Roope Tams

SCADA-JÄRJESTELMÄÄN KOHDISTUVAT KYBERHYÖKKÄYKSET JA NIILTÄ SUOJAUTUMINEN

JYVÄSKYLÄN YLIOPISTO
INFORMAATIOTEKNOLOGIAN TIEDEKUNTA
2020

TIIVISTELMÄ

Tams, Roope
SCADA-järjestelmään kohdistuvat kyberhyökkäykset ja niiltä suojautuminen
Jyväskylä: Jyväskylän yliopisto, 2020, 33 s.
Tietojärjestelmätiede, kandidaatintutkielma
Ohjaaja: Riekkinen, Janne

Tämä kandidaatintutkielma suoritettiin kirjallisuuskatsauksena. Tarkoituksena oli luoda yleinen katsaus SCADA-järjestelmien kyberturvallisuuteen. SCADA-järjestelmä on teollinen hallintajärjestelmä, joka vastaa automaatiosta laajalla maantieteellisellä alueella. Ne vastaavat myös kriittisen infrastruktuurin toiminnasta, kuten sähkönsiirrosta. Laajat ongelmat sähkönsiirrossa tulisivat yhteiskunnalle erittäin kalliiksi, joten SCADA-järjestelmien kyberturvallisuudesta on pidettävä huolta.
Teolliset hallintajärjestelmät ovat perinteisiä kuluttajajärjestelmiä pitkäikäisempiä, joten niiden kyberturvallisuudesta huolehtiminen on vaikeampaa, sillä teknologia kehittyy jatkuvasti. Järjestelmien kyberturvallisuutta pidetäänkin yllä esimerkiksi hyökkäyspuilla. Hyökkäyspuiden avulla voidaan arvioida hyökkääjän tavoitteita ja sitä kautta hyökkäystapoja, josta suoritettavan arvioinnin perusteella järjestelmääkin voidaan kehittää.
Tunnetuimpana SCADA-hyökkäyksenä pidetään usein Stuxnetiä. Se oli hyökkäys, joka kohdistui iranilaiseen uraanin rikastamoon. Uraanin rikastamo oli internetistä eristetty kokonaisuus, mutta siihen saatiin tartutettua haittaohjelma saastuneella USB-muistitikulla. Haittaohjelma ei suinkaan pysynyt piilossa, vaan se oli ohjelmoitu hajottamaan uraanin rikastuksessa käytettäviä sentrifugeja.

Asiasanat: SCADA-järjestelmä, kyberturvallisuus, kriittinen infrastruktuuri

ABSTRACT

Tams, Roope
SCADA-systems cyberthreats and protecting SCADA-system
Jyväskylä: University of Jyväskylä, 2020, 33 pp.
Major subject, Information Systems, Bachelors Thesis
Supervisor: Riekkinen, Janne

Goal of this bachelors thesis was to get overall look in SCADA-system cybersecurity field and get basic knowledge of SCADA-systems and cyberattacks which create threat to SCADA-systems. This study was made as a literature review. Name SCADA comes from words Supervisory Control And Data Acquisition. Its basic goal is to maintain automation in large geographical area, for example in electric grid.

Cybersecurity of SCADA-system is essential because it maintains critical infrastructure. SCADA-systems have very long life compared to basic consumer electronics. That makes it important to keep improving systems cybersecurity. Improving cyber security is not easy, one main reason is that system itself might be quite old, but technologies around SCADA have evolved and new kind of threats have emerged.

Stuxnet is usually being kept as the most famous SCADA attack. Attack was targeted to Iranian uranium plant. Stuxnets was programmed to break centrifuges in the plant and therefore slow down Irans nuclear program.

Keywords: SCADA system, cyber security, Critical Infrastructure

SISÄLLYS

1 Johdanto

SCADA-järjestelmä on teollinen hallintajärjestelmä (Industrial Control System, ICS). Sen tehtävä on kerätä dataa ja valvoa automaatiota alueella, joka voi olla jopa tuhansien kilometrien kokoinen, kuten sähköverkko. Se voi koostua useista tuulivoimaloista, vesivoimaloista, ydinvoimaloista ja perinteisistä voimaloista ja sitä johdetaan yhdestä keskitetystä valvomosta. (Sullivan, Luiijf & Colbert, 2016.)

SCADA-järjestelmät ovatkin älykkään sähköverkon ytimessä oleva alijärjestelmä. Electric Power Research Instituten (EPRI) mukaan yksi suurimmista haasteista koskien älykkäiden sähköverkkojen kehittämistä on järjestelmän kyberturvallisuus (Metke & Ekl, 2010). Sähkönsaanti on yhteiskunnan kannalta kriittistä ja tämänkaltaiset muutokset ovatkin lisänneet turvallisuusnäkökulmiin fyysisten vaatimusten lisäksi kyberturvallisuuden ulottuvuuden (Ericsson, 2010).

Tässä kandidaatintutkielmassa tutkitaan SCADA-järjestelmissä esiintyviä kyberhyökkäyksiä ja niihin varautumista. Kyberhyökkäyksellä tarkoitetaan tämän tutkielman puitteissa hyökkäystä, ”joka kohdistuu kybertoimintaympäristöön ja sen mahdollisesti ohjaamiin fyysisen maailman toimintoihin”, kuten ydinvoimalan ohjausjärjestelmään (Sanastokeskus TSK, 2020).

Tämä kandidaatintutkielma on toteutettu kirjallisuuskatsauksena. Aineistoa on haettu JYKDOK:sta, IEEE Xplore:sta ja Google Scholarista. Aineiston arvioinnissa on hyödynnetty julkaisufoorumia, viittausten määrää ja tutkijan nimeä, mikäli tutkija on julkaissut julkaisufoorumin johtavan tai korkeimman tason tutkimuksia. Pääasiassa lähteinä käytetyt tutkimukset ovat julkaisufoorumin johtavan ja korkeimman tason tutkimuksia. Poikkeuksena tutkielmasta löytyy myös joitakin perustason ja julkaisufoorumin arvostelemattomia tutkimuksia. Tutkimukset, joita ei löydy julkaisufoorumista, ovat pääosin konferenssijulkaisuja. Niitä on käytetty sen takia, että monet tietoturvaongelmista ja niihin liittyneistä ratkaisuista on esitelty vain konferensseissa, eikä niistä löydy lainkaan muita tutkimuksia. Hakusanoina on käytetty kaikkien lukujen nimiä englanninkielelle käännettynä ja mitä tahansa SCADA-järjestelmiin ja niiden turvallisuuteen liittyvää.

Tutkielman tutkimusongelmana on ottaa selvää, voiko kriittisen infrastruktuurin lamauttaa hyökkäämällä SCADA-järjestelmään kyberhyökkäyksellä. Huoltovarmuuskeskus (2005) on määritellyt kriittisen infrastruktuurin seuraavasti: ”Kriittinen infrastruktuuri käsittää ne rakenteet ja toiminnot, jotka ovat välttämättömiä yhteiskunnan jatkuvalle toiminnalle. Kriittiseen infrastruktuuriin (Critical Infrastructure, CI) kuuluu sekä fyysisiä laitoksia ja rakenteita että sähköisiä toimintoja ja palveluja.” (Huoltovarmuuskeskus, 2005). Tarkentavia tutkimuskysymyksiä ovat:

1. Minkälaisia kyberhyökkäyksiä SCADA-järjestelmiin voidaan kohdistaa?
2. Miten SCADA-järjestelmiin hyökätään?
3. Miten SCADA-järjestelmissä voidaan suojautua kyberhyökkäyksiä vastaan?
4. Minkälaisia vaikutuksia SCADA-järjestelmään kohdistuva hyökkäys aiheuttaa järjestelmän toiminnalle?
5. Millä tasolla SCADA-järjestelmien turvallisuus on tällä hetkellä?

Tämän tutkielman toisessa luvussa pureudutaan SCADA-järjestelmään. Siinä käydään läpi mikä SCADA-järjestelmä on, mitkä ovat järjestelmän ominaispiirteet ja minkälaisista osista järjestelmä muodostuu. Järjestelmän osista tutkielmassa esitellään vain tutkielman kannalta relevantit. Tutkielman kolmannessa luvussa esitellään SCADA-järjestelmiin kohdistuneita kyberhyökkäyksiä ja hyökkäystapoja, joilla järjestelmään voidaan hyökätä. Erityisesti toteutuneista hyökkäyksistä käydään läpi todennäköisesti tunnetuinta SCADA-hyökkäystä, eli Stuxnetiä. Neljännessä luvussa paneudutaan SCADA-järjestelmän puolustautumiseen kyberhyökkäyksiä vastaan esittelemällä aiheeseen liittyviä kyberturvallisuusstandardeja sekä riskienarviointimetodeja. Viimeisessä luvussa on yhteenveto tutkielmasta sekä pohdintaa SCADA-järjestelmän turvallisuudesta ja tulevista tutkimusaiheista.

