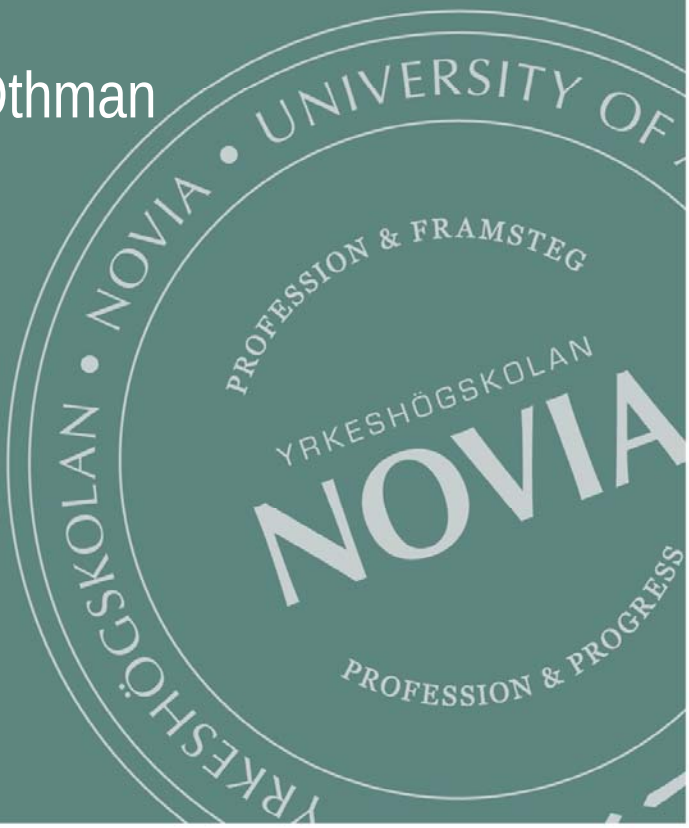
YRKESHÖGSKOLAN
NOVIA

Kyberturvallisuus ja tiedonsiirron turvaaminen autonomisten alusten kehittämisessä ja operoinnissa

Ville Leppänen, Sampsa Rauti, Kalle Rindell, Johannes Holvitie

Redaktör: Teresia Othman



Serie R: Rapporter

Ville Leppänen, Sampsa Rauti, Kalle Rindell, Johannes Holvitie, redaktör: Teresia Othman

Kyberturvallisuus ja tiedonsiirron turvaaminen autonomisten alusten kehittämisessä ja operoinnissa

Utgivare: Yrkeshögskolan Novia, Wolffskavägen 33, 65200 Vasa, Finland

Redaktör: Teresia Othman

Novia Publikation och produktion, serie R: Rapporter 3/2019
ISBN 978-952-7048-51-1 (online)
ISSN 1799-4179

Sammanfattning:

Fi:

Tämän kirjoituksen tavoitteena on käydä oppikirjan tapaan läpi keskeisiä asioita autonomisten alusten kyberturvallisuudesta. Kirjoituksen kohderyhmänä on alusten kehittämiseen, operointiin ja hankintaan liittyviä tahoja. Koska tätä kirjoitusta tehtäessä ensimmäisiä autonomisia vesiliikenteen aluksia ollaan vasta kehittämässä/ostamassa käyttöön, kirjoitus on kohdistettu erityisesti aluksia kehittäville toimijoille.

Kyberturvallisuus tarkoittaa tietoteknisen turvan rakentamista ympäristölle, joka koostuu sen käyttäjistä, tietoliikenneverkoista, laitteista, kaikesta ohjelmistosta, prosesseista, taltioidusta ja välitettävästä tiedosta, sovelluksista, palveluista ja ylipäänsä kokonaisuuteen suoraan tai epäsuoraan liitetyistä järjestelmistä. Autonomisen aluksen tapauksessa autonomisuus saavutetaan ohjelmistollisin keinoin – *ohjelmistojärjestelmien tietoturvallisuus* on yksi keskeisistä asioista. Toisaalta autonominen alus koostuu useasta osasta, joiden tulee olla keskenään vuorovaikutuksessa. Autonomisia aluksia valvotaan ja joissakin tilanteissa myös ohjataan etäältä. Näiden seikkojen takia *tietoliikenneratkaisujen tietoturva* on toinen keskeinen kyberturvallisuuden osa-alue autonomisten alusten kontekstissa. Tietoturvalla ymmärretään resurssien suojaamista oikeudetonta toimintaa vastaan.

Luvussa 2 luodaan katsaus autonomisten alusten resursseihin, joista monet liittyvät aluksen ominaisuuksiin, mutta osa on itse asiassa autonomisuuden mahdollistavien ICT-ratkaisuihin liittyviä resursseja.
Luvussa 3 tarkastellaan tietoturvan yleisiä tavoitteita. Tietoturvan saavuttamiseksi on erilaisia menettelyitä (prosesseja). Esimerkiksi tietoturvavaatimusten muodostaminen, uhkakuvan kartoittaminen ja tietoturvatestaus ovat menettelyiden keskeisiä rakennuselementtejä. Toisaalta on myös kehitetty runsaasti erilaisia teknisiä ratkaisuja kyberturvallisuuden saavuttamiseksi. Kehitysmenetelmät ja tekniset ratkaisut ovat ensisijaisesti suunnattu kehittäjille, mutta toisaalta niiden soveltaminen jatkuu operointivaiheessa, jolloin vastuuta turvallisuuden saavuttamisesta delegoidaan myös operoiville tahoille, viime kädessä ihmisille loppukäyttäjinä. Näitä tietoturvan teknisiä seikkoja tarkastellaan kattavasti luvussa 3.

Kun luvuissa 2 ja 3 on muodostettu näkemys autonomisten alusten keskeisistä turvallisuusratkaisuja kaipaavista resursseista ja ratkaisujen elementeistä, käydään luvussa 4 läpi ryhmittäin resursseihin kohdistuvaa uhkakuvaa ja ratkaisujen kirjoa. Uhkakuvaa muodostetaan kolmesta näkökulmasta: järjestelmän, tietoliikenteen ja operoivien loppukäyttäjien näkökulmista.

Tämän kirjoituksen eri kohdissa pyritään antamaan erilaisia ohjenuoria alussa mainituille kolmelle kohderyhmälle: autonomisten alusten valmistukseen osallistuvat yritykset, operointiin osallistuvat tahot ja hankintaa suorittavat sekä toimintaedellytyksiä kehittävät päättäjät. Tämä kirjoitus on osaa ÄlyVESI-hanketta.

Älykäs kaupunkivesiliikenne eli ÄlyVESI-hanke oli 1.10.2016 – 31.5.2018 toteutettu kaupunkien, yritysten sekä korkeakoulujen välinen konseptointi, tuotekehitys- ja innovaatioprojekti. Hankkeessa tutkittiin, kehitettiin ja testattiin uutta teknologiaa ja älykkäitä kaupunkivesiliikenteen ratkaisuja ja palveluita. Hankkeen toteuttivat yhteistyössä Yrkeshögskolan Novia, Turun ammattikorkeakoulu, Aalto-yliopisto ja Turun kaupunki. Hanke sai päärahoituksen Euroopan aluekehitysrahaston 6Aika-ohjelmasta. Lisäksi hanketta rahoittivat Liikenteen turvallisuusvirasto sekä Helsingin ja Espoon kaupungit.

Eng:

The goal of this report is to cover the central concepts for achieving cyber security for autonomous vessels in a textbook manner. The target audience are the parties involved with the development, operation and purchase of vessels. As at the time of writing this report, the first autonomous water transport vessels are currently only under development / early adoption, the most central target audience for the report are parties that develop vessels.

Cyber security as a term is more comprehensive than the information security of computers or individual devices/systems. Cyber security refers to building data and technology security for the environment, which consists of its users, the data traffic networks, devices, all software, processes, saved and communicated data, applications, services and the systems directly or indirectly connected to the entity. Achieving autonomy with autonomous vessels is achieved through software – *software system security* is a central factor. In contrast, an autonomous vessel consists of several components that must interact with one another. Several different levels have been defined for autonomy in literature and people still have a role in many of them. There is a desire to monitor autonomous systems and in some situations the monitoring party must be able to perform remote manoeuvers on the vessel. For these reasons, *information security of data traffic solutions* is another important cyber security area in the context of autonomous vessels.

Section 2 provides an overview of the resources of autonomous ships, many of which are associated with the ship's features, but some are in fact resources associated with ICT solutions that enable autonomy.

Section 3 further discusses the general goals of information security, one of which is *non-repudiation*. There are different types of procedures (processes) for achieving information security. They aim to ensure the information security of the solutions being developed using a directed appropriate development process. For example, identifying information security requirements, evaluating the threats and information security testing are central building blocks of the procedures. In contrast, a lot of different types of technical solutions have also been developed in order to achieve information security. Older information security methods rely on encryption methods, cryptography, but many other types of solutions that increase information security have been developed through ICT systems. Development methods and technical solutions are primarily directed to developers, but applying them continues during operation. This results in responsibility for achieving security being also delegated to operators - ultimately people as end users. These technical aspects of information security are discussed comprehensively in Section 3.

While sections 2 and 3 form a view of the resources and solution elements needing security solutions for autonomous ships, Section 4 covers the various threats and solutions by resource groups. The threat is evaluated from three perspectives: the system, data communication and the operator end users.

This report aims to provide various guidelines for the three target groups mentioned in the beginning: companies participating in the manufacture of autonomous ships, those who

participate in their operation, entities responsible for procurement and decision-makers. This report is part of the ÄlyVESI project.

Smart City Ferries, the ÄlyVESI project, was a conceptualisation, product development and innovation project realised by cities, businesses and universities 1.10.2016 – 31.5.2018. The project explored, developed and tested new technologies and intelligent urban waterborne traffic solutions and services. Novia University of Applied Sciences, Turku University of Applied Sciences, Aalto University and the City of Turku carried out the project in co-operation. The project was funded by the 6Aika-program of the European Regional Development Fund. In addition, the project was funded by the Finnish Transport Safety Agency and the cities of Helsinki and Espoo.

Sök- och nyckelord:

Novia UAS, Älyvesi, älykäs kaupunkiliikenne, tietoliikenne, kyberturvallisuus, tiedonsiirto, autonominen alus

Novia UAS, Älyvesi, smart city ferries, cyber security, data transfer, autonomous vessels, data traffic

ÄLYKÄS KAUPUNKI VESILIIKENNE

Euroopan unioni
Euroopan aluekehitysrahasto

Vipuvoimaa EU:lta 2014–2020

6Aika

KYBERTURVALLISUUS JA TIEDONSIIRRON TURVAAMINEN AUTONOMISTEN ALUSTEN KEHITTÄMISESSÄ JA OPEROINNISSA

Ville Leppänen, Sampsa Rauti, Kalle Rindell, Johannes Holvitie

Kirjoittajista

Ville Leppänen on ohjelmistotekniikan ja ohjelmistoturvallisuuden professori. Hänellä on yli 200 kansainvälisesti arvioitua julkaisua ohjelmistotekniikan ja -turvallisuuden alalta, joissa keskitytään mm. ohjelmistokehitysmetodologioihin, projektijohtamiseen, työkaluihin ohjelmistolaadun ja turvallisuuden tehostamiseksi, ohjelmointikieliin, rinnakkaisuuteen sekä arkkitehtuurisuunnitteluun. Leppäsen tutkimus- ja kehitysprojektiportfolio on merkittävä. Se kattaa mm. CyberTrust-projektin ohjelmistollisen diversifioinnin ja introspektion käyttämisestä tietoturvan lisäämiseksi. Muutaman viime vuoden aikana Leppänen on vetänyt paikallisesti 5 Tekesin ja Puolustusministeriön rahoittamaa tietoturvaprojektia – näistä edelleen menossa on Tekesin ja F-Securen rahoittama introspektiomekanismien kehittämisprojekti osana NSF:n Suomessa olevaan kyberturvallisuuden co-center:iä. Leppänen toimii Turku Centre for Computer Science ohjelmistokehityslaboratorion johtajana ja Turun yliopiston Tulevaisuuden teknologioiden laitoksen varajohtajana.

Sampsa Rauti on ohjelmistotekniikan DI, jatko-opiskelija ja yliopisto-opettaja. Hänen tutkimusalansa on ohjelmistojen tietoturva, erityisesti proaktiiviset haittaohjelmien torjunta- ja havainnointimenetelmät. Rautilla on yli 30 julkaisua tietoturvan alalta.

Kalle Rindell on teollisuuden ja julkishallinnon tehtävissä työskentelevä tietoturvakonsultti ja tutkijakoulutettava. Koulutukseltaan hän on tietojärjestelmätieteen FM.

Johannes Holvitie on ohjelmistotekniikan tohtori. Hänen tutkimusalansa kattaa laajasti ohjelmistotuotantomenetelmiä, projektinhallintaa sekä tietoturvallisuutta, joista hänellä on yhteensä yli 30 julkaisua. Hän on myös SFS/SR 314 ”Ohjelmistotuotanto ja järjestelmäkehitys” standardisointikomitean jäsen, konsultti sekä aktiivinen elinkoinelämän toimija.

Tämä raportti on toteutettu osana **Älykäs kaupunkivesiliikenne** -hanketta. Älykäs kaupunkivesiliikenne, eli ÄlyVESI-hanke, on kaupunkien, yritysten sekä korkeakoulujen välinen konseptointi, tuotekehitys- ja innovaatioprojekti. Hankkeessa tutkitaan, kehitetään ja testataan uutta teknologiaa ja älykkäitä kaupunkivesiliikenteen ratkaisuja ja palveluita. Hanketta toteuttavat yhteistyössä Yrkeshögskolan Novia, Turun ammattikorkeakoulu, Aalto-yliopisto ja Turun kaupunki. Hanke saa päärahoituksen Euroopan aluekehitysrahaston 6Aika-ohjelmasta. Lisäksi hanketta rahoittavat Liikenteen turvallisuusvirasto sekä Helsingin ja Espoon kaupungit.

Sisältö

Luku 1

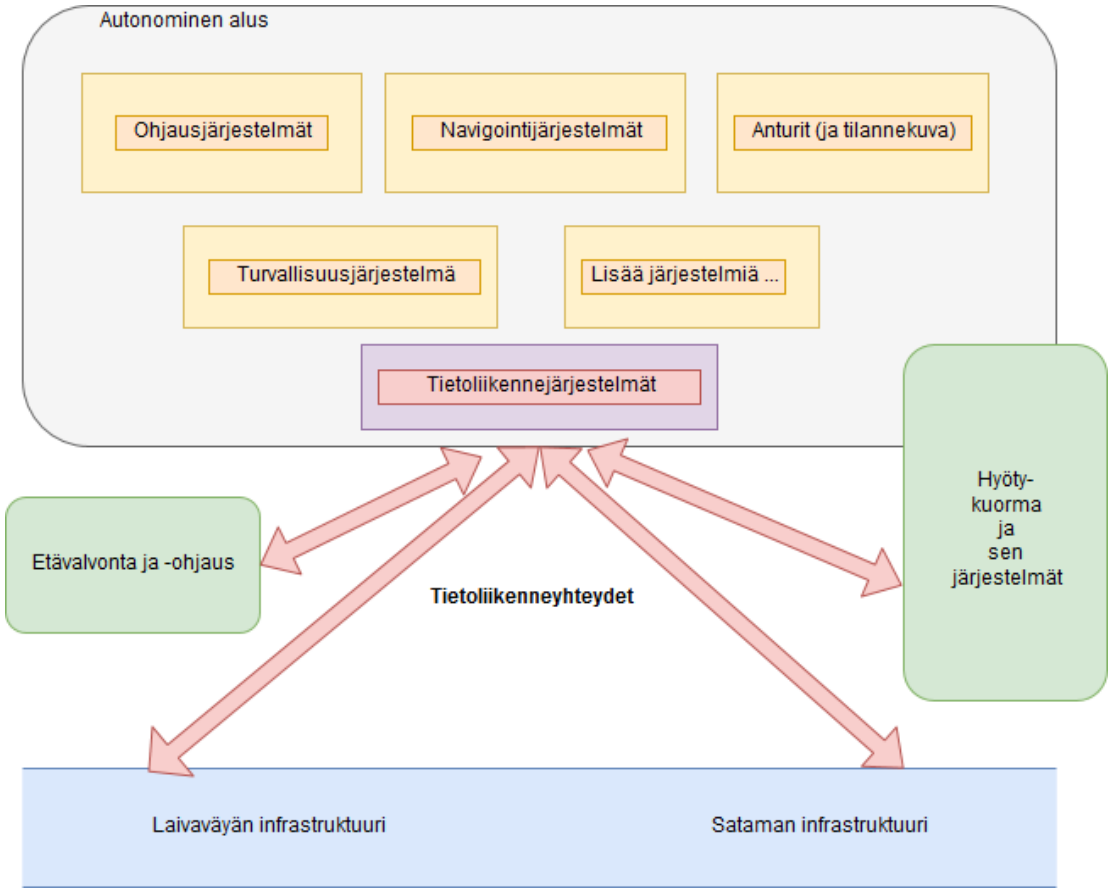
Johdanto

Tämän kirjoituksen tavoitteena on käydä oppikirjan tapaan läpi keskeisiä asioita autonomisten alusten kyberturvallisuuden saavuttamisen kannalta. Kohdeyleisönä pidetään alusten kehittämiseen, operointiin ja hankintaan liittyviä tahoja. Koska tätä kirjoitusta tehtäessä ensimmäisiä autonomisia vesiliikenteen aluksia ollaan vasta kehittämässä / ottamassa käyttöön, kohdeyleisöstä keskeisin on aluksia kehittävät toimijat.

1.1 Kyberturva ja suojattavat resurssit

Kyberturvallisuus (engl. cyber security) on terminä laajempi kuin tietokoneiden tai yksittäisten laitteiden/järjestelmien tietoturva. Kyberturvallisuus tarkoittaa tietoteknisen turvan rakentamista ympäristölle, joka koostuu sen käyttäjistä, tietoliikenneverkoista, laitteista, kaikesta ohjelmistosta, prosesseista, taltioidusta ja välitettävästä tiedosta, sovelluksista, palveluista ja ylipäänsä kokonaisuuteen suoraan tai epäsuoraan liitetyistä järjestelmistä. Autonomisen aluksen tapauksessa autonomisuus saavutetaan ohjelmistollisin keinoin – *ohjelmistojärjestelmien tietoturvallisuus* on yksi keskeisistä asioista. Toisaalta autonominen alus koostuu useasta osasta, joiden tulee olla keskenään vuorovaikutuksessa. Autonomisuudella on kirjallisuudessa määritelty useita erilaisia tasoja, joissa monissa ihmisellä on jokin rooli (esim. [26] luettelee 10 tasoa). Autonomista järjestelmää halutaan valvoa ja joissakin tilanteissa valvovan tahon pitää voida suorittaa etäältä ohjaustoimia autonomiseen alukseen. Näiden seikkojen takia *tietoliikenneratkaisujen tietoturva* on toinen keskeinen kyberturvallisuuden osa-alue autonomisten alusten kontekstissa.

Tietoturvalla ymmärretään *resurssien* suojaamista oikeudetonta toimintaa vastaan. Resurssi on käsite, jolla on autonomisen aluksen puitteissa paljon erilaisia ilmentymiä. Usein tietoturvan yhteydessä ajatellaan, että resurssit tarkoittavat erilaisia tietosisältöjä informaatiojärjestelmien sisällä (esimerkiksi aluksen matkustajaluettelo, lastin tietot tietokannassa tai tieto autonomisen aluksen sijannista). Tämä on kuitenkin liian rajoittunut tulkinta. Resurssina voi toimia myös toiminnallisuus, jonka jokin tietotekninen ratkaisu tuottaa (esimerkiksi kyky muuttaa autonomisen aluksen suuntaa tai nopeutta). Kuva 1.1 havainnollistaa osaa autonomiseen alukseen suoraan ja epäsuorasti liittyviä resursseja. On välttämätöntä ymmärtää suojattavien resurssien olemus, jotta voidaan arvioida niihin kohdistuvat uhkat ja suunnitella toteutettavissa olevia suojaavia vastatoi-

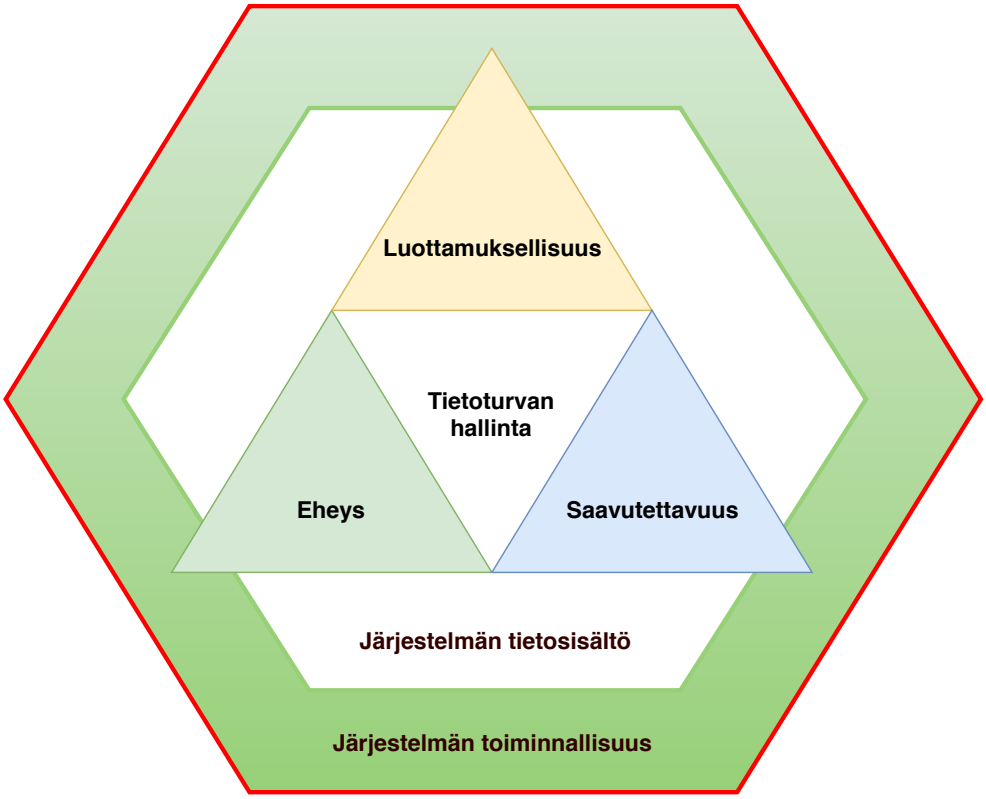


Kuva 1.1: Aluksen ja sen toiminta ympäristön resurssit muodostavat kokonaisuutena autonomisen aluksen kyberturvallisuuden kannalta merkittävät kohteet. Kuvassa havainnollistettu osaa tunnistettavissa olevista resursseista.

mia ja menettelyitä. Luvussa 2 luodaan katsaus autonomisten alusten resursseihin, joista monet liittyvät aluksen ominaisuuksiin, mutta osa on itse asiassa autonomisuuden mahdollistavien ICT-ratkaisuihin liittyviä resursseja.

Autonomisilla aluksilla voi olla autonomisuuden lisäksi myös runsaasti muunlaisia resursseja, jotka liittyvät laivan hyötykuorman olemukseen. Hyötykuorman tyyppeinä voidaan pitää matkustajia (esim. matkustaja-alus, autolautta), lastia (esim. konttialus, tankkeri), havainnointi (esim. tutkimusalus) ja vaikuttavuutta (esim. sota-alus). Luvussa 2 käsitellään myös tätä resurssiominaisuutta, mutta pääsääntöisesti tämän kirjoituksen puitteissa hyötykuorman tietoturvallisuuteen liittyvät seikat on rajattu käsittelyn ulkopuolelle. Autonomiset alukset toiminnallaan vaikuttavat myös aluksen ulkopuolella oleviin resursseihin – niiden turvallisuuteen. Alukset kulkevat satamasta toiseen käyttäen meriväyliä ja erilaista muuta infrastruktuuria. Myös tätä puolta tarkastellaan luvussa 2.

1.2 Tietoturvan perusulottuvuudet



Kuva 1.2: CIA: Luottamuksellisuus, eheys ja saavutettavuus.

Usein tietoturvassa mielletään olevan kolme perusarvoa tai -ulottuvuutta, ks. kuva 1.2. Nämä ovat *luottamuksellisuus* (engl. confidentiality), *eheys* (engl. integrity) ja *saavutettavuus* (engl. availability). Usein kirjallisuudessa käytetään tästä termiä CIA-kolmio (engl. CIA triagle). Seuraavassa laajempi kuvaus termien merkityksestä:

Luottamuksellisuus tarkoittaa, että vain käyttöoikeuden omaavat tahot (henkilöt, prosessit, järjestelmät, viranomaistahot, jne) voivat kohdistaa toimia resurssiin. Kun resurssina on tietosisältö, on luottamuksellisuuden käsite lähellä yksityisyyden käsitettä, ollen sen eräs komponentti. Toiminnallisuuksia edustavien resurssien tapauksessa luottamuksellisuus liittyy voimakkaasti oikeuteen käyttää jotain järjestelmää tai sen tiettyjä toimintoja.

Eheys tarkoittaa resurssin sisällön täsmällisyyttä ja täydellisyyttä koko resurssin elinkaaren aikana. Tietokeskeisen resurssin tapauksessa tarkoitetaan, että mikään oikeudeton taho ei ole missään vaiheessa päässyt vääristämään tietosisältöä. Esimerkkinä vääristämisestä on vaikka laivan sijaintitiedon muuttaminen valheelliseksi. Toiminnallisen resurssin, vaikkapa tietokoneen osana autonomista alusta, tapauksessa tarkoitetaan, että oikeudeton taho ei ole (edes hetkellisesti) muuttanut järjestelmän toiminnallisuutta. Esimerkiksi haittaohjelman toimiminen tietokoneen sisällä tarkoittaa järjestelmän eheyden menettämistä.

Saavutettavuus tarkoittaa, että resurssin pitää olla saavutettavissa silloin kun sitä tarvitaan. Saavutettavuus on ominaisuus, joka ensisijaisesti täydentää luottamuksellisuuden ja eheyden käsitteitä. Ilman saavutettavuuden vaatimusta, voitaisiin tehdä täydellisen luottamuksellisia (estettäisiin kaikkien pääsy järjestelmään) ja eheitä (kryptataan kaikki tietosisältö ja tehdään järjestelmän sisällön muuttamisesta mahdotonta) ratkaisuja, jotka olisivat samalla täydellisen käyttökelvottomia. Suuri ja käytännöllinen uhka saavutettavuutta vastaan ovat erilaiset palvelunestohyökkäystilanteet (engl. denial-of-service attacks), joiden avulla tyypillisesti pyritään estämään tietoliikenneyhteydet etäältä resurssiin.

Luvussa 3 tarkastellaan lisää tietoturvan yleisiä tavoitteita, joista yksi on *kiistämättömyys* (engl. non-repudiation). Tietoturvan saavuttamiseksi on erilaisia menettelyitä (prosesseja), jotka tähtäävät ohjatun kehitysprosessin avulla takaamaan kehitettävien ratkaisujen tietoturvan tarkoituksen mukaisella tavalla. Esimerkiksi tietoturvavaatimusten muodostaminen, uhkakuvan kartoittaminen ja tietoturvatestaus ovat menettelyiden keskeisiä rakennuselementtejä. Toisaalta on myös kehitetty runsaasti erilaisia teknisiä ratkaisuja kyberturvallisuuden saavuttamiseksi. Vanhimmat tietoturvamenetelmät nojaavat salakirjoitusmenetelmiin, kryptografiaan (engl. cryptography), mutta ICT-järjestelmien myötä on kehitetty myös muunlaisia tietoturvaa lisääviä teknisiä ratkaisuja. Kehitysmenetelmät ja tekniset ratkaisut ovat ensisijaisesti suunnattu kehittäjille, mutta toisaalta niiden soveltaminen jatkuu operointivaiheessa, jolloin vastuuta turvallisuuden saavuttamisesta delegoidaan myös operoiville tahoille, viime kädessä ihmisille loppukäyttäjinä. Näitä tietoturvan teknisiä seikkoja tarkastellaan kattavasti luvussa 3.

Kun luvuissa 2 ja 3 on muodostettu näkemys autonomisten alusten keskeisistä turvallisuusratkaisuja kaipaavista resursseista ja ratkaisujen elementeistä, käydään luvussa 4 läpi ryhmittäin resursseihin kohdistuvaa uhkakuvaa ja ratkaisujen kirjoa. Uhkakuvaa muodostetaan kolmesta näkökulmasta: järjestelmän, tietoliikenteen ja operoivien loppukäyttäjien näkökulmista.

Tämän kirjoituksen eri kohdissa pyritään antamaan erilaisia ohjenuoria alussa mainituille kolmelle kohderyhmälle: autonomisten alusten valmistukseen osallistuvat yritykset, operointiin osallistuvat tahot ja hankintaa suorittavat sekä toimintaedellytyksiä kehittävät päättäjät. Seuraavassa ohjeet 1.1 – 1.3 havainnollistavat esitettävien ohjeiden muotoa.

Ohje 1.1 (Kehittäjä) Kehityshenkilökunnan riittävästä tietoturvavalmiuksien koulutuksesta tulee huolehtia.

Ohje 1.2 (Operaattori) Etävalvonta/-hallinta järjestelmien operoivien henkilöiden tietoturvakoulutuksen tulee olla järjestelmällistä ja ajanmukaista.

Ohje 1.3 (Päättäjä) Vaadi autonomisen aluksen operoijalta uhkakartoitus riskiarviointien kera.

1.3 Luottamuksen merkityksestä ja olemuksesta

Ennen varsinaisen käsittelyn aloittamista vielä muutama sana tietoturvaratkaisujen ja *luottamuksen* (eng. trust) välisestä suhteesta sekä toisaalta riippuvuuksista ja niiden aiheuttamasta uhkasta.

Filosofisesti voi sanoa, että tietoturvaratkaisuja ei tarvita, jos meillä on täydellinen luottamus kaikkien osapuolten kesken. Jos mikään taho ei muodosta uhkaa, ei tarvita tietoturvaratkaisuja torjumaan uhkia. Toisaalta, filosofisesti voi myös todeta, että hyvin harva (jos mikään) tietoturvaratkaisu ei sellaiseen – ilman luottamusta – voi taata turvallisuutta. Järjestelmissä on operoivia tahoja, joille pitää antaa oikeudet suorittaa toimenpiteitä. Tahot voidaan tunnistaa vahvoja tunnistautumismenetelmiä käyttäen, mutta tunnistaminen ei takaa sitä, että operoivat tahot eivät voisi tehdä turvallisuutta vaarantavia toimenpiteitä. Käyttäjien oikeuksia ja osajärjestelmien toimintakykyä voidaan rajoittaa, mutta järjestelmien perusluonteeseen kuuluu (järjestelmän käyttäjien) mahdollisuus tehdä valintoja ja osa valinnoista voi olla turvallisuutta vaarantavia. Viime kädessä pitää siis luottaa, että oikeutetut tahot eivät käytä oikeuksiaan väärin.

Luottamuksen tarve näkyy erityisesti operatiivisessa toiminnassa, mutta luottamusta tarvitaan myös järjestelmien kehittämisen yhteydessä. Autonomisen aluksen kehittämisen kannalta keskeistä on ymmärtää, että koska kaikkia ratkaisuja ei voida tehdä itse alusta loppuun asti, ollaan teknisen ratkaisun osalta riippuvaisia muista toimittajista ja ratkaisuista – pitää voida luottaa niiden toimintaan. Teknisten ratkaisujen yhteydessä käytetään myös termiä 'luotettu perusta' (engl. trusted computing base) tarkoittamaan sellaista komponenttien ja ratkaisujen perustaa, joiden oletetaan toimivan ilman, että niitä vastaan tulisi suojautua. Ne toimivat luottamuksen teknisenä ankkurina, jonka päälle ratkaisut rakentuvat.

Luottamusta lopulta aina tarvitaan, mutta teknisten ratkaisujen tarvitseman luottamuksen määrää voidaan riippuvuussuhteiden tapauksessa rajoittaa. Juuri luottamuksen tarpeen määrän rajoittaminen on tietoturvaratkaisujen keskeinen tavoite. Toisaalta osaamisen lisäämisen kautta voidaan myös pyrkiä lisäämään osaamistaan soveltavien tahojen luotettavuutta. Autonomisten alusten yhteydessä riippuvuuden ja siihen liittyvän luottamuksen rajoittamisen toimia ovat mm. seuraavat:

Käyttäjien määrän rajoittaminen: Pyritään tunnistamaan tarpeelliset käyttäjät, jotta luotettujen tahojen määrä voidaan pitää pienenä. Esimerkiksi autonomisten alusten tapauksessa valvontakeskuksen henkilöstön roolit tulee miettiä tarkaan ja samoin myös väyliä ja satamia operoivien tahojen roolit.

Osapuolten oikeuksien rajoittaminen: Eri käyttäjille / osajärjestelmille voidaan antaa erilaisia rajoitettuja oikeuksia noudatellen heidän toimintaprofiilistaan tulevia tarpeita. Usein on myös tarpeen rajoittaa sitä, mistä käsin oikeuksia saa soveltaa (esim. vain tietystä toimipisteestä käsin).

Järjestelmien toiminnan rajoittaminen: Osajärjestelmiä voidaan koventaa poistamalla niistä tarpeettomia toiminnallisuuksia ja suojaamalla tietosisältöjä kryptografisesti. Osajärjestelmien väliin voidaan asettaa esim. palomuureja tai eristää ne mahdollisuuksien mukaan kokonaan toisistaan.

Toimintojen monistaminen ja diversifiointi: Luottamusta yhden ratkaisun toimivuuteen voidaan rajoittaa tuottamalla erilaisia (diversifiointuja) ratkaisuja. Esimerkiksi autonomisen aluksen yhteydessä kommunikaatiokanava valvontakeskuksen suuntaan on keskeisen tarpeellinen toiminto – yksittäisen kanavan toiminnan luottamuksen tarvetta voidaan vähentää monistamalla kanavia ja erityisesti tekemällä niistä keskenään erilaisia (vrt. erilaiset radio- ja satelliittiyhteydet). Sama koskee toiminnallisia järjestelmiä ja niiden komponentteja. Varajärjestelmien olisi hyvä olla keskenään erilaisiin toteutuksiin nojautuvia – jotta ei tarvitse niin paljon luottaa yhteen järjestelmään ja sen komponentteihin.

Toimittajien määrän rajoittaminen: Kun autonomisen aluksen fyysisiä ja ohjelmistollisia järjestelmiä koostetaan, rajoitutaan vain joihinkin luotettuihin toimittajiin ja heidän luotettuihin alihankintaketjuihinsa. Ohjelmistoratkaisujen yhteydessä alihankkijoiden ohjelmat voidaan myös allekirjoittaa, jolloin niiden eheyteen on suurempi luottamus.

Operointiin liittyvän alihankintaverkoston rajoittaminen: Vastaavasti operoinnin kohdalla myös alihankintaverkoston osalta tulee miettiä, miten luottamuksen määrää voidaan hallinta. Tämä koskee ylipäänsä operointiympäristön (laivaväylien ja satamien infran) toimijoita.

Vastaavasti osaamistason noston kautta luottamusta lisääviä toimia ovat mm. seuraavat:

Käyttöhenkilöstön koulutus: Järjestelmien tekninen käyttö nojaa luottamukseen, että henkilöstö osaa oikealla tavalla käyttää järjestelmää. Tällaisen luottamuksen tarvetta voidaan vähentää lisäämällä käyttöhenkilökunnan koulutusta, jolloin luottamus korvautuu tiedolla käyttöhenkilöstön osaamistasosta.

Kehitysstandardien soveltaminen: Uusien kompleksisten autonomisten alusten kehittämisen tietoturvallisuutta voidaan lisätä pakottamalla kehitystoiminta noudattamaan jotain annettua turvallisuutta lisäävää kehitysmenetelmää – jolloin tarve luottaa kehittäjien henkilökohtaisiin kykyihin on vähäisempää.

Testaus: Luottamusta järjestelmien oikeelliseen toimintaa voidaan lisätä testausmenettelyillä (esim. penetraatiotestaus, tyypillisesti osana edellistä).

Tekninen valvonta: Järjestelmien toimivuutta voidaan erikseen valvoa ns. tunkeutumisen havainnointijärjestelmillä (engl. intrusion detection system). Esimerkiksi kun alus muutetaan autonomiseksi alukseksi korvaamatta kuitenkaan kaikkia vanhoja ratkaisuja uusilla, voidaan vanhoja osajärjestelmiä sijoittaa valvonnan alaisuuteen. Yleisesti tarjolla on verkon ja koneen valvontaratkaisuja (network / host intrusion detection). Samaa valvontaa voidaan pyrkiä soveltamaan myös esim. valvontakeskuksissa henkilöstön toimien suuntaan, jos riittävää rajoittavaa logiikkaa ei ole voitu integroida etäoperointiratkaisuihin.

Hankintaosaaminen: Jos oikeita ratkaisuja ei osata vaatia, ei niitä usein tule myös toteutettua. Tämä pätee myös autonomisten alusten hankintaprosessiin. Hankintaosaamista tulisi tarvittaessa kouluttaa.

1.4 Riippuvuudet ja hyökkäyspinta-ala

Autonomiset alukset ja niiden toimintaympäristöt ovat lopulta hyvin kompleksisia ja moninaisia. Käytännössä mikään yksittäinen taho ei voine valmistaa alusta ja sen osia itse kokonaan. Vastaavasti mikään yksittäinen taho ei hallitse autonomisen aluksen koko ympäristöä ja sen operointia. Erityistä huomiota tulee kiinnittää alihankintaverkoston suuntaan sekä valmistuksen että operoinnin ja huollon osalta. Operoinnin osalta tulee ymmärtää, että aluksen ja sen ympärivän infrastruktuurin eroavuudet tietoturvan näkökulmasta.

