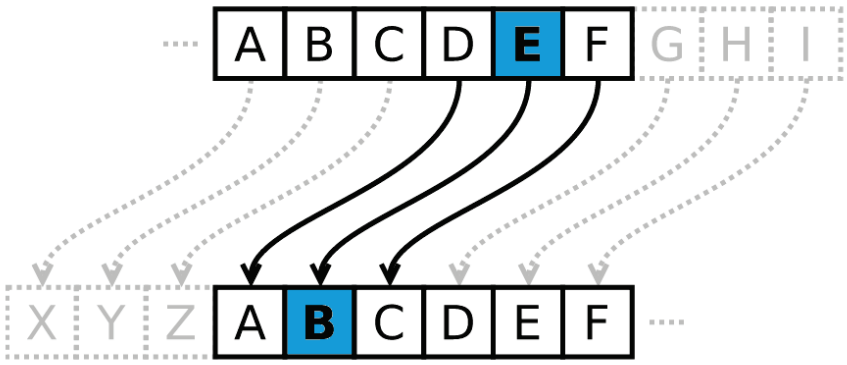
Som en del av inntrengingssystemet kan man benytte seg av såkalte honningkrukker (*en: honey pots*). Disse er designet for å fremstå som attraktive mål inne i organisasjonens nettverk, men er egentlig veldig godt overvåkede feller laget for å finne ut mest mulig om angriperen og metodene som benyttes. Videreutviklingen av honningkrukker kalles «tap and trace», hvor det i tillegg til overvåking

av angriperens aktivitet inne i honningkrukken blir forsøkt å holde angriperen inne i honningkrukken mens man sporer opp angriperens lokasjon.

7.8 Kryptering

Det kan ved mange anledninger være ønskelig at informasjon som utveksles mellom to eller flere parter ikke skal være mulig å lese for en tredjepart, og i disse situasjonene kan kryptering være løsningen.

Kryptering har sine røtter langt tilbake i tid og en av de kjente tidligere krypteringsalgoritmene er Caesarchifferingen. Denne krypteringen ble blant annet benyttet under Julius Ceasars tid for å kryptere meldinger slik at de ble uleselige for fienden. Ceasarchifferingen er en svært enkel, men til sitt bruk en effektiv algoritme. Hvordan denne krypteringen funger er vist i figur 7.11. Ceasarchifferingen forskyver alle bokstavene et gitt antall plasser i alfabetet, og dekrypteringen skjer ved å flytte bokstavene i den krypterte meldingen tilsvarende antall plasser tilbake i alfabetet.



Figur 7.11: *Figuren viser en av de tidlige krypteringsalgoritmene, Ceasarcifferingen [21]. Algoritmen baserer seg på at bokstavene i byttes ut med bokstaver som er flyttet et gitt antall plasser i alfabetet. Som kan sees i figuren, vil for eksempel bokstaven «D» bli byttet ut med bokstaven «A», bokstaven «E» med «B» og så videre. Bildet er hentet fra Wikipedia.*

Siden Ceasarcifferingens tid har krypteringsalgoritmene blitt vesentlig mer komplekse og dette skyldes både at maskinvaren har blitt så god at enkle krypteringer lett kan knekkes ved bruk av «brute-force»-metoder[9] samt at mange krypterings-

[9]Brute-force: Alle muligheter prøves ut. Krever svært mange operasjoner i en datamaskin, men kan gjøres raskt på en moderne maskin dersom algoritmene er svake.

metoder har blir knekt og har allment kjente løsninger. Den økende mengden tilgjengelig datakraft har heldigvis også muliggjort den krevende oppgaven det er å kryptere og dekryptere meldinger som bruker avanserte algoritmer.

Manglende kryptering kan være svært risikabelt i sikkerhetkritisk kommunikasjon da dette kan medføre at en tredjepart kan lese, sende og modifisere datapakker. Dette kan potensielt være farlig for organisasjonen da slik informasjon kan benyttes til å få innsyn i programvare, organisasjonens interne informasjon, kommunikasjonsprotokoller, nettverkstopologier og komponenter som inngår i nettverket [38]. Ved bruk av denne informasjonen kan en potensielt angriper mye lettere tilpasse og spisse sitt angrep mot systemets svakheter.

7.8.1 Prosedyrer for kryptering

For å få best mulig kryptert kommunikasjonen bør algoritmene som velges være nøye utprøvd og dokumentert sikre. Videre bør arkitekturen for systemet som utfører krypteringen være designet på en slik måte at krypteringsalgoritmene er modulære. Dette innebærer at de uten større komplikasjoner kan byttes ut dersom sikrere algoritmer dukker opp eller dersom den benyttede algoritmen blir knekt [38].

Det er også sentralt at organisasjonen har gode rutiner rundt håndteringen av krypteringsnøkler slik at ikke noen kan få tak i disse. En situasjon hvor noen har en gyldig krypteringsnøkkel uten at dette er kjent er potensielt mer skadelig enn ingen kryptering i det hele tatt da en slik situasjon vil medføre en falsk trygghet [38].

7.8.2 Problemer med kryptering

Et av de sentrale problemene rundt kryptering i industrielle nettverk er at disse nettverkene gjerne transporterer tidskritisk kommunikasjon. Da både kryptering og dekryptering av trafikk kan være tidkrevende om det brukes avanserte algoritmer, er det ikke alltid mulig å benytte seg av så avanserte krypteringsalgoritmer som ønskelig [38].

Da algoritmer fra tid til annen blir knekt, er det viktig at noen i organisasjonen følger med på nyheter innen fagfeltet, og at algoritmen raskt blir revurdert dersom noen finner svakheter i algoritmen. Dersom dette ikke blir prioritert er det lett at algoritmene blir for svake til å gi tilstrekkelig beskyttelse.

Kryptering fjerner også mulighetene til enkelt å kunne logge trafikken på nettverket, en viktig kilde til informasjon i arbeidet med å vedlikeholde og avdekke feil eller innbrudd i nettverket [38].

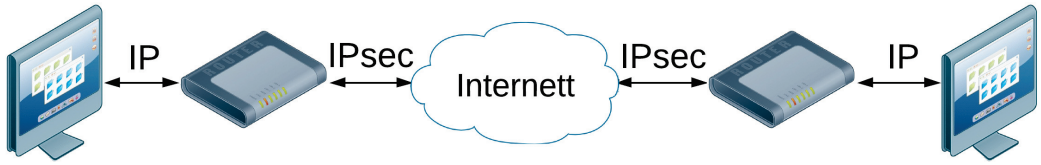
Uavhengig av hvor god krypteringsalgoritmen implementert i kommunikasjonen er, vil systemet med kryptering aldri være sikrere enn endepunktene. Man kan for eksempel oppleve et «mann-i-midten»-angrep (se kapittel 5.2.5) hvor et av endepunktene blir kompromittert, og krypteringen vil være nytteløs da angriperen kan lese av datatrafikken før den er kryptert eller etter at den er dekryptert. Dette gjør at kryptering ikke fjerner behovet for andre former for sikkerhetsmekanismer som er diskutert i denne oppgaven.

7.9 Virtuelle privatnettverk (VPN)

VPN, virtuelle privatnettverk (*en: Virtual Private Network*) er kryptert «punkt-til-punkt»-kommunikasjon mellom to lokalnettverk, og visualiseres gjerne som en tunnel gjennom det åpne Internett for kun din datatrafikk.

Alternativet til VPN-tilkoblinger er dedikerte «punkt-til-punkt»-kommunikasjonslinker, men dette er svært kostbart og vanskelig å realisere over lange avstander, så VPN er en mye billigere løsning. VPN pakker datapakken inn i en IPsec-pakke (IPsecurity-pakke) [21] som er en vanlig IP-pakke hvor innholdet (både hode og data) har blitt kryptert og pakket inn med et nytt IPsec-hode som blant annet inneholder detaljer rundt krypteringen og sjekksummer for å sjekke at pakken ikke har endret seg. For å få best mulig sikkerhet benytter IPsec seg av flere forskjellige krypteringer [44].

Innpakkingen av IP-pakker til IPsec-pakker kan enten gjøres ved hjelp av maskinvare (for eksempel rutere med innebygget VPN-funksjonalitet) eller ved bruk av programvare, en enkel figur over hvordan kommunikasjonen mellom to forskjellige lokalnettverk blir med VPN og IPsec kan sees i figur 7.12.



Figur 7.12: *Figuren viser VPN-koblingen mellom to forskjellige lokalnettverk. IP-pakkene blir i dette tilfellet pakket om til IPsec-pakker ved bruk av rutere med VPN-funksjonalitet. På mottaker-siden blir IPsec-pakken dekryptert og konvertert tilbake til en vanlig IP-pakke før den blir videresendt til mottaker.*

7.9.1 Fordeler og problemer ved bruk av VPN

VPN har flere sikkerhetsmessige fordeler sammenlignet med å sende vanlige IP-pakker. Det vil blant annet ikke være mulig å lese av attributter som sender- og mottakeradresse, hvilken kommunikasjonsprotokoll som benyttes eller datainnholdet i pakken. Det vil heller ikke ha noen hensikt å gjennomfør et «mann-i-midten»-angrep med VPN-kommunikasjon dersom angriperen ikke har krypteringsnøkkelen da IPsec-pakker har integritetssjekker som gjør at mottakeren vil være i stand til å undersøke om pakken har blitt endret (spoofet) under sendingen [44].