Käytännössä SCADA-järjestelmä toimii siten, että operaattori valvoo prosessia käyttöliittymän (human machine interface, HMI) kautta ja antaa tarvittaessa käskyjä, jotka lähetetään fyysisille laitteille, jotka sijaitsevat jopa tuhansien kilomerien päässä valvontapaikasta. SCADA-järjestelmä ohjaa siis teollista automaatiota laajalla alueella ja sen toiminta on esimerkiksi sähkönsiirrolle elintärkeää. (Sullivan ym., 2016.)

Luonteenomaista SCADA-järjestelmille on niiden tarve luotettavuudelle, sillä niillä hallitaan fyysisiä laitteita ja järjestelmän kaatuminen voi aiheuttaa vakavia vahinkoja, tuotannolle, ympäristölle tai jopa ihmisille (Galloway & Hancke, 2013). Järjestelmien suojauksessa hyödynnetään erilaisia kyberturvallisuusstandardeja, jotka keskittyvät vahvasti vastatoimiin, kuten uhka-analyyseihin (Sommestad, Ericsson & Nordlander, 2010). Järjestelmien pitkäikäisyydestä johtuen SCADA-järjestelmät vaativat turvallisuuden evaluointia läpi niiden elinkaaren. Yksi tutkielmassa esitellyistä menetelmistä on hyökkäyspuu, jonka avulla voidaan arvioida järjestelmän heikoimpia osia erilaisia hyökkäyksiä vastaan (Byres, Franz & Miller, 2004).

Tutkielmasta käy ilmi, että hyökkäystapoja SCADA-järjestelmään on lukuisia. Hyökkäyksen onnistuminen riippuukin pitkälti järjestelmän suojauksesta. SCADA-järjestelmien kyberturvallisuuden tutkiminen ja analysointi onkin siten ensiarvoisen tärkeää, jotta kriittistä infrastruktuuria valvovia järjestelmiä voidaan tulevaisuudessa suojata paremmin.

2 SCADA-Järjestelmät

Tässä tutkielman luvussa kerron yleistä SCADA-järjestelmästä ja SCADA-järjestelmien toiminnasta, sekä tutkielman ymmärtämisen kannalta tärkeimmistä järjestelmän osista. SCADA-järjestelmällä ohjataan aina fyysisiä laitteita, jonka seurauksena järjestelmän palvelutaso on korkea. SCADA-järjestelmien täytyykin olla erittäin luotettavia, samalla tavoin kuin muidenkin teollisten hallintajärjestelmien, koska niillä hallitaan fyysisiä laitteita ja järjestelmän kaatuminen voi aiheuttaa vakavia vahinkoja tuotannolle, ympäristölle tai jopa ihmisille, kuten kävi esimerkiksi Fukushiman ydinvoimalaonnettumuudessa 2011. (Galloway & Hancke, 2013.)

SCADA-järjestelmä on Windows-ohjelmisto, joka mahdollistaa ihmisoperaattorin valvoa teollisia prosesseja ja varastoida, sekä analysoida arvoja (Langner, 2011). Käytännössä SCADA-järjestelmä toimii siten, että operaattori valvoo prosessia ja antaa tarvittaessa käskyjä, jotka lähetetään fyysisille laitteille, jotka sijaitsevat jopa tuhansien kilomerien päässä valvontapaikasta (Galloway & Hancke, 2013).

2.1 SCADA-järjestelmän tärkeimmät osat

SCADA-järjestelmä on teollinen hallintajärjestelmä (ICS, Industrial Control System), joka kerää dataa ja valvoo automaatiota alueella, joka voi olla jopa tuhansien kilometrien kokoinen, kuten sähköverkko, joka koostuu useista tuulivoimaloista, vesivoimaloista, ydinvoimaloista ja perinteisemmistä voimaloista ja jota johdetaan yhdestä, keskitetystä, valvomosta. (Sullivan, Luiijf & Colbert, 2016.) Seuraavaksi esittelen mistä osista se muodostuu.

2.1.1 PLC (Programmable Logic Controller)

PLC (Programmable Logic Controller) lukee syötteitä sensoreilta ja suorittaa ohjelmoituja ohjeita hyväksikäyttäen syötteitä ja käskyjä valvontaohjaimilta.

Niiden perusteella se luo tulosteen, jonka mukaan toimintaa muutetaan tai pidetään toiminta ennallaan. PLC:n toiminta on reaaliaikaista. Tyypillisesti PLC käyttää yhteydenpitoon esimerkiksi ethernet-yhteyttä (Sullivan, Luiijf & Colbert, 2016). PLC:tä voidaan käyttää myös RTU:n, eli remote terminal unitin, tilalla, ja ne ovatkin käyttömahdollisuuksiltaan RTU:ta monipuolisempia ja luotettavampia, sekä kehittyneempiä, joten niiden avulla on mahdollisuus tehdä tarkempia säätöjä laitteisiin. Myös ongelmanratkaisu sekä diagnostiikka on helpompaa kuin RTU:ssa (Bailey & Wright, 2003). PLC:n kehitys ja käyttöönotto oli ensimmäinen askel kohti toisiinsa kytkettyjä järjestelmiä, joita teolliset järjestelmät nykypäivänä ovat (Galloway & Hancke, 2013).

2.1.2 RTU (Remote Terminal Unit

RTU (Remote Terminal Unit) on elektroninen laite, joka on suunniteltu ankaraan toimintaympäristöön, eli sen tulee kestää lämpötilanvaihtelua, kuumuutta, tärinää ja elektromagneettista säteilyä. Kenttä RTU:t vastaanottavat syötteitä kenttäsensoreilta ja toimeenpanevat ohjelmoitua logiikkaa näiden syötteiden perusteella. Kenttä RTU:t toimivat yhdessä kenttälaitteiden kanssa. (Sullivan, Luiijf & Colbert, 2016.)

Asema RTU:t toimivat puolestaan myös etäisissä sijainneissa ja vastaanottavat dataa kenttä RTU:lta sekä käskyjä valvontaohjaimilta. Asema RTU luo niiden perusteella tulostearvoja, joilla hallitaan fyysisiä laitteita ja niiden kautta fyysisiä prosesseja. Ne molemmat voidaan myös yhdistää yhdeksi fyysiseksi RTU:ksi ja käytännössä ne toimivatkin fyysisen ja digitaalisen maailman välissä. (Sullivan, Luiijf & Colbert, 2016.)

RTU:n päätehtävänä on kerätä dataa ja siirtää data valvonta-asemalle, mutta RTU:t voivat myös kommunikoida keskenään. RTU on yleisesti ottaen siis erikoistunut PLC, joka kommnikoi MTU:n (Master Terminal Unit) kanssa. Silloin kun RTU:ta käytetään, kyseessä on yleensä tapahtumaperustainen prosessi, jolloin toimintaa muutetaan vain kun jokin muuttuu, jolloin vain muutokset raportoidaan MTU:lle. (Galloway & Hancke, 2013.)

RTU:t ovat yleensä käytössä kaukaisissa sijainneissa, joten sähkövirran syöttäminen niille on usein huolenaiheena, jonka takia ne onkin rakennettu varsin virtapiheiksi. Tämä on saatu aikaan rajoittamalla niiden prosessointikykyä tai kytkemällä laite ”uneen” kunnes muutoksia tapahtuu.(Galloway & Hancke, 2013.)

2.1.3 HMI (Human Machine Interface)

HMI (Human Machine Interface) on ohjelmisto, josta operaattori voi nähdä mitä prosesseissa tapahtuu. HMI:n kautta voidaan valvoa ja hallita laitteita ja prosesseja. Sen kautta voidaan myös esimerkiksi avata tai sulkea venttiilejä tai käynnistää tai pysäyttää pumppuja. (Sullivan, Luiijf & Colbert, 2016.) Sitä käytetäänkin siis järjestelmän valvontaan ja hallintaan.

Käytännössä SCADA-järjestelmä toimii siten, että HMI:n kautta operaattori valvoo ja antaa tarvittaessa käskyjä, jotka lähetetään fyysisille laitteille, jotka sijaitsevat jopa tuhansien kilomerien päässä valvontapaikasta. RTU kommunikoi HMI:n ja MTU:n (Master Terminal Unit) välillä. (Galloway & Hancke, 2013)

2.2 SCADA-järjestelmän ominaispiirteitä

Käytännössä SCADA-järjestelmä toimii siten että HMI:n kautta operaattori valvoo ja antaa tarvittaessa käskyjä, jotka lähetetään fyysisille laitteille, jotka sijaitsevat jopa tuhansien kilomerien päässä valvontapaikasta. RTU kommunikoi HMI:n ja MTU:n (Master Terminal Unit) välillä. (Galloway & Hancke, 2013) SCADA-järjestelmä on siis Windows ohjelmisto, joka mahdollistaa ihmisoperaattorin valvoa teollisia prosesseja ja varastoida ja analysoida arvoja. (Langner, 2011)

SCADA-järjestelmät eroavat monin eri tavoin perinteisistä kaupallisista tietojärjestelmistä, suurimpana erona se, että SCADA-järjestelmällä ohjataan aina fyysisiä laitteita, jonka seurauksena palvelutaso on suuresti eroava. SCADA-järjestelmien täytyy olla myös erittäin luotettavia, koska niillä hallitaan fyysisiä laitteita ja järjestelmän kaatuminen voi aiheuttaa vakavia vahinkoja tuotannolle, ympäristölle tai jopa ihmisille, kuten kävi esimerkiksi Fukushiman ydinvoimalaonnettumuudessa 2011. (Galloway & Hancke, 2013.)