Kompleksisuus tarkoittaa, että kokonaisuus muodostuu monista osajärjestelmistä ja osapuolista, jotka ovat vuorovaikutuksessa keskenään. Kun jonkin järjestelmä A käyttää toisesta järjestelmästä B saatavaa tietoa – sen oma toiminta perustuu ja muuttuu toisesta järjestelmästä tulevan tiedon pohjalta – voidaan sanoa A:n olevan *riippuvainen* B:stä. Usein riippuvuussuhteet ovat samanaikaisesti molempiin suuntiin. Tällaisten järjestelmien kokonaisuuden sanotaan muodostavan hajautetun järjestelmän.

Riippuvuuksista on tarpeen olla tietoinen, koska erilaiset häiriötilanteet propagoituvat helposti riippuvuussuhteiden kautta järjestelmästä toiseen. Jos A on riippuvainen B:stä ja B:n toiminta lakkaa tai olennaisesti muuttuu (esim. kyberhyökkäyksen seurauksena), on mahdollista, että A ei enää pysty toimimaan – mahdollisesti sen suunnittelussa ei ole kunnolla varauduttu kyseiseisiin tilanteisiin[1]. Riippuvuuksien ketjut voivat olla hyvin pitkiä järjestelmien välillä, jolloin häiriön lähteen tunnistaminen propagoitumisketjun kautta voi olla vaikeaa (häiriöt vaikuttavat tuolloin ”mystisiltä”). Sama riippuvuuksien rakentuminen pätee myös yksittäisen järjestelmän ja sen osasten välillä. Esimerkiksi yksittäiset ohjelmistoratkaisut saattavat koostua tuhansista ”osasista”, jotka ovat keskenään vuorovaikutuksessa.

Yksittäiseen järjestelmään liittyvä suora *hyökkäyspinta* (engl. attack surface) on sen järjes telmän rajapinta muihin järjestelmiin – siis kaikki ne kanavat ja niiden rajapinnat, joiden kautta ulkoinen järjestelmä (muu maailma) voi vaikuttaa järjestelmään. Suojaamisessa erityistä huomiota tulee kiinnittää siihen, mitä dataa liikkuu rajapinnan kautta ja mitä tapahtuu rajapinnan puitteissa (miten sitä käytetään). Esimerkiksi hyökkääjä voi yrittää käydä läpi kaikkia TCP-portteja systemaattisesti (niitähän on TCP-protokollan määrittelyn mukaan yli 60,000).

Historiallisesti suuri osa hyökkäyspintaan kohdistuvista tunkeutumishyökkäyksistä on ollut variaatioita injektiohyökkäyksen yleistyypistä. Injektiohyökkäyksissä ulkoinen taho toimittaa odotetun syötedatan sijaan – tai osana sitä – koodia, joka järjestelmän toteutuksessa olevan virheen takia päätyy osaksi suoritettavaa koodia (data injektoituu suoritettavaksi koodiksi). Erityisen tavallisia ovat olleet erilaiset SQL-injektiohyökkäykset. Tärkempiä keinoja estää injektiohyökkäykset on olla luottamatta ulkoisen rajapinnan kautta tulevaan dataan ja sen muodon oikellisuuteen.

Autonomisen aluksen järjestelmäkokonaisuuden kannalta oleellista on havaita, että rajapintojen kautta tulevan datan/toiminnon oikeellisuutta ei useinkaan voida todentaa järjestelmän ”ulkokehällä”, vaan edellä kuvattujen riippuvuuksien kautta hyökkääjän toimet voivat propagoitua muihin järjestelmiin. Hyökkäysrajapinnan kannalta tämä tarkoittaa, että välittömän suoran

[1] Kuvaava kommentti hajauttujen järjestelmien ”guru” Lesie Lamporilta on ”You know you have a distributed system when the crash of a computer you’ve never heard of stops you from getting any work done.”

rajapinnan lisäksi on paljon suurempi epäsuora hyökkäysrajapinta, joka koostuu kaikista niitä sisäisten / riippuvien järjestelmien rajapinnoista, joihin ulkoinen taho voi toimillaan vaikuttaa. Esimerkiksi tietoliikennerajapinnan kautta vastaanotettu data päätyy jollekin sovelluskerroksella olevalle palvelulle, joka voi välittää sen edelleen vaikka tietokantapalvelimen moottorille, jne. *Kyberturvallisuuden kannalta riippuvuudet näyttelevät siis merkittävää roolia mahdollisten hyökkäysten eskaloitumisessa ja kyberriskien propagoitumisessa*.

Yksittäisen kohdejärjestelmän, autonomisessa aluksessa vaikka nagivointijärjestelmän, kannalta tämä tarkoittaa, että hyökkäysrajapinnan kautta on mahdollisesti hyvinkin monia ”polkuja”, joita pitkin hyökkääjä voi yrittää toteuttaa hyökkäystään. Suojautumistoimien osalta tämä tarkoittaa, että toimet pitäisi kohdistaa kaikkiin näihin ”polkuihin” ja niiden osiin. Heikoimman lenkin periaatteen mukaan järjestelmä on niin vahva kuin sen ”heikoin lenkki”.

Edellä on käsitelty hyökkäysuhkaa lähinnä ulkoisen rajapinnan ja välittömän vaikuttamisen kautta. *Kohdennetut hyökkäykset* (engl. Advanced Persistent Threats, APT) ovat skenaarioita, joissa vaikuttaminen tapahtuu monivaiheisesti. Tällaista toimintaa on kuvattu monissa tietoturvayhtiöiden ns. white paper -julkaisuissa[2]. Kohdennetuissa hyökkäyksissä pyritään ensin pääsemään sisään järjestelmään – ja sitten pitkänkin tauon jälkeen liikkumaan järjestelmäkokonaisuuden sisällä ”lateraalisesti” kohti varsinaista kohdetta. Autonomiset alukset voivat hyvinkin olla arvonsa takia kohdennettujen hyökkäysten kohteina. Kyberturvallisuuden kannalta tämä tarkoittaa, että järjestelmäkokonaisuuden sisällä ei tule oletusarvoisesti luottaa muihin osajärjestelmiin, vaan suunnittelua pitää ohjata periaate ”Luota, mutta tarkista.” (Trust but verify).

[2] Ks. esim. https://www.f-secure.com/documents/996508/1030745/dukes_whitepaper.pdf

Luku 2

Merenkulku, alukset ja ICT-järjestelmät

Tämän luvun tarkoituksena on alustaa merenkulun ratkaisukonteksti sekä määritellä ne fyysiset ja virtuaaliset resurssit, joihin autonomisen toiminnan ratkaisuja esitetään. Ensimmäisessä aliluvussa 2.1 läpikäydään merenkulun olennainen infrastruktuuri ja erityisesti eri alustyypit hyötykuormineen. Ratkaisukontekstin pohjustus lopetetaan läpikäymällä merenkulun sääntelyä, joka ottaa kantaa alus- ja hyötykuormatyyppeihin. Tarkemmin käsitellään myös merenkulun kyberturvallisuuteen liittyvän sääntelyn nykytila.

Toinen aliluku 2.2 määrittelee autonomisen merenkulun kohderesurssit sekä autonomisaation prosessina. Kohderesurssien läpikäynti pohjautuu nykytilan tarkasteluun ja on jaettu kahteen osaan: aluksen sisäisiin ja ulkopuolisiin järjestelmiin. Autonomisaatioprosessi esitetään yleisesti hyväksytyn ALFUS (engl. Autonomy Levels for Unmanned Systems) tasoluokittelun tasojen tavoitteluna. Luokittelun läpikäynnin jälkeen siirrytään kuvaamaan, miten nykyinen akateeminen ja teollinen tutkimus mieltävät tasoihin liittyvän autonomian nimenomaan merenkulun toimialalle.

Viimeinen aliluku 2.3 käsittelee yleisesti ICT-järjestelmiä ja niiden kehitystä. Autonomisoinnin näkökulmasta ICT-järjestelmät ovat virtuaalisia kohderesursseja, jotka liittyvät olennaisesti fyysisiin vastinpareihinsa. Ne ovat kuitenkin toteutukseltaan ratkaisukontekstiriippumattomia. Aliluvun lopuksi huomioidaan ICT-järjelmistä ne osa-alueet, jotka ovat kyberturvallisuuden toteuttamisen kannalta kriittisiä.

2.1 Merenkulku

Merenkulun logistiikka kattoi jo vuonna 2008 maailman kaupasta 89.6% siirretyn tavaran koolla mitattuna ja 70.1% siirretyn tavaran arvossa mitattuna. Trendi on ollut jatkuvasti ylöspäin kehittyvä. Merkittävin syy on niin kutsuttu skaalaetu (engl. economies of scale). Logistiikan alustana, meri mahdollistaa erittäin suurteen kuljetusjärjestelmien eli alusten käytön. Alus vaatii huomattavan alkuinvestoinnin, mutta matalaoperointiriski kokonaisarvoltaan huomattavan lastin kanssa kuolettaa sen nopeasti. Tästä johtuen kuljetuksen yksikköhinta saadaan vietyä alas ja riippuen jalostuksen määrästä kuluttaja maksaa tuotteen hinnassa ainoastaan muutamia prosesentteja kuljetuksen liitettäviä kustannuksia. Keskustellut esimerkit soveltuvat osittain myös matkustajalii-

kenteeseen [23].

Merenkulun logistiikkaa määrittää skaalautuvuuden lisäksi globalisaatio sekä hitaus. Koska maapallon pinta-alasta noin 71% on vettä, lähes jokaista toimitusketjun osaa on mahdollista operoida logistisesti edullisimmalla tavalla eli merialuksella. Merirahti on täten merkittävä globalisaation edistäjä. Erityisesti voidaan huomioida merirahdin erikoistava vaikutus, kun toimitusketjun osat myös hyödyntävät skaalaetua. Esimerkkinä tästä on vaikka norjalainen sei, joka kalastuksen jälkeen toimitetaan Kiinaan perattavaksi, Intiaan pakattavaksi ja vasta tämän jälkeen takaisin myytäväksi muun muassa Norjaan ja Suomeen. Hitaus määrittää myös merenkulun logistiikkaa. Vaikka läpäisykyky on alusten koon takia merkittävä, yksikkösiirtojen pituus on erityisesti mannerten välillä huomattava. Suurten merialusten keskinopeus on 25 ja 30 solmun välillä parhaan polttoaine-ekonomian saavuttamiseksi. Aluksen päivämatka voi tällöin olla 500 km luokkaa, jolloin useiden kymmenientuhansien toimitusreitti vie viikkoja [17].

Merenkulkua, kuten monia muitakin teollisuuden aloja, odottaa niin kutsuttu neljäs vallankumous. Maailman ekonominen foorumi (engl. World Economic Forum) ennustaa parhaillaan käynnissä olevan neljännen revoluution koostuvan primäärisesti kasvavan älykkään automaation tuomista tehokkuusparannuksista. Saman on nähty koskevan myös merenkulkua älykkäiden ja autonomisten alusten tutkimuksen ja vaiheittaisen käyttöönoton myötä. Tarkastelemme seuraavissa alikappaleissa merenkulun infrastruktuuria, alustyyppejä sekä sääntelyä, joista nykyinen merenkulun toiminta, automaation kohde, koostuu[1].

2.1.1 Infrastruktuuri

Pelkistäen, logistinen verkosto on joukko toimitusketjuja. Toimitusketju on järjestetty jono hyötykuormasiirtoja kahden pisteen välillä. Merilogistiikassa nämä pisteet ovat satamia. Satamia yhdistävät vesialueet, ja vesialueella operoidaan aluksilla. Tarkastelemme seuraavassa lyhyesti jokaista näistä merilogistiikan osista.

Alus

Alus on kulkuväline, jolla voidaan liikkua vesialueella. Kansainväliselle merenkulkujärjestölle IMO (engl. International Maritime Organization) rekisteröityjen aluksien tunnisteisiin lukeutuvat omistajan nimi, aluksen nimi, rekisteröintimaa sekä tunnistenumero. Omistaja on hyötykuorma-aluksissa yleensä varustamoyhtiö. Aluksen nimi on ei-sitova ja hyötykuorma-aluksissa esimerkiksi valmistussarjaa indikoiva. Rekisteröintimaa kertoo sen lipun, jonka alla alus operoi. Lippu on usein tarpeellinen siirryttäessä aluevesille, jolloin maiden väliset järjestelyt ovat voimassa.

Tunnistenumero on kansainvälisesti operoiville laivoille tärkein tunniste. IMO myöntää numeron laivan rungolle ja täten se ei voi muuttua niin kauan kun samaa runkoa käytetään. IMO ylläpitää rekisterikantaa[2], josta tunnistenumerolla pystytään tarkistamaan muun muassa laivan

[1] Merenkulun infrastuktuuria ja sääntelyä koskeva sanasto on osittain kieliriippuvaista. Katso esimerkkejä suomenkielisten ja englanninkielisten sanastojen väliltä:
`https://fi.wiktionary.org/wiki/Luokka:Suomen_kielen_merenkulun_sanasto`
`https://en.wikipedia.org/wiki/Glossary_of_nautical_terms`

[2] `https://ihsmarkit.com/products/imo-ship-company.html`

Taulukko 2.1: IMO alustyypit. [27]

Alustyyppi	Määrittävä tekijä	Sääntely
matkustaja-alus	Kantaa yli 12 matkustajaa	SOLAS[3] I/2
kalastusalus	Pyytää kaupallisesti mereneläviä	SOLAS I/2
ydinalus	Kantaa ydinvoimalähdettä	SOLAS I/2
kuivalastialus	Kantaa kuivaa lastia primääristi	SOLAS IX/1.6 & XII/1.1
öljytankkeri	Kantaa haitallista nestettä tai kaasua	SOLAS 74 & MARPOL[4] ANX I REG 1.5
rahtialus	Kantaa yleisrahtia	MEPC.1/Circ. 681 ANX
suurnopeusalus	Pystyy korkeaan nopeuteen	SOLAS X/1.2, HSC C 2000 P 1.4.30
siirrel. porausyksikkö	Pystyy kaupalliseen merenpohjaporaukseen	SOLAS IX/1, MODU C 2009 P 1.3.40
erikoisalus	Kantaa roolinsa takia yli 12 erikoismiehistöä	SPS Code P 1.3.12

omistus- ja rekisteröintimaahistoria. Valtioilla on myös omat paikallisjärjestelynsä laivarekisteröinneille. Suomen laki säätää esimerkiksi, että kaikki 15 metriä ja sitä pidemmät kauppamerenkulkuun tarkoitetut suomalaiset alukset tulee saattaa Suomen alusrekisteriin.

Aluksilla on useita tyyppiluokituksia. IMO luokittelee alukset yhdeksään kategoriaan taulukon 2.1 mukaisesti.

Aluksen tyyppi usein määrittää, millaista liikennöintiä sillä voidaan tehdä. Liikennöintityypit jaetaan usein kolmeen kategoriaan: linjaliikenne, hakurahti ja sopimusliikenne. **Linjaliikenteellä** tarkoitetaan ennalta määritetyn linja mukaista liikennöintiä, jossa linja määrittää ne ajanhetket, jolloin alus on reitillään tietyissä satamissa. Matkustaja-aluksen tekemä liikennöinti on tästä yleinen esimerkki. **Hakurahti** on toiselta nimeltään tramppiliikennöinti. Hakurahdilla tarkoitetaan yksittäissopimuksia tietyn tavaraerän toimittamista määritetystä satamasta toiseen. Alus liikennöi ainoastaan, jos se saa lastin aloitussatamasta. **Sopimusliikenne** on jatkuvaa rahdin kuljetusta ennaltamääritetyn lähtö- ja saapumissataman välillä. Sopimus kiinnittää rahtisisällön vakioksi.[5]

Vesialueet

Alus voi operoida eri tyyppisillä vesialueilla. *Yhdistyneiden kansakuntien merilaki* UNCLOS (engl. United Nations Convention on the Law of the Sea) määrittää tällaisiksi sisäveden, alueveden, saaristoveden, raja-alueen, yksinoikeudellisen ekonomisen alueen sekä mannerjalustan[6]. Yleisesti sovellettava UNCLOS III määrittää alueet seuraavasti:

Sisävesi kattaa kaikki maan rajojen sekä yleisen määrityslinjan sisäpuolelle jäävät vesimassat. Yleinen määrityslinja on UNCLOSissa tarkennettu matalan laskuveden rajaksi tai saaristorajaa heikosti mukailevan suoran määrittämäksi rajaksi. Valtiot voivat vapaasti säätää sisävesistään.

[4]MARPOL liite: `http://www.marpoltraining.com/MMSKOREAN/MARPOL/Annex_I/ui1.htm`

[4]IMO *kansainvälinen yleissopimus ihmishengen turvallisuudesta merellä* SOLAS (engl. International Convention for the Safety of Life at Sea)

[5]Esimerkki liikennöintityypeistä osana Euroopan komission lainsäädäntöä: `http://eur-lex.europa.eu/legal-content/fi/TXT/?uri=CELEX:52005PC0651`

[6]UNCLOS artikkelien listaus: `http://www.un.org/depts/los/convention_agreements/texts/unclos/UNCLOS-TOC.htm`

Aluevesi ulottuu 12 meripeninkulmaa (noin 22 km) sisävesirajasta merelle päin. Valtiot saavat säätää vapaasti sisävesistään, mutta niin kutsuttu viaton läpikulku sallitaan. Tällä tarkoitetaan selkeää läpikulkua alueen läpi, jonka aikana ei harjoiteta esimerkiksi kalastusta tai vakoilua. Sukellusalusten tulee nousta läpikulussa pintaan ja kaikkien alusten tulee liputtaa rekisteröintimaansa lippua.

Saaristovedet määritetään piirtämällä uloimpien saarten – jotka ovat kuitenkin riittävän lähellä toisiaan – uloimpien pisteiden välille janat. Valtiolla on samat valtuudet säätää rajoittuvasta alueesta kuin aluevesistään, mutta esimerkiksi perinteisiä kalastusyhteistyösääntöjä ei voida ylittää.

Raja-alue tai **aluemeri** on alueveden rajasta 12 meripeninkulmaa merelle päin ulottuva alue. Valtio saa säätää ainoastaan tulli-, verotus-, maahanmuutto- sekä päästöasioissa kyseisellä alueella.

Yksinomainen talousalue EEZ (engl. Exclusive Economic Zone) ulottuu 200 meripeninkulmaa (noin 370km) sisävesirajasta merelle. Valtiolla on yksinoikeus hyödyntää kaikkia tältä alueelta löytyviä luonnonvaroja. Vieraat valtiot voivat operoida meri- ja ilma-aluksia kuin myös vetää merenalaisia putkia ja kaapeleita tällä alueella, kunhan rannikkovaltioiden muita säädöksiä ei rikota.

Mannerjalusta jatkuu joko luonnollisen mannerjalustan tai EEZ:n rajaan. Kauemmas ulottuva on määräävä, mutta 350 meripeninkulmaa (noin 650km) on ääriraja. Valtiolla on oikeus mineraalien ja elottoman materiaalin keräämiseen toisten valtioiden kustannuksella. Valtio voi myös määrätä vedessä olevan elollisen resurssin osalta, mutta ainoastaan EEZ-rajaan asti.

Vesiväylät ja -merkinnät

UNCLOS säätää myös, että jokaisen valtion tulee rekisteröidä vesiväylät ja näiden muutokset IMO:lle. IMO listaa MSC/Circ.1060 säädösdokumentissaan muun muassa seuraavia elementtejä alusliikenteen ohjaamiseksi[7][8]:

Liikenteenerottelusuunnitelma on reititysmenettely, jonka tavoite on vastakkaisen liikenteen erottelu sopivilla menetelmillä ja erityisesti perustamalla liikennöintikaistoja.

Liikennöintikaista on yhden suuntaiselle liikenteelle määritetty alue. Luonnolliset esteet ja toiset liikennöintikaistat voivat muodostaa rajoja tälle alueelle.

Erottelualue tai -kaista erottaa vastakkaissuuntaisia liikennöintikaistoja, liikennöintikaistoja muista vesialueista tai tietyn luokittelun omaavat alukset toisista.

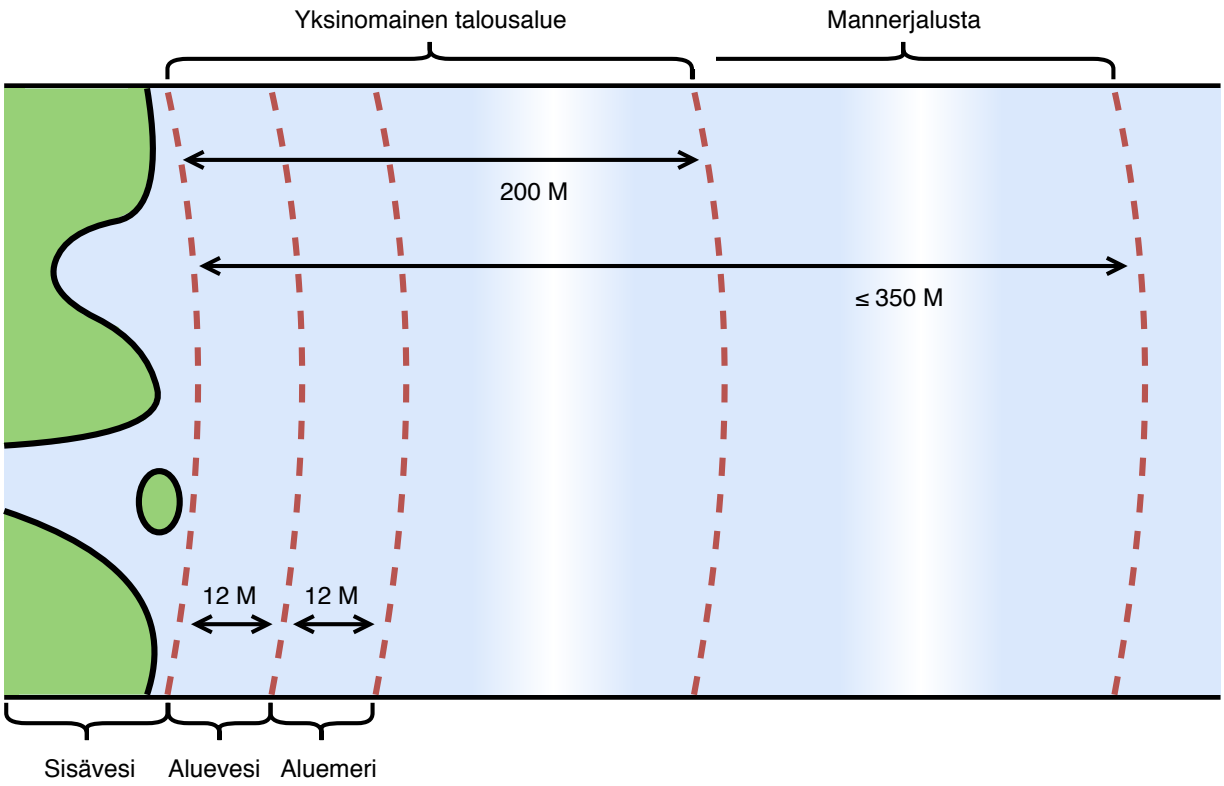
Kiertoliittymä on erottelualueen ja kiertävän liikennöintikaistan yhdistelmä.

Rannikkoliikennöintikaista on alue rannan ja liikenteenerottelusuunnitelman rannanpuoleisen reunan välillä.

Suositeltu reitti on määrittelemättömän leveä reitti, joka koetaan liikennöiville aluksille suotuisaksi. Se osoitetaan usein keskiviiva merkitsevillä poijuilla.

[7] IMOn sääntely vesiväylistä: `http://www.imo.org/en/OurWork/Safety/Navigation/Pages/ShipsRouteing.aspx`

[8] Esimerkki vesiväyläelementtien käytöstä: `http://www.charts.gc.ca/publications/chart1-carte1/sections/m-tracks/examples-eng.asp#section`



Kuva 2.1: Vesialueet ja niiden rajat meripeninkulmissa.

Syvävesireitti on määritellyn levyinen reitti, jolle merenpohjan muodot ja mahdolliset pinnanalaiset kohteet on tarkoin kartoitettu.

Varoalue on rajattu alue, jossa operoivien alusten pitää kiinnittää erityistä huomioita navigointiinsa. Liikenteen suunnasta voidaan antaa tällä alueella suosituksia.

Välttöalue on rajattu alue, jossa liikkuminen on erityisestä syystä vaarallista tai missä onnettomuuksia pitää erityisesti välttää. Alue voi koskea kaikkia tai tietyn tyyppisiä aluksia.

Satama

Satamat muodostavat merireittien päätepisteet. Koska satamat ovat sisävesialueita, niissä sovelletaan ainoastaan kunkin valtion omaa lainsäädäntöä. Satama-alueet ovat usein liikennejärjestelyiltään varoalueita, koska ne ovat ruuhkaisia sekä ahtaita. Riippuen sataman tyypistä ja paikallisesta säännöstöstä, satama voi määrätä luotsipakosta (ts. luotsinkäyttövelvollisuus). Luotsi on ammattinimike, jolla tarkoitetaan henkilöä, jolla on erityistuntemus paikallisesta vesialueesta sekä pitkä kokemus suurten merialusten käsittelystä. Luotsien täytyy suorittaa myös erityinen luotsitutkinto.

Esimerkiksi Suomen laissa on luotsausasetus. Luotsi kutsutaan ja hän astuu laivaan yleensä

huomattavasti ennen varsinaisen satama-alueen alkamista. Vaikka luotsi toimii toimii aluksen päällikön neuvonantajana, käytännössä luotsi on usein silti komentovastuussa aluksesta. Luotsin tehtävänä on ohjata alus tuloväylän ja satama-alueen läpi laiturille tai terminaalille ja auttaa laivan ankkuroinnissa. Aluksen lähtiessä satamasta, luotsi on vastuussa vastakkaisista toimista satama- tai vesistöalueen rajalle asti.

Ohje 2.1 (Kehittäjä) Satamat, vesiväylät ja vesialueet muodostavat autonomiselle alukselle dynaamisesti muuttuvan infrastruktuurin, jota aluksen kehittäjä ei kontrolloi ja jonka kanssa aluksen täytyy olla vuorovaikutuksessa, erityisesti autonomisuuden osalta. Dynaamisuudesta seuraa, että operointivaiheessa rajapintaa käyttäviä osapuolia on monia. Rajapinta infrastuktuuriin muodostaa potentiaalisen hyökkäysrajapinnan ja sitä kautta kyberuhkia alusta kohtaan. Autonomisten alusten kehittäjien tulee hallita aluksen aiotun toimintaympäristön infrastruktuurin olemus ja huomioida siihen liittyvät kyberuhat.

Ohje 2.2 (Päättäjä) Satamien, vesiväylien ja vesialueiden infrastukruurin säätelyn (toimintojen muoto, luotettavat toimijat) kautta voidaan pyrkiä rajaamaan autonomisiin aluksiin kohdistuvia kyberriskejä. Aluksen huono kyberriskien hallinta toteutusvaiheessa aiheuttaa uhkia ympäröivälle infrastruktuurille.

2.1.2 Sääntely

Luvun 2.1.1 määrittelyissä on jo laajalti viitattu muun muassa IMO:n – SOLAS ja MSC (engl. Maritime Safety Committee) liitteet – ja YK:n – UNCLOS – määrityksiin. Nämä ovat esimerkkejä kansainvälisestä sääntelystä. Seuraavassa alikappaleessa luodaan lyhyt katsaus merenkulun sääntelyrakenteeseen, joka voidaan karkeasti jakaan kolmeen tasoon: kansainväliseen, talousliittoumaan sekä kansalliseen. Tätä seuraava alikappale kiinnittää erityistä huomiota turvallisuussääntelyyn. Turvallisuus ja erityisesti kyberturvallisuus ovat kriittisessä roolissa, kun autonomisuuden tasoa kasvatetaan merenkulun logistiikassa.

Sääntelyrakenne

Kansainvälinen sääntely voidaan mieltää korkeimmaksi merilogistiikan sääntelytasoksi. Olennaisin säännös tällä tasolla on UNCLOSin kolmas yhteissopimus UNCLOS III, jonka 157 entiteettiä (suurin osa valtioita) on allekirjoittanut. UNCLOS säätää valtioiden vastuista liittyen maailman merien käyttöön sekä yleisohjeista kaupankäyntiin, luonnonsuojeluun ja merien luonnonvarojen käyttöön liittyen. YK ei kuitenkaan itse täytäntöönpane tai valvo sopimusta.

Kansainvälisten merilogistiikan säädösten täytäntöönpanossa YK:n alaisilla järjestöillä on merkittävä rooli. Näistä olennaisin on IMO, joka panee täytäntöön kansainvälisiä sopimuksia

SOLASin lisäksi MARPOLin (Kansainvälinen konventio merialusten päästöjen ehkäisemiseksi; engl. International Convention for the Prevention of Pollution from Ships) kautta. SOLASia on käsitelty jo aiemmin muun muassa vesialueiden ja laivatyyppien määrittäjänä. MARPOL asettaa rajoituksia liittyen alusten polttoaineeseen, jätteenkäsittelyyn sekä vaaralliseen rahtiin.

Talousliittouman sääntely riippuu siitä, kuuluuko vesialueita hallitseva taho niihin. Merkittävin esimerkki talousliittoumasääntelystä on Euroopan unioni (EU). EU:n perustuskirja takaa ihmisten, tavaran, palveluiden sekä kapitaalin vapaan liikkumisen talousliittouman alueella. Merilogistiikalle aluevesiltä toisille siirtyminen ilman tullikäytäntöjä on merkittävä helpotus. Lisäksi EU:lla on useita toimenpideohjelmia, jotka ovat syntyneet reaktioina talousalueen tapahtumiin[9]. Koska useimmat näistä ovat päästövahinkoja, merkittävä osa toimenpideohjelmista säätää lisärajoituksista ja -varotoimista merilogistiikalle. Esimerkki tästä on PCS (satamavaltiokontrolli; engl. Port State Control), joka määrää satamat tekemään tarkastuksia aluksille, jotka eivät operoi kyseisen valtion lipun alla.

Kansallinen sääntely on sitä vallitsevampaa, mitä lähempänä valtion rantaa ollaan (ks. luvun 2.1.1 vesialueet). Kansallinen sääntely on primääristi valtion lainsäädännön kautta toteutettua ja täten kansallisessa merenkulun sääntelyssä on huomattavia eroja. Suomessa liikeenne- ja viestintäministeriön alaisuudessa toimiva liikenteen turvallisuusvirasto Trafi *"on liikennejärjestelmän sääntely- ja valvontatehtävistä vastaava hallinto- ja turvallisuusviranomainen, jonka tehtävänä on edistää liikenteen turvallisuutta ja kestävää kehitystä liikennejärjestelmässä"*. Trafin toiminnasta esimerkkinä on määräys 15.03.2010 TRAFI/7106/03.04.01.00/2010[10], joka määrittää ja kohdentaa kotimaanliikenteen liikennealueet ja näiden rajat. Tästä aluekategoria I esimerkiksi rajaa rannikon sisävedet (em. vesialueet määritetään sisävesirajasta).

Turvallisuussääntely

Kaikilla tasoilla tapahtuva merenkulun sääntely on primääristi turvallisuussääntelyä. IMO listaa ensisijaiseksi tehtäväkseen turvallisen merenkulun mahdollistamisen ja vasta tätä seuraa esimerkiksi liiketaloudelliset tavoitteet. Pohjana kaikelle turvallisuussääntelylle voidaan pitää IMOn SOLASta. Tämän luvussa XI-2 asetetaan erityiskäytänteitä merenkulun turvaamiselle. Nämä sisältyvät kansainväliseen alus- ja satamalaitos turvallisuuskoodistoon (ISPS; engl. International Ship and Port Facility Security). ISPS velvoittaa UNCLOS:n ratifioineita valtioita[11]. Se muun muassa velvoittaa laivan kapteenin varmistamaan aluksen turvallisuus aina; ylittäen jopa työnantajan tai sopimuksen sanelemat käytänteet.

Talousaluetasolla, esimerkiksi Euroopan Unionilla on lainsäädäntökokonaisuus meriturvallisuudesta, jonka Euroopan komissio on merkittävimmiltä osiltaan säätänyt. 1) Regulaatio numero 725/2004 (Euroopan komission säätämä) on osoitettu parantamaan alusten ja satamien turvallisuutta. Käytännössä se implementoi SOLAS:n ja ISPS:n osia. Erityisesti se tekee pakolliseksi useita ISPS:n käytänne-ehdotuksena esittelemiä osia. 2) Direktiivi 2005/65 on lisäys edelliseen ja määrittää kaikki satamat ja niiden kokonaisuudessa kattamat alueet turvasääntelyn kohteek-

[9] https://ec.europa.eu/transport/sites/transport/files/modes/maritime/security/doc/legislation_maritime_security.pd

[10] https://www.finlex.fi/fi/viranomaiset/normi/501001/35559

[11] http://www.imo.org/en/OurWork/Security/Pages/MaritimeSecurity.aspx

si. Käytännössä, jäsenvaltioiden tulee valvoa esitettyjen ohjeiden ja ennaltaehkäisymekanismien implementointia satamissa. Tästä olennaisena esimerkkinä on turvallisuussuunnitelman luominen satamalle ja sen liikenteelle. Suunnitelma ottaa kantaa muun muassa kaikkiin olennaisiin riskeihin ja niiden todennäköisyyksiin. 3) Regulaatio 324/2008 määrittää, miten Euroopan komissio tekee tarkastuksia meriturvallisuuteen liittyen. Primäärinä tehtävänään Komissiolla on edellisten kohtien ylläpito. Sillä on myös apunaan regulaatioryhmä Meriturvallisuuden komitea (MARSEC; engl. Maritime Security Committee), jossa jokaisen jäsenvaltion on mahdollista vaikuttaa talousalueen meriturvallisuuteen.

Kansallisella tasolla meriturvallisuussääntely on usein osa yleistä turvallisuuteen liittyvää lainsäädäntöä. Suomessa Trafi ylläpitää ISPS- sekä EU-lähtöisiä sääntöjä suorittaen muun muassa satamien suojauskatselmuksia kuin myös Komission vaatimia ulkomaan lipun alla olevien laivojen satamatarkastuksia. Trafi, kuten moni muu valtiotason sääntelyviranomainen, valvoo meriturvallisuusilmoitusten tekoa. Ilmoitus on seikkaperäinen kertomus tapahtuneesta tai lähellä piti tilanteesta, joka on kohdistunut alukseen tai muuhun rakenteeseen. Pakollisia liitteitä ovat muun muassa laiva- sekä konepäiväkirjat.

Tietoturvasääntely on toistaiseksi meriturvallisuudessa vasta ohjeistustasolla[12]. Koska merenkulun sääntelyrakenne on riippuvainen kansainvälisen sääntelyn muuttumisesta, IMOlla merkittävä rooli merilogistiikan tietoturvallisuuden edistämisessä. Muiden tasojen sääntelytoimijat voivat toimia vasta tämän jälkeen. IMO on reagoinnut jatkuvasti kasvavaan kyberturvallisuusuhkien määrään ja se on viime aikoina julkaissut useita korkean tason ohjeistuksia (ei kuitenkaan velvoittavaa lainsäädäntöä) asiaan liittyen. 1) IMOn meriturvallisuus komitean 98. istunnon raportti MSC 98/23 julkaistiin vuonna 2017. Se listaa useita kyberturvallisuusuhkia sekä näihin reagointia tarkoituksenaan tuottaa asiaan liittyvää ohjeistusta. Tämän seurauksena IMO julkaisi hieman myöhemmin Ohjeistuksen merenkulun kyberriskien hallintaan (MSC-FAL.1/Circ.3.; engl. Guidelines on maritime cyberrisk management)[13].

Ohjeistus merenkulun kyberriskien hallintaan tunnistaa, että laivan järjestelmistä ainakin komentosilta-, lastinkäsittely-, propulsio-, koneistonhallinta-, alukselle pääsynhallinta-, matkustajapalvelu-, matkustajaviihde-, hallinto-, miehistöhyvinvointi- sekä kommunikaatiojärjestelmät ovat kyberriskialttiita. Selvitys kertoo, että merkittävin uhka-alue on nimenomaan virtuaalisen ja fyysisen toteutuksen rajapinnat. Ohjeistuksen implementointiosuus koostuu kyberriskihallinnan suorittamisesta aluksella. Tämä toimitetaan hallintasuunnitelman kautta, jonka koostamiseen ohjeistus listaa viisi askelta kyberriskikohtaisesti: [riskitodennäköisyyden kasvun] tunnistaminen, [ennaltaehkäisevä] suojelu, [riskin tapahtumisen] havainnointi, [riskin poistamiseen pyrkivä] vaste ja palautuminen. Ohjeistusta tukeviksi dokumenteiksi listataan muun muassa BIMCOn tuottama, Kyberturvallisuusohjeistus laivoilla[14], ISO/IEC 27001 yleisstandari IT-järjestelmien

[12] http://www.imo.org/en/OurWork/Security/Guide_to_Maritime_Security/Pages/Cyber-security.aspx

[13] http://www.imo.org/en/OurWork/Security/Guide_to_Maritime_Security/Documents/MSC-FAL.1-Circ.3%20-%20Guidelines%20On%20Maritime%20Cyber%20Risk%20Management%20%28Secretariat%29.pdf

[14] http://www.ics-shipping.org/docs/default-source/resources/safety-security-and-operations/guidelines-on-cyber-security-onboard-ships.pdf?sfvrsn=16

tietoturvasta sekä Yhdysvaltain standardisointielin NIST:n kyberturvallisuuskehys (engl. National Institute of Standards and Technology's Framework for Improving Critical Infrastructure Cybersecurity)[15]. Huomion arvoinen on myös Lloyd's Registerin katsaus tietoteknisesti kytkeytyneiden alusten kehitykseen[16].

Ohje 2.3 (Päättäjä) Autonomisten alusten kehitystyössä tulee vaatia kyberturvallisuuteen liittyvän sääntelyn huomiointia kehitystyön aikana. Järjestelmäkehityksen aikana tulee määrittää mitä kyberturvallisuusohjeistusta on seurattu ja miten sen noudattaminen on validoitu järjestelmän tuottamisen aikana.