Det er lett å se at VPN kan være svært hensiktsmessig å benytte mellom for eksempel en operasjonssentral og en servicetilbyder, men dette krever en svært grundig konfigurering og testing, da gode oppsett av VPN-kommunikasjon med tilstrekkelig sikkerhet er krevende å få til. Det finnes mange gode hjelpemidler organisasjonen kan benytte for å implementere sikker VPN, et eksempel er IEC18028-5, «Informasjonsteknologi - sikkerhetsteknikker, Del 5: Sikker kommunikasjon over nettverk ved bruk av VPN» (se tabell 7.2).

Innføringen av VPN i en organisasjon bør gjøres i kombinasjon med innføring av rutiner som tvinger brukere som skal sende data ved hjelp av VPN til å autentisere seg. Autentisering kan for eksempel gjøres ved bruk av smartkortlesere, fingeravtrykk eller ekstra passord.

Det bør også settes strenge krav til IT-sikkerheten i organisasjonene som skal bruke VPN. Spesielt viktig er gode systemer og rutiner for å oppdage virus på utstyr koblet til nettverket da virus potensielt kan bli kryptert og innkapslet i IPsec-pakker og sendt inn i et annet nett. Faren med dette er at IPsec-pakkene

potensielt kan komme seg forbi noen av organisasjonen tradisjonelle sikkerhetsmekanismer da VPN-kommunikasjon er ansett som sikker [44].

7.10 Hardening

Begrepet systemhardening eller bare hardening går ut på at systemer kun skal kunne brukes til tiltenkte oppgaver og dette prinsippet harmoniserer blant annet med de retningslinjer gitt i OLF104 (#6). Hardening gjør det lettere å holde systemene sikre da færre ting potensielt kan gå galt.

Det finnes en rekke tiltak som kan gjennomføres for å gjøre systemene «hardere», og noen av de mest sentrale vil nå bli gjennomgått. Et sentralt tiltak er å fysisk hindre tilkobling av utstyr til systemene. For å oppnå dette kan eventuelle porter for tilkobling av fysisk utstyr som USB-kontakter og overflødige nettverksporter kobles ut. Et mer ekstremt, men effektivt tiltak er plassere alt utstyr i låste skap. Slike tiltak reduserer trusler som for eksempel minnepinner viste seg å være mot industrielle systemer etter Stuxnet.

Andre tiltak kan være å fjerne programmer som nettlesere, e-postklienter og Office-programmer fra disse systemene. Det er også anbefalt å deaktivere alle tjenester og programmer i operativsystemet som ikke benyttes på maskinene da dette kan redusere potensielle sikkerhetshull i systemet.

Det bør også sperres for at alle andre enn administratorer kan gjøre endringer i systeminnstillingene eller installere ny programvare på maskiner. Hardening kan med disse tiltakene hjelpe til å redusere risikoen for at interne aktører kan klare å få gjort skade på systemet.

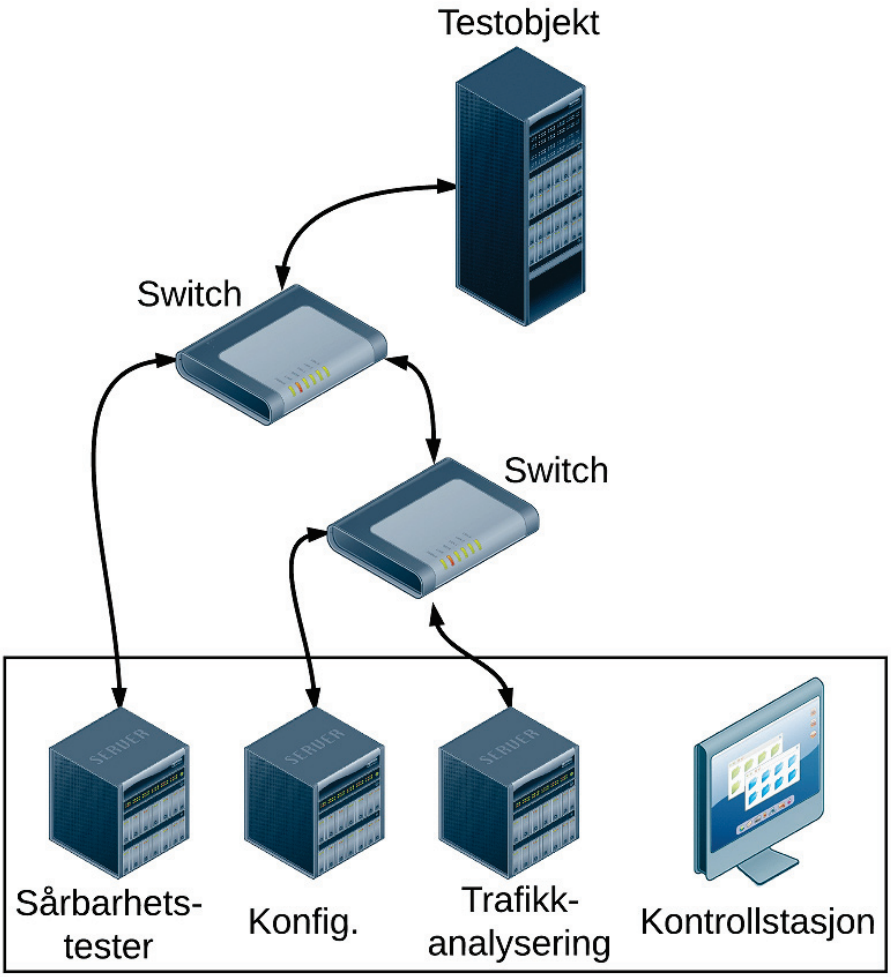
For å hjelpe systemadministratorer finnes det sjekklister fra forskjellige sikkerhetsinstanser over hvilke tjenester som bør deaktiveres, avhengig av hvilket operativsystem som benyttes og hvilke tjenester man trenger.

7.11 Testing av komponenter og systemer

Som nevnt i kapittel 5.2.3, så kan nettverksadministratorer benytte seg av forskjellige verktøy for å analysere blant annet sikkerhet og robusthet i nettverket. Tester vil kunne avsløre svakheter i systemet som sårbarheter, åpne nettverksporter, installerte trojaner og manglende robusthet mot nettstorm.

7.11.1 TOCSSiC

Da systemene skulle testes for svakheter hos CERN ble TOCSSiC (*en: Teststand On Control System Security in CERN*) utviklet [41]. TOCSSiC benytter lett tilgjengelig maskinvare som høyhastighets-switcher og PC-er for å realisere et testoppsett som kan brukes til å teste et system for svakheter. Oppsettet som ble bruk hos CERN i [23], kan sees i figur 7.13, og består av tre datamaskiner og to høyhastighets nettverksswitcher samt en maskin for å styre testen.



Figur 7.13: *Skisse over maskinvareoppsettet av TOCSSiC for å finne svakheter i systemer. Brukeren kan konfigurere hvilke tester som skal kjøres på konfigureringsmaskinen, en maskin benyttes til analyse av trafikken på nettverket og en siste maskin til å kjøre de faktiske testene. To høyhastighets-switcher binder alt sammen.*

TOCSSiC består av flere moduler laget for å teste forskjellige aspekter av sikkerheten i systemet [41]:

- Nessus: Modul for å teste systemet for svakheter. Nessus er bygget opp av over 14 000 tilleggsmoduler [41], som hver og en tester forskjellige typer svakheter. Testene utført med Nessus kan enten være bredspektret eller spisses inn mot spesielle typer svakheter.
- Nmap: Nmap[10] er kanskje den mest kjente nettverksskanneren som finnes, og inngår som en modul i TOCSSiC. Nmap brukes til å skanne nettverk for å avdekke maskiner bak en eller serier av IP-adresser. Etter å ha skannet de utvalgte IP-adressene returnerer Nmap detaljer som hvilke applikasjoner som blir kjørt på de forskjellige maskinene, hvilket operativsystem systemene kjører, hvilken type brannmurer som er i bruk, hvilke nettverksporter som er åpne og hvilke som er strengt [41]. En skjermdump av hvordan Nmap ser ut kan sees i figur 7.14.
- Netwox: Netwox er som Nessus en verktøyboks med forskjellige verktøy for å teste over 200 ulike problemområder[11] i nettverk. Netwox kan for eksempel brukes til sniffing og spoofing, men benyttes i TOCSSiC spesielt til å utføre nettstorm. I TOCSSiC går testprosedyrer for nettstorm ut på å kjøre verktøy 74 i én time, og etter nettstormen er ferdig verifiseres det om målet fremdeles lever ved å bruke en vanlig «ping»-kommando. PLS-er som responderer på dette pinget vil passere testen.
- Ethereal: Ethereal er et gratisprogram for å overvåke og analysere trafikk på et nettverk. I TOCSSiC benyttes Ethereal for å overvåke trafikken mellom PC-ene og målet i testen. Disse dataene benyttes senere for å sette sammen resultatene fra de forskjellige testene.