SCADA-järjestelmille tunnusomaista on niiden tarve luotettavuudelle. Järjestelmät onkin rakennettava siten, että esimerkiksi yhden RTU:n menetys ei aiheuta vaaraa järjestelmän kokonaisuuden toiminnalle. Järjestelmän kannalta kriittisiä toimintoja ovat: CPU (Control Processing Unit), päämuisti ja puskurin uudelleentulostin, ajurit ja kommunikaatioväylät. (Bailey & Wright, 2003.)

SCADA-järjestelmä on vain ohjelmisto, joten informaatioteknologian kehitys vaikuttaa niihin suuresti ja informaatioteknologian kehitys voikin nopeasti tehdä SCADA-järjestelmästä vanhentuneen, vaikka SCADA-järjestelmän itsensä käyttämä tekniikka on puolestaan erittäin pitkäikäistä verrattuna normaaliin informaatioteknologiaan. (Galloway & Hancke, 2013.)

SCADA-järjestelmät ovat maantieteelliseltä kooltaan valtavia ja järjestelmien arkkitehtuuri koostuukin useista eri kerroksista. Esimerkiksi laitteiden kontrollointi tapahtuu yhdellä tasolla, niiden yhteen liittäminen omalla tasollaan, HMI voi olla sijoitettuna niiden yläpuolelle ja vielä niiden yläpuolelle on arkkitehtuurin taso jossa on datan keräys ja ulospäin suuntautuva kommunikointi. Jokaisella tasolla on käytössä omat tiedonsiirtoprotokollansa.(Galloway & Hancke, 2013.)

Järjestelmän prosessien tulee olla reaaliaikaisia ja järjestelmän tuleekin lähettää, prosessoida ja vastata niin nopeasti kuin mahdollista. Mitä korkeammalla tasolla järjestelmän hierarkiassa liikutaan, sitä alhaisemmat aikavaatimukset ovat (Galloway & Hancke, 2013). Älykkään sähköverkon tapauksessa maksimaalinen lähetykseen käytetty aika on 12-20 millisekuntia, eikä järjestelmän

ohjauskeskuksessa käytettävä data saa olla vanhempaa kuin 15 sekuntia, kun se saapuu valvomoon (Ericsson, 2010). Datapaketit, joita viesteissä lähetetään, ovat varsin pieniä, varsinkin järjestelmän alhaisilla tasoilla. Ne saattavat olla kooltaan vain muutaman bitin kokoisia, jos järjestelmä esimerkiksi lähettää vain yksittäisen mittarin yksittäisiä arvoja (Galloway & Hancke, 2013).

3 SCADA-järjestelmään kohdistuvat kyberhyökkäykset

Tässä tutkielman luvussa käsitellään SCADA-järjestelmään kohdistuvia kyberhyökkäyksiä. Ensin tutkaillaan yleisemmällä tasolla SCADA-järjestelmään kohdistuneita hyökkäyksiä, erityisesti Stuxnetiä, jotta saataisiin käsitys miten hyökkäys tapahtuu, ja mitä vaikutuksia sillä on järjestelmän toiminnalle. Tämän jälkeen käydään vielä läpi yksittäisiä hyökkäystapoja.

3.1 Toteutuneita hyökkäyksiä SCADA-järjestelmiin

Ensimmäinen tunnettu SCADA-hyökkäys tapahtui Neuvostoliitossa vuonna 1982, kun Siperiassa sijaitseva kaasuputki räjähti troijalaisen seurauksena. Ohjelmisto, joka ohjasi pumppuja, turbiineja ja venttiilejä oli ohjelmoitu sekoamaan ja nostamaan putken sisällä olevat paineet niin korkeaksi, että se aiheutti valtavan räjähdyksen. (Evancich & Li, 2016.)

Turkissa puolestaan hyökättiin vuonna 2008 öljyputken SCADA-järjestelmään. Räjähdyksen seurauksena maahan joutui yli 30 000 tynnyrillistä öljyä. Hyökkääjät pääsivät järjestelmään langattoman kameran kommunikointi-järjestelmän haavoittuvuuksien kautta ja siitä syvemmälle sisäiseen verkkoon. Tämän jälkeen hyökkääjät muuttivat yksiköitä, joita käytettiin hälyttämään operaattorit valvomossa, sekä rikkoivat PLC:t venttiiliasemilla, jonka seurauksena paine öljyputkessa nousi aiheuttaen räjähdyksen. (Mclaughlin, ym., 2016.)

Tunnetuimpana SCADA-hyökkäyksenä yleisesti pidetään vuonna 2010 löydettyä Stuxnet-nimistä haittaohjelmaa. Se löytyi iranilaisesta uraanin rikastamosta. (Evancich & Li, 2016.) Sen päätavoite oli muokata koodia, joka ohjaa PLC:tä (Programmable Logic Controller) ja muuttaa sen odotettua toimintaa (Karnouskos, 2011). Toisaalta on myös mainittava, että esimerkiksi Lagnerin (2011) mukaan, Stuxnet ei ollut lainkaan SCADA-hyökkäys, vaikka siinä hyödynnettiinkin SCADA-järjestelmää levityksessä, itse hyökkäys kohdistui teollisiin hallintajärjestelmiin.

Uraanin rikastamo, johon hyökättiin, on internetistä eristetty kokonaisuus, joten Stuxnet saatiin eristettyyn ympäristöön USB-muistien välityksellä. Haittaohjelma aloitti toimintansa internetistä saastuttaen USB-laitteen, johon se loi näkymättömän ja raportoimattoman ikkunan, johon se kopioi kuusi tiedostoa. Tämän jälkeen USB-muisti saastutti tietokoneen, joka oli kohdeorganisaation sisällä. Tietokoneen saatua tartunnan Stuxnet päivitti itsensä internetin välityksellä ja jatkoi leviämistä sisäisessä verkossa monistuen USB-muisteihin. Lopulta saastunut USB-muistitikku päätyi uraanin rikastamon sisäiseen verkkoon saastuttaen tietokoneen. (Evancich & Li, 2016.)

Siitä huolimatta, että Stuxnet saastutti kaikki tietokoneet, joihin se sai yhteyden, se oli huomattavasti valikoivampi ohjaimien suhteen, sillä se saastutti ainoastaan Siemensin valmistamia ohjaimia (Langner, 2011) (Evancich & Li, 2016). Saastutettuaan ohjaimet, Stuxnet alkoi hajoittamaan sentrifugeja, joissa uraania rikastettiin. Hajoittaminen tapahtui vaihtelemalla sentrifugien pyörimisnopeutta 1410 hertsistä 2 hertsiin ja siitä 1064 hertsiin. SCADA-järjestelmältä piilotettiin tiedot pyörimisnopeuden muutoksesta ja myös nopeudenvaihtelusta aiheutunut värinä piilotettiin sensoreilta. Tavoitteena ei ollut rikkoa laitteita välittömästi, vaan kuluttaa niitä ja lisätä huomaamattomasti pitkän ajan kuluessa sentrifugien rikkoutumista. (Evancich & Li, 2016.)

3.2 Mahdollisia kyberhyökkäystapoja SCADA-järjestelmiin

SCADA-järjestelmillä on massiivinen hyökkäyspinta-ala johtuen järjestelmien suuresta toimialueesta fyysisesti ja digitaalisesti. Järjestelmien pitkäikäisyys aiheuttaa myös ongelmia, sillä esimerkiksi yleisesti yhä käytössä vuonna 2015 ollut RTU on rakennettu Windows 95-käyttöjärjestelmän päälle. (Evancich & Li, 2016.) Esittelen seuraavaksi tarkempia hyökkäystapoja SCADA-järjestelmään.

3.2.1 Puskurin ylivuoto (Buffer overflow)

Puskurin ylivuoto tarkoittaa hyökkäystä, joka paljastaa kyberelementit, kuten taulukot, liikkumalla määrättyjä rajoja kauemmaksi. Tämä mahdollistaa hyökkääjän murtautumisen pois ohjelman normaalista toiminnasta ja alkaa operoida sitä. (Evancich & Li, 2016.)