Ohje 2.4 (Operaattori) Autonomisen aluksen valvontaan ja etäoperointiin osallistuvien tahojen tulee tuntea kansallinen ja kansainvälinen dynaamisesti muuttuva sääntely, jossa annetaan myös ohjeita kyberriskien kohtaamisen osalta toimintamenettelyistä.

Ohje 2.5 (Kehittäjä) Autonomisten alusten toimintaa säädellään erilaisilla kansainvälisillä ja kansallisilla ohjeilla, joilla on vaikutusta kehityspäätöksiin. Osa näistä liittyy kyberuhkien torjuntaan ja kyberriskien toteutumisesta toipumiseen. Kyberriskien käsittely on saanut ohjeistuksessa viime aikoina lisääntyvässä määrin huomiota.

2.2 Merenkulun järjestelmät ja autonomisuus

Luvussa keskitytään määrittämään merenkulun primääriset ICT-järjestelmät sekä autonomisuuden käsite. Merenkulun pääjärjestelmien käsittely on jaettu kahteen osaan: aluksen sisäisiin ja aluksen ulkopuolisiin järjestelmiin. Autonomisuus-käsitteen määrittelyn lisäksi keskustellaan myös yleisellä tasolla siitä, miten edellisiin järjestelmiin autonomisuus voidaan tuoda sekä konversion että alusta alkaen autonomiseksi suunnittelun kautta.

2.2.1 Aluksen sisäiset järjestelmät

Modernien aluksien mukana toimitetut ICT-järjestelmä tulevat usein kaikki samalta toimittajalta ja järjestelmäkokonaisuus on suunniteltu kyseistä laivaa varten. Vanhempien alusten kohdalla järjestelmät taas ovat usein toimintakohtaisia, eri valmistajilta ja/tai eri aikakausilta riippuen alukseen tehdyistä uudistustöistä. Luvussa 2.1.2 kerrottiin IMOn ohjeistus merenkulun kyber-

[15] https://www.nist.gov/cyberframework

[16] https://www.arbitrage-maritime.org/fr/Gazette/G43complement/lloyds.pdf

riskien hallintaan. Tämän mukaan aluksissa erityisesti komentosilta-, lastinkäsittely-, propulsio-, koneistonhallinta-, alukselle pääsynhallinta-, matkustajapalvelu-, matkustajaviihde-, hallinto-, miehistöhyvinvointi- sekä kommunikaatiotoimiin liittyvät järjestelmät ovat kyberriskialttiita. Seuraavassa käydään läpi nämä sekä muut olennaisimmat alusten järjestelmätyypit.

Navigointijärjestelmät vastaavat aluksen paikan, sijainnin ja suunnan määrittämisestä sekä tämän paikan välittämisestä aluksen miehistön, toisten alusten sekä vesialueen viranomaisten tietoon. IMO listaa olennaisimmiksi navigointijärjestelmiksi kompassin, merikartat, signaalijärjestelmän, komentosillan navigoinnin varoitusjärjestelmän (BNWAS, engl. Bridge Navigational Watch Alarm System), tilannekuvajärjestelmät, automaatisen seuranta-avustimen (AIS-laite) sekä suunnanpitojärjestelmän ja elektronisen karttapiirto- ja tietojärjestelmän (ECDIS, engl. electronic chart display and information system). Edellisen listauksen myöhempänä listatut järjestelmät tarvitaan ainoastaan suurempiin aluksiin.

Turvallisuusjärjestelmien tarkoitus on estää alusten yhteentörmäyksiä sekä toimittaa rannikovaltioille tarvittavaa tietoa aluksesta, jotta ne voivat ylläpitää turvallisuutta alueella. IMOn listaamia olennaisimpia järjestelmiä ovat kansainvälinen merenkulun hätä- ja turvallisuusjärjestelmä (GMDSS, engl. Global Maritime Distress and Safety System), automaattinen tunnistusjärjestelmä (AIS, engl. Automatic Identification System), pitkän kantaman tunnistus- ja seurantajärjestelmä (LRIT, engl. Long-Range Identification and Tracking), tilannekuvajärjestelmät sekä retrospektiivinen mustalaatikko-järjestelmä (VDS, engl. Voyage Data Recorder). GMDSS:än vaaditut komponentit määräytyvät sen mukaan, millä vesialueella alus liikkuu. Niihin lukeutuvat EPIRB-hätäpaikannuslähetin (engl. Emergency position-indicating radio beacon), Navtex-järjestelmä muunmuassa pelastus- ja säätietojen välittämiseen sekä SART-pelastuspaikannustransponderi. IMO määrittää myös, että laitteiston komponenttien pitää käyttää digitaaliselektiivikutsutoimintoa (engl. Digital Selective Calling) tai satelliitteja viestintäänsä. INMARSAT on tällä hetkellä ainut GMDSS-hyväksytty satelliittijärjestelmä.

AIS toimii usein ECDIS-järjestelmän päällä ja integroi GPS-paikkatiedon (engl. Global Positioning System) viestiäkseen aluksen paikan, nopeuden ja suunnan esimerkiksi rannikkovalvontaan. AIS toimii primääristi kommunikoimalla maanpäällisten asemien kanssa, mutta se on laajennettu myöhemmin käytännössä aukottomaan satelliittikommunikaatioon S-AIS-järjestelmänä (engl. Satellite-AIS). AIS:n kattavuuden takia se on tärkein meriliikenneohjauksen väline yli 300 tonnin aluksille sekä kaikille matkustajalautoille. LRIT-järjestelmä on myöhemmin ratifioitu käyttöön ja kuvaa varsinaisen satelliittipohjaisen paikannus- ja tunnistautumisjärjestelmän edellä mainituille alusluokille pakolliseksi.

Matkatallenninjärjestelmä VDR (engl. Voyage Data Recorder, ns. ”musta laatikko”) tarvitaan kaikkiin yli 3000 tonnin aluksiin. Se tallentaa aluksen navigointioperaatiot sekä AIS-tyyppiset tiedot koko reitin ajalta. Mustalaatikko on yleisesti käytetty dokumentointijärjestelmä, jolla voidaan tarkentaa mahdollista turman jälkeistä tutkimusta.

Tilannekuvajärjestelmät on laaja kategoria. Siihen kuuluvat kaikki sellaiset komponentit, jotka tuottavat tietoa aluksen tai sen ympäristön tilasta. IMO vaatii, että aluksilla on niiden tyypistä riippuen kaikuluotain, tutka, ECDIS, AIS, nopeus- ja etäisyysmittari, peräsimien, ruoripotkurien, työntövoimien sekä kallistuskulmien osoittimet. Osoittimien tuottama tieto pitää myös olla muodossa, jossa niiden tuottama tieto on nopeasti ymmärrettävää ja oikealla paikallaan, esimerkiksi komentosillalla.

Nykypäivänä kasvanut ICT-järjestelmien laskentateho mahdollistaa kompleksisten data-analytiikkamenetelmien soveltamisen tilannekuvaan. Edellä kuvattujen vaadittujen mittausarvojen yhdistäminen esimerkiksi kaaviohin tai liikkuvan kuvaan mahdollistaa ihmisluettavan ja jalostetun tiedon palvelun osaksi aluksen operointia. Tästä esimerkkeinä vaikkapa Rolls-Roycen esittämät konseptit propulsiojärjestelmän etäohjatusta vian paikannuksesta ja korjauksesta[17] tai aluksen navigointia helpottava parannettu kokonaistilakuva [22], jossa liikkuvan kuvan, luotauksen yhdistäminen sensorifuusioon tuottaa ajantasaista tietoa horisontista sekä vesiväylällä olevista esteistä.

Aluksen sisäisen tilannekuvan kannalta hyötykuorman ja/tai matkustajien tilanne on myös tärkeä. Hyötykuorman tyypistä riippuen aluksilla käytetään yleisesti ruumaluotaimia, joiden tarkoitus on kertoa, miten kuorma on järjestynyt aluksen sisällä esimerkiksi painolastin tasaamisen avustamiseksi. Vaarallisten aineiden kohdalla myös tarvittavat turvallisuusvälineet, kuten kaasuuntumismittarit, kuuluvat tähän. Öljytankkerit esimerkiksi pumppaavat tankkien tyhjään tilaan jalokaasuja, syttymisriskin minimoimiseksi. Matkustaja-aluksilla, palo- ja häkäilmaisimet ovat yleisen turvallisuusmääritysten mukaisesti pakollisia.

Muut alusten järjestelmät ovat hyvin moninaisia ja aluksen käyttökohteesta riippuvaisia. Aluksilla, joilla operoi useita henkilöitä on yleisesti oma sisäverkko, jossa toimivat aluksen kriittisistä järjestelmistä erilliset yleishyödylliset järjestelmät. Näistä yleisimpiä ovat vapaa-ajan tietoliikennejärjestelmät. Tutkimus- ja erikoistoiminta-aluksilla on usein tutkimuksen tai erikoistoimen suorittamiseen liittyviä järjestelmiä.

IMO on säätänyt, että kaikkien pakollisten komentosiltajärjestelmien (navigointi, turvallisuus sekä tilannekuva) asennus pitää olla sellainen, että yhden järjestelmän rikkoutuminen saatetaan välittömästi hälytysjärjestelmän (ääni- sekä visuaaliset merkit) kautta aluksen vahtipäällystön tietoon. Lisäksi yksittäisen järjestelmän vikaantuminen ei saa aiheuttaa toimintahäiriöitä toiselle järjestelmälle.

Ohje 2.6 (Kehittäjä) Autonomisen aluksen sisäisiä järjestelmiä on tarve valvoa ja tarpeen vaatiessa niihin tulee soveltaa etäohjausta. Tällöin sisäiset ICT-järjestelmät altistuvat ulkoisen rajapinnan kautta tuleville kyberriskeille, jotka tulee hallita jo kehitystoiminnassa.

Ohje 2.7 (Kehittäjä) Autonomisella aluksella voi olla alustyypistä riippuen ICT-järjestelmiä, jotka on suunnattu hyötykuorman (lasti, ihmisiä, ...) suuntaan. Kyseiset sisäiset hyötykuorman järjestelmät muodostavat mahdollisen kyberriskin varsinaisille autonomisuuteen liittyville aluksen järjestelmille. Ensisijainen turvallisuusohje on eristää näitä järjestelmätyyppejä mahdollisimman paljon toisistaan.

[17] Rolls-Royce, Future Shore Control Centre `https://www.youtube.com/watch?v=vg0A9Ve7SxE`

2.2.2 Aluksen ulkopuoliset järjestelmät

Alusten ulkopuolella toimii huomattava määrä tukijärjestelmiä, jotka ovat yhteydessä aluksiin eri tyyppisten kommunikaatioväylien läpi. Järjestelmät voidaan jakaa karkeasti kansainvälisen sääntelyn vaatimiin ja muihin järjestelmiin.

Sääntelyn vaatimia järjestelmiä ovat edellisessä luvussa 2.2.1 keskustellut IMOn määrittämät navigointi-, turvallisuus- sekä tilannekuvajärjestelmät. Osa niistä vaatii maissa olevan tukijärjestelmän toimiakseen. Esimerkiksi S-AIS- ja LRAT- järjestelmät identifiontiin ja paikannukseen kuin myös INMARSAT-satellittikommunikaatiojärjestelmä vaativat maissa olevat palvelinjärjestelyt. S-AIS ja LRAT vaativat tämän primääristi maiden välisen tiedonvälityksen fasilitoimiseksi. INMARSATin ylöslinkitys ja ohjausjärjestelyt tapahtuvat kansainvälisen dynaamisten satelliittien ohjauskeskusten kautta.

IMO määrittä, että rannikkovaltioiden pitää ylläpitää VTS-alusliikennöintipalveluja (engl. Vessel Traffic Services). VTS on laaja määritys[18] ja se kattaa kaikki maissa olevat järjestelmät, jotka toimittavat ulkovesialueilla oleville aluksille viestejä muista liikennöivistä aluksista, säätiedoista sekä vaara- ja pelastustilanteista. AIS-, S-AIS-, LRAT- sekä DSC-pohjaisen viestinnän fasilitointi lukeutuu tähän. Sisävesialuilla VTS:n vastuulle kuuluu kokonaisvaltainen liikennöinninjärjestely satama- ja kanava-alueilla.

Säädösten vaatima kommunikaatio alusten ja toisten alusten tai maa-asemien välillä on määriteltyä. Puhutun kielen kommunikaation tulisi noudattaa ulommilla vesialueilla IMOn SMCP-kommunikaatiosanastoa (engl. Standard Marine Communication Phrases). Ne määrittävät englanninkielisen sanaston ja lauserakenteen, jolla navigointi- ja turvallisuustilanteista pitää kommunikoida aluksen sisällä, alusten välillä ja maihin. Samoin konekielisille turvallisuus- ja navigointijärjestelmilleen on omat määrityksensä. Esimerksi Navtex (engl. Navigational Teletext), joka on osa GMDSSää, määrittää koodiston, jolla laivat voivat lähettää toisilleen viestejä.

Muut järjestelmät on erittäin laaja kategoria ja käytännössä varustamo-, operaattori- ja viranomaiskohtaista. Varustamot ovat usein kiinnostuneita diagnosoimaan laivojen toimintaa mahdollisten huolto- ja lisäpalveluiden tuottamiseksi. Operaattorit haluavat analysoida laivastonsa liikkeitä, lastausta ja reititystä toiminnan tuottavuuden optimoimiseksi. Viranomaiset vastaavat liikennöinnistä, mutta esimerkiksi Euroopan alueella tai paikallisesti aluevesillä, viranomainen voi haluta tehostetusti seurata esimerkiksi laivan polttoaineenkäyttöä tai jäte- ja pilssivesipäästöjä.

Alusten siirtyessä etäoperoinnin, autonomisen toiminnan sekä älykkään toiminnan alle tulee järjestelmien kehityksessä eteen kaksi merkittävää pullonkaulaa, joista molemmat liittyvät tiedonsiirtokanavaan: kaistanleveys sekä saatavuus. Kaistanleveysvaatimukset kasvavat, koska uudet järjestelmät vaihtavat jatkuvasti enemmän tietoa keskenään, esimerkiksi liikkuvaa kuvaa. Saatavuus (engl. availability) on yleinen mittari kanavan laadulle ja se tarkoittaa sitä aikaa, jonka kanava on käytettävissä ja pystyy siirtämään nominaalisen kaistanleveytensä verran informaatiota. Esimerkkinä TIA-942 -standardi (engl. Telecommunications Industry Association) määrittää neljä saatavuuden tasoa palvelininfrastruktuureille. Näistä jo ensimmäinen taso odottaa 99,671% saatavuutta, joka merkitsee, että palvelu voi olla saavuttamattomissa ainoastaan vajaat 29 tuntia

[18] Alusliikennepalveluja ovat tiedotukset, navigointiapu ja alusliikenteen järjestely: `https://www.liikennevirasto.fi/ammattimerenkulku/meriliikenteen-ohjaus/vts#.Wv6xZWcUmP8`

Taulukko 2.2: Tietoliikennejärjestelmiä.

Järjestelmä	**Kaistanleveys (Mb/s)**	Toimintasäde (km)
HF	≤ 0.3	≤ 10000
VHF	0.2-0.3	≤ 85
WAVE (802.11p)	27	≤ 1
WLAN (802.11 gen.)	500-1000	≤ 0.1 (erik. järj. ≤ 10)
4G	1000	≤ 2 km (erik. järj. ≤ 70)
5G	3500 - 20000	2-≥ 20

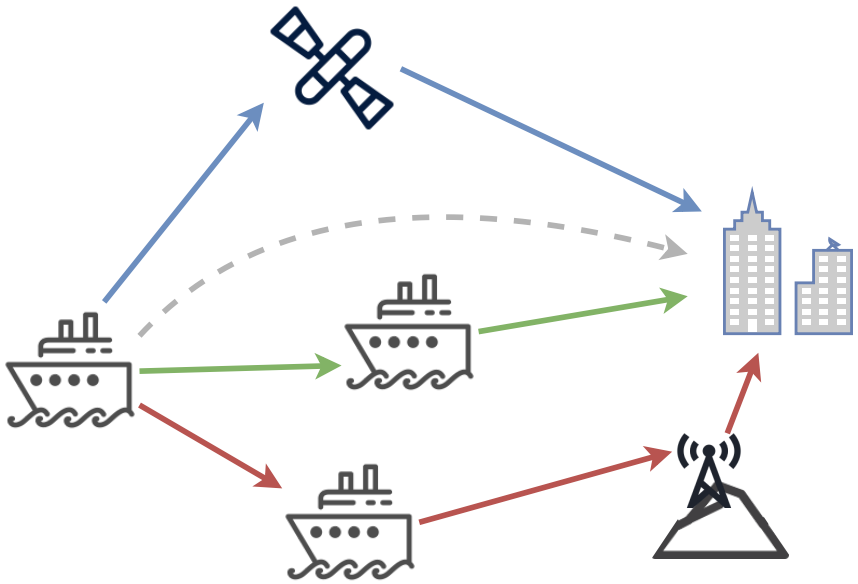
Taulukko 2.3: Arvioita viestityyppien vaatimuksista.

Viestityyppi	Paketin koko (Mb)	Taajuus (Hz)	**Kaistanleveystarve (Mb/s)**
Plotteri (Tutka)	≤ 0.4	0.4	0.1
Infrapuna	≤ 0.4	1-10	0.3-1
Sensorit	≤ 0.02	0.1-1000	0.001-1
SD-video	≤ 0.5	1-10	0.2-1.5
HD-video	≤ 3	1-10	1-5
LiDAR	≤ 200	1	1-2

vuoden aikana.

Yleisesti tietoliikennöintiä koskien seuraava taulukko 2.2 kerää yhteen meriliikennöinnin kannalta relevantteja tietoliikennekanavia sekä näiden yleisiä ominaisuuksia. Näitä voidaan verrata eri viestijärjestelmien vaatimiin tietoliikennejärjestelmiin, jotka on esitetty taulukossa 2.3 [7].

Kun tietoliikennöintiin liittyviä sekä muita haasteita ratkaistaan enenevissä määrin, etäoperointi sekä autonomisen ja älykkään toiminnan seuranta yleistyy. Tämän johdosta edellä mainitut varustamo-, operaattori- sekä viranomaistahot tulevat perustamaan erityisiä autonomisten alusten ohjaus- ja valvontakeskuksia. Ohjauskeskukset ja laivat linkittyvät nyt ja tulevaisuudessa tietoliikennereittien läpi toisiinsa. Tietoliikennereitti on tässä kontekstissa adaptiivisesti päätelty ja automaattisesti ylläpidetty paras tapa toteuttaa reitin päässä olevien viestitahojen tietoliikennetarpeet eli kaistanleveys- ja saavuttavuusvaatimukset. Taulukosta 2.2 nähdään, että edellisten vaatimusten kasvaessa reitti tullaan rakentamaan dynaamisena alus–alus–maa-asema ”monihyppyverkkona”(engl. multi-hop). Verkossa koneellisesti ylläpidetään vaihtoehtoisia, redundantteja, mutta ominaisuuksiltaan heikompia reittejä, jotka on esitetty kuvassa 2.2. Yhden reitin toimituksen estyessä, siirrytään seuraavaksi ominaisuuksiltaan parhaan reitin käyttöön. Mikäli kanava on täysin estynyt, riippuen toiminnon tyypistä, viestinvaihtajat siirtyvät autonomiseen päätöksentekoon tai noudattavat niin sanottuja fail-to-safe -protokollia, kuten hallittua toimintojen alasajoa.



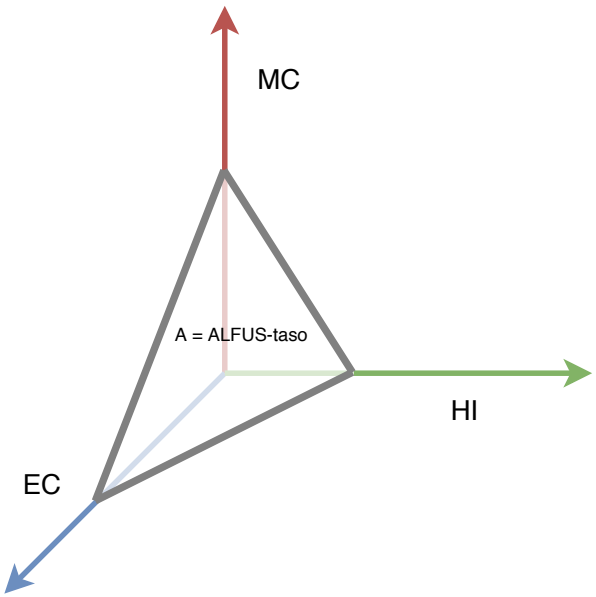
Kuva 2.2: Eri tyyppisiä tapoja tiedonsiirtoreitin perustamiseksi aluksen ja maa-aseman välillä.

Ohje 2.8 (Kehittäjä) Autonomisen aluksen ulkopuoliset järjestelmät ovat selkeä kyberriskien lähde. Alus ei voi välttää kommunikaatiota ulkopuolisten järjestelmien suuntaan määritettyjen rajapintojen kautta. Rajapintojen käytön turvallisuuteen autonomisen aluksen puolella tulee kiinnittää huomiota.

2.2.3 Autonomisuus

Automatisoinnilla tarkoitetaan jonkin tehtävän tai tehtäväosan saattamista koneohjaukseen (eli ICT-järjestelmän logiikan ja vastinparina toimivan fyysisen järjestelmän alle). Jos koneella on tieto tehtävän eri vaiheista ja niistä tiloista, joihin eri vaiheista voidaan oikeellisesti siirtyä, sillä on tarvittava tieto tehtävän läpiviemiseksi (virhetilat on myös saatu koneen tietoon). Yleensä tehtävän korkeasta kompleksisuudesta johtuen päädytään tilanteeseen, jossa tehtävän eri vaiheita ja tiloja ei voida tyhjentävästi mallintaa etukäteen. Tällöin tehtävän suoraviivainen automatisointi ei ole mahdollista. Mikäli tehtävä halutaan kuitenkin saattaa koneohjauksen alle, tulee koneen pystyä viemään tehtävää eteenpäin ennalta mallintamattomasta tilasta toiseen. Autonomisuudella tarkoitetaan koneen kykyä itsenäiseen päätöksentekoon. Jotta siirtyminen tehtävän tilasta toiseen, ilman tyhjentävää ennakkotietoa olisi mahdollinen, koneen pitää pystyä tulkitsemaan ympäristöään, tehtävän nykytilaa ja muodostamaan priorisoitu seuraavien tehtävän tilojen lista.

Koneoppiminen on merkittävä tietojenkäsittelytieteen ala, muun muassa autonomisuuden



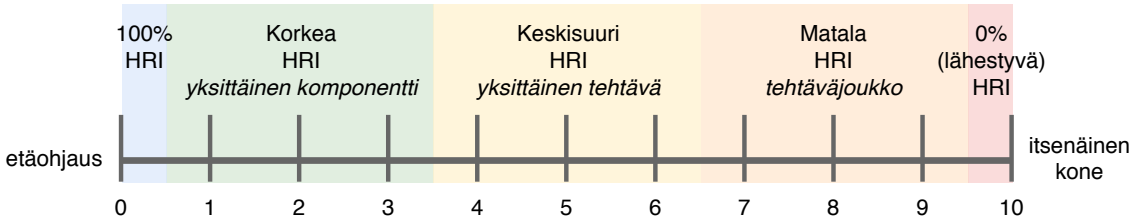
Kuva 2.3: ALFUS-metriikan laskeminen.

mahdollistajana. Koneoppimiselle tarkoitetaan yleisesti erityyppisten ympäristömuuttujien kvantittamista koneluettavaan muotoon sekä erinäisten ennustealgoritmien soveltamista tälle tiedolle mahdollisimman kattavan ja tarkan ennustemallin luomiseksi. Yleisin koneoppimistapa on historiatietoon perustuva, jossa koneelle annetaan raakadatan lisäksi tieto, siitä mitä tilasiirtymiä ihminen on kullakin hetkellä tehnyt. Algoritmisesti tästä voidaan tehdä ennustemalli, jossa tietyt ympäristöstä observoidut kvantit ja aiempien tilojen sarja johtaa tietyn seuraavan tilan valintaan.

Parasuman ja Sheridan ovat keskustelleet jo toistakymmentä vuotta sitten eri automaation ja autonomian tasoista ja niiden vaikutuksista tehtävien suorittamiseen [19]. Tämän työn päälle on sittemmin tehty huomattava määrä parannuksia ja kontekstiperustaisia tarkennuksia. Viimeisimpänä ja merenkulkuun hyvin soveltuvana voidaan pitää ALFUS-mallia (Autonomian tasot miehittämättömille järjestelmille, engl. Autonomy Levels for Unmanned Systems)[19], joka on Yhdysvaltain kansallisen teknologia- ja standardointijärjestön NIST:n johtaman projektin tuotos. Kuva 2.3 listaa ALFUSin käyttämät metriikat järjestelmän autonomian tason määritykseen ja kuva 2.4 listaa metriikoista aggregoitavat mahdolliset autonomian tasot.

ALFUS käyttää kolmea metriikkaa autonomian tason määritykseen kuten kuvasta 2.3 nähdään. Tehtävän kompleksisuus MC (engl. Mission Complexity) ottaa kantaa muun muassa siihen, montako alitehtävää tulee suorittaa itse tehtävän valmistumiseksi, paljonko ennakkoymmärrystä tehtävän suoritus vaatii ja miten tärkeää organisoituminen ja yhteistyö muiden tahojen kanssa on tehtävän suorituksen kannalta. Ihmisriippumattomuus HI (engl. Human Independence) kuvaa,

[19] https://www.nist.gov/sites/default/files/documents/el/isd/ks/ALFUS-BG.pdf

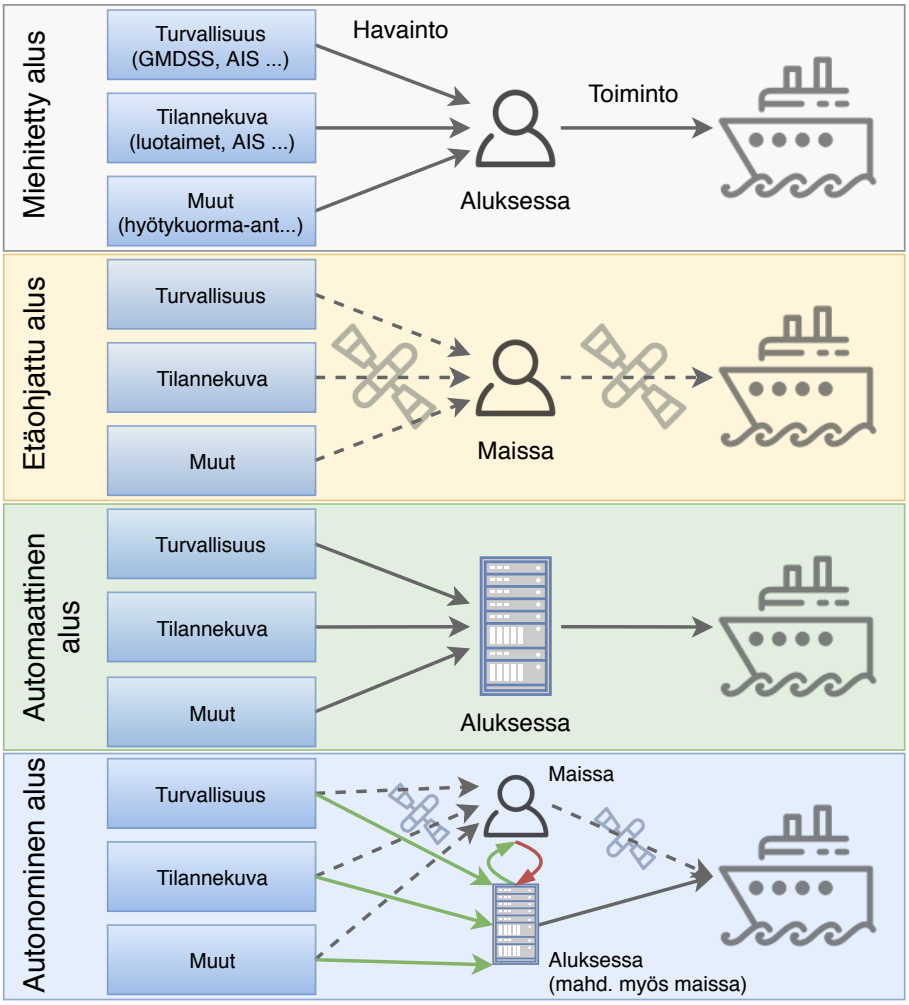


Kuva 2.4: ALFUS-tasot.

miten usein ihmiseltä tarvitaan tietoa tai reaktiota, jotta koneen edistämä tehtävä saadaan suoritettua ja miten kuormittavaa ja vaativaa työn suoritus on ihmiselle, kun kone on vastuullinen tehtävän suorituksesta. Ympäristön kompleksisuus EC (engl. Environmental Complexity) mittaa kuinka haastellinen toimintaympäristö on esimerkiksi ilmaston, pinnanmuotojen, seurattavien objektien haastavan määrän ja/tai muodon takia sekä onko tiedonsiirtoreitti kaistanleveydeltään ja saatavuudeltaan riittävä. Metriikoiden määritys tehdään usein matalamman tason tehtäville ja se aggregoidaan ylemmälle tasolle kokonaisuuksien muodostamiseksi. Riippuen tehtävän kriittisyydestä matalampi pisteytys voi olla täysin hyväksyttävää osalle tehtävistä.

ALFUS-mallin määrittämät autonomian tasot (ks. kuva 2.4) saadaan aggregoimalla mallin määrittämistä metriikoista. Tasot ovat varsinaisesti pisteitä jatkuvalta lukuväliltä 0 - 10. Taso nolla tarkoittaa täydellistä HRI-riippuvuutta (ihmis-robotti-vuorovaikutus, engl. Human Robot Interaction), eli etäohjausta. Siirryttäessä tasovälille 0.5 - 3.5, HRI-vaatimus on edelleen korkea, mutta kone pystyy käyttämään yksittäistä aktuaattoria tai alitoimintoa itsenäisesti yksinkertaisessa ympäristössä. Tasovälillä 3.5 - 6.5 HRI-vaatimus on osittaista, mutta kone pystyy jo vastaamaan yksittäisestä selkeästi määritetystä toiminnosta itsenäisesti normaalissa ympäristössä. Tasovälillä 6.5 - 9.5 järjestelmä on enää matalasti riippuvainen HRIstä ja se pystyy vastaamaan useamman toiminnon muodostamista tehtäväkokonaisuuksista, itsenäisesti, haastavissa ympäristöissä ja se pystyy myös yhteistyöhön: huomioimaan ja käyttämään hyödykseen muiden itsenäisten järjestelmien tuottamia toimintoja. Tasoväli 9.5 - 10 merkitsee täydellisen HRI-riippumattomuuden lähestymistä, jossa järjestelmä pystyy saumattomasti tekemään monien muiden autonomisten järjestelmien kanssa haastavissa ympäristöissä kompleksisia tehtäviä.

Ohje 2.9 (Päättäjä) Autonomisuuden taso vaikuttaa myös kybertuvallisuuden vaatimuksiin ja riskeihin – ALFUS:n kaikki kolme ulottuvuutta ovat tässäkin mielessä merkityksellisiä. Turvallisuuden toteuttamista ei voi tehdä täysin ilman päätöstä siitä, minkä tason mukaista autonomista alusta ollaan kehittämässä.tieto



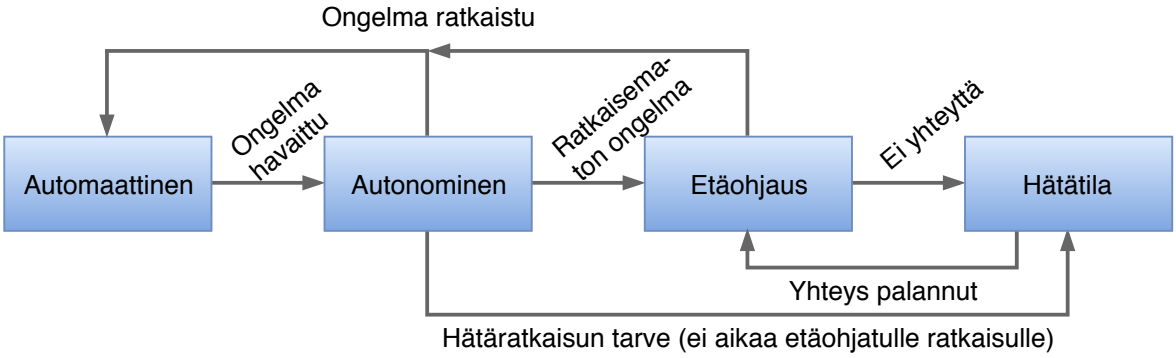
Kuva 2.5: Autonomisuuteen liittyvät alustyypit.[20]

2.2.4 Alusten kehittäminen autonomiseksi

Merenkulun autonomisuustason korotuksesta on tehty viimeaikoina useampia selontekoja [21, 13]. Selontekojen konseksuna on syntynyt tiekartta sekä alusjärjestelmien että alusjärjestelmien tehtävien autonomisoinnille. Alusjärjestelmien autonomisoinnin tasot ovat kuvassa 2.5 ja alusjärjestelmien tehtävien autonomisointiprosessi on kuvassa 2.6.

Kuvassa 2.5 ensimmäinen ryhmä *miehitetyt alukset* kuvaa nykytilaa, jossa miehistön jäsenet käyttävät nykyisin vaadittuja järjestelmiä, kuten tutkaa, ECDISää sekä suoraa ympäristön havainnointia aluksen operoinnissa. Tälle operoinnille on kaksi vaihtoehtoista toteutusta: *etäohjattu alus* ja *automaattinen alus*. Etäohjattu alus tarkoittaa, että aluksella ei ole miehistöä vaan aluksen järjestelmistä sen operoinnin kannalta tarpeelliset tiedot siirretään tietoliikennejärjestelmien kautta ohjauskeskuksessa olevan miehistön tietoon. Miehistö operoi alusta ohjauskeskuksessa,

[20] `http://www.unmanned-ship.org/munin/about/the-autonomus-ship/`



Kuva 2.6: Alusjärjestelmien autonomisointiprosessi.

josta tietoliikennejärjestelmät siirtävät operointikäskyt takaisin alukseen toteutettaviksi. Automaattinen alus on täysin itsenäinen. Se havainnoi edellämainituista järjestelmistä tai niiden koneluettavaan muotoon tarkoitetuista korvaavaista toteutuksista aluksen operoinnin kannalta olennaiset tiedot ja operoi alusta. Automaattinen alus on kuitenkin edellisen kappaleen perusteella hyvin vaikea toteuttaa – ainakin nykyisellä ICT-järjestelmien laskenta- ja tiedonsiirtokapasiteetilla – tehtävien ennaltamäärittelemättömyyden takia. Tämän vuoksi tarvitaan autonomisia aluksia, etäohjatun ja automaatisen aluksen symbioosia. Autonominen alus hoitaa osan tehtävistä suoraan itse, eli automaattisesti. Toinen osa tehtävistä siirretään maa-asemalle raskaampaa laskentaa varten, mutta HRI:tä ei tarvita päätöksentekoon. Jäljelle jäävien tehtävien osalta vastuu on HRI:n tai etäoperoinnin kautta ihmisellä.

Edellinen tehtävien vastuujako perustuu alusjärjestelmien tehtävien autonomisointiprosessiin, joka on kuvassa 2.6. Autonomisointiprosessin voidaan olettaa tavoittelevan tehtävän automatisointia. Jos tehtävä on automaattinen, koneella on tieto siitä miten tehtävä tulee hoitaa. Mikäli tehtävän suorituksen aikana tulee ongelma tai ennalta tuntematon tila, pitää koneen tehdä päätös. Koneoppiminen ja muut keskustellut tavat mahdollistavat konepohjaisten päätösten tekemisen. Mikäli tämä ratkaisee tilanteen, tulee kyseisestä päätöksestä ja sen premisseistä osa automatisointia. Mikäli päätöstä ei pystytä tekemään, tarvitaan HRItä, eli ihminen avustaa robottia tai autonomista järjestelmää päätöksen teossa. Koska alus on kaukana ihmisestä, tämä tarkoittaa ihmisen suorittamaa etäoperointia. Jälleen, ihmisen soveltamasta ratkaisumallista – mikäli mahdollista – yritetään tehdä osa automaatiota. Koska aluksella ei ole miehistöä ja autonomisen operoinnin ja etäoperoinnin tiloissa on mahdollisuus, että tehtävää ei pystytä ratkaisemaan oikein, pitää tehtävään olla assosioitu hätätila. Hätätila määrittää yleensä tavan keskeyttää tehtävän suoritus automaattisesti, ja niin, että syntyy mahdollisimman vähän haittaa alukselle ja sen ympäristölle.

Alus ja sen järjestelmät voidaan saattaa autonomiseksi kahdella eri tavalla: konversiolla tai rakentamalla. Konversiossa aikaisemmin miehitettynä operoinut alus muunnetaan autonomisesti operoitavaksi. On hyvin todennäköistä, että konversioita tullaan tekemään huomattavasti, sillä alusten tekninen ikä on hyvin pitkä – 20-60 vuotta tyypistä riippuen – ja konversion vaatimat perusjärjestelmät voivat merkata ainoastaan joidenkin prosenttien lisäsijoitusta alukseen. Konver-

siolla on kuitenkin haasteensa. Huomattavin haaste on järjestelmän kattava uudelleensuunnittelu vasten niitä tehtäviä, joista aluksen pitäisi konversion jälkeen pystyä selviämään autonomisesti. Lähihistoriassamme on useita esimerkkejä siitä, miten suunnittelussa sattuneet huomiottajätöt ovat johtaneet huomattaviin vahinkoihin [38]. Merkittävä haaste suunnittelussa on ymmärtää, miten aikaisemmin mekanisesti operoitu tehtävä muunnetaan digitaaliseen, koneen operoimaan, muotoon. Jo fyysisen vivun poistuminen merkitsee, että vivun tietyssä asennossa oleminen ei enää implisiittisesti sulje tiettyjä toimintoja pois käytöstä, vaan kaikki implisiittiset tilat pitää suunnittelussa osata mallintaa ja syöttää koneluettavaan muotoon; kone ei pysty tekemään vaadittua päätöstä, koska se ei voi tulkita mallintamatonta tilaa. Toinen merkittävä haaste alusten konversion kohdalla on soveltumattomuus ja tästä johtuva tehottomuus. Jollei suunnittelu keksi kaikille miehistölle tarkoitetuille tiloille ja järjestelyille uudelleenkäyttöä tai pysty poistamaan niitä, autonominen alus kärsii näiden rakenteiden olemassaolosta. Autonomisuudella saavutettavat moninaiset hyödyt oletettavasti kuitenkin nostavat kokonaistehokkuuden positiiviseksi.