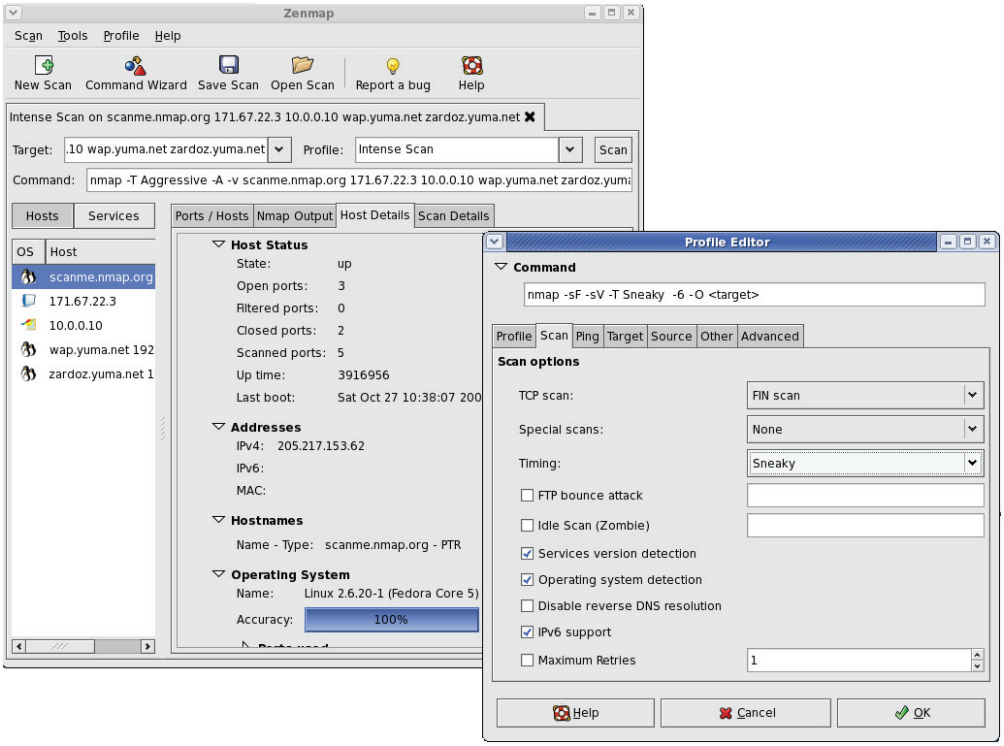
7.11.2 CSET

Testprogramvaren CSET (*en: Cyber Security Evaluation Tool*) kommer fra Department of Homeland Security, og gjort fritt tilgjenglig[12] for å hjelpe nettverksadministratorer med å bestemme sikkerhetsnivået til sine industrielle system. Programvarens grunnleggende funksjonalitet er å kunne bestemme sikkerhetsnivået til systemene gjennom å stille mange detaljerte og generelle spørsmål om syste-

[10] `http://nmap.org/`

[11] Uttømmende liste over alle de tilgjenglige verktøyene i Netwox: `http://www.vulnerabilityassessment.co.uk/netwox.htm`

[12] `http://ics-cert.us-cert.gov/Downloading-and-Installing-CSET`



Figur 7.14: *Dialogvindu i nettverksskanneren Nmap. Til høyre sees oppsettet av skann, mens resultatet sees i det venstre vinduet. Her kan man blant annet se åpne porter, adresser, tjenester og operativsystem fra maskinene som er skannet.*

mene og prosedyrene rundt disse. Når alle spørsmålene har blitt besvart, sammenlignes svarene med forskjellige standarder og retningslinjer innen fagfeltet, for å så gi en tilbakemelding på hvordan systemet harmoniserer med disse. Responsen fra programvaren kategoriserer avvikene etter type og viktighet, og kommer med råd om hvordan sikkerheten kan forbedres.

Fordelen med CSET fremfor for eksempel TOCSSiC er at programvaren kun baserer seg på spørsmål som kan besvares uten at man trenger spesielt utstyr, nedetid eller testoppsett.

7.11.3 CRIOP

CRIOP (*en: Crisis Intervention in Offshore Production*) er en sjekklistebasert metode for å teste sikkerhetsnivået i blant annet kontrollrom offshore og CRIOP ble publisert første gang tidlig på 90-tallet [18].

Etter den første utgaven av CRIOP har metoden blitt videreutviklet i samarbeid

med industrien, og metoden omfatter i dag spørsmål rundt flere av de sårbarheter som kan innføres med IO. Spørsmålene CRIOP stiller rundt IO deles inn i følgende fire hovedtemaer [24]:

- Hvordan de forskjellige aktørenes forståelse er for organisasjons mål og visjoner.
- Hvordan det blir tilrettelagt for å oppnå en best mulig endringsprosess.
- Hvordan det tilrettelegges for å oppnå en god situasjonsforståelse på tross av den fysisk avstanden mellom on- og offshore.
- Hvilke tiltak som gjøres rund håndteringen av informasjonssikkerheten i IO.

Benyttes CRIOP under design av nye systemer anbefales det at metoden benyttes flere ganger under designprosessen slik at feil kan lukes bort underveis. Blant annet bør metoden brukes under det konseptuelle og detaljerte designfasen samt etter at systemet har blitt satt i drift.

Resultater fra testing av metoden i [24] peker mot at CRIOP fyller et tomrom og at metoden kan hjelpe til å oppnå bedre sikkerhet både under design- og driftsfasen av system.

8 Dagens situasjon

Du kan ikke slippe unna morgendagens ansvar ved å unnvike den i dag.
— Abraham Lincoln

Dette kapitlet vil kartlegge hvordan dagens situasjon er med tanke på flere av temaene gjennomgått i foregående kapitler, samt diskutere litt rundt de forskjellige funnene gjort i forskjellige tilsynsrapporter.

Det vil først bli gitt en gjennomgang av kildene benyttet for å kartlegge dagens situasjon før det blir gitt en gjennomgang av hva disse artiklene sier om situasjonen innenfor følgende hovedtemaer: avhengigheter, angrep og systemenes sikkerhet.

8.1 Kilder benyttet for å kartlegge dagens situasjon

Her er en kort oversikt over kildene benyttet for å kartlegge dagens situasjon med tanke på sikkerhetsarbeidet i industrielle nettverk og systemer.

- **Uavhengighet av sikkerhetssystemer [12]**. Skrevet for SINTEF, 2009. Baserer blant annet sine tall på tilsyn i form av møter med operatører og leverandører av SAS fra fire installasjoner. Det har i forbindelse med denne artikkelen også blitt gjennomført en spørreundersøkelse som omfatter hele 46 installasjoner.
- **Uavhengighet av sikkerhetssystemer offshore - status og utfordringer [13]**. Skrevet for SINTEF, 2006. Inneholder blant annet status på koblinger i moderne SCADA-systemer, og oppsummerer resultater fra telefonintervjuer med fagpersonell fra forskjellige installasjoner.
- **In the Crossfire [3]**. Skrevet for McAfee, og gjennomgår bruk og status for kritisk infrastruktur i cyberkrigstider. Rapporten bygger på resultater fra 600 besvarelser[13] av sikkerheten i industrielle nett.
- **Common Cybersecurity Vulnerabilities in Industrial Control Systems [43]**. Skrevet av DHS, CSSP[14] 2011. 18 tilsyn av industrielle kontrollsystemer. De fleste av disse tilsynene er gjennomført i perioden fra 2004-2008, men artikkelen inneholder også noen tilsyn gjennomført i tidsrommet 2009-2010.

[13] 100 respondenter fra USA, 50 respondenter hva hver av de følgende landende: Japan, Kina, Tyskland, Frankrike, Storbritannia og Italia. 30 respondenter fra hver av de følgende landene: Russland, Spania, Østerrike, Brasil, Mexico og India. 20 respondenter fra Saudi-Arabia.

[14] Department of Homeland Security, Control Systems Security Program

- **Telefonintervju [40]** Det har i tillegg til de overstående tekster blitt gjennomført telefonintervju med personell fra en stor leverandør av SAS-systemer.

Det fremste ankepunktet mot disse artiklenes relevans, er at de ikke utelukkende omhandler den norske oljesektoren, men inkluderer respondenter fra flere forskjellige land og sektorer. Dette kan tilsi at resultatene ikke trenger å være representative for det arbeidet som blir gjort innen norsk oljesektor. Selv om litteraturen på enkelte punkter viser til sprikende resultater, blir det generelt observert samsvar mellom de forskjellige artiklenes funn. Dette er naturlig da alle systemene omtalt er sikkerhetskritiske systemer, som i bunn og grunn har mange av de samme kravene.

Da det er en krevende jobb å utføre tilsyn og undersøkelser er flere av artiklene noen år gamle. Dette kan være et problem da utviklingen innen feltet går relativt fort, noe som gjør at tallene ikke nødvendigvis gjenspeiler den virkeligheten man har i dag.