Käytännössä puskurin ylivuoto tapahtuu, kun ohjelman suorittaessa toimintaa puskuriin ohjautuu liian paljon dataa, jonka seurauksena data ylikirjoitetaan viereisiin muistipaikkoihin ja riippuen siitä, mitä niihin paikkoihin on tallennettu, saattaa itse ohjelmankin toiminta muuttua. (Kuperman ym. , 2005.) Fyysiset SCADA-järjestelmän osat ovat immuuneja tämän kaltaisille hyökkäyksille, mutta serverit, jotka hallitsevat PLC:tä ja RTU:ta ovat haavoittuvaisempia. Kaiken lisäksi monet SCADA-järjestelmän osat ovat 8- tai 16-bittisiä, ne vuota-

vat helposti yli, jonka seurauksena hyökkääjät voivat kirjoittaa järjestelmään itse haitallista koodia. (Evancich & Li, 2016.)

3.2.2 Ohjelmakoodin injektointi (Code injection)

Ohjelmakoodin injektointi on hyökkäys, jossa hyökkääjä pääsee kriittisiin prosesseihin esimerkiksi puskurin ylivuodon avulla ja sitten pakottaa järjestelmän suorittamaan uutta koodia. (Evancich & Li, 2016) Yleisin ohjelmakoodin injektointi tapahtuu siten, että hyökkääjä ylikirjoittaa parhaillaan ajettavan ohjelmakoodin paluuosoitteen, ohjaa sen haluttuun osoiteeseen, jossa haluttu ohjelmakoodi sijaitsee, jonka jälkeen hyökkääjä voi suorittaa itse kirjoittamansa koodin. (Salamat ym. 2011.)

3.2.3 Use-after-free

Use-after-free on haavoittuvuus, joka ei tarkista onko resurssi vapautettu. Tämä haavoittuvuus mahdollistaa hyökkääjän mielivaltaisen koodin ajamisen. (Evancich & Li, 2016) Use-after -free -hyökkäyksessä käytännössä uudelleenkäytetään osoitin joka viittaa muistialueeseen sen jälkeen kun se on vapautettu. Tämän seurauksena ohjelma joko kaatuu tai luo roikkuvan osoittimen. Mikäli järjestelmä luo roikkuvan osoittimen, hyökkääjä pystyy suorittamaan omaa ohjelmakoodiaan tai kaappaamaan ohjelmiston virtauksen. (Saito, ym., 2017)

3.2.4 DLL kaappaus (Dynamic Link Library hijacking)

DLL kaappauksessa hyökkääjä korvaa vaaditun Dynamic Link Libraryn uudella DLL:llä, joka sisältää haitallista koodia. DLL tarkoittaa moduulia, joka sisältää funktioita ja dataa, jota muut moduulit kuten sovellukset tai DLL:t voivat käyttää. Se on siis ikään kuin jaettu kirjasto, josta jaetaan dataa useiden ohjelmien kesken. (Microsoft, 2020.) Tämän mahdollistaa se, että monet SCADA-järjestelmät käyttävät yhä Windows 95-käyttöjärjestelmää, jossa tämänkaltainen hyökkäys on helppo toteuttaa, koska Windows 95-käyttöjärjestelmä ei vaadi DLL-kirjastojen todentamista. (Evancich & Li, 2016).

3.2.5 Syötteen todentaminen (Input validation)

Syötteen todentaminen on hyökkäys, joka suoritetaan lähettämällä syöte järjestelmään, jota järjestelmä ei voi vahvistaa. Esimerkiksi hyökkääjä lähettää järjestelmään syötteen, jossa ei ole lopettamiskäskyä. Sen seurauksena on helposti puskurin ylivuoto ja siitä seuraa järjestelmän kaatuminen tai hyökkääjä pystyy muuttamaan järjestelmän hallinnan virtausta. Toinen esimerkki on SQL-injektiohyökkäys, jossa lähetetään haitallisia SQL-kyselyitä. Kyselyt voivat antaa hyökkääjälle suoran hallinnan SQL-tietokantaan. (Evancich & Li, 2016)

Käytännössä SQL-injektiohyökkäyksessä täytetään piilotettuja tai näkyvissä olevia kenttiä SQL-kyselyillä, jotka lähetetään tietokantaan. SQL-injektiohyökkäys voi olla myös uinuva, jolloin sen laukaisee jokin tietty tapahtuma. (Fonseca, Vieira & Madeira, 2014)

3.2.6 Pääsy valvomoon (Access control)

Pääsy valvomoon on hyökkäys, jonka mahdollistaa SCADA-järjestelmien hajanainen sijainti ja sillä tarkoitetaan hyökkääjän mahdollisuutta päästä alijärjestelmiin menemällä fyysisesti syrjäiseen paikkaan, jossa SCADA-järjestelmä sijaitsee. Fyysinen pääsy järjestelmään esimerkiksi ala-aseman kautta mahdollistaa hyökkääjälle lähes rajoittamattomat hyökkäysmahdollisuudet järjestelmää vastaan. (Evancich & Li, 2016.)

Aikaisemmin ala-asemilla oli vain rajoitetut mahdollisuudet kommunikoida muiden järjestelmien kanssa, mutta nykyään ne ovat esimerkiksi älykkäässä sähköverkossa yhdistetty kriittisiin fyysisiin resursseihin, kuten katkaisijoihin, muuntajiin, vaihekiskoihin ja voimansiirtojohtoihin. Onnistuneet hyökkäykset ala-asemille, voivatkin pahimmassa tapauksessa johtaa sähkökatkokseen ja vakaviin rahallisiin vahinkoihin. (Hong ym., 2019)

3.2.7 Kovakoodatut tunnukset ja salasanat (Hard-coded credentials and passwords)

Lukuisissa SCADA-järjestelmissä on hyvin tiedossa olevia ohjelmakoodiin ohjelmoituja salasanoja, jotka mahdollistavat hyökkääjälle helppoja pääsypaikkoja järjestelmään. Järjestelmät sisältävät myös kovakoodattuja tunnistetietoja, jotka mahdollistavat hyökkääjälle sisäänpääsyn järjestelmään, ilman että järjestelmän ylläpitäjälle on mahdollisuutta estää tätä. Siitä huolimatta, että kyberturvallisuudessa olisi tärkeää, että prosessit suoritetaan mahdollisimman pienillä käyttöoikeuksilla, useat SCADA-järjestelmät vaativat pääkäyttäjänoikeuksia (admin). Sen lisäksi ne voivat myös vaatia, että ne täytyy ajaa juuresta. Seurauksena on se että jos järjestelmään pääsee, hyökkääjällä on vapaa pääsy kaikkialle järjestelmään. (Evancich & Li, 2016.)

Aiemmin esitellyssä Stuxnet-hyökkäyksessä Siemenes oli kovakoodannut tietokannan käyttäjänimen ja salasanan toimittamaansa järjestelmään. Lisäksi kovakoodatut kirjautumisnimet, kuten ”Admin”, lisäävät kiistämisen mahdollisuutta siinä tapauksessa että hyökkääjä olisikin työntekijä yrityksen sisältä. Sen takia olisikin tärkeää käyttää järjestelmien kirjautumistietoina käyttäjän yksilöivää tunnusta. (Nicholson ym., 2012.)

3.2.8 Todennus (Authentication)

Monien SCADA-järjestelmien vanhasta iästä johtuen järjestelmissä esiintyy todennusongelmia. Modernit protokollat edellyttävät käskyjen todentamista, eli järjestelmän täytyy todentaa käskyn antajalla olevan oikeudet siihen. Kuitenkin, RTU:t ja PLC:t eivät todenna käskyjä, joten kuka tahansa voi antaa käskyn niille. (Evancich & Li, 2016.) Jotta järjestelmä huomaisi datassa olevia pieniä virheitä, SCADA-järjestelmän kommunikaatioprotokollat sisältävät RTU:n laskemaa koodia, jonka tehtävä on löytää virheitä. Tätä koodia ei kuitenkaan todenneta mitenkään. (Vukovic ym., 2012.)

3.2.9 Raaka voima (Brute force attacks)

Raa'allakin voimalla on mahdollista hyökätä SCADA-järjestelmään. Pääasiassa tämäkin ongelma johtuu suurelta osin järjestelmien vanhasta iästä ja siitä että SCADA-järjestelmissä on usein huonot salauskirjastot, johtuen siitä, että RTU:t ja PLC:t ovat usein 8- tai 16-bittisiä. Järjestelmissä ei ole oikeaa satunaisgeneraattoria, joten salattu data ei ole tarpeeksi satunnaista, mikä vähentää salauksen tehoa. (Evancich & Li, 2016.) Heikko salausalgoritmi voi mahdollistaa raa'alla voimalla tapahtuvan salauksen murtamisen, jos hyökkääjä arvaa, mikä merkkijono tuottaa saman hajautustaulun kuin oikeakin salasana. (Zhu, Joseph & Sastry, 2011.)