Toinen vaihtoehto autonomisten alusten toteuttamiseen on alusta asti autonomiseksi rakentaminen. Tällöin vältytään konversion haasteilta koskien kattavaa uudelleenmallintamista ja tässä mahdollisesti tapahtuvilta huomiotta jättämisiltä. Rakentamisen merkittävin hyöty on, että alus on täysin autonomiseksi suunniteltu ja sen ei tarvitse fasilitoida miehistön toimintaa. Aluksen kapasiteettia voidaan kasvattaa, sen säänkestävyyttä voidaan parantaa (alus voi olla jopa pieniä aikoja sukelluksissa) ja siihen kohdistuvat ihmisperäiset riskit madaltuvat: piratismi on kansainvälisen meriliikenteen ongelma, jossa tavoite on pyytää lunnaita miehistöstä tai ottaa aluksen lasti haltuun. Autonomisessa aluksessa – tyypistä riippumatta – ei ole miehistöä ja autonomiseksi suoraan rakennettu alus voidaan tehdä erittäin vaikeaksi murtautua ja kaapata. Autonomiseksi rakennetuilla aluksilla on myös haasteita. Ensisijaisesti monet kansainvälisen merilogistiikan infrastruktuurin osista toimivat oletuksella, että aluksia operoi ja ne on rakennettu miehistöille. Tämä voi aiheuttaa yhteensopivuusongelmia. Toiseksi, alusten optimointi tulee olemaan haastavaa, sillä nopeuden, säänkestävyyden tai lastikapasiteetin maksimointi voidaan tietyissä tapauksissa saavuttaa ainoastaan esimerkiksi heikentämällä aluksen korjausmahdollisuuksia.

Ohje 2.10 (Kehittäjä) Autonomisen (ja automaattisen sekä etäohjatun) aluksen muodostaminen konvertoimalla vanha alus on kybertuvallisuuden kannalta haasteellista alukselle jäävien vanhojen jäjestelmien / kokonaisuuksien rajapintojen osalta. Vanhojen legacy-osien integrointi osaksi uutta autonomisuuden toteuttavaa osiota luo hyökkäysrajapinnan, jossa puuttellisuuksien poistaminen legacy-osioista ei ehkä ole mahdollista. Tällöin kyberturvaa pitää kehittää lähinnä turvautumalla eristämiseen, koventamiseen ja monitorointiin.

Ohje 2.11 (Kehittäjä) Autonomisen aluksen kehittäminen konvertoimalla voi mahdollistaa fyysisen tunkeutumisen alukseen (merellä tai satamassa) liian helposti, jolloin tietoturvaratkaisujen luottamuspohja voi vaarantua ja tämä kyberriski tulee huomioida tietoturvaratkaisujen toteuttamisessa.

2.3 ICT-järjestelmät

ICT-järjestelmät (Informaatio- ja kommunikaatioteknologia, engl. Information and Communication Techology) kattavat kaikki sellaiset analogista ja digitaalista tietoa käsittelevät ja välittävät loogiset järjestelmät, jotka toimivat fyysisen vastinjärjestelmän päällä. Looginen järjestelmä voi samanaikaisesti toimia tai siirtyä usean fyysisen järjestelmän päällä. Seuraavat alikappaleet kuvaavat, miten ICT-järjestelmiä tuotetaan sekä miten ICT-järjestelmien osat muodostavat kokonaisjärjestelmän, kuten merialuksen, tietoturvakriittiset resurssit. ICT-järjestelmien tuoton osalta kuvataan kehityksen vaihejakomalli sekä modernit kehitysmenetelmät hankintahuomioineen.

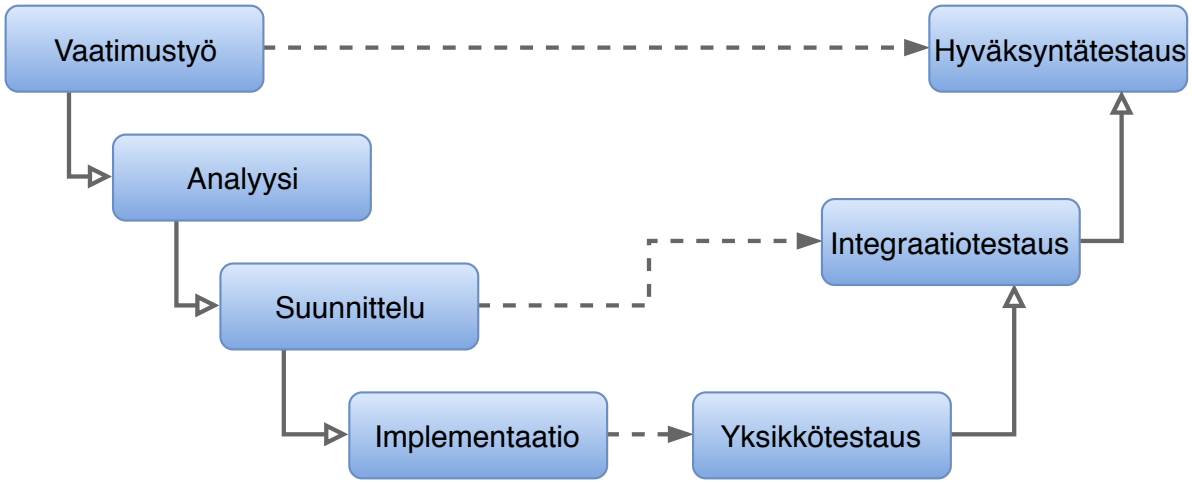
2.3.1 Järjestelmäkehitys

ICT-järjestelmien kehityksestä voidaan toimintakontekstista ja -tavasta riippumatta tunnistaa tietyt vaiheet, jotka toistuvat käytännössä aina. Näiden vaiheiden kuvaamasta prosessista käytetään termiä järjestelmäkehityksen vaihejakomalli. Vaihejakomallista löytyy myös tarkkoja standardeja, kuten ISO/IEC/IEEE 12207:2017 (engl. Systems and software engineering – Software life cycle processes). Vaihejakomallissa on viisi päävaihetta: vaatimustyö, analyysi, suunnittelu, implementaatio sekä testaus.

Vaatimustyövaiheessa asiakasedustajalta kerätään joko uusia vaatimuksia, joita ICT-järjestelmän tulisi tukea, tai tarkennetaan aiemmin kerättyjä vaatimuksia, perustuen esimerkiksi jo valmiin osatuotteen kokeiluun. Koska kaikki myöhempien vaiheiden järjestelmäkehitys perustuu tässä vaiheessa kerätyille vaatimuksille, on äärimmäisen tärkeää, että kerätyt vaatimukset ovat mahdollisimman kattavia, täydellisiä sekä virheettömiä.

Menetelmiä kerätä ja tarkentaa vaatimuksia on huomattavasti. Yleinen tapa kerätä vaatimuksia on kuvata ne käyttötapauksina. Käyttötapaus kertoo mitkä aktorit (ulkoiset järjestelmät tai henkilöt) käyttävät kuvatun järjestelmän vaadittua ominaisuutta, missä tilassa ympäristön ja järjestelmän tulee olla ennen ja jälkeen käyttötapauksen sekä mitä ei-toiminnallisia vaatimuksia kuvaukseen liittyy. Ei-toiminnallisilla vaatimukset ovat laadullisia attribuutteja toimintojen toteutukselle. Esimerkiksi vaatimus järjestelmän toteuttamasta käyttöliittymästä saattaa sisältää ei-toiminnallisia vaatimuksia koskien käyttöliittymän vasteaikaa.

Analyysivaiheessa toteutusorganisaatio käsittelee keräämiään vaatimuksia tarkoitetuksenaan tuottaa alustava analyysimalli. Analyysimallin on tarkoitus kuvata abstraktilla, ei toteutusteknologiaan sidotulla tasolla, millainen looginen järjestelmä voi toteuttaa kaikki kerätyt vaatimukset. Analyysimallissa on yleensä staattinen ja dynaaminen osuus. Staattinen osuus kuvaa järjestelmän



Kuva 2.7: ICT-järjestelmien kehityksen vaihejakomalli.

rakenteen. Dynaaminen osuus kuvaa, miten järjestelmän osat kommunikoivat keskenään, eli logiikan, toimintojen tuottamiseksi. Analyysivaiheessa voidaan huomata tarkennustarpeita vaatimuksille, jolloin siirrytään edelliseen vaiheeseen.

Suunnitteluvaiheessa analyysimalliin kiinnitetään tekninen toteutus. Yleisesti suunnittelu jaetaan kahteen tasoon: korkeaan ja matalaan. Korkean tason suunnittelulla tarkoitetaan ICT-järjestelmän arkkitehtuurin määrittämistä, joka kertoo, minkälaisista moduuleista järjestelmä koostuu, mitkä moduulit keskustelevat keskenään ja mitkä eivät. Korkean tason suunnittelu on tärkeää järjestelmän muokattavuuden ja ylläpidettävyyden takia. Heikosti ryhmitelty järjestelmä on kallis ylläpitää, sillä muutoksia joudutaan riippuvuuksien takia ratkomaan huomattavasti. Matalan tason suunnittelulla tarkoitetaan yksittäisten moduulien toteutusteknologian valintaa ja logiikkakoodin suunnittelua.

Implementaatiovaiheessa suunnitteluvaiheen tuottama järjestelmä toteutetaan. Usein korkean tason suunnittelu mahdollistaa, että moduuleja toteutetaan toisistaan riippumatta.

Testausvaiheessa valmista järjestelmäimplementaatiota testataan. Testauksen on tarkoitus löytää odotetun ja havainnoidun toiminnallisuuden väliltä eroja. ICT-järjestelmissä on niiden loogisen kompleksisuuden ja fyysisen vastinjärjestelmäsidonnaisuuden takia lähes aina vikoja. Kattava testaus on ainut tapa todentaa, että järjestelmä pystyy toimittamaan siltä odotetut toiminnallisuudet tiettyjen raja-arvojen puitteissa. Testauksen tasoja on useita. Matalimmaksi tasoksi usein mielletään yksikkötestaus, joka todentaa yksittäisten koodilauseiden aritmeettistä oikeellisuutta, kun taas korkein taso on hyväksyntätestaus, missä esimerkiksi asiakasedustaja käy vaatimusvaiheen käyttötapauksia itse järjestelmän kanssa läpi.

Ohjelmistokehitysmenetelmä on usein kehitysorganisaation itse valitsema ja muokkaama. Nykypäivänä vallalla ovat niin kutsutut iteratiivis-inkrementaaliset (myös termit Agile ja Lean) menetelmät. Näillä tarkoitetaan menetelmiä, jotka tuottavat ICT-järjestelmästä valmiin osan, eli

inkrementin, aina tietyn kiinnitetyn aikaperiodin, eli iteraation, välein. Näin on tarkoitus taata, että asiakasedustaja pystyy valmiin osatuotteen kautta ymmärtämään mihin suuntaan järjestelmä on menossa, paremmin käsittämään mitä vaatimus valmiina todellisuudessa tarkoittaa ja mahdollisesti mukauttamaan omaa tilaustaan.

Iteratiivis-inkrementaaliset menetelmät ovat viime vuosina muuttaneet ICT-järjestelmien toimitusfilosofiaa huomattavasti. Kehitysorganisaatioiden vastuu asiakkaan osallistamisessa kehitysprosessiin on kasvanut. Asiakkaan tulee kommentoida osatoteutuksen laatua ja toimivuutta sekä ymmärtää, että osatoteutus on osa toimitussopimuksen täyttöä. Samalla asiakkaan vastuu ymmärtää ero fyysinen ja loogisen järjestelmän tilauksen välillä on kasvanut. Fyysinen järjestelmä voidaan mallintaa ja suunnitella valmiiksi ennen varsinaista toteutustyötä. Loogista järjestelmää ei voida mallintaa etukäteen, koska mallinnus vastaa järjestelmän toteutusta.

Edelliseen perustuen asiakkaan tulee ymmärtää, että järjestelmätilausta ei voida tehdä täydellisen etukäteissuunnittelun pohjalle. Toimitussopimuksen tulee sisältää estimaattihaarukka vaadittujen iteraatioiden määrästä, jolla yleistasoisesti määritellystä järjestelmästä saadaan kaikkein olennaisimmat toiminnallisuudet toteutettua sekä optiot uusille iteraatioille myöhemmin tarkentuvien tominnallisuuksien toteuttamiseksi. Tarkemman alussa tehtävän suunnittelun on todistettu olevan täydellistä resurssien haaskausta keskimääräisten resurssipoikkeamien ollessa jopa 800% luokassa.

Ohje 2.12 (Kehittäjä) Edellä kuvattu vaihejakomalli kehitystoiminnan osalta ei sijoita tietoturvaan tai tietoliikenteen turvaamiseen liittyviä asioita johonkin tiettyyn vaiheeseen. Ylipäänsä kybertuvallisuuden käsittely liittyy kehitystoiminnan osalta kaikkiin vaiheisiin – kyberturvaan ei tule suhtautua vain laadullisena asiana, joka voidaan lisätä kokonaisuuteen kehitystoiminnan lopuksi.

Ohje 2.13 (Päättäjä) Autonomisten alusten kyberturvallisuuden saavuttaminen lähtee oikeiden vaatimusten muodostamisessa tilaamisvaiheessa. Tilaajien tulee osata vaatia ratkaisuja, joissa on mahdollista huomioida tieto- ja tietoliikenteen turva hyvin.

Ohje 2.14 (Päättäjä) Nykyään interatiiviset kehitysmenetelmät osallistavat myös tilaajaa. Tilaajalta usein edellytetään edustajaa projektin toteutuksen ohjaamiseen – kyberturvallisuudenkin ymmärtämystä vaaditaan sellaiselta toimijalta.

2.3.2 ICT-järjestelmä tietoturvakriittisenä resurssina

Edellisessä kappaleessa keskusteltiin ICT-järjestelmien kehittämisestä. Nähtiin, että asiakasedustajan vaatimuksiin vastattiin suunnittelemalla ja implementoimalla ICT-järjestelmä, jossa lo-

giikkakoodi käsittelee digitaalisessa muodossa olevaa informaatiota. Logiikkakoodin päätelmät toteutetaan fyysisen vastinparinjärjestelmän kautta toisen järjestelmän tai ihmisen tulkittavissa olevaan muotoon. Tässä luvussa käsitellään lyhyesti sitä, miten näistä järjestelmistä muodostuu tietoturvakriittisiä resursseja ja mikä osuus järjestelmästä on erityinen tietoturvatoimien kohde.

Logiikkakoodilla tarkoitetaan digitaalisen järjestelmän sisällä olevaa koneluettavassa muodossa olevien käskyjen sarjaa. Osa käskyistä käyttää fyysistä vastinparilaitetta ja osa ohjaa uusiin loogista koodia sisältäviin lohkoihin. Logiikkakoodi ja sen rajapintaus fyysiseen vastinparilaitteeseen muodostaa ensimmäisen tietoturvakriittisen resurssin ICT-järjestelmissä. Loogiikkakoodin tulee suorittua oikein ja sen tulee ilmetä fyysisen vastinjärjestelmän läpi odotetulla tavalla.

Logiikkakoodi operoi datalla. Data on digitaalisessa muodossa olevaa tietoa, jonka informaatioarvo vaihtelee. Ennalta kuvatut tehtävien läpiviennit ovat esimerkkejä erittäin korkean informaatioarvon datasta; data lähentelee ymmärrystä. Yksittäisen anturin, esimerkiksi konetilasta mikrofonin ottamatta äänite, voi olla informaatioarvoltaan matala; se on melkein kohinaa. Operoitavan datan oletetaan kuitenkin tulevan, informaatioarvostaan huolimatta, tietystä, tunnetusta lähteestä ja tähän perustuen logiikkakoodissa voidaan tehdä huomattavia olettamia. Täten data, logiikkakoodin ajon mahdollistavana elementtinä muodostaa toisen tietoturvauksen kannalta kriittisen resurssin.

Molemmille keskustelluista tietoturvakriittisistä resursseista, logiikkakoodille sekä datalle, tietoturvanhallinta perustuu luvun 1 esittämän CIA-mallin soveltamiselle. Luvussa 3.3 keskustelemme siitä, miten tämä toteutetaan täysimittaisesti erilaisille ICT-järjestelmille ja niiden osille.

Ohje 2.15 (Kehittäjä) Järjestelmien kyberturvaratkaisut muodostavat myös resurssin, joka vaatii turvaamista. Erityisesti operoivien tahojen (käyttäjät, toiset järjestelmät) tunnistamiseen / todentamiseen tarvitaan tietoa, joka on taltioitava jonnekin osaksi järjestelmäkokonaisuutta. Tällainen tietosisältö voi olla vaikka salasanoja. Näitä tietoresursseja ja toimintoja tulee suojata luottamuksellisuuden ja eheyden saavuttamiseksi. Menetelmiä tähän tarjoavat mm. kryptografia, luotetut laskentaympäristöt ja fyysinen suoja luottamuksellisen tiedon säilyttämiseksi. Automisen aluksen puitteissa eräs haaste on fyysisen suojan luotettavuus.

Luku 3

Kyber- ja ohjelmistoturvallisuus

Tässä luvussa tarkastellaan, miten johdannossa mainitut tietoturvan tavoitteet – luottamuksellisuus, eheys ja saavutettavuus – toteutetaan ICT-järjestelmien kehityksessä ja käytössä. Luvussa luodaan katsaus tietoturvan hallintaan sekä tietoturvaproseissihin, joiden avulla kehitetään järjestelmien kyber- ja ohjelmistoturvallisuutta. Lisäksi käydään läpi myös uhkakuvien tunnistamista ja riskienhallintaa sekä alihankintaketjun ja luottamuksen merkitystä tietoturvan saavuttamisen kannalta. Lopuksi tarkastellaan vielä tietoturvan toteuttamista kyberturvan ja ohjelmistoturvallisuuden teknisten rakennuselementtien kautta. Näitä ovat esimerkiksi tiedon salaaminen, pääsynhallinta, järjestelmien koventaminen ja luotetut suoritusympäristöt. Tietoturvan toteuttamista käytännössä pohditaan kolmella eri tasolla: ICT-järjestelmässä, tiedonsiirrossa sekä loppukäyttäjän näkökulmasta.

Tässä luvussa annetaan erityisesti ohjeita kehittäjille. Ohjeita on lopulta hyvin runsaasti – edelliseen lukuun verrattuna vain osa näistä ohjeista kootaan erikseen listatuiksi ohjeiksi. Itse käsittely on painottunut ohjelmistojärjestelmien laatimiseen liittyvään tietoturvatarkasteluun. Syynä on se, että autonomiset alukset sisältävät lopulta hyvin paljon ohjelmistoilla toteutettuja ratkaisuja[1].

3.1 Tietoturvauhat ja tietoturvan hallinta

Tietoturvauhat voidaan jakaa tahallisiin ja tahattomiin tietoturvan loukkauksiin. Nämä puolestaan voivat olla joko kohdennettuja tai kohdistamattomia. Tahattomia tietourvauhkia ovat mm. virheelliset tietoturva-asetukset tai ohjelmistovirheet, jotka sallivat järjestelmien luvattoman operoinnin tai jättävät osia siitä suojaamattomaksi. Hyökkäykset voidaan suorittaa joko organisaation sisä- ja tai ulkopuolelta, joten tietoturvaa ei voida toteuttaa vain eristäytymällä ulkomaailmasta.

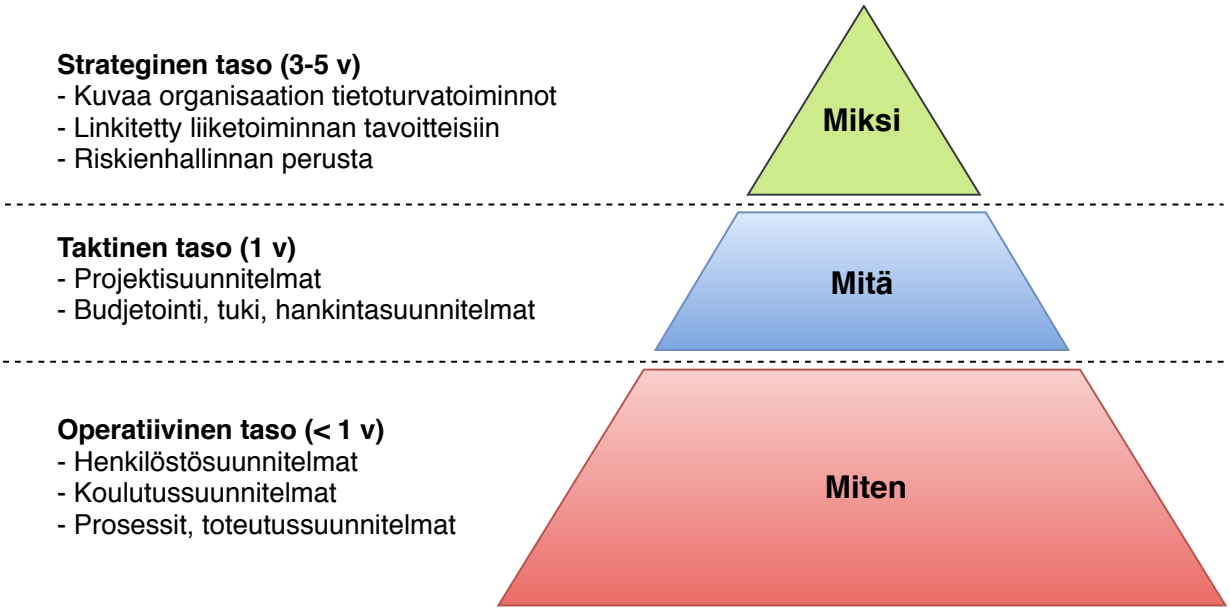
Tahatonkin tietouturvaloukkaus saattaa aiheuttaa tuntuvaa taloudellista, aineellista tai ympäristöllistä haittaa; jopa pelkällä imagohaitalla voi olla pitkäkestoisia vaikutuksia. Tahalliset hyökkäykset puolestaan tehdään yleisesti seuraavista syistä:

[1] Esimerkiksi autonomisia aluksia kehittävä Rolls-Royce Turku Oy kertoo toimivansa ohjelmistoalalla.

- sotilas- ja tiedusteluhyökkäykset
- teollisuusvakoilu
- taloudelliset hyökkäykset
- terrorismi
- kosto tai kauna
- aktivismi, vandalismi, muu rikollisuus

Merenkulku on strategisena ja yhteiskunnalle kriittisenä toimialana altis jokaiselle yllä luetelluista uhkatyypeistä.

Mitä ammattimaisemmin hyökkäys on suoritettu, sitä vähemmän siitä jää todistusaineistoa. Menestyksellinen tietoturvahyökkäys onkin joko sellainen, jota ei huomata lainkaan, tai joka huomataan vasta, kun se on jo tapahtunut. Hyökkääjät saattavat myös pyrkiä peittämään jälkiään ja suuntaamaan ylläpitäjien huomion muualle esim. samanaikaisilla palvelunestohyökkäyksillä tai muulla häirinnällä, pyrkien samalla hankaloittamaan suojautumistoimia. Vastaavasti puolustatuvalla osapuolella saattaa olla omat järjestelmäkohtaiset keinonsa estää ja tutkia hyökkääjien toimintaa vielä hyökkäyksen ollessa käynnissä.



Kuva 3.1: Tietoturvan hallinnan tasot.

Uhat voivat kohdistua itse alukseen, sen lastiin, matkustajiin, miehistöön ja operoitsijoihin, varustamoon, omistajiin, ympäristöön – tai jopa näihin liittyviin valtioihin. Tietoturvan toteutukseen ja sen hallintaan laaditaan tietoturvasuunnitelma, joka määritellään kolmella tasolla (katso myös kuva 3.1):

- Strateginen taso, joka kertoo tavoitteet ja jota muutetaan harvoin.
- Taktinen taso, joka tarkentaa strategiaa ja esittää keinoja siihen pääsemiseksi.
- Operatiivinen taso, joka määrittää toteutetuksen.

Uhkien torjunnan perustana toimii hallintamalli, jonka avulla tietoturvauhat pyritään kartoittamaan ja ennakoimaan, uhkien ja haavoittuvuuksien hyväksikäyttö estämään sekä minimoimaan niiden vaikutukset, jos estotoimet kuitenkin epäonnistuvat.

3.1.1 Riskienhallinta

Riskien hallinta on keskeinen osa tietoturvan hallinnan suunnittelua ja se ohjaa myös työn käytännön toteutusta. Tietoturvariskit tulee aina huomioida erillään yleisistä liiketoiminnan ja esim. merenkulun yleisistä riskeistä. Yleiset liiketoiminnan riskienhallintaprosessit saattavat kuitenkin auttaa tunnistamaan myös tietoturvaan liittyviä riskejä. Tietoturvariskit ovat poikkeuksetta negatiivisia, eli niitä pyritään aktiivisesti välttämään ja niiden vaikutuksia pienentämään. Riskienhallintaprosessi jaetaan yleisesti viiteen päävaiheeseen:

Vaihe 1: Riskien tunnistaminen. Tietoturvariskit tunnistetaan, listataan ja kirjataan ylös. Riskien tunnistamiseen voidaan käyttää tietoturva-asiantuntemuksen lisäksi tekniikoita, joilla voidaan havaita piileviä riskejä. Riskejä varten tulee perustaa oma tietovarastonsa.

Vaihe 2: Riskien analysointi. Tässä vaiheessa arvioidaan kunkin riskin todennäköisyys ja sen seuraukset pyritään ymmärtämään. Tässä vaiheessa on tärkeää ymmärtää riskien luonne ja vaikutukset. Analyysin tulos lisätään kunkin tunnistetun riskin tietoihin.

Vaihe 3: Riskien arviointi ja luokittelu. Riskeille annetaan prioriteetit niiden todennäköisyyden ja seurausten vakavuuden perusteella. Tässä vaiheessa tehdään myös päätös siitä, voidaanko riskin olemassaolo hyväksyä vai vaatiiko se toimenpiteitä. Riskien prioriteetit kirjataan ylös.

Vaihe 4: Riskien valvonta ja torjunta. Riskit, jotka vaativat toimenpiteitä, käsitellään prioriteettijärjestyksessä ja niille pyritään määrittelemään toimenpiteet, joilla ne joko vältetään tai ainakin niiden mahdolliset haittavaikutukset ja jälkiseuraamukset saadaan hallintaan. Torjuntakeinoja ovat riskienhallintastrategiat ja -suunnitelmat, sekä esimerkiksi erilaiset varautumisjärjestelyt. Nämä keinot lisätään riskienhallinnan tietovarastoon.

Vaihe 5: Riskienhallinnan seuranta ja arviointi. Riskienhallinnan tietovaraston sisältöä tulee seurata ja päivittää aina tarvittaessa. Tämä voidaan tehdä määräajoin tai esimerkiksi uusien tietojärjestelmien käyttönoton tai aina järjestelmäpäivitysten yhteydessä. Myös toimintaympäristön muutokset voivat muuttaa riskien arviointiperusteita, tai luoda uusia tai jopa kokonaan uudentyyppisiä riskejä.

Riskienhallinta pienentää riskien todennäköisyyttä, mutta tietoturvauhka voi silti toteutua: tarvitaan siis varautumistoimenpiteitä, joilla torjutaan ns. *jäännösriski*. Näissä tapauksissa tietojärjestelmien tietoturva nojaa yleisiin ohjelmistojen turvallisuutta lisääviin toimenpiteisiin ja varmistusmekanismiehin. Näitä ovat mm. varmuuskopiot, jotka varastoidaan mieluiten järjestelmästä erillään, sekä fyysiset varajärjestelmät.

Riskienhallinnan malleja ja käytäntöjä ovat esimerkiksi Process Hazard Analysis (PHA) sekä Failure Modes and Effects Analysis (FMEA) ja sitä ohjaavat ISO 31000-sarjan standardit. Riskienhallintaa suunnitellessa jäännösriskin olemassaolo pitää tiedostaa: hyväksyttävä riskitaso on sellainen, että riskin torjumisesta aiheutuva lisäkustannus ylittäisi riskin seuraamusten aiheuttamat menetykset, riskin todennäköisyys huomioiden. Menetykset voivat olla myös epäsuoria tai niille voi olla vaikea määrittää suoraa rahallista arvoa.

3.1.2 Hallintamallit

Tietojärjestelmiä, niiden käyttöympäristöjä, niihin kohdistuvia muutoksia sekä esimerkiksi käyttäjien sähköistä identiteettiä voidaan hallita lukuisilla eri tavoilla. Hyvin hallittua tietojärjestelmäympäristöä voidaankin pitää yhtenä keskeisistä toimintaedellytyksistä ja hyvä hallintomalli huomioi myös tietoturvan kaikki osa-alueet. Yleisesti käytettyjä tietojärjestelmien hallintomalleja ovat mm. COBIT (Control Objectives for Information and Related Technologies), avoimesti saatavilla oleva OSSTMM (Open Source Security Testing Methodology Manual) sekä tietotekniikan ja sen palveluiden hallintamalli ITIL (Information Technology Infrastructure Library). Hallintomalleja on myös standardoitu. Näistä tietoturvan kannalta eräänä keskeisimmistä on ISO/IEC 27000-sarja. Prosesseja voidaan lisäksi hallita ja pyrkiä parantamaan erilaisilla laadullisilla menetelmillä (esim. Six Sigma), tai kypsyysmalleilla kuten CMMI (Capability Maturity Model Integration).

Hallintomallit määrittävät prosesseja, käytäntöjä ja konkreettisia tehtävälistoja järjestelmien ylläpitoon ja niiden muutoksien hallintaan. Lisäksi niissä voidaan kuvata esimerkiksi tietoturvan kannalta olennaisia rooleja (kts. Ohje 3.1) sekä näiden tehtävät tietoturvan hallinnan eri vaiheissa.

Tietoturvan hallinta käsittää kaikki tietoturvan osa-alueet aina prosessien määrittelystä salausmenetelmien valintaan. Tietoturvasuunnittelussa on huomioitava mm. lait ja säädökset, teknologiset komponentit, yhteistoiminta alihankkijoiden kanssa, järjestelmien fyysinen turvallisuus, tiedonsiirto, henkilöstöön liittyvät tietoturvariskit sekä esimerkiksi eettiset arvot. Erityisesti muutostilanteet, kuten tietojärjestelmien tai niiden osien päivitykset tai kokonaan uusien järjestestelmien käyttöönotto ovat tietoturvakäytäntöjen kannalta kriittisiä tapahtumia. Myös varautumistoimet voidaan lukea tietoturvatoiminnan piiriin.

Eräs keskeisimmistä tietoturvan hallinnan tehtävistä on tietoturvahäiriöiden ja -poikkeamien hallinta (information security incident management). Tämä on osa ns. jäännösriskin hallintaa, ja sillä havaitaan ja rajataan häiriön aiheuttama vahinko sekä pyritään minimoimaan sen vaikutukset. Joissakin tapauksissa tietoturvahäiriöiden hallintaan voi sisältyä myös ei-teknisiä komponentteja, kuten sisäistä tai julkista tiedottamista sekä muita erikoistilanteiden hallintaan liittyviä toimenpiteitä.

Ohje 3.1 Tietoturvahallinnon rooliesimerkkejä

Tietoturvajohtaja, joka on viime kädessä vastuussa kaikista tietoturvaprosesseista sekä esimerkiksi tietoturvaohjeiden ajantasaisuudesta ja saatavuudesta.
Tietoturva-asiantuntija, joka huolehtii mm. tiedon luokittelusta ja käsittelystä sen elinkaaren eri vaiheissa, sekä tietoturvakäytäntöjen toteutuksesta ja tehokkuudesta.
Tiedon omistaja, esimerkiksi liiketoimintayksikön johtaja. Vastaa mm. tietojen vaatimuksenmukaisuudesta ja käyttökelpoisuudesta.
Tiedon hallinnoija, järjestelmän ylläpitäjä ja operaattori. Huolehtii mm. käyttöoikeuksien hallinnasta ja järjestelmän fyysisestä ja loogisesta toiminnallisuudesta.
Tiedon käyttäjä. Huolehtii siitä, että tietoturvakäytäntöjä noudatetaan ja mm. raportoi havaitsemansa poikkeamat.

Kokonaisriskin pienentämiseksi tietoturva on toteutettava kaikissa tietojärjestelmiin liittyvissä toimissa, periaatteista ja strategioista lähtien aina käyttäjien toimintoihin asti. Asianmukaiset ja liiketoiminnan kannalta tärkeät tietoturvatoimet yhdessä muiden laadunvarmistusmekanismien kanssa auttavat arvioimaan ja rajaamaan tietoturvahyökkäyksen vaikutuksia. Lisäksi tietoturvaloukkauksen sattuessa organisaation on joissakin tapauksissa kyettävä osoittamaan, että asianmukaiset käytännöt on määritelty ja niitä on myös noudatettu asianmukaista huolellisuutta ja tarkkaavaisuutta noudattaen.

3.1.3 Määritelmiä

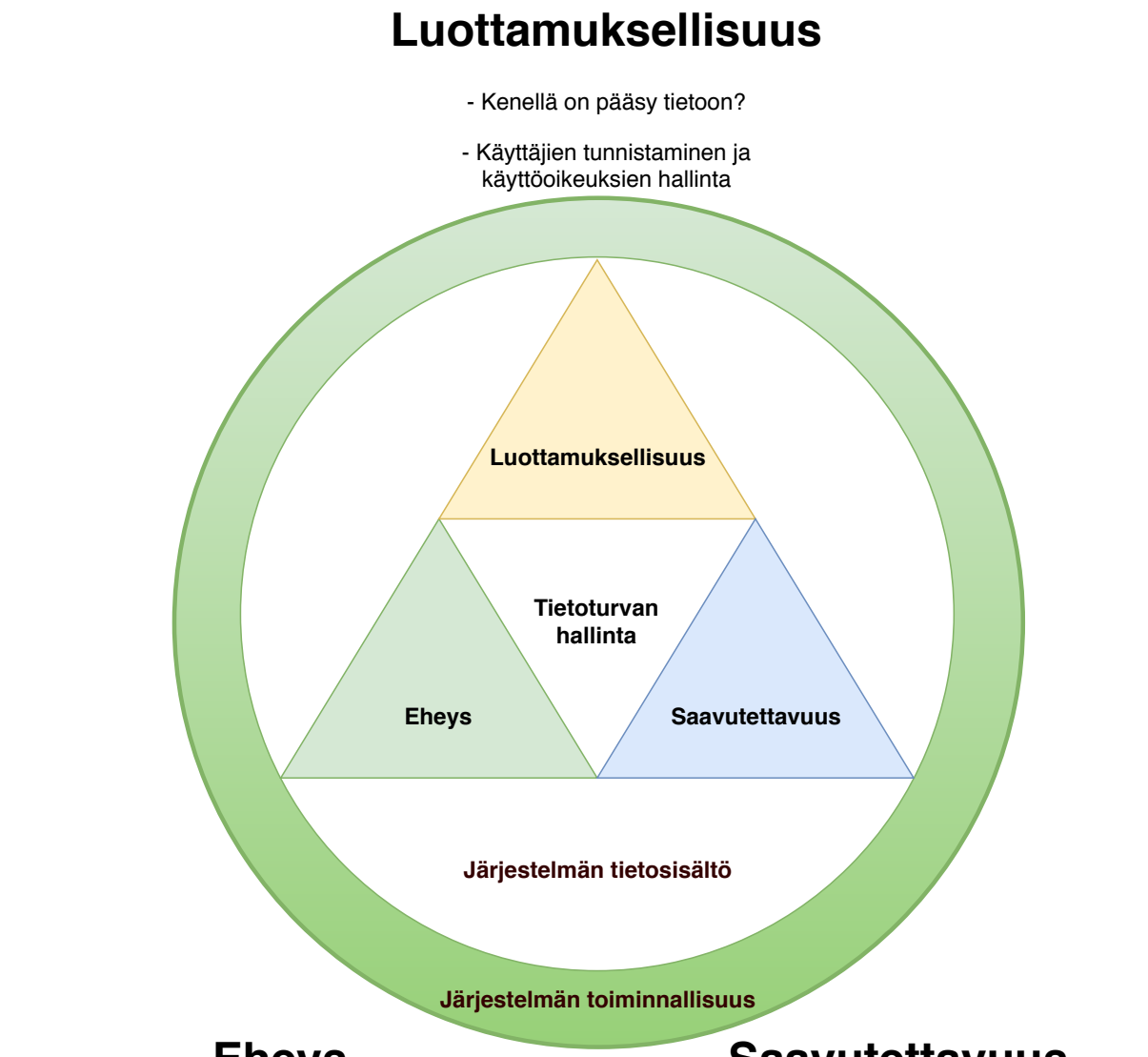
Luvussa 1 käsiteltiin kolmea tietoturvan ytimessä olevaa perusarvoa: luottamuksellisuutta, eheyttä ja saavutettavuutta (ks. kuva 3.2). Lyhyesti näiden käsitteiden merkitykset ovat seuraavat:

Luottamuksellisuus tarkoittaa, että resurssia voivat käyttää vain sellaiset tahot, joilla on siihen oikeus.

Eheys toteutuu, kun resurssin sisältö pysyy täsmällisenä ja täydellisenä sen elinkaaren ajan, eikä mikään oikeudeton taho pääse muuttamaan sitä hyökkäyksessä tai tahattomasti.

Saatavuus merkitsee, että resurssi on saavutettavissa ja käytettävissä, kun sitä tarvitaan.