En grunn til at det er vanskelig å avdekke hvilke sikkerhetsmekanismer organisasjonene benytter seg av er at «security by obscurity»-prinsippet benyttes aktivt. Dette prinsippet går ut på å beskytte systemet ved at så få som mulig kjenner hvilke sikkerhetsmekanismer og strukturer som blir benyttet i organisasjonen. Det kom fra [40] frem at dette er et mye brukt prinsipp, og at mange systemer ville vært relativt enkle å knekke dersom man hadde hatt kunnskap om sikkerhetsmekanismene benyttet.

8.2 Avhengigheter mellom delsystemer

Det ble i kapittel 2.2.4 gjennomgått hvordan SCADA-systemene deles opp i flere mer eller mindre uavhengige delsystemer, og kravene til uavhengighet mellom disse delsystemene ble presentert i kapittel 7.1.

Dette delkapittelet vil gjennomgå situasjonen vedrørende avhengigheter mellom disse delsystemene, og det blir i [13] i flere av tilsynene funnet avhengigheter mellom delsystemer, disse avhengighetene vil nå bli beskrevet nærmere.

- **Koblinger mellom PCS og PAS.** Det ble i flere av intervjuene i [13] avdekket av det i mange tilfeller finnes koblinger mellom PCS og PAS.
  - PAS skal i utgangspunktet være et separat system, men blir av flere aktører sett på som en del av eller som en utvidelse av PCS, og dette synspunktet benyttes til å forsvare koblingen.

  - Det ble funnet tilfeller hvor delsystemene benytter samme programvare og nettverk, noe som skaper avhengigheter mellom systemene.
  - Det ble i enkelte tilfeller funnet at noe av datatrafikken gikk feil vei mellom systemene (PCS → PAS).

- **Koblinger mellom PAS og NAS.**
  - Funksjonalitet som skal være separat for NAS-systemet hadde blitt lagt innunder PAS.
  - Det ble observert at noe datatrafikk gikk feil vei mellom delsystemene (PAS → NAS).
  - Det kom gjennom intervjuene i [13] frem at NAS i enkelte tilfeller benyttet felles kommunikasjonskanal eller felles strøm med PAS.

- **Felles operatørstasjoner for delsystemene**
  - De ulike delsystem opereres enten fra samme fysiske kontrollstasjonen eller fra kontrollstasjoner som baserer seg på den samme programvaren.
  - De samme delene av operatørstasjonene brukes til mange forskjellige oppgaver slik som: å sette verdier, lese av verdier eller å vise alarmer.
  - Det ble under tilsynene gitt tilbakemeldinger om at det var vanskelig å holde oversikt over hvor kritisk inn- og utdata fra systemene tok veien. Dette skyldtes i stor grad at mange av disse signalene ble sendt ut på store nettverksbusser, i motsetning til de gamle «punkt-til-punkt»-kablene hvor hvert delsystem eller hver komponent hadde sin dedikerte link inn mot kontrollrommet.

Det er generelt anbefalt, og i enkelte tilfeller også påkrevd, at sikkerhetskritiske delsystemer har tilstrekkelig uavhengighet fra hverandre, men det er ikke naturligvis lagt føringer på hvor uavhengige disse systemene trenger å være. I mange tilfeller vil for eksempel felles kommunikasjonskanal eller felles strømforsyning mellom forskjellige delsystemer medføre sterke avhengigheter, men dette er ikke nødvendigvis alltid tilfellet. Dersom det er satt i gang tilstrekkelige sikkerhetstiltak slik som redundante kommunikasjonskanaler eller tilstrekkelig robusthet i de enkelte enheter, vil dette kunne resultere i en tilfredsstillende uavhengighet mellom systemene. Det ble for eksempel gjennom et telefonintervju [40] avdekket at denne leverandøren bevisst benyttet samme kommunikasjonskanal både for PCS og SIS, og at uavhengigheten ble ivaretatt av enheter som var robuste og benyttet en sikker kommunikasjonsprotokoll.

8.3 Dagens situasjon vedrørende angrep mot industrielle nettverk

I dette delkapittelet vil det bli gitt en kort gjennomgang av situasjonen vedrørende angrep mot industrielle nettverk.

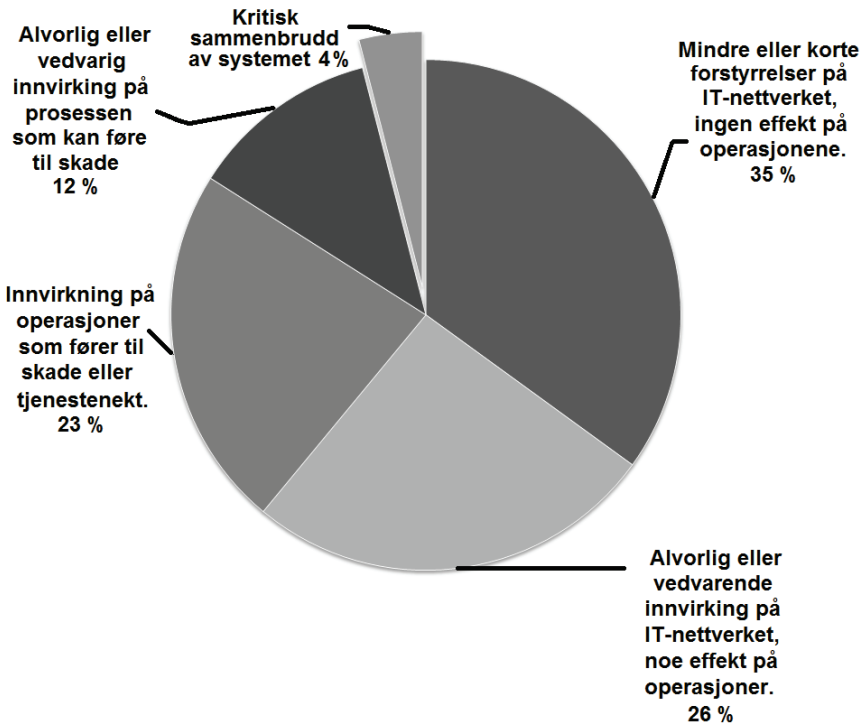
Nettstorm

Nettstorm, beskrevet i kapittel 5.2.2, er et utbredt problem i SCADA-systemer og i de omfattende tilsynene gjennomført i [3] svarte hele 54 % av alle spurte at deres systemer hadde blitt utsatt for nettstorm. De som hadde blitt utsatt for nettstorm ble så bedt om å kategorisere konsekvensene av nettstormene i fem forskjellige alvorlighetsgrader. Resultatene kan sees i figur 8.1.

En annen indikasjon på SCADA-systemenes robusthet mot nettstorm finnes i [23], hvor det i 2006 ble gjennomført kontrollerte tester på CERNs SCADA-system for å blant annet teste systemenes motstandsdyktighet mot intern nettstorm. Testene ble gjennomført ved hjelp av TOCSSiC-verktøyet (kapittel 7.11.1) og resultatene fra denne testen viste at hele 26 % av alle PLS-ene i testen sluttet å respondere etter angrepet. Uttalelser gitt i [12] om at SAS-systemene brukt i mange av dagens installasjoner verken er testet eller sertifisert for å takle den mengden datatrafikk som teoretisk kan sendes på moderne nettverk indikerer en grunn til systemenes svikt under nettstorm-testingen.

Nettstorm er derimot ikke et like stort problem for alle moderne systemer, og under telefonintervju [40] kom det frem at eksterne nettstormangrep ikke var veldig kritisk for denne leverandøren. Dette skyldtes at de nye SAS-installasjonene denne leverandøren leverte benyttet seg av komponenter som er tredjepartsertifisert for å kunne håndtere forskjellige kjente typer nettstorm, og et angrep vil derfor ikke kunne påvirke prosessen i like stor grad. Dette gjelder derimot ikke for anlegg hvor store deler av systemene består av eldre komponenter som ikke er tilpasset moderne nettverkstrafikk.

Det kom frem i de samme telefonintervjuene [40] at de farligste nettstormene for systemet er de som oppstår internt under en eventuell feiltilstand da datamengden generert fra slike feil kan bli svært stor.



Figur 8.1: *Figuren viser alvorlighetsgradene av nettstormangrepene som hele 54 % av de spurte i [3] svarte at de hadde blitt utsatt for.*

Inputvalidering

Inspeksjonene i [43] slo fast at hele 47 % av alle sårbarhetene i kontrollsystemene skyltes dårlig inputvalidering, noe som kan føre til blant annet bufferoverflyt eller kommandoinjeksjoner. Fra tilsynene gjennomført i perioden 2004-2010 stod disse feilene for 29 % av de totale feilene, mens de i perioden 2009-2010 stod for hele 45 % av alle feil, noe som viser en økende trend av feil i systemene på grunn av dårlig inputvalidering.

Disse funnene kan enten peke mot at den økende kompleksiteten til systemenes programvare skaper flere slike feil, men det kan også skyldes at de som utfører tilsynene har blitt mer observante på slike feil eller at andelen feil fra andre deler av systemet har blitt lavere.