3.2.10 Riittämätön salauksen vahvuus (Inadequate encryption strength)

Salauksen riittämätön vahvuus on kytköksissä aiemmin esiteltyyn raa'an voiman hyökkäyksiin. Järjestelmässä on siis käytännössä dataa, joka on salattu liian heikosti, koska järjestelmän kehittäjät luulivat sen olevan turvassa. Esimerkiksi osa vanhemmista SCADA-järjestelmistä säilöö salasanat selväkielisenä tekstinä. (Evancich & Li, 2016.) Syynä tähän datan selväkielisyyteen on, se että järjestelmän osat voivat olla vanhoja, ajalta jolloin SCADA-järjestelmät olivat täysin suljettuja systeemejä, eikä niitä ollut tarkoitettukaan yhdistettäväksi internettiin. Tämän lisäksi vähätehoiset SCADA-järjestelmän laitteet, kuten RTU:t, eivät ole kykeneviä vahvoihin kryptologisiin algoritmeihin, jonka lisäksi vahvat salaustavat saattaisivat rikkoa järjestelmän aikarajoituksia. (Nazir & Kaleem, 2018.)

3.2.11 Resurssien ehdyttäminen ja hallinta (resource exhaustion and management)

Resurssien ehdyttäminen on hyökkäys, jossa hyökkääjä kuluttaa enemmän resursseja kuin mitä järjestelmällä on käytettävissä. SCADA-järjestelmässä tämänkaltainen hyökkäys on toteutettavissa esimerkiksi lähettämällä päivityksiä järjestelmään nopeammin kuin datanhallintaserveri voi niitä

prosessoida. Tällä hyökkäyksellä ei yleensä saada järjestelmää hallintaan, mutta se vähentää järjestelmän toimintoja. (Evancich & Li, 2016.) Resurssien eheyttäminen voi koskea myös esimerkiksi virrankulutuksen kasvattamista niin suureksi, että laite, jossa järjestelmä toimii, käyttää akkunsa loppuun paljon suunniteltua nopeammin (Racic, Ma & Chen, 2006).

Resurssienhallintahyökkäys on puolestaan palvelunestohyökkäys, jossa hyökkääjä lähettää komentoja rajoittaakseen resursseja, joita useat SCADA-järjestelmän alijärjestelmät käyttävät, ja jonka seurauksena ne lakkaavat toimimasta (Evancich & Li, 2016). Perinteisen palvelunestohyökkäyksen tavoitteena on aiheuttaa ylivuoto käyttäjän ja kernelin, eli järjestelmän ytimen, puskureissa. Langattomissa järjestelmissä hyökkäys voi olla vielä helpompi, sillä hyökkääjä voi esimerkiksi lähettää elektromagneettista säteilyä lähettimeen ja saavuttaa siten viestin lähetyksen viivästymisen, sillä lähetin luulee olevansa liian kiireinen. Myös viestin vastaanottajan vastaanottimeen voidaan kohdistaa häirintää siten, että laite ei voi vastaanottaa viestejä. Tuonkaltaiset jamming-tekniikalla suoritetut palvelunestohyökkäykset ovat tosin helppo tunnistaa. (Pelechrinis, Iliofotou & Krishnamurthy, 2011.)

3.2.12 Rootkitit (Rootkits)

Rootkit on ohjelmisto, joka on suunniteltu piilottamaan toimintansa vähentääkseen kiinnijäämisen riskiä. Ne myös yleensä helpottavat läpäistyyn prosessiin sisäänpääsyä tulevaisuudessa. Rootkit on moniosaisen hyökkäyksen osa, jollainen aiemmin esitelty Stuxnet-hyökkäyskin oli. Rootkit asennetaan tai toimitetaan kohteena olevaan järjestelmään haavoittuvuuden kautta. Kun rootkit on asennettu tai asentunut, se pyrkii nostamaan omaa pääsyään korkeamman luottamuksen tasoille, poistaen samalla jäljet asennuksestaan samanaikaisesti piilottaen tekemisistään syntyviä jälkiä. Tässä vaiheessa rootkitillä on erittäin pieni todennäköisyys jäädä kiinni. (Evancich & Li, 2016.)

Nykyaikaiset haittaohjelmat ovat yleensä sekoitus useista eri osista ja rootkit onkin yleensä ensimmäinen osa hyökkäystä ja se toimiikin niin sanotusti ovenavaajana. Rootkit voi säilyttää haitalliset tiedostot, ja ne voivat säilyä uudelleenkäynnistyksistä huolimatta ja piilottaa tiedostot ja prosessit, jotta virustorjuntaohjelmistotkaan eivät niitä huomaa. (Rudd ym., 2017.)

Rootkitit voidaan jakaa useampaan eri sukupolveen. Ensimmäisen sukupolven rootkitit pysyivät levykkeillä, eivätkä ne siten pystyneet asentumaan itse tietokoneeseen. Etuja ensimmäisen sukupolven rootkiteissä olivat helppo asennettavuus ja niiden selviytyminen tietokoneen uudelleenkäynnistyksestä, mutta johtuen rootkitien sijainnista levykkeellä ne olivat helposti huomattavissa ja poistettavissa. (Rudd, Rozsa, Gunther & Boult, 2017.)

Toisen sukupolven rootkitit kaappasivat toimintamuistin ja harhauttivat ohjelmiston virtausta siten, että haitallinen ohjelmakoodi voitiin suorittaa. Tätä tekniikkaa kutsutaan yleisesti nimellä hooking ja se voidaan tehdä usealla eri tavalla. Rootkitit käyttävät hookingia eli koukkaamista muuttaakseen muistia siten, että haitallinen koodi suoritetaan yleensä ennen tai jälkeen oikean järjes-

telmän kutsua. Tämän seurauksena ne voivat suodattaa paluuarvot tai toiminnallisuuden pyynnön järjestelmälle. Etuina toisen sukupolven rootkiteissä on niiden haittaohjelmien erikoistuminen, mutta toisaalta niiden vaikea injektointi järjestelmään. (Rudd ym., 2017.)

Kolmannen sukupolven rootkitit suorittavat suoraan kernelin, eli tietokoneen ytimen, olioiden manipulaatiota. Ne pyrkivät kumoamaan kernelin eheyden ottamalla kohteekseen muuttuvat kernelin tietorakenteet. Verrattuna toisen sukupolven koukkuihin, kolmannen sukupolven kerneliin kohdistuvat hyökkäykset ovat huomattavasti vaikeampia havaita, sillä ne tähtäävät muuttuviin tietorakenteisiin, joiden arvot muuttuvat normaalistikin järjestelmän ajon aikana. Toisen sukupolven koukkaamisissa puolestaan muutettiin staattisia tietoa, joiden ei olisi kuulunut muuttua. Siten kolmannen sukupolven rootkitit ovat vaikeammin tunnistettavissa. Etuina kolmannen sukupolven rootkiteillä on se, että niitä on äärimmäisen vaikea havaita, mutta huonona puolena on niiden rajoittuneet toiminnallisuudet. (Rudd ym., 2017.)

Neljännen sukupolven rootkitit toimivat virtualisoiduilla kerroksilla, BIOSissa ja kovalelyllä. Tämän hetken tietojen mukaan neljännen sukupolven rootkitejä on kokeiltu vain idean toteuttamisen selvittämiseksi. Käytännössä neljännen sukupolven rootkitit sijaitsevat järjestelmätasoa alempana, eikä järjestelmä siten voi havaita niitä. Tämä vaatii kuitenkin käytännössä kustomoidun virtuaaliympäristön, BIOSin, laitteiston tai toimitusketjun murtamisen toimiakseen. (Rudd ym., 2017.) Rootkittien tunnistamiseen on esitelty Gibraltar-niminen työkalu, joka käyttää tunnistamisessa hyväkseen tilastollista päättelyä ja tunnistaa rikkomuksia, joita rootkit tekee kernelissä (Baliga, Ganapathy & Iftode, 2011).

4 SCADA-järjestelmän suojaaminen kyberhyökkäyksiä vastaan

Tässä tutkielman luvussa esittelen SCADA-järjestelmän suojausta. Esimerkkinä käytän suurimmilta osin älykästä sähköverkkoa, pääasiassa siitä syystä, että siitä löytyi paljon tähän tutkielmaan soveltuvaa tutkimustietoa. Ensin esittelen älykkään sähköverkon suojaukseen vaikuttavia vaatimuksia, jonka jälkeen esittelen älykkään sähköverkon vaatimuksia SCADA-järjestelmälle. Lopuksi käyn vielä läpi älykkään sähköverkon kyberturvallisuusstandardeja sekä tapoja parantaa SCADA-järjestelmien turvallisuutta.