Näitä CIA-kolmion käsitteitä täydentävät usein toiset kolme tietoturvan kannalta olennaista käsitettä: kiistämättömyys (engl. non-repudiation), tunnistus (engl. identification) ja todennus (engl. authentication). Näiden termien merkitykset on kuvattu seuraavassa.



Kuva 3.2: Luottamuksellisuus, eheys ja saavutettavuus.

Tunnistus on menettely, jossa järjestelmän käyttäjä tunnistetaan ja erotetaan muista käyttäjistä tunnistetta käyttäen. Esimerkki tällaisesta yksilöivästä tunnisteesta on käyttäjätunnus. Tunnistettava taho voi olla paitsi normaali käyttäjä, myös vaikkapa laite tai organisaatio. Tunnistautuva taho esittää identiteetistään väitteen, joka varmistetaan todentamalla.

Todennus varmistaa tunnistuksen eli järjestelmää käyttävän tahon esittämän väitteen identiteetistään. Kaikille tuttu esimerkki todennuksesta on salasanan syöttäminen kirjauduttaessa sisään järjestelmään. Salasana on esimerkki asiasta, jonka vain tietty käyttäjä tietää. Todennus voi pohjautua myös johonkin, mitä käyttäjällä on hallussaan (esimerkiksi sähköinen avainkortti) tai käyttäjän yksilölliseen fyysiseen ominaisuuteen (esimerkiksi sormenjälki). Tietoturvaa voidaan parantaa kaksivaiheisella todennuksella (two-factor authentication), jossa käyttäjän identiteetti varmistetaan hyödyntäen kahta eri todennusmenetelmää. Yleisesti kahta tai useampaa todennusmenetelmää hyödyntävää todennusta kutsutaan monivaiheiseksi todennukseksi (engl. multi-factor authentication). Pankkikortin ja siihen liittyvän tunnusluvun käyttö on yksi tuttu esimerkki kaksivaiheisesta todennuksesta.

Kiistämättömyys tarkoittaa, ettei käyttäjä tai viestinnän osapuoli voi onnistuneesti kiistää suorittamaansa toimintoa [29]. Tietojärjestelmien käyttöä – esimerkiksi etäohjattavalle alukselle annettavia komentoja – on väärinkäytön ehkäisemiseksi seurattava. Käyttäjän toimet voidaan tallentaa vaikkapa erilliselle lokipalvelimelle, jolloin voidaan kiistämättömästi todistaa käyttäjän tiettyinä ajankohtina suorittamat toimenpiteet. Luonnollisesti kiistämättömyys edellyttää, että kirjatut lokitiedot pystytään pitämään luottamuksellisina ja eheinä.

3.2 Ohjelmistokehityksen tietoturva

Ohjelmistojen tietoturva saa alkunsa jo määrittelyvaiheessa. Järjestelmillä, niiden käyttöympäristöillä ja myös niiden käyttäjillä on erilaisia turvallisuusluokituksia ja tietoturvavaatimuksia. Erityisvaatimusten lisäksi ohjelmistoilla on myös yleisiä tietoturvaan liittyviä ohjeistuksia ja hyviä käytäntöjä, joita tulee noudattaa jo niiden kehitysvaiheessa. Kun järjestelmän käyttöympäristö tai suojattava tieto on erityisen tärkeä, näistä ohjeista muodostetaan erilaisia sääntöjä ja prosesseja, joiden toteutus varmistetaan tietoturvallisilla kehitysprosesseilla. Näiden avulla pyritään varmistamaan itse ohjelmistotuotteen laatu ja tietoturvallisuus. Tarkistuksilla estetään myös kehittäjien tai alihankkijoiden järjestelmiin asentamat kehityksenaikaiset takaportit yms.

Tietoturvan vaatimukset on huomioitava kriittisten järjestelmien kehitystyössä aivan alusta alkaen. Sekä järjestelmän että sen sisältämän tiedon luotettavuus, eheys ja saavutettavuus – tietoturvan kolme perustavoitetta – on pystyttävä varmistamaan koko elinkaaren ajan. Järjestelmän elinkaari voi olla jopa kymmeniä vuosia, joten jokaiseen yksittäiseen uhkaan on mahdotonta varautua etukäteen. Niinpä järjestelmän kehittämisessä tulee noudattaa parhaita käytäntöjä ja periaatteita, jotka kategorisesti estävät mahdollisimman monta tietoturvauhkaa, tekevät mahdolliseksi järjestelmäpäivitysten tekemisen tulevien uhkien välttämiseksi ja onnistuneissa hyökkäystapauksissa tarjoavat ylläpitäjille kontrollit ja keinot vahinkojen jäljittämiseen, minimoimiseen ja korjaamiseen.

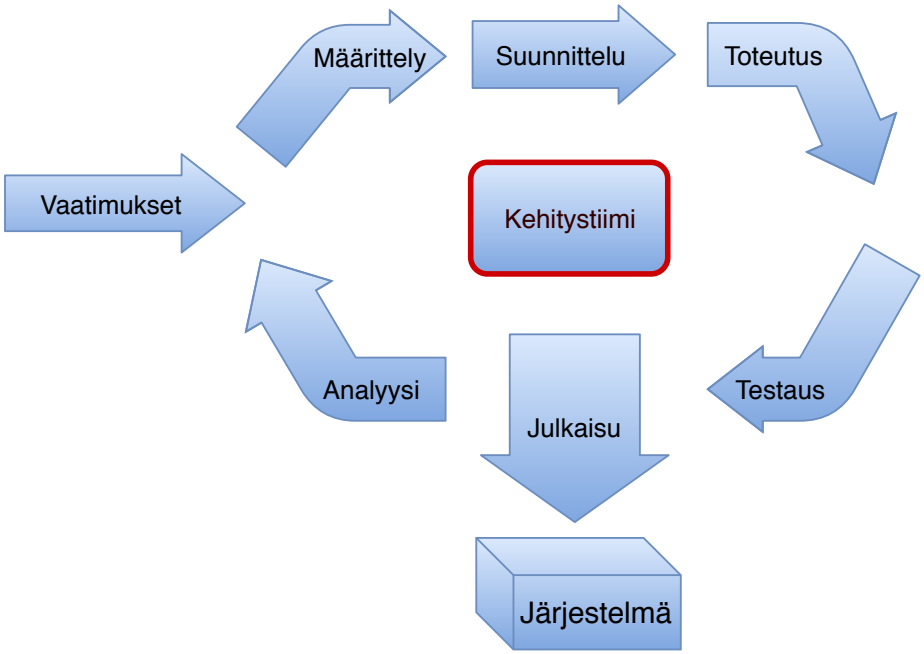
3.2.1 Tietoturvavaatimukset ja toteutusmallit

Tietojärjestelmiin kohdistuu joukko vaatimuksia, jotka jaetaan yleisesti *toiminnallisiin* ja *ei-toiminnallisiin*. Toiminnalliset vaatimukset kuvaavat, *mitä* ohjelmiston tai järjestelmän tulee tehdä. Ei-toiminnalliset vaatimukset puolestaan kuvaavat sen, *miten* nämä ohjelmisto tulee toteuttaa. Tämäntyyppiset vaatimukset koskevat esimerkiksi tietojärjestelmän arkkitehtuuria, laatua sekä tiettyjä keskeisiä tietoturvan osa-alueita. Tietoturvavaatimukset ovat osaltaan myös laadullisia: kuitenkin, vaikka järjestelmän laadulla on keskeinen vaikutus myös tietoturvaan, tietoturva toteutetaan joukolla toiminnallisia ominaisuuksia. Näitä ovat muun muassa käyttäjien tunnistautumistoiminnot, yhteyksien ja tiedostojen salaukset, tietoturvan monitorointiin tarvittavat toiminnot, sekä käyttöympäristön asetusten ja kokoonpanon määrittely. Näiden käyttäjille näkyvien toimintojen lisäksi ohjelmistokehityksen tietoturvalla koostuu joukosta erinäisiä toimintoja, joiden toteutukseen on olemassa yleisesti käytettyjä kehitysmalleja.

Tietoturvan vaatimukset voidaan johtaa paitsi organisaation omista käytännöistä ja ohjeista, myös yleisistä tai toimialakohtaisista standardeista sekä viime kädessä kansallisesta ja kansainvälisestä lainsäädännöstä. Tietoturvavaatimusten huomiointi jo järjestelmän kehitysvaiheessa on varautumista virhetilanteisiin ja tietoturvauhkiin ennalta. Vaikka ohjelmistojen tekeminen onkin keskittynyt siihen perehtyneille asiantuntijaorganisaatioille, on asiakas ohjelmistokehityksessä keskeisessä roolissa. Ohjelmiston tai järjestelmän kehittäminen alkaa vaatimusmäärittelyllä; suurin osa toiminnallisista vaatimuksista saadaan asiakkaalta tai heidän edustajaltaan. Kun nämä vaatimukset yhdistetään muihin ohjelmistotuotteelle asetettaviin vaatimuksiin, niistä muodostetaan joukko ominaisuuksia (feature) jotka ohjelmistokehittäjät sitten toteuttavat. Ohjelmistokehitysprosessi poikkeaa monista muista teollisuudenaloista siinä, että se tapahtuu lyhyissä sykleissä, joissa ohjelmistosta toteutetaan jokin valittu osa, joka testataan ja jonka toteutuksen perusteella asiakkaan on mahdollista antaa palautetta ja tutustua tuotteeseen jo huomattavasti ennen koko tuotteen valmistumista. Tämä niin sanottu iteratiivinen ja *ketterä* kehitysmalli antaa mahdollisuuden muuttaa ohjelmiston määrittelyjä kesken kehitysprosessin siten, että mahdollisimman pieni osa järjestelmästä pitää ohjelmoida uudelleen. Ketteriä ohjelmistokehitysmenetelmiä ovat mm. Scrum [24, 25], extreme programming (XP) [2] ja lean-ajattelua [20] edustava Kanban. Suurissa kehityshankkeissa saatetaan käyttää myös ohjelmistoteollisuudessa yleisiä ketteriin kehitysmenetelmiin perustuvia hallintamenetelmiä, kuten Scaled Agile Framework (SAFe)[2]. *Iteratiivisten* ja *inkrementaalisten* kehitysmenetelmien perusperiaate on esitetty kuvassa 3.3.

Ohjelmistoihin kohdistuu jo kehitysvaiheessa useita erityyppisiä tietoturvavaatimuksia. Näitä ovat itse ohjelmistotuotteeseen kohdistuvien vaatimusten lisäksi kehitysprosessia, kehitysympäristöä myös lopputuotteen käyttöympäristöä koskevat vaatimukset. Ohjelmistokehitykselle ei ole kuitenkaan olemassa yhtä yleispätevää tietoturvaa säätelevää tietoturvastandardia. Tietoturvakäytännöt ovatkin yleensä ohjelmistokohtainen yhdistelmä asiakkaan, kehittäjän ja toimintaympäristön lainsäädännöllisistä vaatimuksista ja standardeista. Yleisiä ja alariippumattomia

[2] https://www.scaledagileframework.com/



Kuva 3.3: Esimerkki iteratiivisesta kehitysprosessista.

tietoturvaohjeita toki löytyy, alkaen ISO/IEC:n ohjelmistokehitysprosesseja ja -toiminnallisuuksia määrittävästä SSE-CMM-standardista (ISO 21827)[9] ja ohjelmistotuotteen tietoturvaa mittaavasta ISO 15408-standardista [8] aina Suomen valtiohallinnon VAHTI-ohjeisiin [35]. Ohjelmistojen suunnittelua, toteutusta, testausta, julkaisua ja ylläpitoa koskevia tietoturvaohjeita ja malleja ovat myös erilaiset tietoturvan toteutusohjeet ja elinkaarimallit. Näitä ovat esimerkiksi Microsoftin Security Development Lifecycle (SDL)[3] [6]. Microsoft tarjoaa mallin lisäksi myös perustyökalut Windows-käyttöjärjestelmälle, joilla esimerkiksi ohjelmiston voi mallintaa työkalun osoittaessa mahdollisia riski- ja haavoittuvuuskohtia. Open Web Alliance Security Projectin (OWASP)[4] [18] avoimet tietoturvallisen ohjelmistokehityksen mallit ja konkreettiset ohjelmointi- ja testausohjeet ja -työkalut ovat myös suosittuja ja käyttökelpoisia. Näistä tärkeimpiä ovat esimerkiksi OWASP Top 10-ohjelmistohaavoittuvuudet ja OWASP Security Assurance Maturity Model (SAMM), joka on avoin tietoturvan kehitys- ja mittausmenetelmä. Muita ajankohtaista ohjelmistokehityksen tietoturvatietoa tarjoavia tahoja ovat mm. SANS-instituutin 25 vaarallisinta ohjelmointivirhettä-listaus[5] ja Suomessa Viestintäviraston kyberturvallisuussivut[6].

3.2.2 Tietoturvallisen ohjelmistokehityksen elementit

Ohjelmistojen elinkaari voidaan jakaa useisiin vaiheisiin, joista jokainen asettaa omat haasteensa myös tietoturvatyölle. Samoin ohjelmistokehitykselle on olemassa omia elinkaarimallejaan.

[3] https://www.microsoft.com/en-us/SDL/
[4] https://www.owasp.org
[5] https://www.sans.org/top25-software-errors
[6] https://www.viestintavirasto.fi/kyberturvallisuus/varoitukset.html

Näiden merkitys on keskeisenä ohjelmistojen tuotantoa ohjaavana elementtinä pienentynyt huomattavasti iteratiivisten menetelmien yleistyessä, mutta vaihejako on edelleen olemassa. Tietoturvan kannalta olennaisia ja lähes jokaisesta elinkaarimallista löytyviä vaiheita ovat vaatimusmäärittely, suunnittelu, toteutus, testaus, julkaisu ja ylläpito. Näiden lisäksi ohjelmistokehitykseen kohdistuu joukko esivaatimuksia.

Seuraavissa osioissa esitetään yleisiä ohjelmistokehityksen aikana toteutettavia tietoturvaa edistäviä toimenpiteitä noudattaen samaa vaihejakoa kuin luvussa 2.3. Tietoturvan osalta elinkaarimallia tulee laajentaa koskemaan myös kehitysprosessin esivaatimuksia, järjestelmän julkaisua ja ylläpitoa sekä tiedon turvallista poistoa sen elinkaaren lopussa.

Esivaatimukset

Ohjelmistojen kehitysympäristöön, sen hallinnointiin, kehitysorganisaatioon, käytettyihin työkaluihin ja itse ohjelmistokehittäjiin kohdistuu mittava määrä erilaisia vaatimuksia ja odotuksia, jotka tulee täyttää jo ennen kuin kehitysprojekti voi edes alkaa. Tähän sisältyvät kaikki organisationaaliset varmistustoimenpiteet sekä esimerkiksi turvastandardoinnit ja muut sertifioinnit. Henkilöstöön kohdistuvista esivaatimuksista ehdottomasti tärkein – taustatarkistuksen lisäksi – on projektiin osallistuvien henkilöiden ja johtoportaan kouluttaminen ja perehdytys tietoturvaasioihin. Asianmukainen, työtehtävään soveltuva ja riittävä tietoturvakoulutus on paras tae tietoturvallisen ohjelmistoprojektin onnistumiselle ja ikävien yllätysten välttämiselle kehitysprojektin myöhemmissä vaiheissa tai ohjelmistopäivityksiä tehtäessä. Koulutusta voidaan antaa myös projektin kuluessa, jos esim. uusi teknologia edellyttää vanhan koulutuksen täydentämistä tai kokonaan uuden oppimista.

Vaatimusmäärittely

Vaatimusmäärittelyssä yhdistetään ohjelmiston kaikki liiketoiminnalliset ja muut vaatimukset. Ns. perinteisessä ohjelmistokehitysmallissa projektin aikataulut määräytyvät tässä vaiheessa tehtävien arvioiden perusteella; iteratiivisessa ohjelmistokehityksessä tähän vaiheeseen voidaan vielä palata kehitysprojektin aikana. Vaatimusmäärittelyssäkin voidaan käyttää apuna standardoituja menetelmiä, kuten esimerkiksi System Quality Requirements Engineering (SQUARE)[15].

Tässä vaiheessa kartoitetaan myös ohjelmiston tietoturvauhat, jotka yhdistetään aiemmin tunnistettuihin riskeihin (kts. luku 3.1.1). Samassa yhteydessä suunnitellaan myös mekanismit, joilla todennetaan ohjelmiston turvallisuus sekä sen arviointiin ja seurantaan käytettävät mittarit. Yksinkertainen esimerkki tällaisesta mekanismista on lokitiedosto, johon kirjataan ohjelmiston käyttäjien suorittamat toimenpiteet.

Suunnittelu

Ohjelmiston tietoturvan suunnittelu koostuu tietoturva-arkkitehtuurin laatimisesta, ohjelmiston rakenteen uhkamallinnuksesta, ohjelmiston yleisen rakenteen kriittisestä katselmoinnista ja tämän perusteella tehtävistä tietoturvakovennuksista. Näillä toimenpiteillä pyritään tukemaan myös ketterässä ohjelmistokehityksessä yleisesti suosittua yksinkertaistetun suunnittelun periaatetta: esi-

merkiksi tarpeettomia toimintoja karsimalla ja suoraviivaistamalla myös ohjelmiston ylläpidettävyys ja tietoturva paranevat.

Tietojärjestelmän tietovarastot, kommunikointirajapinnat ja muut tietoturvan kannalta kriittiset osiot tunnistetaan suunnitteluvaiheessa ja niille laaditaan kattavat tietoturvatestit. Tietoturva-asiantuntijat katselmoivat suunnitelman, joka hyväksytään yhdessä kehittäjien kanssa ennen kuin tietojärjestelmän toteutus voi konkreettisesti alkaa.

Toteutus

Tietojärjestelmän toteutusvaiheessa asiantuntijat rakentavat järjestelmän pala palalta, suunnitelmaa tai ainakin sen pääpiirteitä noudattaen. Kehittäjät ja tietoturva-asiantuntijat tekevät yhdessä tietoturvakatselmointeja, joissa käydään läpi ohjelmistokoodia, konfiguraatiotiedostoja tai vaikkapa tietovarastojen sisäisiä rakenteita. Toteutusvaiheessa voidaan käyttää myös mm. pariohjelmointia, jossa kehittäjät tekevät samaa asiaa yhdessä. Ohjelmistoja tuotettaessa noudatetaan yleisesti myös hyväksi havaittuja ohjelmointikäytäntöjä ja periaatteita, joilla pyritään parantamaan ohjelmistokoodin selkeyttä, ylläpidettävyyttä ja toimivuutta – ja siten myös turvallisuutta.

Lähdekoodi ja muut käytettävät ohjelmistokomponentit talletetaan keskitettyyn ja pääsyltään rajattuun konfiguraationhallintaan, jossa voidaan suorittaa myös automatisoituja testejä. Tällä menettelyllä säilytetään myös itse ohjelmiston rakennusosien eheys ja luottamuksellisuus: komponentteihin pääsevät käsiksi vain tarkkaan määritellyt henkilöt ja jokaisesta muutoksesta jää jälki hallintajärjestelmään. Automatisoidut käännös- ja yksikkötestaustyökalut yhdessä integrointi- ja asennustyökalujen kanssa pitävät huolen siitä, että kehitysympäristössä on aina valmiiksi testattu ja viimeisimmät muutokset sisältävä ohjelmistoversio. Julkaisuprosessissa sille voidaan antaa julkaisulupa ja asentaa se tuotantoympäristöön. Kehitys- ja testiympäristöjen tulee aina sijaita erillään tuotantoympäristöstä.

Testaus

Ohjelmiston toiminnallisuuden verifiointi on keskeisessä roolissa ohjelmiston hyväksynnässä julkaisukelpoiseksi: ohjelmisto joko toimii halutulla tavalla, esimerkiksi tuottaa halutun syötteen, tai muussa tapauksessa testaus todetaan epäonnistuneeksi. Tietoturvan ei-funktionaalinen testaaminen puolestaan kykenee osoittamaan lähinnä vikojen tai puutteiden poissaoloa. Tämän vuoksi kaikkea järjestelmään vastaanotettavaa dataa tulee kohdella tietoturvauhkana. Testaus on myös erityisesti kehittäjien vastuualuetta ja jokaiselle kehitysaikana havaitulle virheelle on laadittava testi.

Erilaisia injektiohyökkäyksiä on jo kauan toteutettu menestyksekkäästi. 1980- ja 1990-luvulla kohteina oli paljon C-kielellä kirjoitettuja palveluita, joiden rajapintaan hyökkääjä kohdisti hyvin epätavallisia syötteitä onnistuen injektoimaan dataksi tarkoitettua sisältöä osaksi suoritettavaa koodia. Tällaisesta ns. puskurin ylivuoto -haavoittuvuutta hyödyntämällä hyökkääjä saa ohjelma tallentamaan annetun syötteen osittain sille varatun muistialueen ulkopuolelle ja tätä kautta suorittamaan hyökkääjän kirjoittamaa vahingollista koodia. Sittemmin sama ongelma on esiintynyt SQL:n ja web-kerroksen yhteydessä. Tätä torjumaan on kehitetty erityinen testauksen

muoto, josta käytetään nimitystä fuzz-testaus (engl. fuzz testing). Siinä ideana on hyvin erilaisilla (satunnaisilla) syötteillä testata rajapinnan toiminnallisuutta.

Tietoturvatestaus on keskeinen osa ohjelmistokehitystä. Testit tulee automatisoida ja ottaa osaksi hyväksyntäkriteereitä. Tietoturvaltaan puutteellinenkin järjestelmä saattaa vaikuttaa toimivan oikein, mutta jättää väärinkäyttäjälle takaportin, jolla ohitetaan muut tietoturvamekanismit. Asianmukaisella ja kattavalla tietoturvatestauksella varmistetaan, että järjestelmän turvallisuus ei ole pelkästään palomuurien tai muiden ulkoisten turvamekanismien varassa. Hyvä järjestelmä kestää ulkoisia uhkia itsenäisesti, vaikka muut varokeinot pettäisivätkin.

Julkaisu

Testit läpäisseelle ohjelmistolle suoritetaan usein vielä tietoturvan kannalta olennaisia toimenpiteitä, ennen kuin se voidaan hyväksyä tuotantokäyttöön. Ohjelmistolle tulee laatia varautumissuunnitelma virhetilanteiden varalle ja sen käyttäjille ja ylläpitäjille suunnatun dokumentaation tulee olla hyväksyttävällä tasolla. Ohjelmiston käyttöympäristölle saatetaan joutua tekemään vielä tietoturvakovennuksia esimerkiksi asentamalla sinne käyttöjärjestelmästä riisuttu tai toiminnoiltaan muutettu versio. Itse julkaistava ohjelmisto voidaan vielä kerran katselmoida ja antaa sille digitaalinen sertifikaatti. Sertifiointi yhdessä käyttöympäristön kovennusten kanssa pyrkii varmistamaan, ettei ohjelmistoa, sen osia tai käyttöjärjestelmää voida peukaloida sen julkaisemisen jälkeen.

Ylläpito

Tietojärjestelmien elinkaari saattaa jatkua vuosia tai joissakin tapauksissa jopa vuosikymmeniä. Tänä aikana käyttäjien tai ympäristön vaatimukset voivat muuttua, jolloin ohjelmistoon on tehtävä päivityksiä. Yhä useammissa tapauksissa järjestelmä voidaan ostaa myös palveluna tai sitä tuotetaan ns. jatkuvan kehityksen mallilla. Tällöin järjestelmän toimintaa voidaan parantaa tai tuoda siihen uusia ominaisuuksia sitä mukaa kun kehittäjät saavat niitä valmiiksi. Luotettujen työkalujen, osaratkaisujen (kirjastojen) ja prosessien merkitys on tässä toimintamallissa ensisijaisen tärkeä.

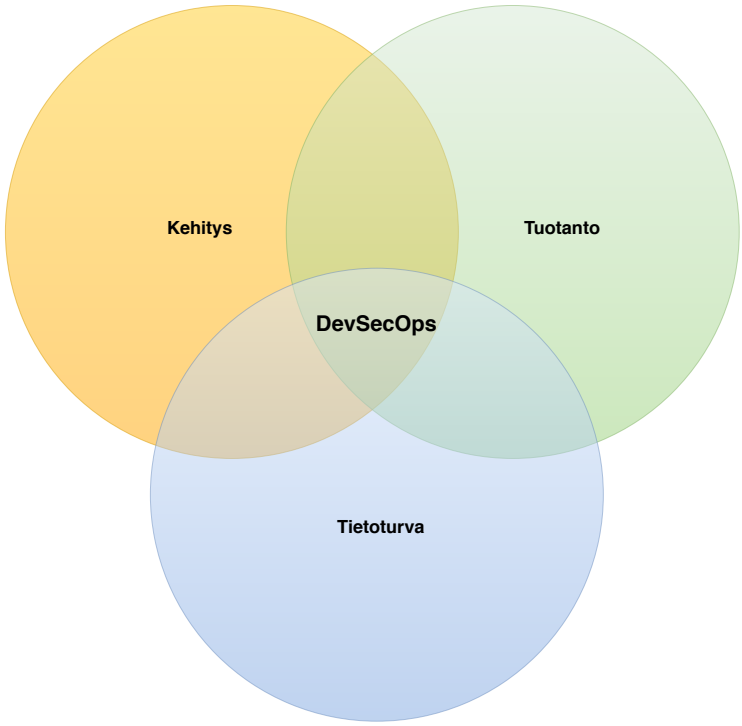
3.2.3 Käytöstä poistaminen

Kun tiedon tai tietojärjestelmän elinkaari päättyy, se poistetaan. Tietoa saatetaan kuitenkin joutua arkistoimaan vielä sen aktiivisen käyttöiän jälkeenkin. Käytöstä poistettukin tieto on usein luottamuksellista, eikä saa joutua vääriin käsiin. Täten sen tuhoaminenkin on tietoturvan kannalta merkityksellinen toimenpide. Toisaalta esimerkiksi yksityisyyttä suojaavat säädökset vaativat henkilötietojen poistamista, kun niillä ei ole enää varsinaista käyttötarkoitusta. Tällainen tieto on pysyttävä tunnistamaan ja poistamaan luotettavasti. Organisaatiolla on oltava selkeät ohjeet ja käytännöt tiedon käsittelylle ja säilyttämiselle siten, että se voidaan poistaa luotettavasti ja pysyvästi. Myös mahdolliset varmuuskopiot tulee tällöin hävittää.

3.2.4 DevOps ja DevSecOps

Jatkuvan ohjelmistokehityksen malli edellyttää ylläpidon ja ohjelmistokehityksen yhteistoimintaa. Tässä ns. DevOps-mallissa (development ja operations, eli kehitys ja tuotanto yhdistettynä) tietoturvatoimien merkitys korostuu erityisesti verifiointi- ja julkaisuvaiheissa. Vaatimuksenhallinnan kautta tulleet uudet tai muuttuneet tietoturvavatimukset verifioidaan ennen julkaisua ja ylläpidon ja käyttäjien ohjeistus päivitetään aina ajanmukaiseksi. Kun tähän toimintamalliin tuodaan mukaan tietoturva (Security), mallista voidaan käyttää nimitystä DevSecOps korostamaan tietoturvan keskeistä roolia. DevOps- ja DevSecOps-malleissa prosessit ovat pitkälle automatisoituja, joten kehitysinfrastruktuurin turvaaminen on erityisen tärkeää. Mallin etuna oikein toteutettu automatisointi ja turvattu kehitysympäristö takaavat paitsi nopeammat kehityssyklit, myös tuotantoympäristön nopean uudelleenrakentamisen häiriötilanteista toipuessa.

Autonomisten alusten yhteydessä olisi jossain määrin toivottavaa, että DevOps- tai DevSecOps-toimintaa ei olisi tarve soveltaa. Valitettavasti ohjelmistoratkaisut ovat usein niin kompleksia, että jälkikäteen (satamissa) suoritettavaa päivitystoimintaa lienee odotettavissa. Toisaalta luvussa 2 tuli myös esiin, että meriliikenteessä alusten toimintaympäristöä rajoittavat monet tahot ja sääntely on myös dynaamista, jolloin DevOps-toiminnalle on ilmeisesti lähtökohtaisesti tarvetta.



Kuva 3.4: Kehityksen, tuotannon ja tietoturvan yhdistäminen.

3.2.5 Yhteenveto

Tietojärjestelmäkehitys on usein jatkuva prosessi tai ohjelmisto ostetaan palveluna. Näissä tapauksissa myös muuttuvat tietoturvavaatimukset voidaan huomioida nopeammin ja ilman lisäkustannuksia. Tietojärjestelmiä kehitettäessä riittävä osaamistaso myös tietoturva-asioissa on edellytys onnistuneelle järjestelmähankkeelle ja tavoitteiden saavuttamiselle. Tiukat aikataulut ja niukat henkilöstöresurssit johtavat lyhyen aikavälin kustannussäästöihin, mutta mutkissa oikominen ja vaikkapa tietoturvavaatimusten huomioimatta jättäminen kehitysvaiheessa voi pakottaa mittaviin ylläpito- ja korjaustoimiin myöhemmin. Tietoturva ei ikinä ole ”jonkun muun” huolenaihe ja seuraamukset sen laiminlyönnistä koskevat jokaista.

Ohje 3.2 Tietoturva järjestelmäkehityksessä

Tietoturvan huomioiminen järjestelmäkehityksessä yhdistää organisaation liiketoimintatavoitteet, riskienhallinnan ja järjestelmään kohdistuvat tietoturvavaatimukset. Kehitysvaiheessa tehdyt laadulliset ja tietoturvakeskeiset parannukset helpottavat järjestelmän ylläpitoa, lisäävät sen vikasietoisuutta ja tekevät häiriötilanteista toipumisesta helpompaa.

Ohje 3.3 Asiakkaana ohjelmistokehitystyössä

Asiakkaan edustaja osallistuu kehitysprosessiin määrittelijänä, lopputuloksen hyväksyjänä, sekä usein myös päivittäisenä yhteistyökumppanina, jolta järjestelmän kehittäjät voivat kysyä lisätietoja ja tarkennuksia. Asiakas voi pyrkiä ulkoistamaan osan osaamisesta ja vastuusta yhteistyökumppaneille: tietoturvatyössä voi olla usein kannattavaa käyttää puolueetonta tietoturva-asiantuntijaa tai organisaatiota, joka varmistaa järjestelmän turvallisuuden ja hyväksyttävyyden. Tietoturvasertifiointi saattaa kriittisissä järjestelmissä olla myös viranomaisvaatimus. Oma osaaminen on kuitenkin tärkeää myös tietojärjestelmien kehittämisessä ja ennen kaikkea tietoturva-asioissa. Tietosuoja- ja tietoturva-asioissa luottamus on aina syytä voida tarvittaessa varmistaa ja todistaa.

3.3 Kyberturvallisuuden toteutus järjestelmän kannalta

Seuraavaksi tarkastellaan, miten tietoturvan kolme perusarvoa, eli luottamuksellisuus, eheys ja saavutettavuus toteutetaan käytännössä ICT-järjestelmän tasolla. Luvuissa 3.4 ja 3.5 tarkastellaa

samaa tietoliikenneyhteyksissä ja loppukäyttäjän näkökulmasta. Esitys painottuu lähinnä kyber- ja ohjelmistoturvallisuuden teknisiin rakennuselementteihin.

ICT-järjestelmän tietoturvan heikkenemisellä voi olla negatiivisia vaikutuksia sen palvelujen luottamuksellisuuteen, eheyteen ja saatavuuteen ja tätä kautta myös organisaation maine voi vaarantua. Järjestelmän tietoturvaa voidaan sen kehityksen ja käytön aikana lisätä monilla erilaisilla teknisillä rakennuselementeillä. Järjestelmässä käytettävien teknisten tietoturvaratkaisujen tehtävä on esimerkiksi käyttäjien toimien tarkoituksenmukainen rajoittaminen järjestelmässä, järjestelmässä olevan luottamuksellisen tiedon salaaminen, tietojen oikeudettoman käytön estäminen ja valvominen mm. oikeilla tietoturva-asetuksilla ja lokitiedostoilla, ohjelmien suorituksen ja niiden käsittelemän datan eriyttäminen toisistaan sekä haittaohjelmien toiminnan havaitseminen ja estäminen.

Tähän käytettäviä ratkaisuja ovat muun muassa kryptografia, pääsynhallinta, koventaminen, luotetut suoritusympäristöt, tunkeutumisen havaitsemisjärjestelmät ja virustojuntaohjelmat. Vaikkei tämä teknisten ratkaisujen joukko olekaan tyhjentävä, se antaa kuvan olennaisimmista teknologioista ja kyberuhkien torjuntaan käytettävien tekniikoiden kirjosta.

3.3.1 Kryptografia

Yksi kyberturvallisuuden olennaisimmista teknisistä rakennuselementeistä on kryptografisten menetelmien hyödyntäminen. *Kryptografia* (engl. cryptography) viittaa turvallisen ja salatun viestinnän harjoittamiseen tilanteessa, jossa kolmas osapuoli saattaa pyrkiä vakoilemaan tai muokkaamaan viestien sisältöä. Tavoitteena on luoda sellaisia protokollia eli yhteiskäytäntöjä, jotka takaavat turvallisen viestinnän hyökkääjän tai vastustajan läsnäolosta huolimatta. Kryptografisin menetelmiin voidaan myös turvata tallennetun tiedon luottamuksellisuus ja eheys. Kryptografisiin menetelmiin liittyy siis olennaisesti tietoturvan perusarvojen kuten luottamuksellisuuden, eheyden ja kiistämättömyyden varmistaminen. Moderni kryptografia perustuu paljolti matemaattiseen teorian olettamuksiin[7] laskennellisesta vaikeudesta; oletusten seurauksena on, ettei pahantahtoinen kolmas osapuoli pysty murtamaan kryptografista menetelmää – esimerkiksi selvittämään salattua viestiä tai purkamaan salattua tiedostoa – järkevässä ajassa [14].

Kryptografiaan liittyy läheisesti *salauksen* (engl. encryption) prosessi. Salauksessa *selväkielinen teksti* (engl. plaintext) muunnetaan *salatuksi tekstiksi* (engl. ciphertext), jota ei voi tulkita. Salaus takaa siis tiedon luottamuksellisuuden. Päinvastaista prosessia eli salatun tekstin muuntamista selkokieliseksi kutsutaan *purkamiseksi* (engl. decryption). Salausmenetelmä on pari algoritmeiksi kutsuttuja laskennallisia menetelmiä, joilla salaukseen ja purkamiseen vaaditut muunnokset tehdään. Salausalgoritmi ottaa salattavan tekstin ohella syötteekseen salausavaimen (engl. encryption key), jonka vain lähettäjä ja vastaanottaja tuntevat symmetrisen kryptografian tapauksessa. Salatun viestin voi purkaa selväkieliseksi ainoastaan, jos tietää oikean avaimen. Symmetrisissä kryptografisissa menetelmissä salaamiseen ja purkamiseen käytetään samaa avainta. Itse sa-

[7]Esimerkiksi kuuluisa julkisen kryptografian RSA-menetelmä perustuu siihen, että on laskennallisesti hyvin vaikeaa toteuttaa suurten lukujen kohdalla niiden alkutekijöiden (alkulukujen) laskenta. Tekijöiden jakoa pidetään yleisesti matemaattisesti erittäin vaikeana ongelmana, mutta sitä ei ole kyetty todistamaan sellaiseksi. Vaikeille laskennallisille ongelmille on määritelty erilaisia kompleksisuusluokkia, esim. NP-täydellisten ongelmien luokka, mutta tekijöiden jaon ei tiedetä todistettavasti kuuluvan NP-täydellisten ongelmien joukkoon ...

lausmenetelmä on siis tavallisesti niin sanotun Kerchoffin periaatteen mukaisesti julkinen, mutta salausavaimet pidetään salaisina. Kryptografisten menetelmien turvallisuus on vahvasti riippuvainen niiden parametreina käytettyjen salausavaimien pituudesta eli bittien lukumäräästä.

Salauksen luotettavuuden kannalta tärkeä osa-alue on avaintenhallinta. Kryptografian periaatteisiin kuuluu, että salaus kestää, vaikka hyökkääjällä olisi käytössään kaikki salatut viestit ja tieto käytetyistä salausalgorimeista. Salaisten avaimien vuotaminen kuitenkin mitätöi parhaatkin kryptografiset menetelmät. Tämän takia onkin syytä laatia järjestelmää varten avaintenhallintasuunnitelma, jossa esimerkiksi avainten luontiin, turvalliseen säilytykseen, jakeluun ja voimassaoloaikoihin otetaan kantaa. Avaintenhallinnasta on esimerkiksi avainten luomisen, elinkaaren, jakelun ja suojaamisen osalta lisätietoa VAHTI:n salauskäytäntöjä koskevassa ohjeistuksessa [34].

Myös avainten aitouden varmistaminen on tärkeää. Yhtenä avaintenhallinnan haasteena epäsymmetrisessä salausjärjestelmissä on se, miten yleisessä levityksessä olevan julkisen avaimen aitouteen voi luottaa. Yksi ratkaisu tähän on julkisten avainten hallintajärjestelmä PKI (Public Key Infrastructure) [1]. PKI perustuu siihen, että *varmentaja* (engl. Certificate Authority, CA) eli luotettava kolmas osapuoli allekirjoittaa julkisen avaimen ja sen tunnistetiedot digitaalisesti. Näin syntynyttä *varmennetta* (engl. sertificate) voidaan sitten jakaa eteenpäin. Varmentajan luotettavuuden sekä digitaalisen allekirjoituksen oikeellisuuden perusteella voidaan periaatteessa varmistua siitä, että julkisen avaimen avaimen tunnistetiedot ovat oikeat. Usein varmenteita käytetään esimerkiksi todistamaan verkkopalvelun käyttäjälle, että hän asioi oikean toimijan kanssa.