Virus

Det blir i [3] undersøkt om eierne av systemene har opplevd noen form for virus eller tilsvarende angrep mot SCADA-systemet i organisasjonen, og hele 89 % bekreftet at deres systemer en eller flere ganger hadde blitt infisert av slik programvare. Selv om svært mange responderte at det hadde forekommet ondsinnet programvare i deres systemer, vises det i [12] til at det i flere tilfeller ikke eksisterte programvare eller rutiner for å detektere og hanskes med virus i organisasjonen.

Et av de store problemene med disse tallene er at alle er funnet i tilsyn gjennomført før Stuxnet brøt ut. Dette angrepet beviste for alle at også industrielle systemer kan være mål for ondsinnet programvare, og det er naturlig å anta at flere har blitt observante på disse farene etter dette angrepet.

Tilsynene sier heller ikke noe om hvilke systemer som ble utsatt for virusene, noe som betyr at disse kan ha forekommet på mindre kritiske systemer som ligger utenfor noen av organisasjonens sikkerhetsmekanismer og dermed ikke nødvendigvis utgjør en like stor trussel mot sikkerheten i systemene.

8.4 Status for sikkerhetsmekanismene til systemene

Dette delkapitlet vil ta for seg i hvilken grad organisasjonene benytter seg av sikkerhetsmekanismene beskrevet i kapittel 7.

Hardening

Hardening er beskrevet i kapittel 7.10 og det er i [3] stilt flere spørsmål til systemeierne som handler om hvorvidt organisasjonen arbeidet for å oppnå dette. Av respondentene svarer kun 19 % at det er gjennomført systemhardening på SCADA eller IT-systemene i organisasjonen. I disse undersøkelsene svarte rundt 1/3 at organisasjonen hadde gjennomført tiltak for å forhindre tilkobling av USB-minnepinner eller tilsvarende utstyr.

Hardening er også tatt opp i [43] og det blir her påpekt at flere av installasjonene hvor det ble gjennomført tilsyn manglet fysisk sikring av utstyr og at for eksempel ledige nettverksporter både var tilgjengelige og aktive.

Resultatene viste altså at et klart mindretall av organisasjonene benyttet hardening i systemene, og det kan være flere grunner for dette. En grunn kan være

at det er tidskrevende arbeid å utføre hardening samtidig som at det vil medføre ekstraarbeid dersom det senere viser seg at noen har behov for deaktiverte tjenester eller tilkoblinger i systemene.

Noen leverandører benytter seg også av andre filosofier, hvor nytt utstyr kan kobles i ledige nettverksporter, men utstyret vil kun ha mulighet til å sende eller motta på nettverket dersom enheten benytter seg av en bestemt og sikker proprietær kommunikasjonsprotokoll.

Brannmurer

Bruken og rutinene rundt brannmurer blir tatt opp i flere av de nevnte artiklene, og det kommer blant annet frem i [43] at dårlige regler for hvilken trafikk som skal stoppes i brannmurene er den oftest forekommende svakheten i industrielle nettverk.

Brannmurer er viktige for å kunne oppnå god segmentering i nettverkene, men det ble i [12] avdekket at det vanligste er å kun benytte brannmurer mellom bedrifts- og prosessnettverket (PCS), mens SIS og PCS gjerne befinner seg på det samme usegmenterte nettverket.

I [3] blir det også avdekket at kun 77 % av alle spurte benytter tilstrekkelig sikre brannmurer mellom organisasjonens nettverk og Internett, og hele 76 % svarte at deres SCADA-system var koblet til Internett. Av disse 76 % svarte rundt halvparten, 47 %, at denne koblingen mot Internett kunne føre til sikkerhetsproblemer som det ikke var tatt hensyn til i organisasjonen.

Disse tallene fanger ikke nødvendigvis opp tilfeller der man har flere sikkerhetslag bak brannmuren, noe som kan rettferdiggjøre at man slipper gjennom litt mer trafikk enn det som er strengt nødvendig. Grunnen til de høye tallene funnet under tilsyn kan også komme av at mange tilsyn vil fokusere ekstra på brannmurer da disse tradisjonelt har vært svært viktige for å ivareta sikkerheten, og dette kan føre til at det blir funnet flere feil i disse komponentene.

Monitorering

Tallene for i hvor stor grad nettverkene og systemene blir monitorert spriker, i [3] svarer hele 92 % av de spurte at systemene blir monitorert, et tall vesentlig høyere enn det som ble svart under spørreundersøkelsen gjennomført i [12] hvor bare 37 % (17 av 46 installasjoner) svarer at trafikken blir overvåket.

Patching og oppgradering

Oppgradering og implementering av nye patcher i systemer blir i henhold til funn gjort i [3] bare gjennomført på en jevnlig basis av 43 % av de spurte.

Heller ikke operativsystemene som ligger i bunn for kontrollsystemene oppdateres jevnlig, og [12] påpeker at det sjeldent blir observert oppdateringer av disse systemene. Dette til tross for at sårbarheter i disse operativsystemene i [43] utgjør hele 24 % av de totale sårbarhetene.

Det finnes flere grunner til at oppdateringer av systemene ikke blir gjort så ofte som ønskelig, og et sentralt moment her kan være at organisasjonen ikke har gode rutiner for å gjennomføre oppdateringer. Dersom man bestemmer seg for å iverksette en oppdatering, er man også ofte prisgitt at leverandøren har gjort en grundig jobb med å teste denne slik at man ikke risikerer at oppdateringen fører til lengre nedetid enn nødvendig.

I tillegg vil oppdateringer gjerne føre med seg at en blir nødt til å rekonfigurere systemene med de endringene som har blitt gjort på systemet, noe som kan medføre betydelig ekstraarbeid for administratorene.

Robusthet

Flere installasjoner mangler en overordnet risikoanalyse av hele systemet slik som beskrevet i 7.4, og i [12] ble det funnet at kun fem av 46 installasjoner hadde gjennomført grundige risikoanalyser for systemene.

Det ble i [12] i samtlige tilsyn observert at løsningene for å rapportere og registrere systemfeil eller sikkerhetsbrudd i IT-systemene var dårlige eller manglende. Det kan være mange grunner til dette, men det er nærliggende å anta at det ikke er ønskelig å innrømme at systemet har en feil eller et sikkerhetshull.

8.5 Overordnet oversikt over dagens situasjon

Som det kan leses ut i fra de foregående delkapitler har systemene benyttet i dag mange sårbarheter som kan føre til flere sikkerhetsmessige problemer. Da IO vil øke kompleksiteten til systemene ytterlige, vil dette kunne føre til at sårbarhetene forverres eller at nye problemer kommer til.

Dagens situasjon viser at det arbeidet som blir gjort for å forbedre sikkerheten ikke nødvendigvis oppfyller verken de krav eller anbefalinger som blir gitt i sen-

trale standarder og retningslinjer, og dette kan tyde på at organisasjonene enten ikke følger eller kun følger deler av disse.

Det blir i [38] vist til at det generelt er dårlig kommunikasjon og kunnskapsdeling mellom prosess- og IKT-avdelingene. Det dårlige samarbeidet mellom disse fagfeltene kan skape problemer i sikkerhetsarbeidet, og dårlige rutiner for oppdateringer og dårlig konfigurerte brannmurer kan være resultater av dette.

Resultatene fra tilsyn viser at det er manglende testing av systemene både på delsystemnivå og helhetlig nivå. Dette kan blant annet bidra til at den andel av feil som har konsekvenser for prosessen og er forårsaket av nettstorm er så stor som den er. Tilstrekkelig testing av systemene kunne ha avdekket avhengigheter mellom delsystemer og dårlig inputvalidering, og bidratt til å unngått slike problemer i nye system.

9 Anbefalte tiltak

Det er én ting som er verre enn endring, og det er status quo.
— John le Carré

Som vist i foregående kapitler så finnes det et stort antall forskjellige tilgjengelige mekanismer for å sikre systemer mot angrep, men at utnyttelsen av og rutinene rundt disse mekanismene er ikke alltid så gode som de burde være, selv om organisasjonen i henhold til styringsforskriftens § 4 er nødt til å velge sikre tekniske løsninger.

Dette kapittelet vil ta for seg flere tiltak som forfatteren av denne oppgaven mener at organisasjoner vil ha mest nytte av sikkerhetsmessig, ved en innføring av IO.

9.1 Tverrfaglig samarbeid

Det første og viktigste tiltaket vil være å opprette en god gjennomgående sikkerhetskultur i hele organisasjonen. For å få til en slik sikkerhetskultur er man avhengig av at mange aktører klarer å spille på lag. De viktigste er organisasjonens ledelse, IT-avdelingen, prosessingeniørene, leverandørene av SAS-systemer og de regulatoriske myndighetene innen feltet. Disse faggruppene har ikke tradisjonelt hatt sterke tradisjoner for samarbeid [38], men innføringen av IO vil som nevnt i kapittel 3.2 føre til organisatoriske endringer som gjør at organisasjonen kan legge til rette for slike samarbeid.

Gevinsten med en slik tverrfaglig arbeidsgruppe er blant annet at man kan oppnå bedre forståelse for de forskjellige fagfeltenes krav til sikkerhet, samt at ledelsen kan få et godt innblikk i hvilke sikkerhetsmessige utfordringer organisasjonen står ovenfor.