4.1 Älykkään sähköverkon suojaukseen vaikuttavat vaatimukset

Viimevuosina älykkäät sähköverkot ovat saaneet paljon huomiota ja niiden odotetaan kehittyvän lisää tulevina vuosina. SCADA-järjestelmät ovat älykkään sähköverkon ytimessä oleva alijärjestelmä, joihin tässäkin tutkielmassa keskitytään. Electric Power Research Instituten (EPRI) mukaan yksi suurimmista haasteista koskien älykkäiden sähköverkkojen kehittämistä on järjestelmän kyberturvallisuus (Metke & Ekl, 2010). Sähkönsaanti on yhteiskunnan kannalta kriittistä ja tämänkaltaiset muutokset ovatkin lisänneet turvallisuusnäkökulmiin fyysisten vaatimusten lisäksi kyberturvallisuuden ulottuvuuden (Ericsson, 2010).

Sähköverkkoon on lisätty uusia ominaisuuksia, kuten jaettua älyä ja laajakaistayhteyksiä, jotka lisäävät sähköverkon luotettavuutta ja tehokkuutta, mutta ne kasvattavat myös järjestelmän haavoittuvuutta. Mikäli turvallisuusnäkökulmia ei oteta riittävästi huomioon, näin laajassa järjestelmässä jää avoimia ikkunoita kyberhyökkäyksille. (Metke & Ekl, 2010.)

Kriittisimmät osat älykkäässä sähköverkossa ovat sähköverkon kommunikointi (Power System Communication, PSC), SCADA-järjestelmä ja alaasemat, eli tässä tapauksessa esimerkiksi tuulivoimalat tai muut ei-jatkuvassa fyysisessä valvonnassa olevat asemat ja siellä sijaitsevat järjestelmät. Järjestel-

män kommunikaatiovaatimukset täytyy luokitella, koska se helpottaa vaatimusten käsittelyä ja järjestystä. Ericssonin (2010) mukaan vaatimukset voidaan jakaa kolmeen kategoriaan: 1) reaaliaikaiset operationaaliset vaatimukset, 2) hallinnolliset operationaaliset vaatimukset ja 3) hallinnolliset kommunikaatiovaatimukset. Järjestelmän kolmijaon Ericsson esitteli jo vuonna 2001/2002 ja se on laajasti käytetty sekä Ruotsissa että Ruotsin ulkopuolella. (Ericsson, 2010.)

Reaaliaikaiset operationaaliset kommunikaatiovaatimukset määrittävät kommunikaation minimivasteajat, jotka järjestelmä vaatii pysyäkseen toiminnassa. Kommunikaation on tapahduttava todella tiukissa vaatimuksissa, ja vaatimukset määrittelevätkin käytettävän tekniikan. Viestiyhteyden suojaamiseksi maksimiaika, joka on sallittua käyttää viestin lähettämiseen, on 12-20 millisekuntia, riippuen kuitenkin käytössä olevasta suojaustavasta. Tiukka vaatimus tulee siitä, että vikavirran katkaisu toimii noin 100 millisekunnissa. Mitatut arvot, jotka toimitetaan ohjauskeskukseen, saavat olla korkeintaan 15 sekuntia vanhoja ja virrankatkaisusta tiedon tulee olla ohjauskeskuksessa korkeintaan 2 sekunnin päästä tapahtumasta. Lisäksi viestintään käytetään perinteistä matkapuhelinta työntekijöiden välillä, jotta pystytään ratkaisemaan ongelmatilanteita ja jakamaan ohjeita. (Ericsson, 2010.)

Hallinnolliset operationaaliset vaatimukset eivät ole yhtä tiukkoja kuin edellä mainitut vaatimukset. Käytännössä tätä kommunikaatiokanavaa käytetään kun tarvitaan enemmän tietoja ja tukea siihen, mitä on tapahtunut. Esimerkkejä ovat esimerkiksi turvajärjestelmä, vianpaikannukset, resurssien hallinnointi, mittaukset ja tiedonsiirto, sekä ala-asemien kameravalvonta. Tällaisen kommunikaation ei tarvitse tapahtua reaaliajassa ja aikavaatimukset ovat maltillisia. (Ericsson, 2010.)

Hallinnolliset kommunikaatiovaatimukset pitävät sisällään äänikommunikaation ja faksinkäytön myös toimistojen tai toimipisteiden välillä niissä tilanteissa, kun kommunikaatiolla on hallinnollinen tarkoitus. (Ericsson, 2010.)

4.2 SCADA-järjestelmän kyberturvallisuusstandardit

Vuonna 2010 tehdyssä tutkimuksessa vertailtiin SCADA-järjestelmien kyberturvallisuusstandardeja. (Sommestad, Ericsson & Nordlander, 2010) Tutkimukseen valittuja standardeja oli kaiken kaikkiaan kahdeksan ja niiden valinnassa oli neljä kriteeriä:

1. Standardien tuli olla saatavilla englanninkielellä.
2. Standardit on julkaissut standardointiin erikoistunut taho tai valtiollinen toimija.
3. Standardin täytyy keskittyä SCADA-järjestelmän turvallisuuteen, eikä esimerkiksi IT-turvallisuuteen yleisesti.
4. Standardien täytyy keskittyä SCADA-järjestelmiin kokonaisuutena, eikä vain alajärjestelmään tai komponentteihin kuten älykkäisiin elektronisiin laitteisiin.

Taulukko 1 Tutkimukseen valitut standardit

Dokumentti	Julkaisija
Good Practice Guide, Process Control and SCADA Security	Centre for the Protection of National Infrastructure (CPNI)
Cyber Security Procurement Language for Control Systems	Department of Homeland Security (DHS)
21 steps to Improve Cyber Security of SCADA Networks	U.S. Department of Energy (DOE)
CIP-002–1 - CIP-009–1	North American Electric Reliability Corporation (NERC)
Guide to Industrial Control Systems (ICS) Security	National Institute of Standards and Technology (NIST)
System Protection Profile - Industrial Control Systems	National Institute of Standards and Technology (NIST)
ANSI/ISA-99.00.01–2007 Part 1–3	The International Society of Automation (ISA)
Cyber security for Critical Infrastructure Protection	U.S. Government Accountability Office (U.S. GAO)

SCADA-standardit keskittyivät vahvasti vastatoimiin ja avainsanoista vastatoimiin liittyneet sanat esiintyivätkin tutkimuksissa kaiken kaikkiaan 8222 kertaa, kun taas avainsanat jotka liittyvät uhkiin mainittiin vain 876 kertaa. Tutkimuksen mukaan, tätä eroa voisi selittää se, että on vaikea olla varma, mitkä uhat todellisuudessa uhkaavat SCADA-järjestelmiä. Tutkimuksen mukaan standardit ehdottivatkin uhka-analyysiä vastatoimeksi. Toisaalta haitallinen koodi oli useimmiten mainittu uhka ja se saikin peräti 40% standardien uhka-osuuksien huomiosta, mutta selvänä vastatoimena toimiva virustorjuntaohjelmisto sai vain 1,6% huomion vastatoimista. Tätä epäkohtaa selitettiin SCADA-järjestelmään kohdistuvien uhkien epävarmuudella, sillä varmistettuja tapauksia ei ole kovin montaa, ainakaan julkisessa tiedossa. (Sommestad, Ericsson & Nordlander, 2010.)

4.3 SCADA-järjestelmän riskienarviointi

SCADA-järjestelmän riskienarviointi on tärkeä osa varautumista. Sen ansiosta voidaan parantaa järjestelmää analysoimalla, mikä voi mennä pieleen, millä todennäköisyydellä ja mitkä ovat seuraukset (Cherdantseva, ym., 2016). Riskienarviointiin on kehitetty useita eri metodeja. Valitsin tähän tutkielmaan Cherdantsevan ym. (2016) tutkimuksessa analysoiduista riskienarviointimenetelmistä kolme eniten viitattua.

4.3.1 Hyökkäyspuut (Byres, Franz & Miller, 2004)

Vuonna 2004 esiteltiin hyökkäyspuut, joita käytetään haavoittuvuuksien arviointiin SCADA-järjestelmässä. Hyökkäyspuu mahdollistaa strukturoidun

näkökulman tapahtumista, jotka johtavat hyökkäykseen, ja helpottaa niiden tunnistamisessa, sekä sopivien vastatoimien valinnassa. Byresin ym. mukaan riski riippuu järjestelmän arkkitehtuurista ja tilasta, vastatoimista, hyökkäyksen vaikeudesta, huomatuksi tulon todennäköisyydestä ja hyökkäyksen hinnasta. Arvion tarkoitus on laskea korkeimmat hyökkäyksen ominaisuudet ja tunnistaa mahdolliset tavat saavuttaa hyökkäyksen lopullinen päämäärä. (Byres, Franz & Miller, 2004.)