Kun halutaan parantaa tietojärjestelmän tietoturvaa kryptografian keinoin, käytetään alhaisen tason rakennuselementteinä kryptografisia primitiivejä. Aiemmin esitelty todennus on yksi tällainen primitiivi. Muita yleisesti käytettyjä *kryptografisia primitiivejä* (engl. cryptographic primitives) ovat esimerkiksi seuraavat:

- *Symmetrinen salaus* (engl. symmetric encryption) on salausmenetelmä, jossa viestin lähettäjä ja vastaanottaja käyttävät viestin salaamiseen ja salakirjoitetun viestin purkamiseen samaa salausavainta [29, 14]. Symmetrinen salaus toimii selvästi epäsymmetristä salausta nopeammin ja soveltuu suurten tietomäärien salaamiseen. Symmetrisen salauksen heikkous on se, että avain tulee toimittaa turvallisesti viestinnän kaikille osapuolille. Avaimen paljastuminen tarkoittaisi, että kaikkien sillä salattujen viestien salaus voitaisiin purkaa. Autonomisten alusten kohdalla avaimia joudutaan säilyttämään myös aluksessa, jolloin avainten fyysinen turvallisuus on myös asia, joka pitää huomioida muuten digitaalisessa avainten säilytyksessä. Symmetrisen salauksen periaate on esitetty kuvassa 3.5.

- *Epäsymmetrinen salaus* (engl. asymmetric encryption) puolestaan on salausmenetelmä, jossa käytettyjä avaimia ei salausalgoritmin laskutoimitusten vaativuuden ansiosta voi johtaa toisistaan. Salausavaimet muodostavat parin, joista toinen pidetään salassa ja toinen julkaistaan [29, 14]. Kuka tahansa voi lähettää viestin avainparin haltijalle käyttäen hänen julkista avaintaan. Tämän jälkeen viestin vastaanottaja purkaa viestin salauksen yksityisellä avaimellaan. Kuvassa 3.6 näkyy asymmetrisen salauksen periaate. Epäsymmetrisen menetelmän etuna on, ettei viestin purkamiseen tarvittavaa avainta tarvitse välittää vies-

tinnän osapuolilla samoin kuin symmetrisen salauksen tapauksessa. Epäsymmetrisen menetelmän heikkoutena on sen hitaus ja toteutusten monimutkaisuus. Epäsymmetristä ja symmetristä salausmenetelmää käytetään usein yhdistettynä *avaimenvaihtoon* (engl. key exchange). Viestinnän toinen osapuoli luo ensin symmetrisen avaimen, salaa sen vastaanottajan julkisella avaimella ja lähettää sen vastaanottajalle. Näin symmetrinen avain saadaan välitettyä turvallisesti ja osapuolet voivat käyttää sitä keskinäisten viestiensä salaamiseen symmetristä menetelmää soveltaen.

- *Yksisuuntaisella tiivistefunktiolla* (engl. one-way hash function) voidaan luoda viestistä tai yleisesti bittijonosta vakiomittainen tiivistearvo. Tätä tiivistettä voidaan käyttää tarkistussummana, josta käyttäen voidaan tarkistaa, onko viesti tai tiedosto muuttunut esimerkiksi tiedonsiirtovirheen tai haittaohjelman toimien seurauksena. Tiivistefunktiota ei siis käytetä tiedon salaukseen vaan tiedon eheyden tarkistamiseen.

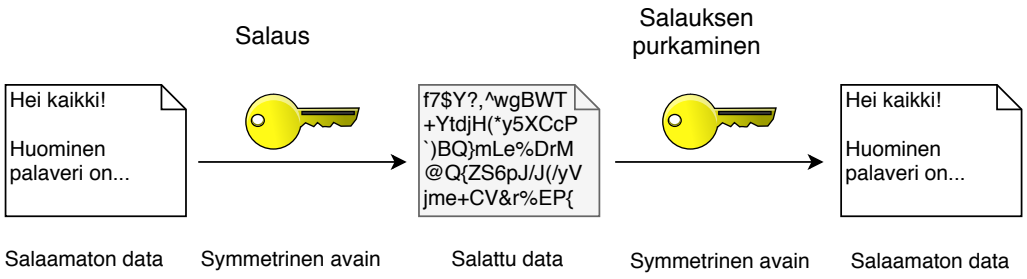
- *Digitaalinen allekirjoitus* (engl. digital signature) on sähköinen sähköiseen viestiin liitettävä varmenne, jolla voidaan tarkistaa viestin sisällön eheys tai allekirjoittajan identiteetti [10]. Digitaalista allekirjoitusta käytetään siis tiedon todentamiseen, sen oikeellisuuden ja alkuperäisyyden toteamiseen. Lähettäjä allekirjoittaa viestin yksityisellä avaimellaan ja vastaanottaja voi tarkistaa allekirjoituksen lähettäjän julkista avainta käyttäen. Jos hyökkääjä peukaloi viestiä, matemaattinen allekirjoitus ei täsmää ja viestin eheyden rikkova väärennys paljastuu.

Kryptografiset protokollat (engl. cryptographic protocols) ovat käytänteitä, joilla suoritetaan tietoturvaan liittyviä toimintoja käyttäen kryptografisia menetelmiä, yleensä hyödyntäen rakennusaineina useita kryptografisia primitiivejä. Protokollat kuvailevat, kuinka algoritmeja ja kryptografisia primitiivejä pitäisi hyödyntää. Kryptografisia protokollia käytetään laajasti suojaamaan sovellustason tiedonsiirtoa.

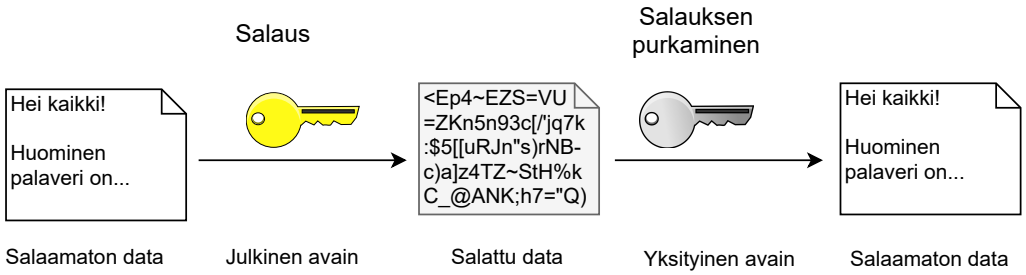
Esimerkiksi TLS (Transport Layer Security)[8] on kryptografinen protokolla, jota käytetään yhteyksien suojaamiseen muun muassa web-ympäristössä. Siihen sisältyy useita kryptografisia primitiivejä: todennusmekanismi, symmetristen avainten vaihto asymmetristä kryptografiaa käyttäen sekä verkon yli siirrettävän datan salaus symmetrisellä kryptografialla. TLS takaa tiedonsiirron eheyden ja luottamuksellisuuden.

Uuden uhan salausalgoritmeja vastaan muodostavat kehitteillä olevat kvanttitietokoneet [10]. Jos kvanttitietokoneet edistyvät tarpeeksi, melkein kaikki nykyään yleisessä käytössä olevat epäsymmetriset salausmenetelmät voidaan murtaa ja symmetristä salausta heikentää. Tavalliset tietokoneet käyttävät laskuoperaatioihin bittejä, joilla on kaksi mahdollista tilaa: yksi ja nolla. Suoritetut laskutoimitukset muuttavat bittien arvoja. Kvanttitietokone sen sijaan käsittelee bittien asemesta kubittejä. Kubitin ei digitaalisen bitin tavoin tarvitse olla diskreetisti yksi tai nolla. Sen sijaan kubitin arvo voi kvanttimekaniikassa käytetyn superposition käsitteen mukaisesti olla yhtäaikaisesti yksi ja nolla. Lisäksi tärkeä kubitin normaalista bitistä erottava ominaisuus on, että sen tilat lomittuvat useiden kubittien välillä. Useat kubitit yhdessä voivat siis ilmaista samaan

[8] https://tools.ietf.org/html/rfc5246



Kuva 3.5: Symmetrisen salauksen periaate.



Kuva 3.6: Epäsymmetrisen salauksen periaate.

aikaan monien binäärilukujen superposition. Kvanttitietokone pystyy näin katsomaan laskutoimituksen useita vaihtoehtoja samanaikaisesti. Tästä seuraava kvanttitietokoneen kyky suorittaa rinnakkaisesti suuria määriä laskutoimituksia mahdollistaa ainakin teoriassa monien salausmenetelmien, kuten paljon käytetyn epäsymmetrisen RSA-menetelmän, murtamisen.

Kvanttitietokoneiden kehityksessä on toistaiseksi monia haasteita. Tätä kirjoitettaessa esimerkiksi IBM on saanut kehitettyä 50 kubitin kvanttitietokoneen[9], mikä ei vielä riitä salauksen murtamiseen nykyisillä avainpituuksilla järkevässä ajassa. Arviot salauksia tehokkaasti murtavan kvanttitietokoneen valmistumisesta vaihtelevat, eikä sellaista välttämättä koskaan valmistu. Uusille kvanttilaskennan kestäville salausmenetelmille on joka tapauksessa tilausta, ja niiden tulee olla käytössä jos ja kun tarpeeksi tehokas kvanttitietokone saadaan kehitetyksi. Tällaisia ovat esimerkiksi hilateoriaan pohjautuvat salausmenetelmät (engl. lattice-based cryptography) kuten NTRU-salaus, jota vastaan ei nykytiedon valossa ole kvanttitietokonetta hyödyntävää hyökkäystä.

[9]https://www.research.ibm.com/ibm-q/

Ohje 3.4 (Kehittäjä) Ohjeita kryptografisten menetelmien käyttöön.

- Kaikki verkon yli kulkeva ja järjestelmässä säilytettävä luottamuksellinen tieto tulisi salata.
- Salaukseen tulee käyttää asianmukaisia salausmenetelmiä – esimerkiksi huomioiden epäsymmetrisen salauksen hitaus ja toteutusten monimutkaisuus.
- Käytettyjen avainten tulee olla tarpeeksi pitkiä, sillä salauksen turvallisuus riippuu vahvasti käytettävän avaimen pituudesta.
- Asianmukaisesti avaintenhallinnasta on huolehdittava esimerkiksi salaisten avainten turvallisen säilömisen ja avainten ajoittaisen vaihtamisen osalta.
- Kryptografisten menetelmien vahvuus on vaikeasti todennettava asia ja siksi kehittäjien tulee harkiten suhtautua uusien menetelmien luontiin – suositeltavaa on keskittyä jo määritettyjen kryptografisten menetelmien soveltamiseen.
- Hyvin pidetty kryptografinen menetelmä on helposti mahdollista toteuttaa väärin. Suositeltavaa on katsoa saatavilla olevia luotettuja toteutuksia kryptografisille menetelmille ja tarkkailla laajasti käytössä olevien menetelmien totetuksen mahdollisia todennettuja heikkouksia. Esimerkkinä äskettäinen Heartbleed-vaavoittuvuus TLS-toteutuksissa.
- Kryptografisten menetelmien sekä niiden murtamiseen käytettävien keinojen (esim. kvanttitietokone) kehitystä on seurattava ja tarpeen vaatiessa kasvatettava avainpituuksia tai vaihdettava käytettyjä salausmenetelmiä.

3.3.2 Pääsynhallinta

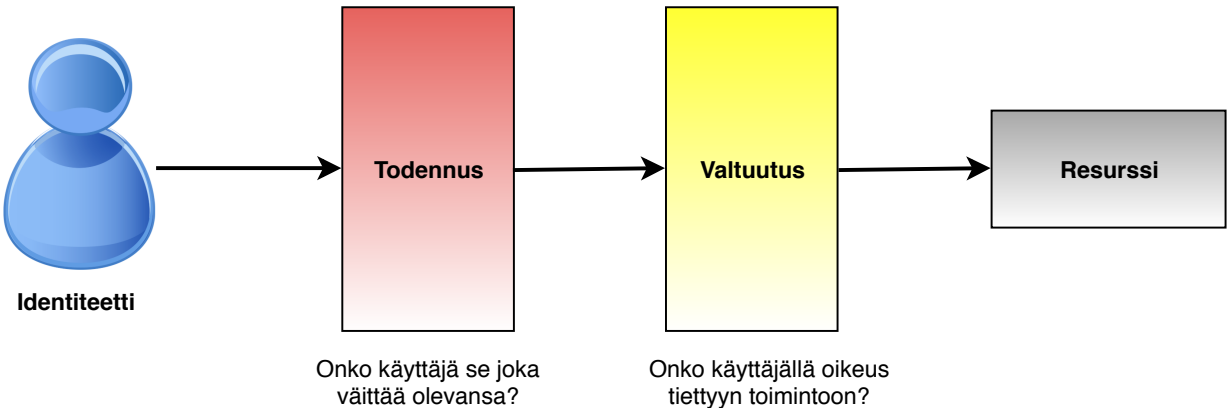
Luottamuksellisuutta, eheyttä ja saatavuutta tavoitellessa lähtökohtana on, että on olemassa tietty joukko tahoja, jotka saavat tarkastella ja muokata tietoa. Toisaalta on olemassa ulkopuolisia tahoja, joilla ei ole tätä oikeutta. *Pääsynhallinnalla* (engl. access management) tarkoitetaan menettelyjä, joilla varmistetaan, että asiankuuluvat tahot – käyttäjät, järjestelmät ja sovellukset – pääsevät käyttämään tietojärjestelmässä olevia resursseja ja tietoa käyttäjäroolinsa mukaisesti. Pääsynhallinnan katsotaan yleensä sisältävän todennuksen, *valtuutuksen* (engl. authorization) ja käyttäjän toimista koostuvan *tapahtumaketjun* (engl. audit trail) kirjaamisen lokiin.

Todennus todistaa käyttäjän tahoksi, jolla on oikeus järjestelmän käyttämiseen. Tyypillisesti todennus tapahtuu, kun käyttäjä antaa käyttäjätunnuksen ja salasanan yhdistelmän järjestelmään sisäänkirjautumiseksi. Todennus ei kuitenkaan vielä ota mitään kantaa siihen, mitä resursseja käyttäjä saa järjestelmässä käyttää.

Kullakin käyttäjällä on tietyt käyttöoikeudet eli hänelle myönnetyt oikeudet järjestelmän tiet-

tyjen resurssien käyttöön. Käyttäjillä on järjestelmässä erilaisia rooleja; kaikilla ei luonnollisestikaan ole oikeuksia tarkastella ja muokata kaikkia järjestelmän resursseja. Roolipohjaiset järjestelmät ovat varsin suosittuja. Käyttäjän todennuksen jälkeen järjestelmä käyttää toimintansa aikana pääsynhallintasääntöjä määrittämään, onko käyttäjä valtuutettu tietyn resurssin tarkasteluun tai muokkaamiseen vai evätäänkö pääsy resurssiin. Kuvassa 3.7 on esitetty todennuksen ja valtuutuksen periaate pääsynhallinnassa.

Kaikkien käyttäjän suorittamien toimenpiteiden kirjaaminen ylös järjestelmän lokitietoihin varmistaa kiistämättömyyden ja mahdollistaa tarvittaessa käyttäjien aiheuttamien tapahtumien myöhemmän todistamisen [32]. Lokitietojen tallentaminen on tärkeää esimerkiksi silloin, kun epäillään valtuutetun henkilön kiellettyä tai sopimatonta toimintaa.



Kuva 3.7: Todennus ja valtuutus pääsynhallinnassa.

Järjestelmän elinkaaren aikana ylläpitäjän tulee huolehtia *käyttöoikeuksien hallinnasta* (engl. access control). Käyttöoikeuksien myöntämiseen ja poistamiseen tulisi olla olemassa selkeä käytäntö. Samalla on olennaista seurata, mitä käyttöoikeuksia kullakin käyttäjällä on ja kuka oikeudet on myöntänyt ja milloin. Huonosti kontrolloitu ja ajan tasalla pysymätön käyttöoikeuksien hallintaprosessi heikentää tietoturvaa. On tietoturvan kannalta olennaista antaa käyttäjälle aina niin suppeat käyttöoikeudet kuin mahdollista; kenellekään (tai toisella osajärjestelmällä ”käyttäjänä”) ei tulisi olla oikeuksia sellaisiin resursseihin, joita hän ei työssään tarvitse.

Pääsynvalvontaa varten on useita olemassaolevia teknologiaoita ja valmiita kehyksiä. Esimerkiksi OpenID[10] on sähköisen identiteetin todentamisen web-palveluissa mahdollistava teknologia. Käyttäjä rekisteröityy yhteen OpenID-tunnisteita tarjoavaan palveluun, ja muut web-palvelut voivat varmentaa käyttäjän identiteetin tämän luotetun palvelun kautta. OAuth[11] puolestaan on valtuuksien hallintaan web-palveluissa keskittyvä ohjelmistokehys. Tunnistetun ja todennetun käyttäjän käyttöoikeuksien perusteella hänet voidaan valtuuttaa käyttämään tiettyä web-palvelimen resurssia.

[10] http://openid.net/
[11] https://oauth.net/2/

3.3.3 Luotetut suoritusympäristöt

Luotetun suoritusympäristön (engl. trusted platform) ajatus on taata sisältämänsä datan eheys sekä se, että kyseisessä ympäristössä sisällä ajettavien ohjelmien suoritus sujuu odotetusti ja häiriöittä. Näiden periaatteiden toteuttamiseksi luotettu suoritusympäristö hyödyntää laitteistoon integroituja kryptografisia avaimia. Se voi perustua esimerkiksi niin sanottuun TPM-moduuliin (engl. Trusted Platform Module) eli laitteistossa olevaan mikrokontrolleriin, johon säilötään tietyn laitteen identiteetin määrittteleviä avaimia ja salasanoja. TPM toimii eräänlaisena perustana, jonka päälle voidaan rakentaa luotettuja tietoturvaratkaisuja ja joka takaa ylempien kerrosten eheyden ja luottamuksellisuuden. Luotetussa suoritusympäristössä ajettavat ohjelmistot voivat käyttää TPM:n palveluja esimerkiksi säilömällä siihen turvallisesti luottamuksellista tietoa ja kryptografisia avaimiaan.

TPM:n lisäksi tunnettuja luotettuja suoritusympäristöjä ovat Intel SGX ja AMD TrustZone. Intel SGX:n (Software Guard Extensions) pääidea on, että luotetun laitteiston päälle voidaan muodostaa turvallinen säiliö (engl. enclave), jossa voidaan suorittaa laskentaa niin, että eheys ja luottamuksellisuus toteutuvat datan ja sillä operoivan koodin osalta. Toiselle osapuolelle voidaan luotettavasti todistaa, että säiliön sisällä suoritetaan ohjelmaa turvallisesti näiden tietoturvan perusarvojen vaarantumatta. Vastaavasti AMD:n TrustZone-teknologia jakaa järjestelmän normaaliin ja turvalliseen osioon (engl. normal and secure worlds). Normaalista osiosta ei pääse käyttämään turvallisen osion resursseja. Teknologiaa käytetään takaamaan esimerkiksi käyttöjärjestelmän ytimen eheys.

Luotetut suoritusympäristöt mahdollistavat seuraavat teknologiat:

- *Suojattu tietovarasto* (engl. sealed storage). Suojattuun tietovarastoon voidaan säilöä tietoa turvallisesti siten, että sen voi purkaa vain tietyn järjestelmäkofiguraation (laitteiston ja ohjelmien yhdistelmä) vallitessa. Näin ollen esimerkiksi haittaohjelma, joka rikkoo tämän eheän tilan, ei koskaan voi päästä käsiksi tietoihin.

- *Muistialueiden eristäminen* (engl. memory curtaining). Prosesseja estetään kirjoittamasta ja lukemasta toistensa muistiavaruuksia. Näin prosesseja voidaan suorittaa luotetussa suoritusympäristössä ilman, että mikään toinen prosessi tai järjestelmän käyttäjä pääsee haitallisesti muokkaamaan tai vakoilemaan niitä tai häiritsemään niiden suoritusta.

  Tähän liittyviä teknologioita edustavat ns. konttiteknologiat (engl. container technologies), joiden ideana on tarjota turvallisia, keveitä virtuaalisia suoritusympäristöjä perustuen virtuaalikoneiden ja niiden prosessien eristämiseen toisistaan.

- *Turvallinen siirräntä* (engl. secure input/output). Prosessi voi turvallisesti vastaanottaa syötteitä ja esittää tietoa käyttäjälle ilman, että jokin toinen prosessi muokkaa tai tarkastelee tätä tietoa.

- *Etätodistaminen* (engl. remote attestation). Toiselle osapuolelle voidaan todistaa luotettavasti, että luotetussa suoritusympäristössä suoritetaan alkuperäistä, muuntelematonta versiota tietystä ohjelmasta.

- *Homomorfinen salaus* (engl. homomorphic encryption). Homomorfisessa salauksessa data salataan siten, että siihen voidaan soveltaa laskentaoperaatioita purkamatta salausta. Vaikkapa autonominen alus voisi lähettää toiselle taholle salatun kyselyn lähimmän aluksen sijainnista, ja vastaanottaja voisi laskea ja palauttaa kyseisen tiedon saamatta koskaan tietää, mitä tietoa lähettäjä halusi laskettavan. Homomorfisen salauksen varjopuoli on, että se on toistaiseksi liian tehotonta useisiin käyttötarkoituksiin.

3.3.4 Koventaminen

Koventaminen (engl. hardening) tarkoittaa sellaisten järjestelmän ominaisuuksien, palvelujen, ja ohjelmien poistamista tai käytön estämistä, joita ei tarvita järjestelmän käyttötarkoituksen toteutumiseen [31]. Usein koventamisessa tehdään myös muutoksia konfiguraatioon siten, ettei järjestelmän ominaisuuksia voida väärinkäyttää. Käytännön esimerkkejä tästä ovat pääsyoikeuden rajoittaminen tiettyihin tiedostojärjestelmän osiin ja järjestelmässä säilytettävien arkaluontoisten tietojen salaaminen. Samoin käyttöjärjestelmän tarpeettomat palvelut ja avoimet verkkoportit, joita ei tarvita, on syytä kokonaan sulkea. Näin pienennetään sitä hyökkäyspinta-alaa, jota haittaohjelmat voivat hyökkäyksissään hyödyntää.

Eräänlaiseksi koventamiseksi voidaan myös ajatella järjestelmässä suoritettavien ohjelmien eli prosessien käyttäminen muistialueiden eristäminen toisistaan. Aiemmin mainittujen luotettujen suoritusympäristöjen lisäksi prosessorien eristämiseen voidaan käyttää *säiliöitä* (engl. containers), jotka on käyttöjärjestelmätason virtualisointitekniikka, jolla prosessit voidaan suorittaa toisistaan erotetuissa muistiavaruuksissa ilman, että niillä on mahdollisuutta havaita tai häiritä toisiaan. Vaikka hyökkääjä pääsisi käsiksi yhteen eristettyyn prosessiin, tästä eristetyt järjestelmän osiot ovat turvassa. Esimerkki paljon käytetystä säiliöiden luotiin tarkoitetusta teknologiasta on Docker[12].

Koventamistekniikaksi voidaan katsoa myös *järjestelmän rajapintojen diversifiointi* (engl. interface diversification) [12]. Haittaohjelmien täytyy järjestelmässä toimiakseen päästä käsiksi sen kriittisiin resursseihin (kuten esimerkiksi verkkoyhteyksiin ja tiedostojärjestelmään), ja tämä saavutetaan käyttöjärjestelmässä tiettyjen hyvin tunnettujen rajapintojen, kuten systeemikutsujen avulla. Jos nämä hyvin tunnetut rajapinnat muutetaan järjestelmäkohtaisesti salaisiksi niin, että tietyt luotetut ohjelmat järjestelmässä osaavat käyttää niitä eli tietävät niin kutsutun diversifiointisalaisuuden, eivät haittaojelmat enää pysty hyödyntämään järjestelmän rajapintoja eivätkä enää pysty vahingollisesti vuorovaikuttamaan ympäristönsä kanssa. Käytännön esimerkki diversifioinnista olisi esimerkiksi muuttaa käyttöjärjestelmän systeemikutsujen numerot sekä kaikkien niiden kirjastofunktioiden nimet, jotka suorasti tai epäsuorasti käyttävät näitä systeemikutsuja. Näin ikään kuin tukitaan kaikki järjestelmän (sisääntulo)pisteet, joista haittaohjelma voisi päästä käsiksi kriittisiin resursseihin.

Koventaminen vaatii syvällistä tietämystä siitä, mihin kutakin järjestelmän ominaisuutta käytetään. Järjestelmän peruskonfiguraatio tulisi säätää siten, että käyttäjillä ja ohjelmistoilla on vain toiminnallisuudelle täysin välttämättömät käyttöoikeudet. Käyttäjien ei tulisi voida asentaa tai suorittaa asiaankuulumatomia ohjelmia eikä muuttumaan järjestelmän tietoturva-asetuksia. Ko-

[12]https://www.docker.com/

ventamisesta tulisi aina huolehtia ennen järjestelmän käyttöönottoa järjestelmien toimitusvaiheessa.

Palomuurien soveltaminen on myös järjestelmän koventamista verkkotasolla. Käytännössä palomuurilla säädellään, minkälaisia verkkoyhteyspareja verkossa voi olla. Tyypillisesti suljetaan lähes kaikki TCP-tason päällä toimivat sovellustason portit.

Ohje 3.5 (Kehittäjä) Suositeltavia koventamistoimenpiteitä.

- Riisutun käyttöjärjestelmäversion käyttäminen tai järjestelmän käyttötarkoituksen kannalta turhien toimintojen ja palvelujen poistaminen käytöstä
- Ohjelmistojen ja käyttöjärjestelmän asetusten turvattomien oletusarvojen muuttaminen turvallisiksi
- Vain välttämättömien käyttöoikeuksien antaminen sovelluksille ja käyttäjille
- Käyttämättömien verkkoporttien sulkeminen
- Säännölliset turvallisuuspäivitykset
- Vain tiettyjen määriteltyjen sovellusten suorituksen salliminen (valkolistaus)
- Kriittisien prosessien ja niiden tietojen eristäminen toisistaan
- Vahvojen salasanojen vaatiminen

3.3.5 Tunkeutumisen havaitseminen ja monitorointi

Uusia haittaohjelmia, vanhojen haittaojelmien variaatioita ja uudenlaisia kyberhyökkäyksiä syntyy jatkuvasti lisää kasvavalla tahdilla[13]. Tämän takia täytyy varautua myös täysin uudenlaisiin uhkiin, joita ei vielä laajalti tunneta ja joita esimerkiksi virustorjuntaohjelmat eivät kykene tunnistamaan. Hunajapurkit ja tunkeutumisen havaitsemisjärjestelmät kykenevät useissa tapauksissa tunnistamaan myös tällaisia haittaohjelmia ja hyökkäyksiä.

Hunajapurkki (engl. honeypot) tarkoittaa kyberturvallisuuden terminologiassa hyökkääjälle asetettua ansaa, jonka avulla voidaan havaita järjestelmää vastaan tehtävät hyökkäykset ja kerätä tietoa hyökkääjän toimista [28]. Käytännössä hunajapurkki on yleensä huonosti suojattu palvelin, johon hyökkääjän odotetaan murtautuvan. Hyökkääjä kuvittelee tunkeutuvansa oikeaan järjestelmään, ja tämän vaikutelman tukemiseksi hunajapurkissa on usein monenlaisia hyökkääjän näkökulmasta mielenkiintoisilta ja arvokkailta vaikuttavia valeresursseja. Hunajapurkki on todellisuudessa eristetty muista verkossa olevista laitteista ja siihen on asennettu hyökkääjän toi-

[13]Tietoturvayritykset havaitsevat uusia haittakoodeja satoja miljoonia joka vuosi – maailmanlaajuisesti tarkastellen liikutaan yli miljoonassa uuden(laisen) haittaohjelman havainnossa per päivä!

mia huomaamattomasti seuraavaa toiminnallisuutta, kuten esimerkiksi tunkeutumisen havaitsemisjärjestelmä.

Tunkeutumisen havaitsemisjärjestelmä (engl. Intrusion Detection System, IDS) on yksittäisellä laitteella tai verkossa oleva järjestelmä, joka valvoo tietyllä laitteella suoritettavia toimintoja tai verkon liikennettä ja tunnistaa tämän perusteella järjestelmään tai verkkoon kohdistuvat kyberhyökkäykset [4]. Konekohtainen tunkeutumisen havaitsemisjärjestelmä eli HIDS (engl. Host IDS) keskittyy tarkkailemaan järjestelmän tapahtumissa havaittavia poikkeamia, esimerkiksi järjestelmän logeja, epätavallisia prosesseja tai tiedostojärjestelmässä tapahtuvia outoja muutoksia. Verkkoon sijoittuva tunkeutumisen havaitsemisjärjestelmä eli NIDS (engl. Network IDS) puolestaan kerää verkkoliikennettä, josta voidaan etsiä hyökkäyksen merkkejä. Tunkeutumisen havaitsemisjärjestelmät etsivät keräämänsä tapahtumadatan tai verkkoliikenteen perusteella selviä poikkeavuuksia järjestelmän normaaliin toimintaan verrattuna, jolloin voidaan päätellä hyökkäyksen olevan meneillään ja ryhtyä tarpeellisiin toimenpiteisiin – esimerkiksi katkaista hyökkääjän yhteys verkkoon.

Autonomisten alusten yhteydessä NIDS edustaa eräänlaista koventamista verkkotasolla, mikä saattaa olla kovin tarpeellista, jos autonominen alus muodostetaan konvertoimalla perinteinen alus autonomiseksi. Tuolloin NIDS tarkkaillee erityisesti legacy-osioita. Vastaavasti HIDS:n käyttö on myös keino lisätä järjestelmän valvontaa autonomisella aluksella. Tämä on erityisen hyödyllistä ulkoisten rajapintojen ”lähellä”. HIDS auttaa myös legacy-järjestelmien valvonnassa autonomisen aluksen muodostamisessa konversion kautta.

3.3.6 Virustorjunta

Virustorjuntaohjelma on olennainen osa järjestelmän suojaamista haitallisilta ohjelmilta. Ylläpitäjän on tärkeää pitää huolta siitä, että ohjelmisto pidetään ajan tasalla. Virustorjuntaohjelman tulee olla aina aktiivinen eikä sitä tule normaalitilanteessa voida ottaa pois päältä.

Sulautetuissa järjestelmissä – myös autonomisten alusten kontekstissa – on virustorjuntaohjelmien osalta kuitenkin aina huomioitava, että ne voivat esimerkiksi suuren resurssienkäyttönsä takia mahdollisesti aiheuttaa häiriöitä ja viivettä muiden kriittisten prosessien toiminnassa. Virustorjuntaohjelman toimivuutta onkin testattava ennen käyttöönottoa varsinaisessa järjestelmässä, ja testaus on järjestelmän muuttuessa ja virustorjuntaohjelman uusien versioiden ilmestyessä toistettava.

3.3.7 Syötedatan eheystarkistukset ja koodi-injektion estäminen

Siinä missä IDS-järjestelmät keskittyvät tapahtumiin ja virustorjunta tiedostoihin, rajapintojen kautta tulevan tiedon oikeelliseen muotoon on syytä myös kohdistaa tarkastuksia. Yksi keskeinen tietoturvaongelmien mahdollistaja on väärämuotoisen tiedon syöttäminen hyökkääjän toimesta niin, että syötetty data tulkitaan (osittain) suoritettavaksi koodiksi. Tällaiset injektiohyökkäykset onnistuvat kahdesta syystä:

1. rajapinnan läpi saadaan toimitettua väärämuotoista tietoa – vastaanottava taho ei suorita tarpeeksi dataa koskevia eheystarkastuksia (esim. jos data edustaa henkilön etunimeä se ei

voi olla 10,000 merkkiä pitkä sisältäen binäärisiä ei-merkkiä tarkoittavia arvoja, ja

2. väärämuotoinen data onnistutaan injektoimaan paikkoihin, joissa se tulkitaan koodiksi.

Jälkimmäistä vastaan on nykyisissä käyttöjärjestelmissä kaksi varsin tehokasta mekanismia: (a) suoritetun koodin kryptografiset allekirjoitukset, jolloin muuttunut (injektoidun osuuden sisältävä) koodi ei ole enää suorittavan järjestelmän kannalta kelvollinen, ja (b) muistialuemäärityksiin perustuva datan suorittamisen koodina estäminen (engl. data execution prevention, DEP). Ideana DEP:ssä on, että koodia sisältävää muistialueen sisältöä ei voi muuttaa (muistisivumääritykset) ja data sisältävää voi muuttaa mutta ei suorittaa. DEP on viime vuosina levinnyt laajasti mm. Linux- ja Windows-pohjaisiin käyttöjärjestelmiin, mutta esim. IoT-ympäristöissä sitä ei juuri ole käytössä.

Kohtaa (1) vastaten pitää ohjelmistotuotannossa yksinkertaisesti lisätä tarkistuksia koskien vastaanotettavan datan muotoa, eheyttä. Eri konteksteissa tätä varten löytyy valmista tukea, esim. kirjastojen muodossa ja vahvasti tyypitettyjen kielten soveltamisen muodossa.

3.4 Kyberturvallisuus tiedonsiirron kannalta

3.4.1 Tiedonsiirron salaus

Tietoliikenteen tietoturvallisuuden osalta olennaista on varmistaa, että liikenne on aina vahvasti salattua edellä esitetyin kryptografisin keinoin. Näin varmistetaan, ettei hyökkääjä pääse salakuuntelemaan tai muokkaamaan verkon yli siirtyvää tietoa. Salaus tulee aina tehdä vahvaksi todetulla salausmenetelmällä käyttäen tarpeeksi pitkää salausavainta, jota säilytetään asianmukaisella tavalla [34]. Vain sellaiset tietoliikennetavat, jotka voidaan riittävällä tavalla salata ja joille on järjestelmän ja työtehtävien kannalta oikeaa käyttöä, tulisi sallia. Erilaisten tietoliikenteen yhteystapojen sekä salausalgoritmien mahdollisista tietoturvaheikkouksista on jatkuvasti syytä pysyä ajan tasalla, jotta nämä puutteellisen menetelmät voidaan tarvittaessa korvata ja tietoliikenne pysyy turvattuna.

Esimerkiksi aiemmin mainittu TLS on paljon käytetty yhteyskäytänne, jolla liikenne voidaan salata. Alemmalla verkkokerroksella voidaan datan turvaamiseen käyttää IPsec-teknologiaa (IP Security Architecture)[14], joka on koostuu joukosta Internet-yhteyksien turvaamiseen tarkoitettuja protokollia. IPSec varmistaa tiedon luottamuksellisuuden ja eheyden sekä todentaa viestinnän osapuolet. Salausta käytetään tavallisesti myös VPN-teknologian (Virtual Private Network)[15] eli virtuaalisten erillisverkkojen yhteydessä. VPN mahdollistaa näennäisesti yksityisen verkon muodostamisen julkisen Internetin yli, verkon yksityisyys ja tietoturva taataan salauksella.

Tietoliikenteen salaustarpeen osalta tulee ymmärtää, että tietoliikenteessä on monia eri kerroksia ja salauksen tarve on suurinta ylimmillä kerroksilla. Periaatteessa riittää, että salaus suoritetaan vain ylimmällä sovelluskerroksella, jolloin tietoliikenteen hyötykuorma on aina salattuna. Tuolloin on mahdollista, että alemmilla tietoliikennekerroksilla osa (lähinnä osapuolten yhteystiedot) viestinnän sisällöstä näkyy selväkielisenä, mutta silloinkin itse hyötykuorma on salattua,

[14] https://tools.ietf.org/html/rfc2401
[15] https://tools.ietf.org/html/rfc2764

jolloin hyökkääjä ei pysty sitä purkamaan. Näissä tilanteissa tulee varautua toistohyökkäyksiä vastaan – tuolloin hyökkääjä voi pyrkiä toistamaan jotain nauhoittamaansa aiemmin lähetettyä salakirjoitetussa muodossa olevaa tietosisältöä. Tämän estäviä kryptografisia protokollia on kehitetty.

Autonomisten alusten tapauksessa on monestakin syystä tarpeen käyttää tietoliikennekanavien kirjoa. Ylipäänsä kaikki kanavat eivät joka hetki ole käytössä ja toisaalta palvelunestohyökkäyksien takia on tarve olla käytössä vaihtoehtoisia kanavia. Kun tietoliikenteen salaus tehdään ylimmillä kerroksilla (sovelluskerroksella), niin kanavien vaihtaminen toiseen häiriötilanteessa pitäisi olla suoraviivaista ilman, että salausta täytyy uudistaa, sillä vai MAC-taso (Media Access Control) vaihtuu sovellustason pysyessä samana.

3.4.2 Luvattoman liikenteen estäminen ja tarkkailu

Palomuuri (engl. firewall) on järjestelmä, joka estää asiattoman pääsyn verkosta toiseen, esimerkiksi julkisesta internetistä autonomisen aluksen lähiverkkoon [11, 30]. Palomuuri suojaa verkkoa ja järjestelmää ulkomaailmasta tulevia kyberhyökkäyksiä vastaan. Myös järjestelmästä lähtevää liikennettä voidaan suodattaa, jotta esimerkiksi järjestelmään mahdollisesti päässyt haittaohjelma ei voisi häiriköidä ulkomaailmaa. Palomuurin toimintaperiaate on esitetty karkealla tasolla kuvassa 3.8.