9.2 Tiltak for å øke mengden oppdateringer

Med et godt etablert samarbeid mellom nevnte faggrupper vil man også kunne forbedre organisasjonens rutiner rundt oppdateringer av SAS. Oppdatering av disse systemene er svært viktig da flere av dem, i motsetning til IT-systemer, har lang forventet levetid (15-20 år [38]) og oppdateringene er derfor essensielle for at systemene skal kunne stå imot moderne angrep.

Det er viktig å få til et tett samarbeid mellom organisasjonen og leverandøren av SAS da dette kan resultere i at leverandøren kan produsere oppdateringer som er nyttige for organisasjonen, samt at organisasjonen kan få større tiltro til de oppdateringer som leverandøren leverer.

For å gjøre oppdatering enda lettere, bør strukturen på programvaren bygges opp slik at alle installasjonsspesifikke konfigurasjoner, samt konfigurasjoner utført av brukeren lagres i versjonsuavhengige filer. Ved å bruke en slik struktur vil man slippe å rekonfigurere systemene etter oppdateringer, og dette vil redusere arbeidet med, og dermed terskelen for å utføre oppdateringer.

9.3 Tiltak for å forbedre sikkerheten i brannmurer

Organisasjonen bør også utnytte samarbeidet mellom IT- og prosessavdelingene til å utarbeide bedre sikkerhet i brannmurene. Gjennom et slikt samarbeid kan prosessavdelingen definere hvilken trafikk som må gå gjennom brannmurene, mens IT-avdelingen kan implementere disse spesifikasjonene på en så sikker måte som mulig. Eksempler på spørsmål som kan benyttes av organisasjonen for å kartlegge hver enkelt brannmurs regler kan være:

- Hvilken trafikk og hvilke protokoller skal benyttes hver side av brannmuren?
- Hvilken trafikk må få lov til å slippe gjennom brannmuren?
- Hvilket nivå av sikkerhet er assosiert med trafikken på hver side av brannmuren?

Etter gjennomgangen av reglene for alle brannmurene er utført bør det utarbeides gode rutiner for hvor ofte brannmurenes logger skal gjennomgås. For å gjøre dette arbeidet så enkelt som mulig kan det benyttes en løsning hvor man bruker et system som automatisk samler inn logget data fra alle brannmurene og automatisk prioriterer hendelsene etter viktighet.

En slik gjennomgang av alle brannmurer kan også føre til en bedre nettverksstruktur, noe som er svært viktig for avgrensningen av hvor forskjellig typer trafikk får transporteres på nettverket, og kan øke sikkerheten.

9.4 Standarder og retningslinjer som et virkemiddel for å oppnå bedre sikkerhet

Resultatene fra kapittel 8 viser at sentrale retningslinjer som OLF104 og andre sentrale standarder ikke blir benyttet i det omfang som de burde, og dette kan føre til at problemer ikke blir løst på en tilfredsstillende måte.

Et tiltak som kan gjennomføres for å øke bruken av standarder og retningslinjer kan være å arrangere kurs i regi av de regulerende myndighetene som omhandler de tilgjengelige standarder og retningslinjer, og hvordan disse kan benyttes for å øke sikkerheten i systemene. Dette kan være et tiltak som stimulerer flere organisasjoner til å øke bruken av standarder uten at myndighetene trenger å regulere dette eksplisitt.

9.5 Bruk av IDS for å skape sikker kommunikasjon i nettverkssegmenter

For å kunne forbedre kontrollen av hvilken trafikk som sendes i bestemte segmenter av nettverket kan organisasjonen ta i bruk IDS-er med tilstandsspesifikk kommunikasjonsprotokollanalyse (se kapittel 7.7.2), hvor all trafikk som ikke passer inn i et bestemt regelsett vil utløse en alarm i IDS-en. En slik løsning vil kunne være lettere å realisere dersom IT-avdelingen og leverandøren av SAS-systemene har et godt samarbeid, da det typisk vil være leverandørens oppgave å definere settet med regler som benyttes i IDS-en. Ved bruk av et slikt oppsett vil organisasjonen kunne ha svært god kontroll på at det kun sendes godkjent trafikk i de mest kritiske segmentene i nettverket.

9.6 Robusthet og risikoanalyse

I dagens komplekse trusselsituasjon blir robustheten til systemene stadig viktigere for å få kunne realisere sikre systemer. For å øke robustheten vektlegger litteraturen, blant annet i [15], hvor viktig det er å gjennomføre fullstendige risikoanalyser av hele systemet, samt viktigheten med at denne analysen er en kontinuerlig pågående prosess. Ved bruk av en slik analyse kan organisasjonen stadig lære av feil som oppstår og kan etterhvert luke bort både disse og avhengigheter som ikke ble oppdaget under design eller i de første iterasjonene av risikoanalysen. Et slikt tiltak vil kunne føre til at systemet som helhet blir sikrere,

samt at analysen vil kunne føre til at organisasjonen opprettholder et fokus på sikkerhetsarbeidet.

I den sykliske risikoanalysen for å øke robustheten (beskrevet i kapittel 7.4), inngår det som en komponent at nye analyser må iverksettes om nye feil blir oppdaget. For at dette skal kunne utføres er det viktig at organisasjonen utarbeider gode rutiner for å granske hendelser eller ulykker relatert til sikkerhetsbrudd. Ved gode granskninger av slike hendelser vil kilden til problemet kunne avdekkes og problemet vil kunne utbedres.

9.7 Testing av systemet som helhet

Et moment som blir tatt opp både under tilsyn og nevnt i telefonintervju [40], er at mange av dagens eksisterende systemer fremstår som lappetepper med en blanding av gamle og nye komponenter og systemer. Problemet med dette er at komponentene ikke nødvendigvis er designet for å arbeide sammen eller til å brukes slik de faktisk blir brukt. For å være sikker på at systemene som helhet fungerer slik man forventer, er det viktig at organisasjonen gjennomfører helhetlig testing av systemene, tilsvarende [23], da dette kan avdekke skulte svakheter i komponenter eller samspillet mellom komponentene.

For å gjøre verktøy til testing av systemer mer tilgjengelig enn hva det er i dag bør det utvikles tester og testoppsett som er lettere å gjennomføre. Det blir i kapittel 10 presentert et konsept for et testoppsett som kombinerer noe av funksjonaliteten til TOCSSiC (kapittel 7.11.1), CRIOP (kapittel 7.11.3) og CSET (kapittel 7.11.2).

9.8 Aktiv bruk av DMZ for å skape sikre barrierer

Da systemene i mange år fremover sannsynligvis må ivareta sin sikkerhet ved bruk av nettverkssegmentering og sikkerhetskomponenter i nettverkets struktur, er det å anbefale at organisasjonen sørger for at de dataene som må krysse en barriere gjør dette på en så sikker måte som mulig.

Et tiltak som støttes i litteraturen, er å løse dette ved utstrakt bruk av DMZ-er kombinert med en implementering av diodeløsningen (kapittel 7.6.1). Dersom dette blir gjort sammen med gode struktureringer av rettigheter for dataaksess, vil man kunne oppnå god kontroll på trafikken inn og ut av DMZ-ene. Et eksempel på en slik strukturering er blant annet gitt i [13] og er løst gjengitt nedenfor. Disse

rettighetene bør tildeles i på en «need-to-know»-basis, hvor ingen har tilgang til mer data enn det som er strengt nødvendig.

1. **Ingen tilgang:** Eksterne aktører har verken tilgang til å lese eller endre variabler i DMZ-en. Dette tilgangsnivået bør tilordnes alle som ikke har behov for noen av dataene i DMZ-en.

2. **Lesetilgang:** Eksterne aktører har bare tilgang til å lese data fra DMZ-en. Flere moderne kommunikasjonsprotokoller avhenger av at mottakeren sender aksept på mottak av datapakkene, noe som gjør at man ofte i realiteten har to-veis kommunikasjon.

3. **Skrive- og lesetilgang:** Eksterne aktører har tilgang både til å skrive og lese fra DMZ-en. Dersom denne rettigheten brukes er det svært viktig at det eksisterer oversikter over hvilke brukere som har lov til å skive og lese til de forskjellige DMZ-ene. Dersom dette ikke blir tatt hensyn til, vil en enkelt bruker kunne få tilgang til å sende trafikken sin gjennom mange sikkerhetslag, noe som kan utgjøre en trussel for systemet.

9.9 Tiltak for å kunne reagere aktivt på trusler

Tilsyn utført viser at flere organisasjoner har mangelfulle eller uoppdaterte antivirusprogrammerer for deteksjon og fjerning av virus og ondsinnet programvare. Antivirusprogrammer kan være svært nyttige verktøy for administratorer slik at de kan handle aktivt dersom programvaren detekterer noe. Det anbefales derfor at organisasjonen finner en leverandør av slik programvare som passer bruken og trusselbildet organisasjonen står overfor.

Organisasjonen må også utarbeide gode rutiner for oppgradering av programvaren tilsvarende de beskrevet i kapittel 9.2, slik at beskyttelsen alltid er så optimal som mulig.