Toimialan ammattilaisten tehtävä on identifioida potentiaaliset hyökkääjän päämäärät ja suunnitella niiden perusteella hyökkäyspuu, jossa tavoitteet ovat muotoiltuna solmukohdiksi. Tämän jälkeen jokaiselle hyökkäyspuun lehdelle annetaan taso teknisen vaikeuden mukaan. Jokainen tavoite arvioidaan myös sen mukaan, miten vakavia vaikutuksia sillä toteutuessaan olisi ja miten todennäköistä hyökkäys olisi huomata. (Cherdantseva, ym., 2016.)

Tutkimuksessa käytetyssä esimerkissä hyökkäyspuusta pystyttiin seuraamaan polkua, joka oli arvioitu hyökkääjän kannalta helpoimmaksi. Siten selvisi että esimerkin tapauksessa onnistunut hyökkäys ei tapahtuisi internetin välityksellä, vaan saamalla fyysinen pääsy ala-asemalle syrjäisessä sijainnissa. (Byres, Franz & Miller, 2004.)

4.3.2 Riskien arviointi, havaitseminen ja vastatoimet (Cárdenas ym., 2011)

Riskinarviointimetodi sensoriverkoille yhdistettynä hyökkäyksen havaitsemiseen ja automaattiseen vastausmoduuleihin esiteltiin vuonna 2011. Se on sensoriverkon yhteydessä hyväksytty ja tulkittu kaava, jolla lasketaan riskin keskimääräisesti aiheuttamaa tappiota. Mallin perusteella, kun anomalia huomataan, järjestelmä suorittaa automaattisia vastatoimia, samalla kun odotetaan järjestelmänkäyttäjän toimia. Malli helpottaa laskemaan, minkä tyyppisiin hyökkäyksiin ja mille sensoreilla annetaan enemmän rahaa turvallisuusbudjetista. (Cherdantseva, ym., 2016.)

Tutkimuksessa myös ehdotettiin automaattisia vastauksia havaittuihin hyökkäyksiin, vaikkakin ne voivat olla esimerkiksi väärien hälytysten tapauksessa ongelmallisia. Ratkaisuksi ehdotettiin automaattista vastausta, joka on väliaikainen ratkaisu siihen asti, että ihminen tutkii hälytyksen. (Cárdenas, ym., 2011.)

Tutkimuksessa myös varoitetaan laajojen järjestelmien joustamattomuudesta, joka johtuu niiden arkkitehtuurista. Mikäli järjestelmää ei ole alun alkaen rakennettu riittävän joustavaksi ja toipumiskykyiseksi se voi koitua järjestelmän kohtaloksi huomaamattomia hyökkäyksiä ajatellen. Idea onkin toimivampi siten operationaalisessa toiminnassa, sillä vielä ei ole pystytty rakentamaan täysin joustavia hallintarakenteita eikä algoritmejä. (Cárdenas ym., 2011.) SCADA-järjestelmien suojauksessa voidaan hyödyntää myös petri-verkkoja. Petri-verkko on mallinnustapa, joka on tunnettu graafisesta ja analyyttisistä ominaisuuksistaan. (Huang & Kirchner, 2011.)

4.3.3 Kriittisen infrastruktuurin kyberturvallisuus: hyökkäyksen ja puolustuksen mallintaminen (Ten, Manimaran & Liu, 2010)

Vuonna 2010 esiteltiin SCADA-turvallisuuden viitekehys RAIM, joka koostuu neljästä osasta: Reaaliaikaisesta monitoroinnista, anomalioiden havaitsemisesta, iskun analyysista ja lievennysstrategioista. Reaaliaikainen monitorointi ja anomalioiden havaitseminen perustuvat jatkuvaan järjestelmälokien seurantaan. Iskun analyysin tavoite on esitellä hyökkääjän käyttäytymistä ja mahdollisia iskun vaikutuksia SCADA-järjestelmään. (Ten, Manimaran & Liu, 2010)

RAIM-viitekehykseen kuuluva iskuanalyysi koostuu neljästä osasta, joita ovat järjestelmän konfiguraation kaappaaminen, virran simulointi, haavoittuvuusindeksin laskeminen ja turvallisuusparannukset. Lievennysstrategiat puolestaan tarkoittavat vahinkojen minimoimista arvioimalla esimerkiksi, onko mahdollista, että kalliita laitteita rikkoutuu tulevaisuudessa hyökkäyksen seurauksena. (Ten, Manimaran & Liu, 2010.)

Iskuanalyysissä käytetään hyväksi hyökkäyspuuta, mihin on laskettu haavoittuvuusindeksi, joka näyttää todennäköisyyden hyökkäykselle onnistua. Todennäköisyys on laskettu perustuen historialliseen dataan hyökkäyksistä ja tietoon vastatoimista ja salasanakäytänteistä. Hyökkäyspuun lehden haavoittuvuusindeksi riippuu kohteen auditoinneista ja salasanojen vahvuudesta. (Cherdantseva, ym., 2016.)

Anomalioiden havaitseminen perustuu tapahtumien havainnoimiseen ja niiden historiallisen datan vertaamiseen. Tällöin voidaan esimerkiksi huomata, että mikäli järjestelmässä on tavallista enemmän yhteyksiä, kyseessä on palvelunestohyökkäys. (Ten, Manimaran & Liu, 2010.)

5 Yhteenveto

SCADA-järjestelmä on teollinen hallintajärjestelmä, jonka tehtävä on kerätä dataa ja valvoa automaatiota laajalla alueella (Sullivan, Luiijf & Colbert, 2016). SCADA-järjestelmiä käytetäänkin esimerkiksi yhteiskunnan kannalta kriittisissä rakenteissa, kuten sähkönsiirrossa (Ericsson, 2010). SCADA-järjestelmien tuleekin olla luotettavia, sillä järjestelmän kaatuminen voi aiheuttaa vakavia vahinkoja (Galloway & Hancke, 2013). Järjestelmän ohjauskeskukseen kohdistuvat hyökkäykset eivät ole todennäköisiä, sillä niiden fyysinen turvallisuus on yleensä hyvällä mallilla (Ten, Liu & Manimaran, 2008). Hyökkäys järjestelmään tapahtuu siis todennäköisesti esimerkiksi tietojärjestelmiä hyödyntäen tai ala-asemien kautta.

SCADA-järjestelmät ovat massiivisia, joten niiden hyökkäyspinta-alakin on massiivinen (Ericsson, 2010). Kaiken lisäksi järjestelmät ovat pitkäikäisempiä verrattuna kuluttajakäytössä totuttuun informaatioteknologiaan, jonka seurauksena teknologinen kehitys voi tehdä niistä tekniikaltaan vanhentuneita (Galloway & Hancke, 2013).

Tuotannossa olevista SCADA-järjestelmistä on vaikea löytää tietoa. Syynä on varmastikin se, että tiedot järjestelmän haavoittuvuuksista muodostaisivat itse turvallisuusuhan. Kuitenkin esimerkiksi vuonna 2015 yhä yleisessä käytössä ollut RTU on rakennettu Windows 95-käyttöjärjestelmän päälle (Evancich & Li, 2016). Aiemmin esitellyistä hyökkäyksistä, esimerkiksi DLL kaappaus hyödyntää juuri tästä kumpuavaa ongelmaa, sillä Windows 95-käyttöjärjestelmässä sellainen hyökkäys on helppo toteuttaa, koska se ei vaadi DLL-kirjastojen todentamista (Evancich & Li, 2016). SCADA-järjestelmissä on myös useita osia, jotka ovat 8- tai 16-bittisiä, joten ne ovat alttiita puskurin ylivuodolle (Evancich & Li, 2016). Järjestelmiin kovakoodatut kirjautumisnimet ja salasanat vähentävät entisestään järjestelmän turvallisuutta (Nicholson ym., 2012). Heikot salausalgoritmit, joita SCADA-järjestelmistä löytyy, voivat mahdollistaa raa'alla voimalla tapahtuvan salauksen murtamisen (Zhu, Joseph & Sastry, 2011). Tulevaisuudessa laskentatehon kehitys entisestään heikentää heikkojen salausalgoritmien tehoa.

Järjestelmän kannalta kriittisiä toimintoja ovat: CPU (Control Processing Unit), päämuisti ja puskurin uudelleentulostin, ajurit ja kommunikaatioväylät. (Bailey & Wright, 2003.) Myös osa SCADA- järjestelmistä on rakennettu ajatellen järjestelmän olevan suljettu, jolloin järjestelmän tietoturvassakin on suuria puutteita. (Nazir & Kaleem, 2018.) Hyökkäykset SCADA-järjestelmiin voivat vaihdella huomaamattomista rootkiteistä vaikutuksiltaan nopeisiin ja nopeasti huomattaviin palvelunestohyökkäyksiin (Rudd ym., 2017) (Pelechrinis, Iliofotou & Krishnamurthy, 2011). Hyökkäystapoja on lukuisia, ja hyökkääjän onnistuminen riippuukin siitä, miten järjestelmä on suojattu, miten taitava hyökkääjä on ja minkälainen järjestelmä on. Järjestelmän jatkuva kehittäminen onkin ensiarvoisen tärkeää turvallisen SCADA-järjestelmän kehittämisessä. Kuten esimerkiksi Stuxnet-hyökkäyksestä nähtiin, onnistuessaan hyökkäyksellä voi olla dramaattisia seurauksia.