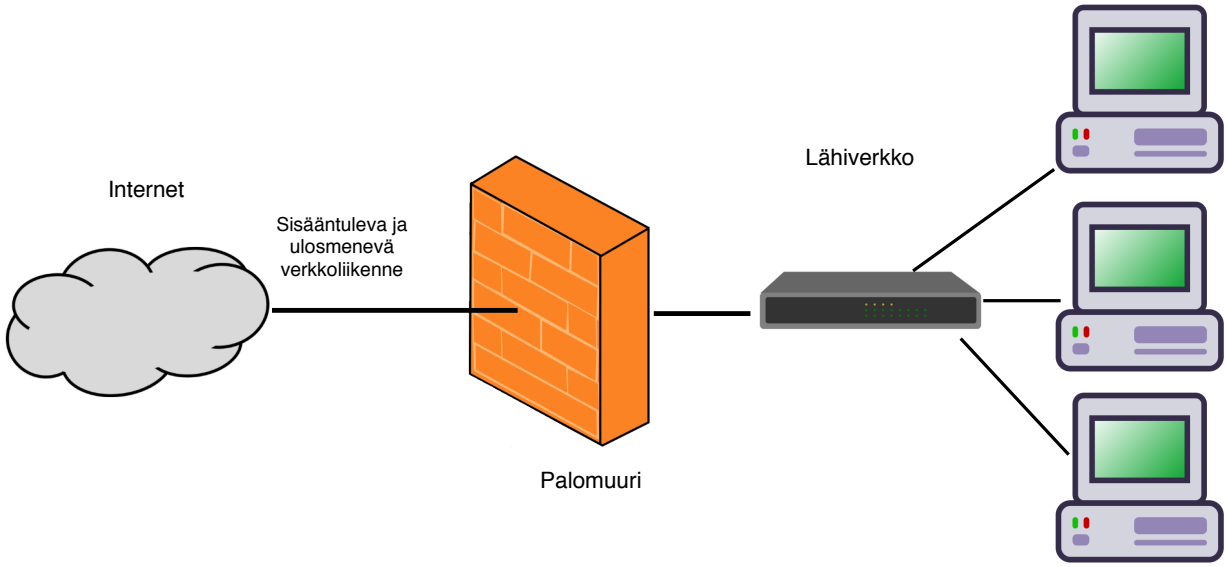
Käytännössä palomuuri suodattaa verkkoliikennettä esimerkiksi sen lähettäjän ja sisällön perusteella. Edistyneemmät palomuurit osaavat usein myös tunnistaa liikenteen, joka sisältää haitallista koodia tai luvattomia komentoja. Palomuuri voi myös saada ohjeita tunkeutumisen havaitsemisjärjestelmältä siitä, millaista liikennettä sen tulisi suodattaa pois. Palomuurin tehtävä on myös estää liikenne sellaisiin verkkoportteihin (järjestelmän palvelupisteisiin), joiden ei ole tarkoitus olla järjestelmässä käytössä.

Palomuuri tulee konfiguroida sallimaan vain asianmukainen liikenne käytössä oleviin palveluihin. On huomattava, että palomuuri suodattaa vain lävitseen kulkevan tietoliikenteen. Hyökkääjä voi päästä käsiksi organisaation verkkoon vaikkapa langattoman lähiverkon tukiasemien avulla tai tulemalla fyysisesti organisaation tiloihin, eikä palomuurista tällöin ole hyötyä.

Organisaation verkoissa tapahtuvaa liikennettä tulee valvoa siinä tapahtuvat epänormaalin tai haitallisen toiminnan varalta. Aiemmin tässä aliluvussa mainitut tunkeutumisen havaitsemisjärjestelmät ovat hyvä apuväline poikkeavan verkkoliikenteen havaitsemisessa. Tapahtumalokeja pitävät myös useat laitteet kuten tukiasemat ja reitittimet. Nämä verkkoliikenteen tapahtumatiedot ovat tärkeitä kyberhyökkäysten ja vikatilanteiden selvittämisessä sekä tulevien ongelmatilanteiden ennaltaehkäisyssä.

3.4.3 Langattomat lähiverkot

Lähes kaikilla organisaatioilla on nykyisin käytössään langattomia lähiverkkoja, joilla verkkolaitteet ja järjestelmän osat voidaan yhdistää toisiinsa käyttämättä kaapeleita. Langaton verkko on aina suojattava hyvällä salasanalla; salasanan on oltava riittävän pitkä ja sisällettävä monipuolisesti merkkejä, sitä tulee vaihtaa riittävän usein. Verkon salasana ei saa olla missään julkisesti esillä eikä oletussalasanoja saa käyttää.



Kuva 3.8: Palomuurin toiminta.

Ellei langaton verkko ole täysin eristetty järjestelmän kriittisestä toiminnallisuudesta, siihen ei koskaan tule päästää laitteita, joiden tietoturvasta ei välttämättä ole huolehdittu asianmukaisesti (esimerkiksi aluksen operointikeskuksen operaattorien omat kannettavat tietokoneet). Verkkoon laiteellaan liittyvältä käyttäjältä tulisi aina myös vaatia todennus. Organisaation tietoverkot on tietoturvan kannalta järkevää segmentoida siten, että kriittinen toiminnallisuus on eristetty omaan osioonsa esimerkiksi virtuaalisia lähiverkkoja eli VLAN-tekniikkaa käyttämällä[16].

Eräissä tukiasemamalleissa on paljastunut takaportteja, jotka mahdollistavat tietomurron[17]. Takaportti (engl. backdoor) on ominaisuus, joka sallii hyökkääjän ohittaa normaalit tietoturvatoimet ja mahdollistaa yleensä laitteen etähallinnan. Tämän mahdollisuuden ja muiden lähiverkkojen ja tukiasemien mahdollisten haavoittuvuuksien tietoturvatiedotteita on syytä seurata ja tarvittaessa vaihtaa verkon laitteita uudempiin malleihin. Tukiasemien ohjelmistot eivät yleensä päivity automaattisesti. Siksi verkon ylläpitäjän tulee myös säännöllisesti tarkastaa, onko laitteille jaossa uusia päivityksiä.

3.4.4 Yhteyksien ja järjestelmien monistaminen

Yhden tietoliikenneyhteyden tai järjestelmän toimivuuteen ei tule luottaa, vaan käytettävissä tulisi aina olla varajärjestelmä kriittiselle toiminnallisuudelle. Järjestelmien toimivuutta valvotaan erillisellä monitorointikomponentilla, ja tarvittaessa vaihdetaan automaattisesti käytössä olevaa aktiivista järjestelmää. Tällainen redundanssi alusten järjestelmien suunnittelussa häiriötilanteiden välttämiseksi on jo valmiiksi tärkeää, mutta autonomisuus ja kyberuhkien läsnäolo tekevät

[16] http://www.ieee802.org/1/pages/802.1Q.html
[17] https://www.viestintavirasto.fi/kyberturvallisuus/tietoturvanyt/2014/09/ttn201409301601.html

siitä entistä olennaisempaa. Lähtökohtana tulee siis olla, että vioista tai hyökkäyksistä huolimatta autonominen alus kykenee turvalliseen toimintaan.

Autonomisten alusten yhteydessä vaikkapa yhteys etävalvomoon ja paikannus ovat toiminnallisuuksia, jotka on erityisen tärkeää monistaa. Etteivät toiminnallisuudet olisi alttiita samoille haavoittuvuuksille, on järkevää käyttää rinnakkain keskenään erilaisia järjestelmiä ja viestintäkanavia, esimerkiksi satelliitti- ja radioyhteydet. Tällainen toiminnallisuuden diversiteetti niin fyysisellä kuin ohjelmistojen rakenteenkin tasolla lisää järjestelmän toimintavarmuutta, kun ei sokeasti luoteta vain yhden järjestelmän ja sen komponenttien toimivuuteen.

Tietoturvan perusarvoista monistaminen liittyy saavutettavuuteen: vaikka yksi tietoliikenneyhteys muuttuisi toimintakyvyttömäksi esimerkiksi palvelunestohyökkäyksen seurauksena, toimisi vaihtoehtoinen, erilaiseen toteutukseen pohjautuva ratkaisu todennäköisesti edelleen. Myös kriittinen data on järjestelmissä syytä monistaa vikasietoisesti; jos datan eheys vaarantuu yhdessä paikassa, on varmuuskopio yhä olemassa toisaalla.

Ohje 3.6 (Kehittäjä) Turvallisen tiedonsiirron olennaisia periaatteita.

- Viestintä verkon yli tulee aina salata.
- Asiattomat yhteydet ulkopuolisista tietoliikenneverkoista järjestelmään tulee estää kattavasti.
- Tietoliikenneyhteydet ja järjestelmät tulee monistaa häiriöiden ja palvelunestohyökkäysten varalta erilaisiin ratkaisuihin pohjautuvilla toteuksilla.
- Verkon kriittiset osat tulee eristää muusta verkosta.
- Verkkoliikennettä tulee jatkuvasti valvoa esimerkiksi tunkeutumisen havaitsemisjärjestelmän avulla ja kerätä tapahtumista lokitietoja.

3.5 Kyberturvallisuus käyttäjän näkökulmasta

Päätelaitteiden ja niiden ohjelmistojen osalta järjestelmän tietoturvan ylläpito on paljolti myös tavallisen loppukäyttäjän varassa. Tietoturvalliseksikaan suunniteltu järjestelmä ei nimittäin voi pysyä kaikilta osin turvallisena, ellei sen tietoturvaa ylläpidetä säännöllisesti [33]. Käyttäjän laiminlyönnit tai järjestelmän käyttäminen väärin voivat altistaa järjestelmän luvattomalla käytölle. Toteutuneissa kyberhyökkäyksissä ja tietomurroissa tekniikka toimii usein oikein, mutta ihminen eli laitteen käyttäjä on huijattu toimimaan väärällä tavalla. Seuraavaksi käydään läpi järjestelmän tietoturvan ylläpitoa käyttäjän näkökulmasta ja annetaan käyttäjille yleisluontoisia ohjeita tietoturvan edistämiseksi.

3.5.1 Salasanat

Salasanat ja niiden hallinta ovat merkittävässä osassa ICT-järjestelmien kyberturvallisuuden toteuttamisessa. Salasanan ja käyttäjätunnuksen avulla rajataan pääsy järjestelmään ja tietoihin vain niille henkilöille, joilla on oikeus käyttää järjestelmää ja käsitellä tietoja. Käyttäjätunnus on yleensä suhteellisen helposti selvitettävissä, mutta salasana on nimensä mukaisesti salassa pidettävä tieto, jolla pyritään estämään luvaton pääsy järjestelmään tietyllä käyttäjätunnuksella. Hyvän salasanan valitseminen on siksi ensiarvoisen tärkeää. Hyvä salasana on [37]:

- *Tarpeeksi pitkä*. Salasanan tulee olla riittävä pitkä, mieluusti vähintään 15 merkkiä. Näin salasanan arvaaminen on hankalaa, vaikka järjestelmään haluava hyökkääjä näkisi salasanan syöttämisen tai hänen hallussaan olisi henkilökohtaisia tietoja käyttäjästä. Pitkä salasana suojaa myös automaattisia salasananarvaushyökkäyksiä vastaan, sillä tarpeeksi merkkejä sisältävän salasanan murtaminen on laskennallisesti haastavaa.

- *Monipuolisesti erilaisia merkkejä sisältävä*. Salasanan arvaaminen tulee hankalammaksi myös silloin, kun siinä käytetään paljon erilaisia merkkejä. Hyvän salasanan tulisikin sisältää eri kirjainkokoja, numeroita ja erikoismerkkejä. Numeroita valitessa ei kannata suosia helposti arvattavia korvauksia, kuten numeron 1 käyttämistä numeron i-kirjaimen sijasta.

- *Hankalasti arvattava*. Omat henkilökohtaiset tiedot, muiden järjestelmien salasanat tai muuten yleisesti käytössä olevat salasanat eivät ole hyvä valinta salasanaksi tai sen osaksi. Normaalien sanojen tai muutamien sanojen yhdistelmien käyttäminen altistaa salasanan sanakirjahyökkäyksille, jossa hyökkäjä arvailee salasanoissa usein esiityviä salasanoja. Tämän takia “salasana123” tai “kissakoira” ovat äärimmäisen huonoja salasanoja.

- *Enemmän kuin yksittäinen sana*. Nimestään huolimatta salasanan ei tulisi olla sana, vaan on parempi käyttää esimerkiksi kokonaista lausetta, joka on muille täysin tuntematon. Vaikkapa lauseesta “Vuoteen 2035 mennessä maailman merillä liikennöi jo useita autonomisia miehittämättömiä aluksia!” voisi muodostaa suhteellisen turvallisen salasanan “V2035mmmljuama!”.

Turvallisten salasanojen luomiseen on olemassa myös apuohjelmia. Niiden avulla pystyy automaattisesti generoimaan itselleen pitkiä ja näennäisen satunnaisia salasanoja. Hyvääkin salasanaa on tärkeä vaihtaa säännöllisesti, ettei hyökkääjä saa sen murtamiseen liikaa aikaa. Salasana tulisi uusia ainakin puolen vuoden välein. Seuraava ohje tiivistää salasanoihin liittyvät olennaiset asiat.

Ohje 3.7 (Operaattori) Salasanoihin liittyviä ohjeita.

- Salasanan on oltava tarpeeksi pitkä, monipuolisesti merkkejä sisältävä ja hankalasti arvattava salasana.
- Eri järjestelmiin tulee valita eri salasanat.
- Salasanaa ei saa tule tallentaa järjestelmän automaattisesti muistettavaksi, esimerkiksi verkkoselaimen automaattinen täyttö -toiminnolla.
- Salasanaa ei tule kirjoittaa muistiin lapulle työaseman viereen eikä kirjoittaa sitä koneella muiden katsellessa.
- Salasana tulee vaihtaa säännöllisin väliajoin.

3.5.2 Ohjelmistojen pitäminen ajan tasalla

Ohjelmistojen ja niiden tietoturvan pysymisestä ajan tasalla huolehditaan säännöllisillä päivityksillä. Käyttäjän tulee huolehtia käyttöjärjestelmän, työasemalle asennettujen ohjelmistojen sekä tietoturvaohjelmiston pysymisestä ajantasaisina. Järjestelmän sisäänrakennetut suojaukset voivat toimia suunnitellusti vain, kun käyttäjä tekee oman osansa niiden ylläpidossa.

Ellei päivityksiä ole säädetty tapahtuviksi automaattisesti, käyttöjärjestelmä ja ohjelmistot muistuttavat siitä, että uusimmat päivitykset tulisi ladata. Tällöin päivitykset on syytä ladata ja asentaa koneelle välittömästi, jotta ohjelmistoissa olevien, uusimpien haittaohjelmien hyödyntämien tietoturva-aukkojen määrä säilyy mahdollisimman vähäisenä.

Nykypäivänä yhä suurempi määrä käyttäjän toimista tapahtuu web-ympäristössä, mikä on kasvattanut verkkoselainten (kuten Chrome, Firefox ja Internet Explorer) merkitystä. Useat haittaohjelmat hyödyntävät verkkoselaimissa olevia haavoittuvuuksia. Tämän vuoksi verkkoselainten pitämiseen ajantasaisina pitää kiinnittää erityistä huomiota, mikäli niitä käytetään kriittisillä työasemilla, esim. autonomisten alusten etävalvontaan ja/tai operointiin.

3.5.3 Järjestelmän ulkopuolelta tulevat ohjelmat ja tiedostot

Järjestelmän ulkopuolelta tulevien ohjelmiin ja tiedostoihin tulee aina suhtautua asiaankuuluvalla varovaisuudella ja niiden alkuperä tulee selvittää. Tuntemattomilta verkkosivuilta ei koskaan tulisi ladata työasemalle ohjelmia ja muutenkin on syytä vierailla vain sellaisilla verkkosivuilla, joiden turvallisuuteen voi luottaa. Samaten koneeseen ei tule liittää tallennusvälineitä, kuten muistitikkuja tai ulkoisia kiintolevyjä, joiden sisällöstä ei ole täysin varma.

Ohje 3.8 (Operaattori) Ohjeita työaseman tietoturvallisesta käytöstä.

- Kriittisillä työasemilla ei tule suorittaa ohjelmia, joiden alkuperä ei ole varma.
- Työasemilla ei tule vierailla kuin käyttötarkoitukseen liittyvillä verkkosivuilla.
- Roskaposti- ja tietojenkalasteluviesteihin tulee jättää reagoimatta.
- Käyttäjän ei tule koskea järjestelmän asetuksiin, joiden merkityksestä hänellä ei ole varmaa tietoa.
- Työasemille ei tule liittää ulkoisia tallennusvälinettä, paitsi jos sitä on käytetty vain asianmukaiseen käyttötarkoitukseen (esim. aluksen operointiin liittyvän datan siirto tai ohjelmistojen asennus) ja sen puhtaudesta voidaan näin olla suhteellisen varmoja
- Käyttöjärjestelmä, ohjelmistot, mahdolliset verkkoselaimet ja virustorjuntaohjelma tulee pitää ajan tasalla
- Operointiin tarkoitettua tietokonetta (ja sen selaimia) tulee käyttää vain operointiin

Tuntemattomasta lähteestä peräisin oleva sähköposti on usein niin sanottua roskapostia. Roskapostiviestit voivat sisältää liitetiedostoina haittaohjelmia ja nämä viestit kannattaa hävittää reagoimatta niihin mitenkään. Vieraalta lähettäjältä tulevia liitetiedostoja ei koskaan pidä avata. Haitallisia ohjelmia leviää nykyään myös sosiaalisen median kautta.

On myös syytä kiinnittää huomiota siihen, ettei tiettyyn käyttötarkoituksen varatulla ja kriittisiä toimintoja ohjaavalla työasemalla tehdä mitään sen käyttötarkoitukseen liittymätöntä. Esimerkiksi autonomisen aluksen operointiin omistettuun työasemaan ei välttämättä koskaan tarvitse liittää ulkoisia tallennusvälineitä, eikä sillä ole syytä vierailla käyttötarkoitukseen liittymättömillä verkkosivuilla tai asentaa siihen uusia asiaankuulumattomia ohjelmistoja (operointiin tarkoitettujen työasemien koventaminen). Nämä toimenpiteet voidaan estää esimerkiksi sulkemalla laitteen USB-portit ja koventamalla järjestelmä. Näin todennäköisyys sille, että hyökkääjä tai haittaohjelmat pääsevät häiritsemään kriittistä toiminnallisuutta, vähenee huomattavasti. Näitä varotoimia täydentää työasemilla oleva ajantasainen tietoturvaohjelmisto.

3.5.4 Tietoturvakulttuuri

Pelkät tekniset ratkaisut, kuten virustorjuntaohjelmisto ja palomuuri, eivät yksinään riitä hyvän tietoturvan saavuttamiseen. Monet tekniset ratkaisut ovat hyödyttömiä vaikkapa silloin, jos työntekijän työhuoneen ovi jää auki ja salasana on työaseman vieressä muistilapulla. Olennaista on siis huolehtia myös organisaation tietoturvakulttuurista, eli organisaatiossa vallitsevasta suhtautumisesta tietoturvaan [36]. Tietoturvakulttuuri perustuu käyttäjien riittävään ohjeistukseen ja

koulutukseen tietoturva-asioissa sekä selkeään tietoturvapolitiikkaan.

Kunkin organisaation tulisi oman toimintaympäristönsä, tavoitteidensa ja tunnistettujen riskien pohjalta luoda omiin tarpeisiin räätälöity tietoturvapolitiikka. Tietoturvapolitiikan linjausten perusteella voidaan antaa tarkempaa teknistä ohjeistusta ja luoda tavallisille loppukäyttäjille tarkoitetut käytännön ohjeet, joissa käyttäjiä opastetaan tietoturvalliseen toimintaan.

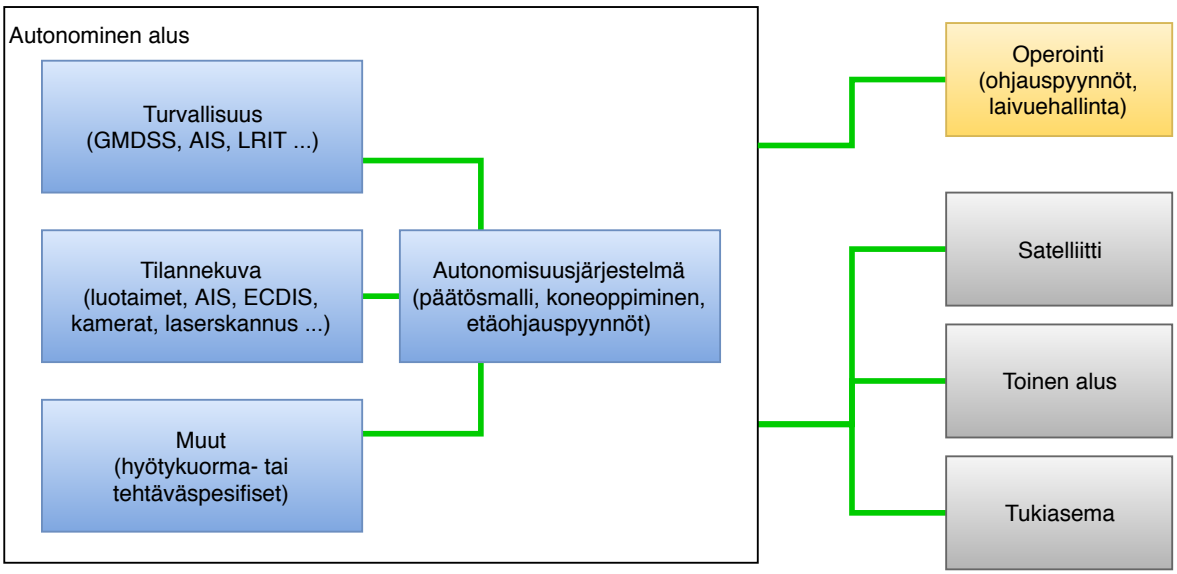
Tietoturvakoulutuksessa käyttäjät tulisi saada ymmärtämään tietoturvan konkreettiset hyödyt käytännön esimerkkien kautta. Tietoturvan tasoa on vaikeaa suoraan nähdä tai mitata, minkä takia organisaatioissa ei usein ymmärretä tietoturvan hyötyjä. Heikon tietoturvan vaikutukset tulevat usein ilmi vasta liian myöhäisessä vaiheessa, kun vahinko on jo tapahtunut. Tietoturvasta tulisi organisaatiossa tehdä osa arkipäivää, ja tietoturvakulttuuria tulee ylläpitää organisaation toimintaympäristön kehittyessä ja riskien muuttuessa.

Tietoturvakulttuurin ja -politiikan luomisessa organisaation johdolla on tärkeä rooli. Johtajien tulee omalla käytöksellään osoittaa pitävänsä tietoturvallisuutta tärkeänä asiana, sillä tämä vaikuttaa myös työntekijöiden tietoturvakäyttäytymiseen. Myös riittävien resurssien allokoiminen esimerkiksi tietoturvakoulutukseen viestii siitä, että tietoturva nähdään organisaatiossa tärkeänä asiana. Hyvä tietoturvakulttuuri takaa sen, että käyttäjän tietoturvatietämys muuntuu tietoturvan huomioimiseksi jokapäiväisissä käytännön toimissa.

Luku 4

Autonomisen merenkulun kyberturvallisuus

Tässä luvussa aiemmin esitettyjä periaatteita ja menetelmiä esitetään sovellettuna autonomiseen merenkulkuun. Luvussa 2.2.4 keskusteltiin alusten autonomisuuden kehittämisestä ja niistä järjestelyistä, joita aluksella pitää olla tai sille pitää toteuttaa aluksen autonomisen operoinnin mahdollistamiseksi. Kuva 4.1 alla kerää nämä uudelleen yhteen.



Kuva 4.1: Autonomisen aluksen merkittävät osat. Sinisellä on merkitty aluksen sisäiset järjestelmät, harmaalla aluksen ulkopuoliset järjestelmät, vihreällä järjestelmien välillä tapahtuva tiedonsiirto sekä keltaisella operointiosuus, johon lukeutuu myös operointikeskus järjestelmänä.

Kuvasta 4.1 voidaan nähdä kolme merkittävää kategoriaa: alusjärjestelmät, tiedonsiirto ja operaattori. Alusjärjestelmiin lukeutuvat luvussa 2.2.1 käsitellyt alusten sisäiset järjestelmät, ku-

ten turvallisuus- ja tilannekuvajärjestelmät sekä luvun 2.2.4 keskustelema autonomisesta päätöksenteosta vastuussa oleva järjestelmä. Tämä käyttää turvallisuus- ja tilannekuvajärjestelmien tuottamaa tietoa tehdessään päätöksiä aluksen navigointioperaatioista. Tiedonsiirtoa tapahtuu sekä alusjärjestelmien välillä kuin myös alusjärjestelmien ja aluksen ulkopuolisten järjestelmien välillä. Ulkopuolisia järjestelmiä ovat muun muassa toiset alukset, rannalla olevat toistinkeskukset, satelliitit, väylään ja satamaan liittyvät tahot sekä etäoperointikeskus. Operointi, viimeinen merkittävä kategoria, on myös esitelty luvussa 2.2.4. Silloin kun autonominen alus ei pysty tai sen ei haluta pystyvän tekemään päätöksiä itse, alusta etäoperoidaan. Tällöin aluksen ja operointikeskuksen välillä viestit siirtyvät tiedonsiirron mahdollistaman linkin yli.

Jokainen keskustelluista pääkategorioista on kyberturvallisuutensa puolesta haavoittuvainen. Luvussa 3.1 on keskusteltu tavoista hallita kyberturvallisuutta yleisesti kun taas luvuissa 3.3 – 3.5 käsiteltiin sen toteuttamista yleisesti järjestelmien, tiedonsiirron ja käyttäjän tai operaattorin kannalta. Seuraavissa aliluvuissa keskustellaan kyberturvallisuuden uhkista sekä hallinnasta näille pääkategoriolle. Tarkemmin luku 4.1 käsittelee järjestelmän uhkia, luku 4.2 tiedonsiirroon uhkia ja luku 4.3 järjestelmien operointirajapinnan uhkia. Alilukujen keskeisenä sisältönä on määrittää erilaisia relevantteja uhkia tai hyökkäysskenaarioita sekä johdattaa lukija pohtimaan (ja toisaalta osittain esittää), miten niille voidaan toteuttaa kyberturvallisuuden hallintaa.

Tarkasteltavat uhkat eivät edusta tyhjentävää luetteloa, mutta pyrkimyksenä on, että ne edustaisivat laajasti erilaisia todellisia uhkia autonomisille aluksille. Esitettävien uhkien toteutumisen todennäköisyys ja vaikuttavuus ovat asioita, jotka riippuvat hyvin paljon autonomisen aluksen olemuksesta – siksi esitetyt uhkaskenaariot ja niiltä suojautumistoimet ovat vain suuntaa-antavia. Esimerkiksi tilanteet, joissa autonominen alus on

- kaupunkivesiliikenteessä ihmisiä kuljettava alus,
- meriväyliä pitkin kulkeva valtameriä ylittävä autonominen rahtialus,
- vaikuttamiskyvyillä varusteltu etäohjattava sota-alus tai
- yhdellä valvovalla henkilöllä varustettu autonominen rahtialus,

ovat kyberriskeiltään ja niiden realisoitumisen osalta kustannuksiltaan hyvin erilaisia. Myös suojautumisen painopiste riippuu lopulta suuresti autonomisen aluksen tyypistä ja sen ympäristöstä.

4.1 Järjestelmä

Seuraavassa muutamia uhkaskenaarioita liittyen autonomisen aluksen sisäisiin järjestelmiin, toimintaympäristön rajapintoihin, vihamielisiin tahoihin ja aluksen antureiden toimintaan. Skenaarioiden vastapainoksi pohditaan niiden uhkilta suojautumista.

4.1.1 Sisäisiä ja infran rajapintaan liittyviä uhkia

Uhkaskenaario 4.1 Aluksen huollon tai hyötykuormaan liittyvän toiminnon kautta joku ulkopuolinen taho onnistuu vaihtamaan/lisäämään jonkin aluksen järjestelmiin liittyvän/liitetyn fyysisen komponentin. Jos uusi komponentti korvaa olemassa olevan, sen avulla voidaan pyrkiä vääristelemään muulle järjestelmälle komponentin oletettua toimintaa (tilannekuva). Sen tarkoitus voi myös liittyä järjestelmän toiminnan vakoiluun (tietojen välittämiseen ulos). Jos uusi komponetti ei korvaa mitään olemassa olevaa, sen tarkoitus voi olla havainnoida muun järjestelmän toimintaa tai pyrkiä estämään sitä.

Uhkaskenaario 4.2 Aluksen valmistamisen yhteydessä on alukseen integroitu komponentti (tai sen osa), jonka koko toiminnallisuus ei olekaan ollut tiedossa. Esimerkiksi, komponenttia voidaan mahdollisesti komentaa ulkopäin vaikuttamaan komponentin kautta järjestelmään epätoivotulla (hyökkäävällä) tavalla.

Uhkaskenaario 4.3 Järjestelmän toteutukseen on jäänyt virhe, ns. tietoturva-aukko (esim. puskurin ylivuodon mahdollistama virhe), jonka kautta järjestelmään on päässyt haittaohjelma (tietokonevirus, troijanhevonen, looginen pommi, kiristysohjelma, . . .).

Uhkaskenaario 4.4 Järjestelmän autentikointiin / oikeuksien määrittelyyn tai ylipäänsä (etä)operoinnin rajapintaan on jäänyt looginen toteutusvirhe, joka mahdollistaa autonomisen aluksen jonkin toiminnon käytön oikeudettomasti, ilman minkäänlaista suojausten murtamista.

Uhkaskenaario 4.5 Autonomisen aluksen etäkäyttö edellyttää, että etäkäyttävä taho voidaan aluksessa todentaa lailliseksi käyttäväksi tahoksi. Toisaalta autonomisen aluksen pitää myös itse todentaa itsensä 'lailliseksi' tahoksi etäkäyttötahojen ja käyttämänsä väylä- ja satamainfrastruktuurin suuntaan. Kaikki tämä edellyttää, että autonomisessa aluksessa on tietoturvaprotokolliin liittyviä salaisia tunnisteita (esim. epäsymmetrisen kryptografian salaisia avaimia). Uhkana on, että joku murtautuu fyysisesti alukseen varastaen salaiset tunnisteet.

Uhkaskenaario 4.6 Skenaarion 4.5 variaationa tilanne, jossa laillinen taho hankkii itselleen autonomisen aluksen ja suorittaa sille takaisinmallinnusta (engl. reverse engineering) saadakseen haltuun yleiskäyttöisiä salaisia tunnisteita tai selvittääkseen salaiseksi tarkoitetut tietoturvamenettelyt (tai autonomisuuden toteutukseen liittyvät IPR:t) – ajatuksena on, että tällainen laillinen mutta vihamielinen taho voisi soveltaa hankitun tiedon perusteella toimia muita saman valmistajan autonomisia aluksia vastaan.

Uhkaskenaario 4.7 Jokin vihamielinen taho on hankkinut hallintaansa autonomisen aluksen tarvitsemaa liikennöinti- tai satamainfrastruktuuria. Operoidessaan autonominen alus on vuorovaikutuksessa vihamielisen tahon hallussa olevan infran kanssa. Uhkana on, että taho väärinkäyttää hallussaan olevan infran kykyä olla vuorovaikutuksessa autonomisen aluksen kanssa.

Uhkaskenaario 4.8 Käyttöön otettu järjestelmä rakentuu muiden valmistamiin tunnettuihin komponentteihin. Jostakin sellaisesta komponentista paljastuu kauan käyttöönoton jälkeen tietoturvahaavoittuvuus.

Suojautuminen

Järjestelmien kyberturvariskeihin voidaan soveltaa yleistä suojautusmenettelyä pohjautuen fyysisten järjestelmien moninkertaistamiseen – tämä on toimivaa järjestelmän toiminnan takaamisen osalta, oikeastaan paljolti riippumatta uhkaskenaarion yksityiskohdista. Tällaista toimintaa katsotaan tämän osion lopuksi, mutta ensin esitetään uhkaskenaariokohtaisia pohdintoja.

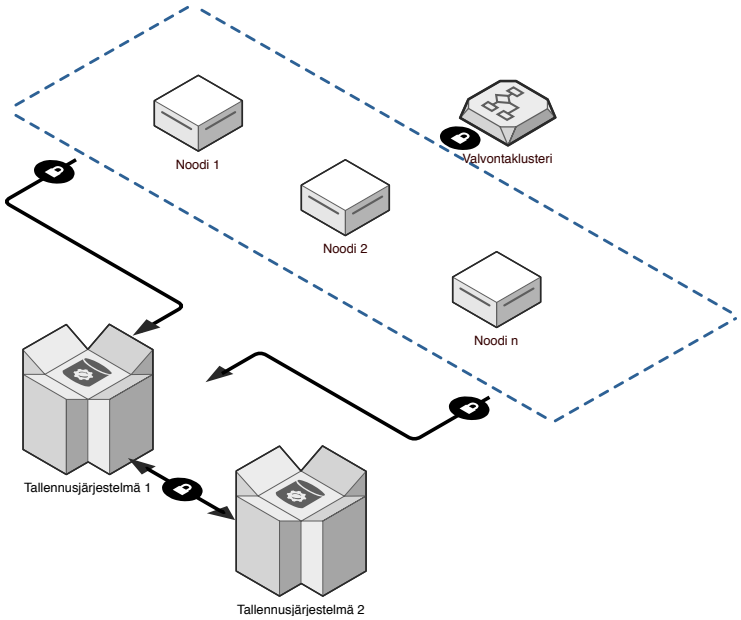
Skenaarioissa 4.1 ja 4.2 lähtökohta suojautumiselle on estäminen operointi- ja valmistusvaiheissa. Valmistusvaiheessa tulee luottaa vain luotettuihin komponenttien toimittajiin. Ja ensimmäistä skenaariota tulee torjua huolehtimalla fyysisestä suojasta. Kummankin kohdalla voidaan myös soveltaa komponenttien ja osajärjestelmien toimintaympäristön koventamista ja valvontaa (IDS). Skenaarioiden 4.3, 4.8 ja 4.4 osalta edellisiä voidaan myös soveltaa, joskin virheitä pitää pyrkiä poistamaan järjestelmästä. Suositeltavaa on tehdä päivitystoimia, kun autonominen alus on satamassa.

Skenaario 4.5 on helppo unohtaa riskikartoituksesta, koska järjestelmien autentikointiin liittyvät tiedot eivät ole primäärisiin järjestelmiin liittyviä tietoja – uhka edustaa suojausmekanismien suojaamista. Perusmenetelmiä ovat fyysiset suojat, kryptografian soveltaminen ja luotetut suoritusympäristöt.

Skenaario 4.6 edustaa lopulta varsin tavallista (mutta kiellettyä) toimintaa, jossa pyritään takaisinmallintamisen keinoin selvittämään, miten jokin (kilpailijan) järjestelmä toimii. Tällöin suojauskeinojen menetelmät helposti paljastuvat, mutta juuri siitä syystä pitää soveltaa ns. Kerckhoffsin periaatetta, jonka mukaan menetelmän paljastaminen ei saa paljastaa itse salausta. Eli, vaikka eri autonomisissa aluksissa sovellettaisiinkin samoja menettelyitä, niiden paljastumisen ei tule sellaisenaan vaarantaa muita analogisesti tehtyjä aluksia, vaan suojaamisen tulee perustua myös menetelmän lisäksi järjestelmäkohtaisiin salasanoihin. Itse suojaamiseen liittyvää menetelmää (tai ylipäänsä IPR:ää) voidaan myös pyrkiä suojaamaan luotettuja suoritusympäristöjä soveltamalla.

Vihamielisen infran skenaario 4.7 on myös varsin todellinen. Siltä suojautuminen lähtee siitä, että mihinkään ei saisi ”sokeasti” luottaa. Kun autonominen alus kommunikoi ulkopuolisen infran kanssa, tulee aina mahdollisuuksien mukaan verifioida vuorovaikutuksen sisältö.

Varautuminen tietoturvariskeihin järjestelmän fyysisen toteutuksen tasolla on varsin samanlaista kuin muut, yleiset ohjelmistoturvallisuuden fyysiset varautumiskeinot. Niillä pyritään en-



Kuva 4.2: Esimerkki vikasietoisesta klusterista, jossa quorum-palvelin ja kahdennettu tallennusjärjestelmä.

sisijaisesti takaamaan tiedon ja toiminnallisuuden saatavuus ja saavutettavuus.

Laitteille ja järjestelmille luodaan toimivuutta tarkkaileva monitorointijärjestelmä. Monitorointi toteutetaan varsinaisesta tuotantoverkosta erilliseen monitorointiverkkoon ja sen yhteydet ulkomaailmaan ja tuotantoverkkkoon ovat tarkkaan rajatut hyökkäyspinnan minimoimiseksi. Monitorointi ei ole virheenkorjausjärjestelmä, vaan sillä voidaan vain havaita viat: komponenttien vikaantumisen varalle järjestelmillä tulee olla toiminnoiltaan samanlainen tai vastaava varajärjestelmä. Tällöin puhutaan ns. *korkean käytettävyyden* järjestelmistä (high availability, HA) tai *klusteroinnista*. Klusteroinnissa itse järjestelmä ja sen tallennus- ja tiedonsiirtojärjestelmät turvataan vähintään yhdellä varajärjestelmällä. Klusterin järjestelmistä tyypillisesti yksi on aktiivinen ja loput passiivisia. Klusterissa on siis useita tietokoneita tai järjestelmiä, joiden toiminnallisuutta tarkkailee niistä erillinen ns. *quorum*-komponentti. Tarkkailijakomponentti määrittää monitorointitiedon pohjalta kulloinkin klusterissa aktiivisena toimivat osat. Klusterin aktiivisen osan vaihto eli ns. yliheitto voidaan tehdä automaattisesti tai manuaalisesti. Ihminen on kuitenkin koneeseen verrattuna hidas päätöksentekijä ja oikein määritelty automatiikka lyhentää varajärjestelmän käyttöönottoa ja siten häiriötilanteen kestoa yleensä merkittävästi, tyypillisesti useista minuuteista muutamiin sekunteihin. Esimerkki klusteroidusta järjestelmästä on esitetty kuvassa 4.2.

Varajärjestelmä voi olla myös ei-klusteroitu tai järjestelmiä voi olla käytössä useampia samanaikaisesti. Varajärjestelmä voi olla tekniseltä ratkaisultaan erilainen kuin varsinaiset toiminnalliset järjestelmät, jolloin toiminta häiriötilanteessa pyritään takaamaan järjestelmien *teknisellä diversiteetillä*. Äärimmäinen esimerkki kriisitilanteeseen varautumisesta useilla järjestelmillä löytyy NASAn avaruussukkulasta: avaruuslennon kriittisissä vaiheissa navigoinnista huolehti

neljä identtistä tietokonetta, joita varmistamassa oli vielä viides, eri tavalla toteutettu laite [16]. Varajärjestelmien ja -järjestelyjen määrä suhteutetaan aina kustannuksiin, järjestelmän tärkeyteen, häiriötilanteen todennäköisyyteen ja toimintahäiriön seuraamusten laatuun.