9.10 Tiltak for å kunne handle proaktivt

For at operatørene til enhver tid skal være oppdatert på de farene som finnes er det svært viktig at det blir satt av både tid og ressurser slik at administratorene kan følge med på dette. Dette kan gi muligheter til å handle proaktivt dersom det dukker opp nye potensielle trusler, noe som kan resultere i at man kan avverge angrep eller uhell før de oppstår. Eksempler på kilder til slik data er «ICS-CERT

Alerts»[15] og «NorCERT»[16], men ideelt bør sektoren på sikt samarbeide for å utvikle sine egne uavhengige kilder til hendelser og trusler som kan true drift og sikkerhet i prosessene.

9.11 Tiltak som kan gjennomføres av regulerende myndigheter

Et siste, men viktig, tiltak for å forbedre sikkerheten, er innføring av obligatoriske retningslinjer og regelverk av de regulerende myndigheter som operatørene er nødt til å forholde seg til med tanke på IT-sikkerheten til sine systemer.

Situasjonen i dag er slik at operatørene selv i stor grad kan diktere egne retningslinjer innenfor denne typen sikkerhetsarbeid, og de kan velge å legge seg på det nivået av sikkerhet de føler er ansvarlig. Dersom et slikt regelverk kommer i stand vil dette også kunne hjelpe operatørene å utføre et skikkelig sikkerhetsarbeid når det først gjøres. Et godt utgangspunkt for et slikt regelverk kan for eksempel være en videreutvikling av OLF104. Inntil et slikt regelverk er på plass bør de regulerende myndighetene gjennomføre flere tilsyn slik at man har god oversikt over sikkerhetsnivået i systemene.

De regulatoriske myndighetene kan også forsøke å stimulere dannelsen av fagnettverk hvor organisasjoner og leverandører kan møtes. Slike fagnettverk vil kunne brukes til å spre ideer, informasjon om farlige hendelser eller nye trusler, og kan også føre til å skape en uformell arena for kunnskapsdeling mellom alle deltakerne.

[15] http://www.us-cert.gov/ncas/alerts
[16] https://www.nsm.stat.no/Arbeidsomrader/Internettsikkerhet-NorCERT/

10 Videre arbeid

Da en masteroppgave skrives i et begrenset tidsrom, vil dette påvirke hvor stort omfanget av den kan bli. Dette kapitlet vil ta opp noen av temaene og områdene forfatteren mener at det kan jobbes mer med dersom oppgaven skal videreføres.

Bedre datagrunnlag

Utviklingen innen feltet skjer fort og nye tekniske muligheter og trusler dukker stadig opp. For å kunne få et enda bedre bilde av hvordan situasjonen til sikkerhetssystemene benyttet i integrerte operasjoner i dag, bør det enten gjennomføres nye tilsyn, eller det må finnes alternative kilder til relevant informasjon.

Det anbefales her at det for eksempel utarbeides et spørreskjema med relevante spørsmål som kan distribueres til forskjellige operatører og leverandører av SAS. Arbeidet med å samle inn data kan være svært tidskrevende, og det anbefales at dette settes igang tidlig dersom man har begrenset tid. Av områder som ville vært spesielt spennende å kartlegge bedre vil forfatteren trekke frem følgende:

- Undersøke nærmere hvilke struktureringer av nettverk som blir benyttet per dags dato, og hvordan man sikrer seg for at det ikke finnes noen «snarveier» i denne strukturen.
- Prøve å kartlegge i hvilken grad det foregår aktiv drifting av brannmurer, IDS-er og gjennomgang av logger.
- Kartlegge i hvilken grad systemer testes for feil. Både testing av delsystemer og av det komplette systemet er av interesse her.
- Hvilke tiltak som faktisk blir gjort for å bedre sikkerheten i integrerte operasjoner.
- Hvordan operatørene ser for seg å løse de sikkerhetsmessige problemene tredjeparts serviceleverandører i integrerte operasjoner kan føre med seg.

Litteraturstudium av standarder og retningslinjer

Det bør gjennomføres et mer detaljert litteraturstudium over flere standarder og retningslinjer som er relevante både for sikkerhet innen prosess- og IKT-systemer

enn det som er gjort i kapittel 7.2. Et slikt studium vil kunne gi en bedre forståelse av hvilke standarder og retningslinjer som blir og burde benyttes av organisasjoner. Det anbefales spesielt at det rettes fokus mot ISA/IEC-62443, da flere organisasjoner allerede har vist en interesse for denne standarden.

Trusselanalyse spesifikk for sektoren

Det kunne vært nyttig å utføre trusselanalyse slik at det kunne kartlegges hvilke aktører som kan ha interesse av å kompromittere systemene, samt hvor stor interessen for disse systemene er i miljøene som driver med slike angrep. Dette ville kunne gi et innblikk i hvilke krefter som kunne tenkes å stå bak et eventuelt angrep, og muligens også hvilken typer angrep som vil bli benyttet.

Andre faktorer for sikkerheten

Det er i denne oppgaven lagt stor vekt på de tekniske faktorene som kan bidra til å øke sikkerheten i systemer og nettverk ved innføringen av IO. Mye av den tilgjengelige litteraturen innen fagområdet legger vekt på at sikkerheten best kan forbedres ved å fokusere på tre faktorer: de menneskelige, de tekniske og de organisatoriske (MTO).

Det bør derfor også gjøres en gjennomgang med fokus på de tiltak som kan øke sikkerheten ved hjelp av de menneskelige og organisatoriske faktorene. Eksempler på slike faktorer kan være: kursing av personell, utvikling og innføring av regler og rutiner rundt bruk av systemer, hvordan organisasjonen jobber for å få til en god sikkerhetskultur og hvilke rutiner man har for autentisering av brukere.

Trådløs kommunikasjon

Trådløs kommunikasjon har allerede gjort sitt inntog i svært mange systemer, og det er naturlig å anta at denne formen for kommunikasjon vil bli mer og mer sentral også i industrielle systemer i tiden fremover. Det bør undersøkes hvilke mekanismer som finnes for å sikre trådløse kommunikasjon, hvor man kan tillate slik kommunikasjon og hvilke problemstillinger utbredt innføring av trådløs kommunikasjon vil kunne føre med seg.

Rutiner for testing av systemene

En videreføring av oppgaven kan være å utvikle testprosedyrer og programvare spesifikt for teste systemers sikkerhet ved IO. Et slik system kan for eksempel tenkes å spinne rundt TOCSSiC-programvaren ved å kombinere funksjonaliteten med den man finner i både den sjekklistbaserte CRIOP-metoden og CSET-programvaren. Dersom man implementerer norske forskrifter, retningslinjer og relevante standarder i en slik programvare, vil programvaren kunne bidra til å hjelpe administratorer med konkrete tiltak og tips som kan føre til økt sikkerhet tilpasset norske forhold.

Det anbefales også en gjennomgang av verktøyet CheckIT [17] som har fellestrekk med både CRIOP og CSET, men som det ikke ble tid til å behandle i denne oppgaven.

Bruk og problemer vedrørende OPC

Det er kjent at OPC (*en: OLE for Process Control*) ofte blir benyttet i SCADA-systemer til både lesing og skriving av prosessvariabler til og fra en OPC-server.

OPC har blitt svært populært da flere leverandører har adoptert protokollen i komponentene sine. Et problem med OPC er at det benytter seg av COM (Component Object Model) for å transportere data. Denne protokollen er kjent som vanskelig å rute gjennom brannmurer og man kan derfor ende opp med å åpne for mye i brannmuren slik at man skal få OPC til å fungere. Dette kan føre til sikkerhetsproblemer, og det bør utforskes om organisasjonen kan forbedre sin sikkerhet enten ved innføring av den nye utgaven av OPC, OPC UA, eller om man ville vært bedre tjent med å benytte DMZ-er fremfor OPC.

[17] `http://www.sintef.no/home/Technology-and-Society/Safety-Research/CheckIT/`

Referanser

[1] Albright, D., Brannan, P., and Walrond, C. (2010). *Did Stuxnet Take Out 1,000 Centrifuges at the Natanz Enrichment Plant?* Institute for Science and International Security.
http://isis-online.org/uploads/isis-reports/documents/stuxnet_FEP_22Dec2010.pdf.

[2] Alcaraz, C., Fernandex, G., and Carvajal, F. (2012). *Security aspects of SCADA and DCS Environments*. Springer-Verlag.

[3] Baker, S., Waterman, S., and Ivanov, G. (2009). *In The Crossfire - Critical Infrastructure in the Age of Cyber War*. McAfee.

[4] Bang, P. and Bjerkebæk, E. (2008). *IO i et risikoperspektiv - Hvordan kan IO påvirke menneskelige og organisatoriske faktorers betydning for storulykkesrisiko?* Petroliumstilsynet.
http://www.ptil.no/getfile.php/Presentasjoner/IO-1.pdf.

[5] Chief of Engineers United States Army (2006). *Survisory Control and Data Acquisition (SCADA) Systems for Command, Control, Communications, Computer, Intelligence, Surveillance and Reconnassance (C4ISR) Facilities*. United States Army.
http://scadahacker.com/library/Documents/SCADA_Basics/SCADA%20Basics%20-%20TM%205-601.pdf.