SCADA-järjestelmien tutkiminen ja sitä kautta tapahtuva kehittäminen on ensiarvoisen tärkeää järjestelmien turvallisuuden kannalta. Järjestelmät ovat pitkäikäisiä, joten niiden turvallisuutta täytyy kehittää läpi järjestelmien elinkaaren. Vuonna 2004 esiteltiin hyökkäyspuut, joita käytetään haavoittuvuuksien arviointiin SCADA-järjestelmässä. Hyökkäyspuu mahdollistaa strukturoidun näkökulman tapahtumista, jotka johtavat hyökkäykseen, ja helpottaa niiden tunnistamisessa sekä sopivien vastatoimien valinnassa. Byresin ym. mukaan riski riippuu järjestelmän arkkitehtuurista ja tilasta, vastatoimista, hyökkäyksen vaikeudesta, huomatuksi tulon todennäköisyydestä ja hyökkäyksen hinnasta. Arvion tarkoitus on laskea korkeimmat hyökkäyksen ominaisuudet ja tunnistaa mahdolliset tavat saavuttaa hyökkäyksen lopullinen päämäärä. (Byres, Franz & Miller, 2004.)

SCADA-järjestelmien kyberturvallisuuden tutkiminen on mielestäni erittäin tärkeää. Monet tutkimuksista keskittyivät jonkin pienen osasen parantamiseen ja vaikka sekin on tärkeää, olisi mielestäni mielenkiintoista tutkia SCADA-järjestelmiä enemmän myös kokonaisuuksina ja esimerkiksi sosiaalista manipulointia hyökkäystapana. Olisi myös perusteltua tutkia enemmän erilaisia kyberturvallisuusstandardeja. Ymmärrettävästi tutkimukset, jotka käsittelevät tuotannossa olevien järjestelmien turvallisuutta eivät ole julkisia. Toivon kuitenkin sellaista tutkimusta tehtävän.

LÄHTEET

Baliga, A. Ganapathy, V. & Iftode, L. (2011). Detecting Kernel-Level Rootkits Using Data Structure Invariants. IEEE Transactions on Dependable and Secure Computing, 670-684.

Byres, E. Franz, M. & Miller, D. (2004). The use of attack trees in assessing vulnerabilities in SCADA systems. Proceedings of the international infrastructure survivability workshop.

Cárdenas, A. Amin, S. Lin, Z.-S. Huang, Y.-L. Huang, C.-Y. & Sastry, S. (2011). Attacks against process control systems: risk assessment, detection, and response. Proceedings of the 6th ACM symposium on information, computer and communications security (ss. 355-366). ACM.

Cherdantseva, Y. Burnap, P. Blyth, A. Eden, P. Jones, K. Soulsby, H. & Stoddart, K. (2016). A review of cyber security risk assessment methods for SCADA systems. Computers & Security, 1-27.

Ericsson, G. N. (2010). Cyber Security and Power System Communication-Essential Parts of a Smart Grid Infrastructure. IEEE Transactions on Power Delivery, 1501-1507.

Evancich, N. & Li, J. (2016). Attacks on Industrial Control Systems. Teoksessa E. Colbert & K. Alexander, Cyber-security of SCADA and Other Industrial Control Systems (ss. 95-110). Springer Nature.

Fonseca, J. Vieira, M. & Madeira, H. (2014). Evaluation of Web Security Mechanisms Using Vulnerability & Attack Injection. IEEE Transactions on Dependable and Secure Computing, 440-453.

Galloway, B. & Hancke, G. P. (2013). Introduction to Industrial Control Networks. IEEE Communications Surveys & Tutorials, 860-880.

Hong, J. Nuqui, R. F. Kondabathini, A. Ishchenko, D. & Martin, A. (2019). Cyber Attack Resilient Distance Protection and Circuit Breaker Control for Digital Substations. Transactions on Industrial Informatics, 4332-4341.

Huang, H. & Kirchner, H. (2011). Formal Specification and Verification of Modular Security Policy Based on Colored Petri Nets. IEEE Transactions on Dependable and Secure Computing, 852-865.

Huoltovarmuuskeskus. (2005). CIP – kriittisen infrastruktuurin turvaaminen. Noudettu osoitteesta

https://cdn.huoltovarmuuskeskus.fi/app/uploads/2016/08/31144136/CIP-raportti_final.pdf

Karnouskos, S. (2011). Stuxnet worm impact on industrial cyber-physical system security. IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society, (ss. 4490-4494).

Kuperman, B. Brodley, C. Ozdoganoglu, H. Vijaykumar, T. & Jalote, A. (2005). Detection and prevention of stack buffer overflow attacks. Communications of the ACM, 50-56.

Langner, R. (2011). Stuxnet: Dissecting a Cyberwarfare Weapon. IEEE Security & Privacy, 49-51.

Mclaughlin, S. Konstantinou, C. Wang, X. Davi, L. Sadeghi, A.-R. Maniatakos, M. & Karri, R. (2016). The Cybersecurity Landscape in Industrial Control Systems. Proceedings of the IEEE, 1039-1057.

Metke, A. R. & Ekl, R. L. (2010). Security Technology for Smart Grid Networks. IEEE Transactions on Smart Grid, 99-107.

Microsoft. (7. 4 2020). Microsoft Documentation. Noudettu osoitteesta https://docs.microsoft.com/en-us/windows/win32/dlls/dynamic-link-libraries

Nazir, S. & Kaleem, M. (2018). Random Network Coding for Secure Packet Transmission in SCADA Networks. 2018 3rd International Conference on Emerging Trends in Engineering, Sciences and Technology (ICEEST), (ss. 1-4). Karachi.

Nicholson, A. Webber, S. Dyer, S. Patel, T. & Janicke, H. (2012). SCADA security in the light of Cyber-Warfare. Computers & Security, 418-436.

Pelechrinis, K. Iliofotou, M. & Krishnamurthy, S. V. (2011). Denial of Service Attacks in Wireless Networks: The Case of Jammers. IEEE Communications Surveys & Tutorials, 245-257.

Racic, R. Ma, D. & Chen, H. (2006). Exploiting MMS Vulnerabilities to Stealthily Exhaust Mobile Phone's Battery. 2006 Securecomm and Workshops, (ss. 1-10). Baltimore.

Rudd, E. M. Rozsa, A. Gunther, M. & Boult, T. E. (2017). A Survey of Stealth Malware Attacks, Mitigation Measures, and Steps Toward Autonomous Open World Solutions. IEEE Communications Surveys & Tutorials, 1145-1172.

Saito, T. Sugawara, R. Yokoyama, M. Kondo, S. Miyazaki, H. Bing, W. & Watanabe, R. (2017). Mitigating Use-After-Free Attack with Application

Program Loader. IEEE 31st International Conference on Advanced Information Networking and Applications (AINA), (ss. 919-924). Taipei.

Salamat, B. Jackson, T. Wagner, G. Wimmer, C. & Franz, M. (2011). Runtime Defense against Code Injection Attacks Using Replicated Execution. IEEE Transactions on Dependable and Secure Computing, 588-601.

Sanastokeskus TSK. (27. 2 2020). Tepa-termipankki. Noudettu osoitteesta http://www.tsk.fi/tepa/fi/haku/kyberhy%C3%B6kk%C3%A4ys

Sommestad, T. Ericsson, G. N. & Nordlander, J. (2010). SCADA system cyber security - A comparison of standards. IEEE PES General Meeting, (ss. 1-8). Providence.

Sullivan, D. Luiijf, E. & Colbert, E. J. (2016). Components of Industrial Control Systems. Teoksessa A. K. Edward J. M. Colbert, Cyber-security of SCADA and Other Industrial Control Systems (ss. 15-28). Springer.

Ten, C.-W. Liu, C.-C. & Manimaran, G. (2008). Vulnerability Assessment of Cybersecurity for SCADA Systems. IEEE Transactions on Power Systems, 1836-1846.

Ten, C.-W. Manimaran, G. & Liu, C.-C. (2010). Cybersecurity for Critical Infrastructures: Attack and Defense Modeling. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 853-865.

Vukovic, O. Sou, K. C. Dan, G. & Sandberg, H. (2012). Network-Aware Mitigation of Data Integrity Attacks on Power System State Estimation. IEEE Journal on Selected Areas in Communications, 1108-1118.

Zhu, B. Joseph, A. & Sastry, S. (2011). A Taxonomy of Cyber Attacks on SCADA Systems. International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing, (ss. 380-388).