Tallennusjärjestelmiin taltioitu tieto suojataan tuhoutumiselta käyttämällä vikasietoisia tallennusjärjestelmiä, kuten Redundant Array of Independent Disks (RAID). Yksittäinen vikasietoinenkaan tallennusjärjestelmä ei yleensä ole yksin riittävä, vaan myös tallennuskapasiteetin RAID-järjestelmät klusteroidaan. Tämän lisäksi tiedosta otetaan varmuuskopioita, jotka säilytetään järjestelmistä erillään. Tallennusjärjestelmiä ja varmuuskopioita koskevat yleiset tiedonsiirron ja tiedon säilyttämisen tietoturvaohjeet. Fyysinen varautuminen on siis myös tältä osin olennainen osa organisaation laajempaa riskienhallintaa.

4.1.2 Anturidata ja julkiset tietolähteet

Uhkaskenaario 4.9 Aluksen ympäristöä havainnoivien antureiden toimintaa häiritään tai se estetään kokonaan.

Uhkaskenaario 4.10 Aluksen ympäristöä havainnoiviin antureihin syötetään väärää tietoa.

Uhkaskenaario 4.11 Aluksen ympäristöä muokataan siten, että aluksen eteneminen tai muu toiminta estyy.

Uhkaskenaario 4.12 Aluksen tilaa havainnoivia antureita häiritään tai niiden toiminta estetään.

Suojautuminen

Autonominen alus ylläpitää tilannetietoa ympäristöstään havainnoimalla sitä erilaisilla antureilla. Näitä ovat erilaiset optiset sensorit (näkyvä valo, infrapuna), tutka, kaikuluotain, LIDAR, radioaallot tai esimerkiksi värinäanturit (”konekuulo”). Tietoja ympäröivästä liikenteestä sekä esimerkiksi jäätilanteesta voidaan lisäksi lähettää alukselle datalinkin kautta.

Anturidatan käsittelyssä olennaista on epäjohdonmukaisuuksien ja poikkeustilanteiden tunnistaminen. Jos anturidataan ei voida luottaa, aluksella on toimintasuunnitelma tällaisen tilanteen varalta, erityisesti tilanteissa, joissa operoijalla ei ole yhteyttä alukseen. Toimintoja voivat olla esimerkiksi paikallaan pysyminen, paluu sijaintiin, jossa viimeksi on saatu varmistettu datayhteys, tai jatkaminen reittisuunnitelman mukaisesti alennetulla nopeudella. Anturidatan luotettavuuden varmistamiseksi myös antureilla tulee olla varajärjestelmät (kts. luku 4.1.1 edellä). Aluksella saattaa olla käytettävissä etähavainnointiin drooneja tai ilmaa kevyempiä aluksia, jotka sääolojen salliessa voivat nousta esim. sumupeitteen tai saarten yläpuolelle. Ne voivat myös toimia radiolinkkeinä horisontin tai muiden esteiden yli.

Aluksen tilaa, kuten esimerkiksi lämpötilaa, kallistusta, syväystä tai mahdollisia vuotoja havainnoivien antureiden häiritseminen tai hämääminen voi olla myös mahdollista. Tämä tulee huomioida varauduttaessa aluksella sattuviin hätätilanteisiin, kuten tulipaloihin tai vakaviin vuotoihin. Myös sijaintitiedon tai automaattisen identifiointijärjestelmän kommunikaatiota voidaan yrittää häiritä tai niiden sisältöä manipuloida. Myös muuta julkisesta tietolähteestä tulevaa tietoa saatetaan pyrkiä peukaloimaan, tai sen käyttö estää kokonaan. Puuttuva tai virheellinen tieto lähestyvästä myrskyrintamasta, tsunamista tai jäävuoresta ei välttämättä ole Suomen kotimaisilla vesillä kovin ilmeinen riski, mutta autonomisen merenkulun kannalta merkittävä kokonaisuhka. Nämä poikkeustilanteet yhdistettynä luotettavan ympäristötiedon puuttumiseen sekä datalinkkien toiminnan estämiseen lienevätkin eräs vakavimmista autonomiseen merenkulkuun kohdistuvia suojautumisuhkista.

4.2 Tiedonsiirto

Seuraavassa muutamia uhkaskenaarioita liittyen tiedonsiirtokanaviin ja niiden väärinkäyttöön kyberturvallisuusmielessä. Skenaarioiden vastapainoksi esitetään pohditaan niiden uhkilta suojautumiseksi.

4.2.1 Palvelunesto

Seuraavassa esimerkinomaisesti kaksi erilaista skenaariota palvelunestohyökkäykseen liittyen.

Uhkaskenaario 4.13 Palvelunestohyökkäys tiedonsiirtokanaviin aluksen ulkopuolelta: Jokin ulkopuolinen taho häiritsee aluksen käyttämää radio-/satelliittiyhteyttä estäen sen toimimisen.

Uhkaskenaario 4.14 Palvelunestohyökkäys tiedonsiirtoon aluksen sisältä: Autonomisen aluksen sisällä on jokin toiminnaltaan vieras komponentti, joka voi tehdä etäältä ohjatusti palvelunestohyökkäyksen kohdistaen sen yksittäiseen/useisiin aluksen sisäisiin tiedonsiirtokanaviin. Vaikutus voi olla tilapäistä häirintää tai suorastaan fyysistä järjestelmän tuhoamista.

Suojautuminen

Merenkulussa käytetään useita eri yhteysmuotoja ja useimmiten niitä on käytettävissä useampia yhtäaikaisesti – se on myös keskeisin keino skenaariota 4.13 vastaan. Aluksen liikkuessa merialueella aluksen valvontaan käytettävää yhteystapaa saatetaan vaihtaa sen ollessa käytössä. Etäohjaukseen ja valvontaan avomerellä saattaa olla käytettävissä vain satelliittiyhteys (ja epäsuorat yhteydet muiden alusten kautta), mutta rannikkoa lähestyttäessä sitä voidaan täydentää muilla yhteystavoilla tai siirtää yhteys käyttämään kokonaan toista tiedonsiirtotapaa. Yhteyden muodostaminen edellyttää osapuolten molemminpuolista tunnistamista ja todentamista eli identiteetin varmistamista. Yhteyden aikana datan sekä mahdollisuuksien mukaan myös yhteyskanavan muut protokollakerrokset on salattava.

Langaton tiedonsiirto on aina käytännössä julkista tiettyyn pisteeseen asti: signaalin olemassaolo voidaan lähes poikkeuksetta todeta ja sen lähetyspiste paikantaa. Autonomisen aluksen kontrollointiin ja valvontaan käytettyjä tiedonsiirtoprotokollia voidaan siis kuunnella ja häiritä, ja niitä voidaan yrittää myös kaapata: häiritsemällä jompaa kumpaa tiedonsiirron loppupistettä tai välisolmua paikallisesti voidaan estää komentojen tai valvontatiedon perilletulo. Tämän estäminen tapahtuu pääasiassa infrastruktuurin tasolla. Autonomisen aluksen sisäinen toimintalogiikka voi tällaisessa tapauksessa toimia itsenäisesti ja joko antureihin perustuvaan tilannetietoon (Situational Awareness) nojaten jatkaa matkaa ennakkosuunnitelman mukaisesti tai jäädä paikalleen (Dynamic Positioning) kunnes yhteys saadaan palautettua.

Skenaarion 4.14 torjuminen lähtee liikkeelle aluksen valmistusprosessin aikaisista toimista: pitää käyttää vain luotettuja komponentteja ja niiden toimittajia. Toisaalta, vieras komponetti voidaan mahdollisesti asentaa valmistamisen jälkeen – joten fyysiseen suojaukseen ja fyysisten tilojen ”käytön” valvontaan pitää kiinnittää huomiota myös täysin autonomisella aluksella. Osin tässä vastuu kohdistuu operointiin ja alihankintaketjujen luotetettavuuteen operoinnin yhteydessä. Valmistusvaiheessa voidaan myös pyrkiä suojatumaan tätä uhkaskenaariota vastaan pitämällä kaikkia osajärjestelmiä ja niiden komponentteja lähtökohtaisesti ei-luotettuina. Tällöin mahdollisen vieraan komponentin aiheuttamaa vaikutusta voidaan pyrkiä rajaamaan järjestelmien koventamis- ja monitorointikeinoin.

4.2.2 Protokolliin liittyviä uhkia

Seuraavassa kaksi hieman erilaista protokolliin liittyvää uhkaskenaariota.

Uhkaskenaario 4.15 Osana autonomisen aluksen ulkoista kommunikointirajapintaa on protokolla, josta paljastuu tietoturvahaavoittuvuus (vrt. Heartbleed -haavoittuvuus taannoin).

Uhkaskenaario 4.16 Toistohyökkäys: Jokin taho pääsee nauhoittamaan autonomisen aluksen ulospäin tai aluksen sisällä suorittamaa salattua viestintää ja pyrkii saamaan aikaan vahingollisia salatun protokollan viestejä ilman kykyä purkaa salausta.

Suojautuminen

Skenaario 4.16 on vältettävissä käyttämällä tiedonsiirrossa krytografisia protokollia, jotka ovat resistenttejä toistohyökkäyksille. Käytännössä haasteena on, että jotkin esim. langattoman tietoliikenteen protokollat ovat osoittautuneet tässä mielessä ongelmallisiksi, vrt. äskettäinen WPA2-protokollan haavoittuvuus. Tämä on ongelmallista, sillä autonomisen aluksen rakentamisen yhteydessä joudutaan laajasti nojaamaan muiden toimittamiin komponentteihin ja ratkaisuihin. Autonomisen aluksen kehittämisen yhteydessä tämä riski tulee mallintaa ja mahdollisuuksien mukaan vaatia komponenteissa sovellettavan vain tässä mielessä ongelmattomaksi osoitettuja protokollia.

Skenaario 4.15 liittyy läheisesti edelliseen skenaarioon. Aina halutaan käyttää vain turvalliseksi osoitettuja menetelmiä ja niiden toteutuksia, mutta käytännössä valmistamisen jälkeen

paljastuu usein ongelmia. Käytössä olevien viestintämenetelmien moninaisuus – useita kanavia, useita protokollia ja useita erillisiä toteutuksia protokollista – on keskeisin keino nopeasti reagoida havaittuun ongelmaan: luovutaan protokollasta, jossa on todettu ongelma.

Yleisemmin paljastuneet haavoittuvuudet pitää pyrkiä korjaamaan. Mahdollisesti korjaaminen on mahdollista etänä – etäasentamalla uusi haavoittuvuuksista vapaa ohjelmisto. Ohjelmistojen etäpäivittäminen on nykyään laajasti käytössä, mutta se on menetelmänä myös hyökkäysrajapintaa laajentava ja siksi on paikallaan myös pohtia päivitysten tekemistä vain kun autonominen alus on satamassa.

4.2.3 Yhteyksien siirrot

Uhkaskenaario 4.17 Yhteystavan vaihdon yhteydessä ulkopuolinen taho kaappaa yhteyden ja esiintyy joko aluksena tai operaattorina.

Yhteyden siirrossa datalinkin yli välitettävän datavirran yhteystapaa vaihdetaan. Tämä voidaan tarvittaessa tehdä siten, ettei datalinkkiä käyttävä sovellus tai käyttäjä edes havaitse yhteystavan vaihtoa muuten kuin käytettävissä olevan kaistanleveyden tai kommunikaatioviiveen muutoksena.

Yhteyksien muodostamiseen kohdistuvat paitsi autentikointiin liittyvät uhkaskeaariot, myös mahdollisuudet palveluestohyökkäyksille. Autonomisessa merenkulussa fyysisen kanavan ja kommunikaatioon käytettävän infrastruktuurin suojaamisen merkitys korostuu.

4.3 Etäoperointikeskus ja käyttäjä operoijana

Seuraavassa esitetään joukko uhkaskenaarioita koskien autonomisen aluksen etäoperointikeskusta ja sitä kautta alusta. Skenaarioita esitetään erikseen myös koskien käyttäjän manipulointia. Taaskin skenaarioiden lista ei pyri olemaan täydellinen – pikemminkin sen tarkoitus on herättää lukija pohtimaan erilaisia skenaarioita ja niiden riskin suuruutta – ja pohdinnat suojautumisesta ovat suuntaa-antavia.

4.3.1 Operointikeskukseen kohdistuvia uhkia

Uhkaskenaario 4.18 Fyysinen turvallisuus operointikeskuksessa on heikko ja hyökkääjä pääsee operointiin käytettävälle koneelle, josta ei ole kirjauduttu ulos tai jonka salasana on helposti arvattava tai esillä lapulla laitteen vieressä. Hyökkääjä voi asentaa koneelle haittaohjelman.

Uhkaskenaario 4.19 Operaattori asentaa operointijärjestelmään siihen kuulumattoman ohjelman – kuten vaikkapa mediasoittimen tai tietokonepelin – joka voi sisältää haittakoodia tai tietoturvahaavoittuvuuden.

Uhkaskenaario 4.20 Haittaohjelma estää operointijärjestelmän toiminnan ja tekee sen saavuttamattomaksi esimerkiksi autonomisen aluksen näkökulmasta. Kyseessä on siis eräänlainen sisältäpäin toteutettu palvelunestohyökkäys.

Uhkaskenaario 4.21 Operaattori muuttaa tahattomasti ohjelmistojen ja käyttöjärjestelmän asetusten turvallisia oletusarvoja turvattomiksi heikentäen näin operointijärjestelmän tietoturvaa. Esimerkki tästä on käyttöjärjestelmän automaattisten päivitysten kytkeminen pois päältä.

Uhkaskenaario 4.22 Operointijärjestelmän käyttöjärjestelmän tai ohjelmistojen päivityksistä ei huolehdita asianmukaisesti, ja hyökkääjä löytää jostakin järjestelmän ulospäin näkyvästä rajapinnasta tietoturva-aukon.

Uhkaskenaario 4.23 Organisaation sisäisenä uhkana (engl. insider threat) voi olla pahantahtoinen operaattori, jolla on käyttöoikeudet operointijärjestelmään. Operaattori saattaa antaa autonomiselle alukselle vahingollisia komentoja, jotka voivat vaarantaa merenkulun turvallisuuden.

Uhkaskenaario 4.24 Hyökkääjä asentaa jollakin aiemmissa skenaarioissa (4.18 ja 4.19) kuvatulla menetelmällä tai käyttäjää manipuloimalla järjestelmään haittakoodia, joka vakoilee alukselta tulevia tilannetietoja sekä sille lähteviä komentoja ja välittää tiedot takaisin hyökkääjälle.

Uhkaskenaario 4.25 Hyökkääjä soluttaa jollakin aiemmissa skenaarioissa (4.18 ja 4.19) kuvatulla tavalla tai käyttäjää manipuloimalla järjestelmään haittakoodia, joka toimii epärehellisenä välittäjänä (engl. man-in-the-middle). Käytännössä tämä tarkoittaa sitä, että haittakoodi sijoittuu osana operointijärjestelmää operaattorin ja autonomisen aluksen väliin niin, että se voi valehdella operaattorille aluksen tilannekuvasta ja toisaalta antaa alukselle virheellisiä komentoja operaattorin nimissä. Kumpikaan viestinnän osapuoli ei voi tietää, että välissä onkin viestejä molempiin suuntiin muokkaava epärehellinen taho – haittaohjelma, joka toimii hyökkääjän ohjeiden mukaan. Skenaarion tilanne näkyy kuvassa 4.3. Koska tietoliikenne verkon yli on vahvasti salattua, on hyökkäys yleensä helpointa toteuttaa kommunikaation päätepisteessä eli esimerkiksi operointijärjestelmässä, jossa data on nähtävillä salaamattomassa muodossa. Kun operointikeskuksesta ohjataan useita aluksia samanaikaisesti, voi hyökkäys yleistyä näihin kaikkiin ellei eri alusten ohjausta ole asianmukaisesti eristetty.

Suojautuminen

Ennaltaehkäisyllä on näiltä skenaarioilta suojautumisessa suuri merkitys. Operaattorien kouluttaminen niin, että operointijärjestelmällä suoritetaan vain sen käyttötarkoitukseen liittyviä toimia

sekä järjestelmän säännöllinen päivittäminen ja operointikeskuksen pitäminen fyysisesti turvallisena auttavat pitkälle siinä, ettei haittaohjelmia pääse asentumaan järjestelmään operaattorin virheestä johtuen. Yleisesti ottaen tietoturvakulttuurin ylläpito ja tietoturvauhkien kuten käyttäjän manipuloinnin tuntemus on tärkeää.

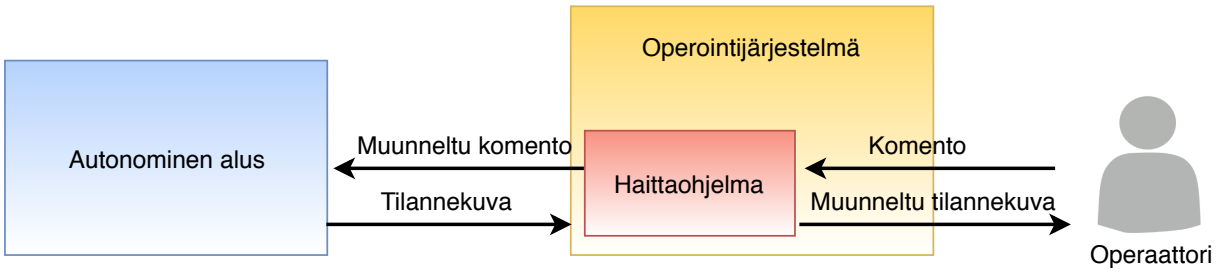
Kehittäjän ja ylläpitäjän kannalta katsottuna taas ennaltaehkäiseviä toimia ovat järjestelmän koventaminen ja siitä huolehtiminen, että käyttäjällä on mahdollisimman vähän mahdollisuuksia mitätöidä näitä järjestelmän kovennustoimia. Prosessien ajaminen toisistaan eristettyinä vaikkapa luotetussa suoritusympäristössä on suositeltavaa, ja vain tiettyjen sovelluksien suorituksen salliminen estää osaltaan haittaohjelmien toimintaa. Konekohtainen tunkeutumisen havaitsemisjärjestelmä voi myös auttaa havaitsemaan niitä epätavallisia tapahtumia, joita epärehellisenä välittäjänä toimiva haittaohjelma järjestelmässä aiheuttaa. Ajantasainen virustorjuntaohjelma on suositeltava, mutta tässä skenaariossa sen hyöty voi olla rajallinen varsinkin, mikäli edistynyt haittaohjelma on räätälöity kyseistä hyökkäystä varten ja on näin ollen ennalta tuntematon haittaohjelma.

Ylipäänsä peruslähtökohta koventamisessa on, että operointikeskuksessa etäoperointiin osallistuvilla koneilla ei tule olla mitään ylimääräisiä toimintoja. Esimerkiksi sähköpostiviestien lähettämiskyvystä tai yleisestä verkkoselailun suorittamisesta tulisi luopua. Työntekijälle ne voidaan toki sallia, mutta täysin erillään olevasta järjestelmästä. Samalla tapaa käyttäjältä pitää vaatia myös epäsuoraa eristäytymistä: samoja salasanoja (tunnisteita) ei operointia tekevä käyttäjä saa käyttää muissa ulkopuolisissa järjestelmissä (esim. sosiaalinen media).

Operaattorin on luonnollisesti hyvin vaikea havaita, että jotakin on vialla, kun epärehellinen välittäjä valehtelee, että kaikki on kunnossa ja näyttää operaattorille valheellisen, mutta uskottavan tilannekuvan alukselta. Mitään tavallisuudesta poikkeavaa on vaikea havaita varsinkin silloin, jos haittaohjelma aktivoituu toimimaan vasta tietyssä tilanteessa (esimerkiksi jättää ilmoittamatta jostakin kriittisestä häiriöstä aluksella tai aktivoituu silloin, kun aluksen törmäys on muuttunut todennäköisemmäksi ja operaattorin huomiota mahdollisesti tarvittaisiin). Tämän takia olisikin syytä toteuttaa operointi niin, että epätavalliset komennot varmistetaan vaihtoehtoista kanavaa pitkin. Kun komennolle pyydetään vahvistus sellaista vaihtoehtoista kanavaa pitkin, joka ei ole yhteydessä epärehellisen välittäjän saastuttamaan järjestelmään, selviää operaattorille, millaisia komentoja alukselle on todellisuudessa yritetty välittää. Näin voidaan päätellä, että operointijärjestelmässä on jotakin vialla ja vaihtaa varajärjestelmään, joka on puhdas haittaohjelmista. Yleensäkin autonomisen aluksen olisi suositeltavaa lähettää kriittisiä tilannekuvaan liittyviä tietoja useaa toisistaan riippumatonta kanavaa pitkin.

4.3.2 Käyttäjän manipulointi

Kaikissa kyberhyökkäyksissä ei tarvita haavoittuvuuksia, edistyneitä haittaohjelmia tai syvällistä teknistä osaamista. Tärkeä autonomisen aluksen etäoperointiin kohdistuvien uhkaskenaarioiden alijoukko ovatkin käyttäjän manipulointiin liittyvät skenaariot. *Käyttäjän manipuloinnin* (engl. social engineering) tarkoituksena on saada käyttäjä antamaan hyökkääjälle pääsy järjestelmään, sen toimintoihin ja dataan [5]. Hyökkääjä ohittaa näin järjestelmän turvajärjestelyt käyttäjän avustuksella kuvassa 4.4 kuvatulla tavalla. Usein hyökkääjä saa käyttäjän luottamaan itseensä esiintymällä jonakin luotettavana tahona, kuten esimerkiksi organisaation teknisen tuen edusta-



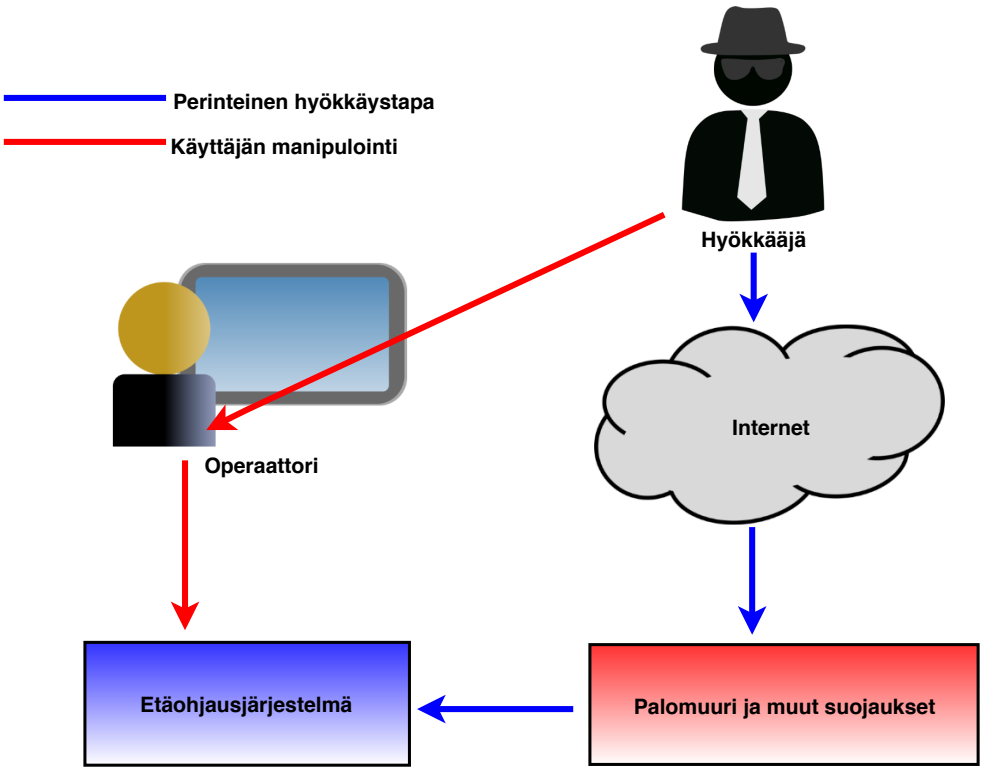
Kuva 4.3: Epärehellisen välittäjän uhkaskenaario.

jana. Hyökkääjä lähestyy useissa tapauksissa uhria sähköpostitse tai vaikkapa sosiaalisen median kautta, mutta myös puhelinsoitto tai kasvokkain tapahtuva huijaus ovat hyvin mahdollisia. Onnistuneen käyttäjän manipuloinnin ehkäisemiseksi autonomisen aluksen operointiin osallistuvien henkilöiden on tärkeää tuntea yleisimmät tavat, joilla käyttäjiä pyritään harhauttamaan.

Uhkaskenaario 4.26 *Haitallisen ohjelman suorittaminen*. Käyttäjän manipuloinnin tavoitteena on usein saada käyttäjä suorittamaan järjestelmässä haittaohjelma. Käyttäjä voi erehtyä suorittamaan esimerkiksi sosiaalisessa mediassa jaettua, sähköpostin liitteenä tullutta tai vaikkapa parkkipaikalta löytyneellä muistitikulla olevaa haittakoodia. Tämä voi johtaa monenlaiseen järjestelmän väärinkäyttöön sekä luottamuksen, eheyden ja saatavuuden vaarantumiseen.

Uhkaskenaario 4.27 *Tietojenkalastelu* (engl. phishing). Tietojenkalastelu tai verkkourkinta on toimintaa, jolla hyökkääjä pyrkii saamaan käyttäjän luovuttamaan itselleen luottamuksellisia tietoja[1]. Autonomisten alusten operointijärjestelmän yhteydessä tällaista tietoa ovat lähinnä käyttäjätunnukset ja salasanat. Hyökkääjä lähestyy uhria mahdollisimman luottamusta herättävällä viestillä, usein sähköpostin, pikaviestimien tai sosiaalisen median kautta. Viestissä on useissa tapauksissa linkki tietojenkalastelusivulle, jossa kysytään käyttäjätunnusta ja salasanaa. Huijaussivu voi olla naamioitu tyypillisen sisäänkirjautumissivun näköiseksi, mutta tavallisesti selaimen osoiterivi paljastaa, ettei kyseessä ole aito sivu (esimerkiksi organisaation oma kirjautumissivu).

[1] https://www.viestintavirasto.fi/kyberturvallisuus/tietoturvanyt/2017/03/ttn201703091241.html



Kuva 4.4: Järjestelmän turvatoimien ohitus käyttäjää manipuloimalla autonomisen aluksen operointijärjestelmässä.

Uhkaskenaario 4.28 *Haittaohjelmanjakelualusta* (engl. exploit kit). Käyttäjä voidaan luottamusta herättävällä viestillä houkutella vierailemaan haitallisella verkkosivulla, jolle on asennettu vierailijan tietokoneen ja verkkoselaimen haavoittuvuuksia selvittävä haittaohjelmanjakelualusta. Haittakoodi murtautuu löytämiään haavoittuvuuksia käyttäen kohdejärjestelmään ja voi tämän jälkeen asentaa järjestelmään haittaohjelmia ja avata takaportteja. Tällaisesta jakelujalustan avulla tapahtuvasta haittaohjelmien leviämisestä käytetään englannin kielessä termiä *drive-by download*[2]. Pelkkä sivulla vierailu voi riittää tartuntaan, eikä käyttäjän tarvitse erehtyä tietoisesti suorittamaan mitään ohjelmaa.

[2]https://www.viestintavirasto.fi/kyberturvallisuus/tietoturvanyt/2015/02/ttn201502101246.html

Uhkaskenaario 4.29 *Etäyhteys* (engl. remote connection). Hyökkääjä voi muodostaa laitteeseen etäyhteyden haltuunsa saamiensa käyttäjätunnusten tai käyttäjän sosiaalisen manipuloinnin tuloksena luoman takaportin avulla. Takaportti voi syntyä esimerkiksi silloin, jos käyttäjä hyökkääjän houkuttelemana sallii ulkopuoliset yhteydet koneelle (esimerkiksi avaamalla verkkoportteja tai etähallintaohjelman) mahdollistaen näin tietoturvajärjestelyiden ohittamisen. Etäyhteyden avulla hyökkääjä voi toimia huijauksen kohteeksi joutuneen operaattorin oikeuksilla ja mahdollisesti etsiä järjestelmästä haavoittuvuuksia, luoda takaovia ja asentaa haittaohjelmia.

Uhkaskenaario 4.30 *Kohdistettu hyökkäys* (Advanced Persistent Threat, APT). Kohdistettu hyökkäys on nimessä mukaisesti rajattu vain tiettyyn organisaatioon ja siinä hyökkääjä käyttää hyväkseen organisaatiosta saamiaan tietoja. Esimerkki kohdistetusta hyökkäyksestä on haitallisen tiedoston lähettäminen työntekijälle sähköpostiviestin liitteenä käyttäen lähettäjänä jonkun toisen saman organisaation työntekijän nimeä ja luottamusta herättävää viestin otsikkoa. Kohdistetuille hyökkäyksille on yleensä ominaista pitkäkestoisuus ja organisaation kriittisen tiedon vakoilu[3], mutta autonomisten alusten yhteydessä hyökkäys voi yhtä hyvin tähdätä esimerkiksi aluksen operoinnin häirintään.

Suojautuminen

Teknisessä mielessä monikerroksinen turvaratkaisujen käyttö suojaa käyttäjän manipuloinnin vaikutuksilta ainakin joiltain osin. Vaikka hyökkääjä olisi esimerkiksi saannut haltuunsa operaattorin salasanan, ei hän pääse helposti asentamaan järjestelmään haittaohjelmaa elleivät operaattorin käyttöoikeudet riitä tähän eikä häiritsemään etäoperointijärjestelmässä ajettavien prosessien suoritusta, jos ne on eristetty toisistaan esimerkiksi luotetulla suoritusympäristöllä. Vastaavasti jos operointijärjestelmä käyttää monivaiheista todennusta – vaikkapa salasanaa ja avainkorttia – ei hyökkääjälle riitä pelkkä salasana. Tapahtumalokit ja tunkeutumisen havaitsemisjärjestelmä auttavat huomaamaan epätavalliset tapahtumat järjestelmässä.

Käyttäjän manipuloinnin uhkaskenaarioiden toteutumisen mahdollisuutta voidaan kuitenkin tehokkaimmin pienentää lisäämällä käyttäjän tietoisuutta näistä uhista ja niiden mahdollisista seurauksista. Operaattorit tulisikin velvoittaa osallistumaan tietoturvakoulutukseen osana autonomisen aluksen operointiin perehdyttämistä ja tämän jälkeenkin säännöllisin väliajoin. Tietoisuuden lisääminen käyttäjän manipulointiin liittyvistä skenaarioista ja niiden vastatoimenpiteistä on tutkitusti ollut huomattavan tehokasta hyökkäyksiä vastaan [3]. On siis ensiarvoisen tärkeää, että operaattorit koulutetaan tunnistamaan käyttäjän manipuloinnin uhat ja olemaan vastaamatta tällaisiin lähestymisyrityksiin. Organisaatioon on luotava toimiva tietoturvakulttuuri sekä selkeät toimintaohjeet käyttäjän manipulointiyritysten varalle.

Edellisessä kohdassa esiin nostetut järjestelmien ja toimintojen eriyttämis- ja koventamistoimenpiteet ovat myös tämän kohdan uhkien torjunnassa keskeisessä roolissa.

[3] https://www.viestintavirasto.fi/kyberturvallisuus/tietoturvanyt/2015/09/ttn201509170939.html

Kirjallisuutta

[1] Carlisle Adams and Steve Lloyd. *Understanding PKI: Concepts, Standards, and Deployment Considerations*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2002.

[2] Kent Beck. *Extreme Programming Explained: Embrace Change*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2000.

[3] Jan-Willem H. Bullée, Lorena Montoya, Wolter Pieters, Marianne Junger, and Pieter H. Hartel. The persuasion and security awareness experiment: reducing the success of social engineering attacks. *Journal of Experimental Criminology*, 11(1):97–115, 2015.

[4] Roberto Di Pietro and Mancini Luigi. *Intrusion Detection Systems*. Springer, 2008.

[5] Christopher Hadnagy. *Social Engineering: The Art of Human Hacking*. John Wiley & Sons, 2011.

[6] Michael Howard and Steve Lipner. *The Security Development Lifecycle*. Microsoft Press, Redmond, WA, USA, 2006.

[7] Marko Höyhtyä, Jyrki Huusko, Markku Kiviranta1 Kenneth Solberg, and Juha Rokka. Connectivity for autonomous ships: Architecture, use cases, and research challenges. *sensors*, 12:0–1, 2017.

[8] ISO/IEC standard 15408-1:2009. *Information technology – Security techniques – Evaluation criteria for IT security*. ISO/IEC, 2009.

[9] ISO/IEC standard 21827. *Information Technology – Security Techniques – Systems Security Engineering – Capability Maturity Model (SSE-CMM)*. ISO/IEC, 2008.

[10] Liikenne ja viestintäministeriö. Sähköisen viestinnän salaus- ja suojausmenetelmät, 2018.

[11] Brian Komar, Ronald Beekelaar, and Wettern Joern. *Firewalls For Dummies*. John Wiley & Sons, 2003.

[12] P. Larsen, A. Homescu, S. Brunthaler, and M. Franz. Sok: Automated software diversity. In *2014 IEEE Symposium on Security and Privacy*, pages 276–291, 2014.

[13] M Laurinen. Remote and autonomous ships: The next steps. *Advanced Autonomous Waterborne Application Partnership, Buckingham Gate, London*, 2016.

[14] Wenby Mao. *Modern Cryptography: Theory and Practice*. Prentice Hall, 2008.

[15] Nancy Mead. System quality requirements engineering, 2012.

[16] NASA. Computers in spaceflight: The nasa experience – computer synchronization and redundancy management, 2018.

[17] United Nations. Review of maritime transport, 2017. Sales No. E.17.II.D.10.

[18] OWASP. OWASP Secure Software Development Lifecycle Project, 2017.

[19] Raja Parasuraman, Thomas B Sheridan, and Christopher D Wickens. A model for types and levels of human interaction with automation. *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and Humans*, 30(3):286–297, 2000.

[20] Mary Poppendieck and Tom Poppendieck. *Lean Software Development: An Agile Toolkit: An Agile Toolkit*. Addison-Wesley, 2003.

[21] Thomas Porathe, Hans-Christoph Burmeister, and Ørnulf Jan Rødseth. Maritime unmanned navigation through intelligence in networks: The munin project. In *12th International Conference on Computer and IT Applications in the Maritime Industries, COMPIT'13, Cortona, 15-17 April 2013*, pages 177–183, 2013.

[22] Dilip K Prasad, C Krishna Prasath, Deepu Rajan, Lily Rachmawati, E Rajabaly, and Chai Quek. Challenges in video based object detection in maritime scenario using computer vision. *arXiv preprint arXiv:1608.01079*, 2016.

[23] Jean-Paul Rodrigue, Claude Comtois, and Brian Slack. *The geography of transport systems*. Routledge, 2016.

[24] Ken Schwaber and Mike Beedle. *Agile Software Development with Scrum*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2001.

[25] Ken Schwaber and Jeff Sutherland. The scrum guide (2017). *http://www.scrumguides.org/index.html*, page 19, 2017.

[26] Several authors, Jokioinen, E (ed.). Remote and autonomous ships-the next steps. Technical report, Rolls-Royce, AAWA Project, 2016.

[27] IMO SOLAS. International convention for the safety of life at sea. london. *International Maritime Organization*, 2003.

[28] Lance Spitzner. *Honeypots: Tracking Hackers*. Addison Wesley, 2002.

[29] William Stallings. *Network Security Essentials: Applications and Standards*. Pearson, 2016.

[30] Michael J. Stewart. *Network Security, Firewalls and VPNs*. Jones and Bartlett Publishers, 2014.

[31] Suomen Automaatioseura. Teollisuusautomaation tietoturva: verkottumisen riskit ja niiden hallinta, 2010.

[32] VAHTI. Lokiohje, 2009.

[33] VAHTI. Päätelaitteiden tietoturva, 2013.

[34] VAHTI. Ohje salauskäytännöistä, 2015.

[35] VAHTI. Vahti-ohje, 2015.

[36] J. F. Van Niekerk and R. Von Solms. Information security culture: A management perspective. *Comput. Secur.*, 29(4):476–486, June 2010.

[37] Viestintävirasto. Salasanalla on väliä, 2014.

[38] Michael Zhivich and Robert K Cunningham. The real cost of software errors. *IEEE Security & Privacy*, 7(2), 2009.

YRKESHÖGSKOLAN
NOVIA

Yrkeshögskolan Novia har ca 3500 studerande och personalstyrkan uppgår till ca 390 personer. Novia är den största svenskspråkiga yrkeshögskolan i Finland som har examensinriktad ungdoms- och vuxenutbildning, utbildning som leder till högre yrkeshögskoleexamen samt fortbildning och specialiseringsutbildning. Novia har utbildningsverksamhet i Vasa, Jakobstad, Raseborg och Åbo.

Yrkeshögskolan Novia är en internationell yrkeshögskola, via samarbetsavtal utomlands och internationalisering på hemmaplan. Novias styrka ligger i närvaron och nätverket i hela Svenskfinland.

Novia representerar med sitt breda utbildningsutbud de flesta samhällssektorer. Det är få organisationer som kan uppvisa en sådan kompetensmässig och geografisk täckning. Högklassiga och moderna utbildningsprogram ger studerande en bra plattform för sina framtida yrkeskarriärer.

Yrkeshögskolan Novia
Wolffskavägen 33, 65100 Vasa, Finland
Tfn +358 (0)6 328 5000 (växel),
www.novia.fi

Ansökningsbyrån
PB 6, 65201 Vasa, Finland
Tfn +358 (0)6 328 5555
ansokningsbyran@novia.fi

Yrkeshögskolan Novia upprätthåller en publikations- och produktionsserie för att sprida information och kunskap om verksamheten såväl regionalt, nationellt som internationellt.
Publikations- och produktionsserien är indelad i fem kategorier:

R - Rapporter • P - Produktioner • A - Artiklar • L - Läromedel • S - Studerandes arbete

Läs våra senaste publikationer på www.novia.fi/FoU/publikation-och-produktion

ISSN 1799-4179
ISBN 978-952-7048-51-1 (online)