[6] Cole, E. (2008). *Official Course Material - Certified Ethical Hacker - Hackers Beware*. EC-Council.

[7] Control Systems Security Program (2009). *Recommended Practice: Improving Industrial Control Systems Cybersecurity with Defense-In-Depth Strategies*. U.S. Department of Homeland Security.

[8] Det Kongelige Olje - Og Energidepartement (2003-2004). *Stortingsmeldingmeld. nr. 38 - Om petroleumsvirksomheten*. Det Kongelige Olje - Og Energidepartement.
http://www.regjeringen.no/Rpub/STM/20032004/038/PDFS/STM200320040038000DDDPDFS.pdf.

[9] Falliere, N., Murchu, L. O., and Chien, E. (2011). *W32.Stuxnet Dossier*. Symantec Security Response.
http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf.

[10] FBI National Press Office (2006). *Moroccan Authorities Sentence Two in*

Zotob Computer Worm Attack. Federal Bereau of Investigation. http://www.fbi.gov/news/pressrel/press-releases/moroccan-authorities-sentence-two-in-zotob-computer-worm-attack.

[11] Grøtan, T. O., ICT), M. G. J. S., Øien, K., and (NTNU), T. O. (2007). *The SeSa Method for Assessing Secure Remote Access to Safety Instrumented Systems*. SINTEF. http://www.sintef.no/upload/Teknologi_og_samfunn/Sikkerhet%20og%20p%C3%A5litelighet/Rapporter/SINTEF%20A1626%20-%20SeSa%20report-final.pdf.

[12] Hauge, S., Johnsen, S. O., and Onshus, T. (2009). *Uavhengighet av sikkerhetssystemer*. SINTEF.

[13] Hauge, S., Onshus, T., Øien, K., Grøtan, T. O., Holmstrøm, S., and Lundteigen, M. A. (2006). *Uavhengighet av sikkerhetssystemer offshore - status og utfordringer*. SINTEF.

[14] International Electrotechnical Commission, IEC (2010). *IEC 61508 - Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems*. IEC, 2 edition.

[15] Johnsen, S. O. (2012a). *An Investigation of Resilience in Complex Socio-Technical Systems to Improve Safety and Continuity in Integrated Operations*. NTNU.

[16] Johnsen, S. O. (2012b). *Presentasjon: Informasjonssikkerhet i industrien*.

[17] Johnsen, S. O. (2012c). *Resilience at Interfaces - Improvement of Safety and Security in Distributed Control Systems by Web of Influence*. NTNU and SINTEF.

[18] Johnsen, S. O., Bjørkli, C., Steiro, T., Fartum, H., Haukenes, H., Ramberg, J., and Skiver, J. (2011). *CRIOP: A scenario method for Crisis Intervention and Operability analysis*. SINTEF.

[19] Johnsen, S. O., Lundteigen, M. A., Albrechtsen, E., and Grøtan, T. O. (2005). *Trusler og muligheter knyttet til eDrift*. SINTEF.

[20] Johnsen, S. O., Skramstad, T., and Hagen, J. (2009). *Enhancing the Safety, Security and Resilience of ICT and SCADA systems Using Action Research*. Critical Infrastructure Protection.

[21] Kurose, J. F. and Ross, K. W. (2011). *Computer Networking: A Top-Down Approach*. Pearson, 5 edition.

[22] Laboratory of Cryptography and System Security (CrySyS Lab) (2012).

sKyWIper (a.k.a. Flame a.k.a. Flamer): A complex malware for targeted attacks. Budapest University of Technology and Economics, Department of Telecommunications.
http://www.crysys.hu/skywiper/skywiper.pdf.

[23] Lüders, S. (2006). *CERN tests reveal security flaws with industrial networked devices*. The Industrial Ethernet Book.
http://cds.cern.ch/record/1000756/files/open-2006-074.pdf.

[24] Lundteigen, S. O. J. M. A. (2008). *Sikrere fjerndrift med CRIOP*. Tapir akademisk forlag.

[25] Moltu, B., Ringstad, A. J., and Guttormsen, G. (2008). *Consequence Analysis as Organizational Development*. SINTEF.

[26] Murchu, L. . and Jackson, D. (2011). *Debate: The Stuxnet authors are behind the Duqu trojan*. SC Magazine.
http://www.scmagazine.com/duqu-variant-uncovered/article/217171/.

[27] Norsk Olje & Gass (2009). *OLF104 - Norwegian oil and gas recommended guidelines for information security baseline requirements for process control, safety and support ICT systems*. Norsk Olje & Gass, 5 edition.

[28] OLF (2007). *HMS og Integrerte Operasjoner: Forbedringsmuligheter og nødvendige tiltak*. OLF.
http://www.ptil.no/getfile.php/PDF/IO%20og%20HMS-%20OLF-rapport.pdf.

[29] Olsen, O. A. (2005). *Instrumenteringsteknikk*. Tapir akademisk forlag.

[30] Onshus, T. (2010). *Prosess og Maskin-sikkerhet*. Tor Onshus.
http://www.itk.ntnu.no/ansatte/Onshus_Tor/PM-Sikkerhet.htm.

[31] Onshus, T. (2011). *Instrumenteringssystemer*. Institutt for teknisk kybernetikk, 5 edition.

[32] Roberts, P. F. (2005). *Zotob, PnP Worms Slam 13 DaimlerChrysler Plants*. eWeek.
http://www.eweek.com/c/a/Security/Zotob-PnP-Worms-Slam-13-DaimlerChrysler-Plants/.

[33] Sanger, D. E. (2012). *Obama Order Speed Up Wave of Cyberattacks Against Iran*. The New York Times.
http://www.nytimes.com/2012/06/01/world/middleeast/obama-ordered-wave-of-cyberattacks-against-iran.html?_r=1&pagewanted=all.

[34] Scarfone, K. and Hoffman, P. (2009). *Guidelines on Firewalls and Firewall Policy*. NIST - National Institute of Standards and Technology.

[35] Scarfone, K. and Mell, P. (2007). *Guide to Intrusion Detection and Prevention Systems (IDPS)*. NIST - National Institute of Standards and Technology.

[36] Seehusen, J. (2008). *Kan spare 300 milliarder*. Teknisk Ukeblad. http://www.tu.no/olje-gass/2008/02/20/kan-spare-300-milliarder.

[37] Statoil (2010). *Praktisk bruk og innføring av OLF104*. Statoil. http://www.ifea.no/wp-content/themes/ifea/kursdokumentasjon/2011/2011_nov_datasikkerhet_i_industrien/LGroteidePraktisk_bruk_og_innforing_av_OLF104_E.pdf.

[38] Stouffer, K., Falco, J., and Scarfone, K. (2007). *Guide to Industrial Control Systems (ICS) Security*. NIST.

[39] Symantec (2011). *W32.Duqu, The precursor to the next Stuxnet*. Symantec Security Response. http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_duqu_the_precursor_to_the_next_stuxnet.pdf.

[40] Telefonintervjuer (2013). *Telefonintervju med personel fra SAS-leverandør*.

[41] The TOCSSiC Team (2007). *TOCSSiC, A Teststand On Control System Security at CERN*. The TOCSSiC Team, 2 edition. https://edms.cern.ch/file/573062/3/The_TOCSSiC_Procedures.pdf.

[42] Tveiten, C. K., Lunde-Hanssen, L.-S., Grøtan, T. O., and Pehrsen, M. (2008). *Hva innebærer egentlig Integrerte Operasjoner? Fenomenforståelse og generisk elementer med mulige konsekvenser for storulykkepotensialet*. SINTEF.

[43] U.S. Department of Homeland Security (2011). *Common Cybersecurity Vulnerabilities in Industrail Control Systems*. Control Systems Security Program, National Cybver Security Division.

[44] Whitman, M. E., Mattord, H. J., Austin, R. D., and Holden, G. (2009). *Guide To Firewalls And Network Security: With Intrusion Detection And VPNs*. Course Technology Cengage Learning, 2 edition.

[45] Young, S. (2001). *Designing a DMZ*. SANS Institute InfoSec Reading Room. http://www.sans.org/reading_room/whitepapers/firewalls/designing-dmz_950.

VEDLEGG

A Innhold på CD

CD-en vedlagt på slutten av oppgaven inneholder følgende:

- Rapporten med alle tilhørende LaTeX-filer.
- Høyoppløselige utgaver av alle figurer.
- Flere av artikkelen benyttet i PDF-format.
- BibTeX-filen med alle referanser benyttet i oppgaven.

Veier bare går og går,
over berg og under tre,
langs elv som havet aldri når,
ved hi som aldri sol får se;
Over vintersnøen ren,
og gjennom junis blomstervell,
over gress og over sten,
og under månens fjerne fjell.

Veier bare går og går,
under sky og under stjerne,
men vandringsmannens føtter trår
til sist mot hjemmet i det fjerne.
Etter syn av sverd og brann
og skrekken stor i berge-slott,
kan øyne skue milde land
med tre og eng de kjenner godt.

— Hobbiten, J.R.R. Tolkien